Ca' Foscari University of Venice

Master's Degree

In Management

**Joint Degree in Financial Analytics
with Stevens Institute of Technology**

Final Thesis

Cryptocurrency

Supervisor
Ch.ma Prof.ssa Chiara Saccon

Assistant supervisor
Ch. Prof. Khaldoun Khashanah,
Stevens Institute of Technology

Graduand
Beatrice Busato
Matriculation
number 885826

Academic Year
2021 / 2022

CONTENTS

Abstract

Cryptocurrencies have become an important topic in the financial industry, which seeks to determine their impact in current transaction spaces. Launched in 2008, the advent of cryptocurrencies has made it possible to send and receive online payments directly between parties without passing through any financial institution. However, cryptocurrency analysis represents a challenge due to the high level of volatility and complexity of these financial instruments. This study is focused on the analysis of the cryptocurrency market taking into consideration cryptocurrencies of the Bitcoin family (BTC, BSV), Ethereum family (ETC, ETH), Dogecoin, and Litecoin. The objective of this study is to better understand the distributed ledger and how it works in real or simulated environments, along with providing insights concerning the most appropriate analysis. In particular, this study targets a natural question that regards whether fundamental analysis can be applied to cryptocurrency time series. This study aims to apply Time-Series analysis and technical and fundamental analysis to cryptocurrencies. The data considered are historical as well as real-time prices, volumes, and flows. The preliminary results show a different probability distribution among the variables' returns which excludes the possibility of treating the variables in the same way when forecasting returns. Technical indicators produced results for each cryptocurrency, and provide different insights in terms of oversold or overbought conditions. Regarding Fundamental analysis, the preliminary results related to correlation among returns, addresses, and volumes provide different results which depend on the variable's nature. The conclusion can be drawn that each analysis provides relevant inputs which can be combined to obtain a complete variable analysis.

1. Introduction

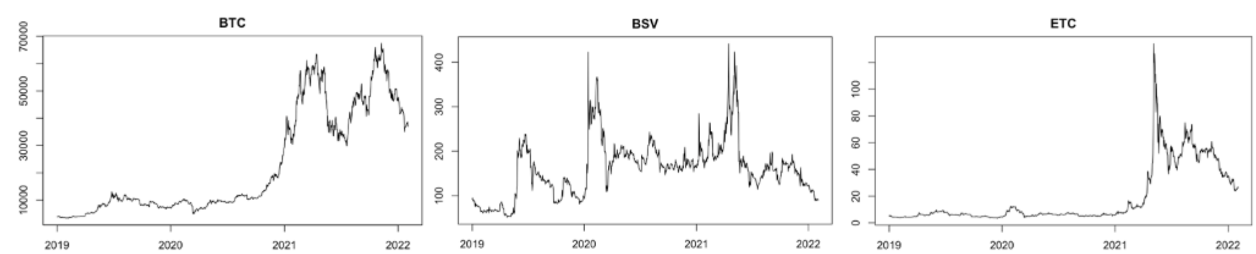
It is widely acknowledged that when analyzing an asset the most relevant analyses are Fundamental and Technical analyses. Fundamental and Technical analysis allows investors to make informed decisions for investing. This study aims to investigate whether cryptocurrency time-series can be analyzed fundamentally and technically, and derive the kind of information provided by those analyses concerning cryptocurrencies.

Cryptocurrencies are defined as a new type of financial instrument that made their first appearance after the publication of the "Bitcoin: A Peer-to-Peer Electronic Cash System"[1] in 2008 written by Satoshi Nakamoto, the creator of the first cryptocurrency, the Bitcoin. The concept underlying cryptocurrencies is the absence of regulations by financial institutions. Financial institutions act as trusted third parties to process transactions, resulting in inherent weaknesses of the trust-based model. Among those weaknesses, there is the inability to carry out irreversible transactions and transaction costs arising from the mediation of financial institutions. Cryptocurrencies respond to these problems by identifying themselves as a Peer-to-Peer tool that allows the exchange of online payments directly between the sender and the receiver, without the mediation of a financial institution.

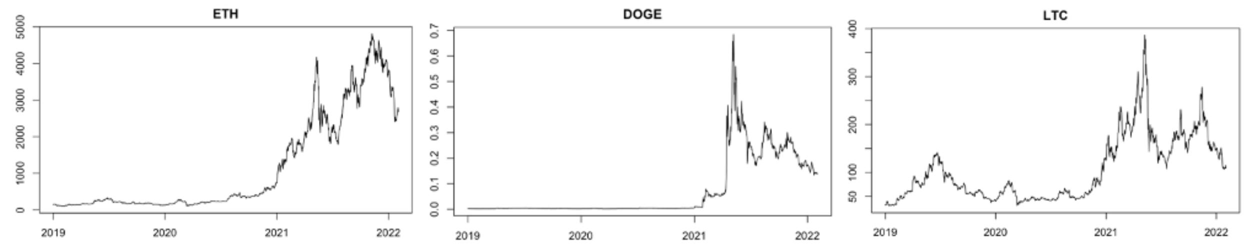
Since cryptocurrencies are peer-to-peer, global, and government-free, they are attractive as a medium of exchange. Users may be concerned, however, about a lack of confidence in the system and the unacceptability of cryptocurrencies as a payment method. Cryptocurrencies were not designed as an investment asset, but as a new type of currency.

The price of cryptocurrencies has increased substantially since 2010. According to Baur et al (2018)[2] and Corbet et al. (2018)[3], cryptocurrency accounts are primarily used as speculative investments and not as alternative currencies. Figure 1 shows the price movements of cryptocurrencies in this study's reference period.

Figure 1: Cryptocurrencies price movements 2019-2022



[1] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf
[2] Baur, D. G., K. Hong, and A. D. Lee (2018). Bitcoin: Medium of exchange or speculative assets? Journal of International Financial Markets, Institutions and Money 54, 177–189.
[3] Corbet, S., A. Meegan, C. Larkin, B. Lucey, and L. Yarovaya (2018). Exploring the dynamic relationships between cryptocurrencies and other financial assets. Economics Letters 165, 28– 34.



Regarding the role of cryptocurrencies in portfolio optimization, using the mean-CVaR[4] approach for US, European, and Chinese portfolio assets, it is important to consider the effect of adding cryptocurrencies to the optimal portfolio. Cryptocurrencies, especially Bitcoins, are shown to increase portfolio returns primarily due to their higher returns rather than lower volatility, and contribute to portfolio diversification.

Cryptocurrencies are based on blockchain technology built on cryptographic evidence rather than trust. To maintain control over the blockchain and stop attacks, major cryptocurrencies, such as Bitcoin and Ethereum, run on the Proof-of-Work[5] blockchain, which requires miners to solve computational puzzles that verify every transaction.

The purpose of this study is to identify the information deriving from Technical and Fundamental analysis applied to cryptocurrencies. Specifically, provide information to technical and non-technical investors to trade market movements with confidence, create informed strategies with better odds of profitability, and forecast the direction of the financial market through the use of technical indicators.

The first approach performed in analyzing these particular financial instruments is time series analysis using statistical tests, residual analysis, and stochastic models such as ARMA, ARIMA, and GARGH to forecast returns for the next month. Subsequently, the study moves to Fundamental analysis by analyzing factors such as Market Capitalization and Volumes as financial metrics to assist in understanding the trading conditions of assets. Along with financial metrics, Fundamental analysis in this study concerns also blockchain metrics, in particular, the Hash Rate, and addresses analysis. As the last approach, this study focuses on technical analysis and uses mathematical indicators such as Relative Strength Index (RSI) and Klinger Oscillator and models on previous price action data to forecast future trends. The approaches performed in this study allow for a

[4] A measure of tail risk is used in risk assessment to measure the amount of tail risk a portfolio faces.
[5] Shi, N. A new proof-of-work mechanism for bitcoin. *Financ Innov* **2,** 31 (2016)

complete view of cryptocurrency analysis and provide information regarding the most appropriate analysis to be performed by technical and non-technical investors to make informed investing decisions.

1.1 Blockchain Technology and Information Security

Cryptocurrencies are based on blockchain technology which works as an information system security. Information system security is the process of securing an automated information system so that the relevant objectives of availability, confidentiality, and integrity of information system resources are attained[6]. Specifically, availability refers to the access and use of information in a timely and reliable manner; confidentiality regards the protection of confidential information aiming at preventing unauthorized disclosure to unauthorized sources, and integrity aims at preventing unauthorized modifications to the information or unauthorized entities from making changes to information. Integrity is strongly related to authenticity which refers to the verification of the information and its origin.

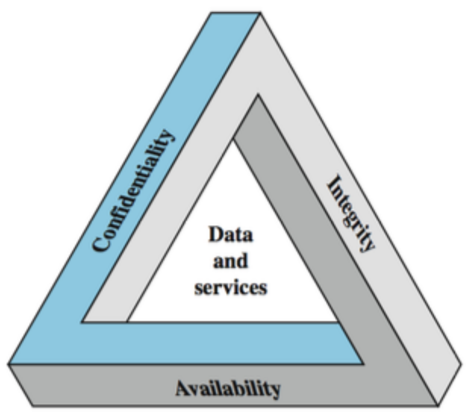


Figure 2: Information Security[7]

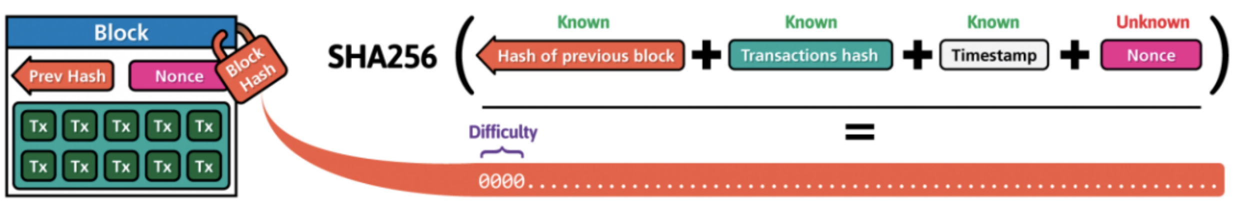
Confidentiality is achieved using Encryption (or Encipherment). Encryption is a process by which protected information is altered or transformed into a form that is incomprehensible to all but

[6] Including hardware, software, firmware, information/data, and telecommunications
[7] Source: NIST SP 800-14 https://doi.org/10.6028/NIST.SP.800-14

authorized receivers[8]. Often referred to as single-key encryption, the permutation and substitution algorithms encrypt and decrypt using the same key. The requirements for secure use relate to the need for a strong underlying encryption algorithm, and the secret keys must be obtained and kept securely by both the sender and the receiver.

Integrity is achieved through Message Authentication Codes (MAC). Message Authentication Codes maintain security through the use of functions that operate on variable-length input data (named "messages"). These functions are parametrized by symmetric key pairs. Authenticity is achieved through a digital signature. Digital signature and Message Authentication Codes both require the use of a cryptographic primitive named Cryptographic Hash Functions[9]. Hash functions optimize the digital signature schemes since without them the signature will have the same size as the message. Through the digital generation algorithm, the sender does not sign the entire message but only the digest of it. Moreover, with the use of hash functions as one-way functions, one can implement a pseudo-random number generator (PRNGs) without the use of symmetric encryption.

Figure 3: Blockchain Proof-of-Work[10]



1.2 Cryptocurrencies strengths

A key characteristic of cryptocurrencies is their strength by design, which makes them viable currencies that have grown in status over the years, particularly owing to the fixed number of cryptocurrencies that are and will be available. A limited number of cryptocurrencies will prevent an oversupply of them from becoming inflated. Moreover, cryptocurrencies are generally

---

8 Stallings, W., Brown, L., Bauer, M.D., and Howard, M., 2012. *Computer security: principles and practice* (Vol. 2). Upper Saddle River: Pearson.

9 Sobti, R. and Geetha, G., 2012. Cryptographic hash functions: a review. *International Journal of Computer Science Issues (IJCSI)*, *9*(2), p.461.

10 Source: cryptographics.info

considered protected by inflation originating from government changes or restrictions (Magro 2016). Despite cryptocurrencies representing a means against inflating national currencies, a variety of external factors can cause the price to fluctuate wildly. The combination of price volatility and demand made cryptocurrencies, in particular Bitcoin, become the best-performing currency of 2015 using the US Dollar Index (Desjardins 2016)[11]. Concerning the use of cryptocurrencies, Argentina records a major cryptocurrency usage due to the high inflatable rate and the elevated number of unbanked citizens (Magro, 2016). To preserve the value of their currency, Argentinians, often converted it into USD. However, due to the high amount of money converted, the Government put restrictions on the maximum amount of conversion. Consequently, the restrictions have brought a black market to purchase USD along with increasing the cryptocurrency adoption. Another particular case is represented by the United Kingdom, especially after the decision to leave the European Union. Before the Brexit, the price of Bitcoin decreased by 15% (Bovaird, 2016)[12], followed by a high growth the day after the Brexit.

Therefore, cryptocurrencies increased in value after the vote. On the other hand, globally traded markets dropped due to the lack of trust deriving from the effects Brexit would have had on financial markets.

A relevant strength regarding cryptocurrency refers to its ability to be bought and sold rapidly and worldwide. Fiat currencies can be converted; however, this process requires a money exchange and they can be spent only if locally accepted. On the contrary, cryptocurrencies can be purchased through an online account and an online exchange. The process comprises a request for cryptocurrency and the transaction can be completed in minutes. Subsequently, the cryptocurrency is kept in a digital wallet allowing the owner to purchase worldwide.

---

[11] Desjardins, J. (2016, January 5). It's Official: Bitcoin was the Top Performing Currency of 2015. Retrieved from The Money Project Website: http://money.visualcapitalist.com/its-official-bitcoin-was-the-top- performing-currency-of-2015/

[12] Bovaird, C. (2016, June 24). Bitcoin Rollercoaster Rides Brexit As Ether Price Holds Amid DAO Debacle. Retrieved June 2016, from CoinDesk Website: http://www.coindesk.com/bitcoin-brexit-ether-price- rollercoaster/

1.3 Cryptocurrencies weaknesses

Cryptocurrency design also provides some inner weaknesses not easily modifiable. In particular, all semi-anonymous users can see all the transactions happening within the blockchain, however, cryptocurrency wallet owners cannot accurately be identified. The blockchain presents some weaknesses in terms of attacks since the ledger is shared among the users (King 2013[13]). Multiple "stress tests" have been conducted on the cryptocurrency network, in particular on the Bitcoin network, which was Distributed Denial-of-Service (DDoS)[14] attacks. A large number of exchange and miners were behind these tests, which were intended to prove a relevant aspect such that the Bitcoin's network cannot support a high volume of transactions. Additionally, there is an unfortunate design feature in the cryptocurrency code that allows users to shut down the network to make a point. These two aspects of cryptocurrency represent an integral part of their functioning and therefore, cannot be altered.

Among the strongest weaknesses of cryptocurrencies is their perceived reputation. An event closely linked to the reputation of cryptocurrencies, especially Bitcoin, is represented as the "Silk Road" case. Silk Road is an online marketplace located on the dark web used by users to carry out illegal transactions. This market is associated with the reputation of cryptocurrencies as these were the main means used to carry out transactions, due to the lack of traceability from institutions and semi-anonymity. For this reason, semi-anonymity is associated with crime and therefore, perceived negatively. The negative image influences the perception of general users regarding cryptocurrencies, which are perceived as being used only for criminal activities. The perception of security is also an issue. A relevant case is represented by the "Magic the Gathering Online Exchange" (Mt GOX). Mt GOX was the leading Bitcoin exchange in the world. Which failed due to a hacker attack in 2011 due to security flows. The failure of Mt GOX caused a reduction in the Bitcoin's value as users, influenced by the fear of being hacked, sold their Bitcoin. Subsequently, the same fate struck Ethereum causing, also in this case, a reduction in its value (Price 2016)[15].

[13] King, R. S. (2013, December 17). By reading this article, you're mining bitcoins. Retrieved from Quartz.com Website: http://qz.com/154877/by-reading-this-page-you-are-mining-bitcoins/.

[14] Multiple and remote resources are used by a threat actor to attack the organization's online operations. The largest problem with DDoS attacks is that they attempt to manipulate network equipment and services like routers, naming, or caching services.

[15] Price, R. (2016, June 17). Digital currency Ethereum is cratering because of a $50 million hack. Retrieved from Business Insider Website: http://www.businessinsider.com/dao-hacked-ethereum-crashing-in-value-tens- of-millions-allegedly-stolen-2016-6.

The lack of continuous updating of security standards has fostered a breeding ground for hacker attacks. Indeed, hacker attacks represent one of the fundamental reasons why the value of cryptocurrencies plummets, damaging their image and hindering their adoption. The ability to treat cryptocurrencies as commodities can also be considered a weakness. Commodity-based markets are characterized by high fluctuations caused by events that affect the market. The movements limit confidence as the occurrence of unforeseen events could lead to significant losses. Furthermore, what determines the price of cryptocurrencies is not clearly defined, causing a high level of intrusion regarding trading. The volatility of prices generates risks, which disadvantage the holding of cryptocurrency in the long term. To conclude, cryptocurrencies are currently not a mature form of currency in their market. However, an increase in confidence and, consequently, greater adoption could be a solution to this issue.

1.4 Cryptocurrencies opportunities

The technology underlying cryptocurrencies represents an innovation for financial systems. The integration of peer-to-peer technology with the financial technology currently in use, to solve traditional problems related to non-bank consumers. An example is Latin America, where 60% of the population does not have access to bank accounts (Magro, 2016). The absence of mediation by financial institutions would allow that 60% to exchange currency. To be able to carry out transactions via Bitcoin, the only device needed is a mobile phone, which in Latin America is owned by 70% of the population (Magro, 2016). The transaction via Bitcoin takes place by scanning the QR codes displayed by the app. The simplicity of the process represents an opportunity for potential application developers as this technology could be adopted by any industry that relies on compensation systems. Cryptocurrencies have a high potential in terms of international transactions especially due to their speed which could be useful in emergencies. The peer-to-peer system could be a solution for the transfer of money in extreme conditions, such as in the case of conflicts and dictatorships where governments control the transactions that take place within the country. Using this technology, money can be transferred for the full amount worldwide in minutes without having to wait days and without transaction costs. In difficult and extreme conditions, speed in transactions is a key element.

Many online stores such as eBay.com have already introduced a payment system similar to cryptocurrencies. PayPal has seen enormous success as it allows to send and receive immediate payments without transaction costs. Returning to the example of Silk Road, this market, albeit illegal, represented a meeting point between buyers and sellers who transacted through Bitcoins. This demonstrated how digital currencies could be the basis for transactions in a market without incurring fees.

In the context of international regulation concerning taxation, cryptocurrencies are considered valid as a form of device. However, laws need to be established to identify specific taxation for cryptocurrencies so that they can be considered a valid form of transaction. An important consideration concerns the decision of the European Court of Justice of 2015, which established the VAT exemption for Bitcoins. The exemption from VAT increases the flow of cryptocurrencies as it represents an advantage over fiat currencies. The continued interest in some level of regulation for cryptocurrency could increase its recognition by general users, who may be skeptical about the use of a currency and the lack of information on how it could affect tax declaration and returns.

A relevant opportunity for cryptocurrencies, particularly for Bitcoin, is that they can be considered a gold-like commodity. External events can affect the market's equilibrium globally and when this happens, the value of gold fluctuates considerably.

As the commodity market represents a formally accepted market, cryptocurrencies have mimicked this commodity and its characteristics. In this context, cryptocurrencies could play an important role in the commodity market. Cryptocurrencies can stimulate fiat currencies by leading investors to increase their perception of validity and improve their reputation.

2. Data Collection and Preparation

This study employs daily historical global prices obtained from Yahoo Finance encompassing 6 variables (BTC, BSV, ETC, ETH, DOGE, LTC) with a date range stemming from January 1, 2019, to February 2022. The data have been used for Time-series analysis and subsequently integrated by volumes, Flows (Inflow, Outflow, Netflow), and addresses (Active, Sending, Receiving) downloaded from CryptoQuant.com using JSON and Python for Technical and Fundamental Analyses.

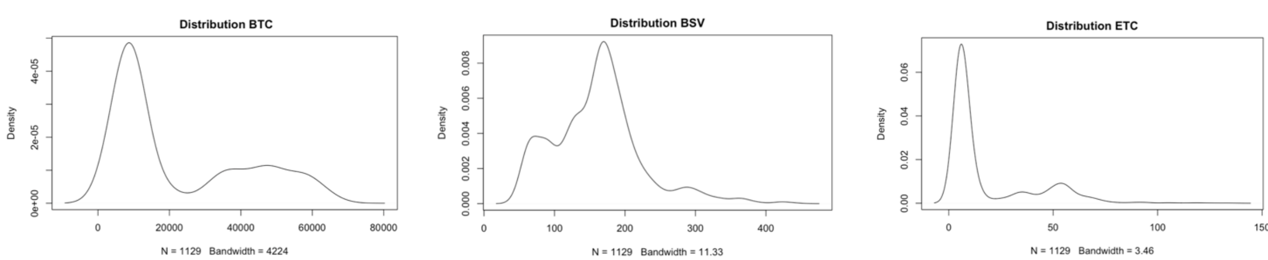
Besides Bitcoin, cryptocurrencies such as Ethereum and Litecoin are interesting to analyze since they represent the largest and most liquid cryptocurrencies in the market (Hudson et al. (2021)[16].

The analysis period has been selected according to the date on which the data regarding each cryptocurrency were published and available on public platforms. All cryptocurrency markets trade for 7 days a week and all 24 hours, therefore the dataset represents a complete timeline and includes all observations during the day, including weekends and holidays. Figure 1 shows the time series chart of each cryptocurrency for the selected sample period. The graphs show the dramatic price increase in each cryptocurrency starting from 2021 accompanied by large volatility.

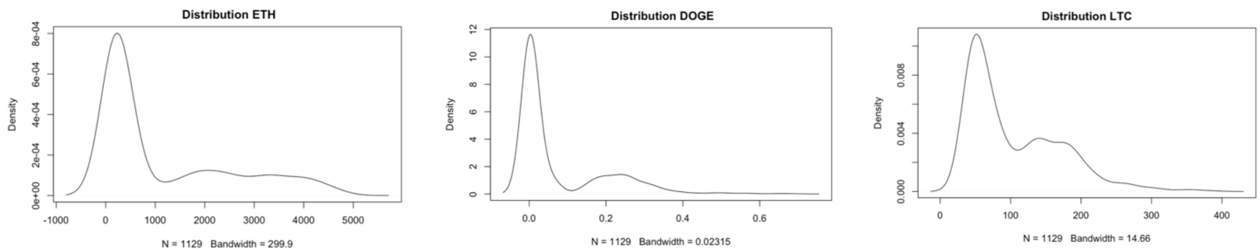
2.1 Experimental design and descriptive statistics

The first approach of the experimental design was the evaluation of the probability density function of the cryptocurrency prices. The probability density function is relevant as it allows for the identification of the probabilities associated with continuous stochastic variables. Figure 4 represents the probability density function of the price of each cryptocurrency. Looking at the plots, prices follow a log-normal distribution, which identifies a random variable with continuous probability distribution whose logarithm is normally distributed.

Figure 4: Cryptocurrencies prices probability density function



[16] Hudson, R. and Urquhart, A., 2021. Technical trading and cryptocurrencies. *Annals of Operations Research*, *297*(1), pp.191-220.



Given the price behavior, this study employs log returns. The basic assumption underlying log-returns is that instead of being compounded across sub-period, there are compounded continuously. Log returns have been computed such that:

$$rt = log\frac{P_t}{P_{t-1}}$$

where P_t represents the cryptocurrency's price at time *t* and P_{t-1} represents the first lag of the cryptocurrency price, therefore at time *t-1*.

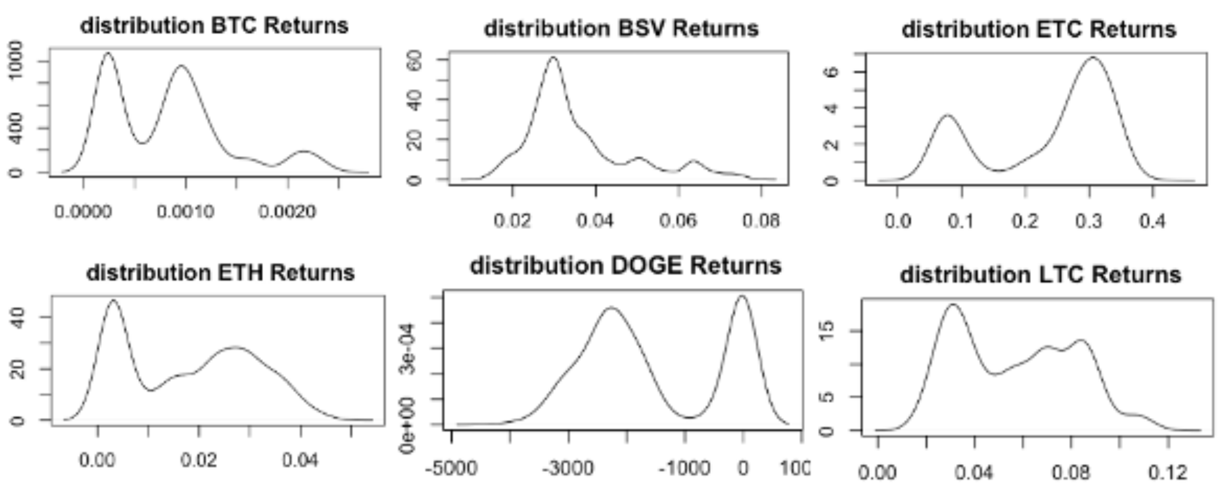
Table 1 reports the descriptive statistics of the returns of the cryptocurrencies object of this study. All cryptocurrencies have a positive mean return. Concerning skewness, it indicates a distortion from the normal distribution. The majority of returns is moderately skewed and therefore, the deviation from the normal distribution is moderate. The exception is BSV which shows a skewness of 1.20 and therefore is highly skewed. Regarding the Kurtosis, it measures the combined weight of the distribution's tails for the distribution's center. BTC and BSV present a value greater than 3 therefore the returns have heavier tails than the normal distribution. For the remaining cryptocurrencies, the returns present lighter tails than the normal distribution which corresponds to a lower kurtosis value.

Table 1: Descriptive statistics of the returns of BTC, BSV, ETC, ETH, DOGE, and LTC from 2019 to 2022

Cryptocurrency	Mean	Std. Dev	Variance	Skewness	Kurtosis
BTC	0.000825	0.0005710	0.00000	0.8403	3.2094
BSV	0.035614	0.0129522	0.00016	1.2065	3.8726
ETC	0.237054	0.0999971	0.00999	-0.7369	1.9593
ETH	0.0178965	0.0127371	0.00016	0.1148	1.6346
DOGE	1537.862	1178.877	1389752	0.2222	1.5798
LTC	0.05671564	0.02457324	0.00060	0.2129	1.8484

Upon computing log returns for each variable, the first approach has been the establishment of the probability density function to identify the distribution of log returns. Figure 5 shows that log returns do not follow a Log-normal distribution.

Figure 5: Cryptocurrencies log-returns probability density function



Due to this uncertainty, to provide the best fit to the data for the distribution, the two-sample Kolmogorov Smirnov test represents an efficient tool. The two-sample Kolmogorov Smirnov test determines whether two samples follow the same distribution[17] (Berger et al. (2014)). The intuition behind this is to match two cumulative distribution functions. The distributions tested using the Kolmogorov-Smirnov test are Gaussian distribution, T-student distribution, LaPlace distribution, and T-Sallis distribution. The t-student distribution is a probability distribution that presents similar characteristics to the normal distribution, however, the shape presents heavier tails. LaPlace distribution presents a higher dispersion of the data around the mean for the normal distribution. Finally, the T-Sallis distribution is a probability distribution derived from the maximization of the T-Sallis entropy[18]. The choice among the various distribution with the respect to the most appropriate fit to the data is based on the test value. The best fit is represented by the distribution that provides the lower test value. A preliminary descriptive analysis is detailed below in Table 2 showing that the T-Sallis distribution provides extreme test values concerning the other

[17] Berger, V.W. and Zhou, Y., 2014. Kolmogorov–Smirnov test: Overview. *Wiley statsref: Statistics reference online*.
[18] Entropy: a measure of uncertainty of a random variable. It is used in decision trees for heterogeneity estimation.

distributions tested. The test results provided evidence regarding the best distribution. For BTC the most appropriate distribution is LaPlace, for BSV is Gaussian, ETC is T-student, ETH is Gaussian, Doge is LaPlace, and LTC is T- student.

Table 2: Kolmogorov-Smirnov two-sample test results

Distribution	BTC	BSV	ETC	ETH	DOGE	LTC
Gaussian	0.51596	0.49379	0.51684	0.49025	0.97606	0.52128
T-Student	0.53546	0.52394	0.48493	0.55321	0.97606	0.50266
LaPlace	0.50887	0.50975	0.53280	0.50841	0.96277	0.53280
T-Sallis	1.00000	0.99645	0.93440	0.99911	1.00000	0.99557

Subsequently, linear regression was performed to describe the relationship among the different cryptocurrencies. In particular, the regression was based on regressing each variable on the remaining ones to have a clearer view regarding how strong their relationship is. This aspect is relevant to understanding if a decrease in price or return of one cryptocurrency could result in an increase or decrease of another cryptocurrency, depending on the relationship between the variables. A strong relationship between the variables means that they move in the same direction. On the contrary, a lack of association between the variables means that changes in one variable do not affect the remaining variables. Figure 6 shows the relationship of each variable included in this study with the remaining ones. Looking at the linear regressions outputs, the BTC shows a strong relationship with the other cryptocurrencies which means that a change in BTC will result in changes to the other cryptocurrencies. Concerning BSV, the variable shows a strong relationship with all variables besides LTC which shows a low association. A particular case is represented by ETC, which shows strong relationships with all variables besides LTC as for BSV in the previous case. The particular element in this regression is the lack of relationship between ETC and ETH, which is strange because both variables are from the Ethereum family. The regression of ETH on the remaining cryptocurrencies confirmed the relationship absence with ETC and confirmed a strong association with the remaining variables. DOGE is strongly associated with all variables while LTC shows a low relationship with BSV and ETC.

Figure 6: Linear regression

```
Call:
lm(formula = R$BTC ~ R$BSV + R$ETC + R$ETH + R$DOGE + R$LTC,
    data = R)

Residuals:
       Min         1Q     Median         3Q        Max
-5.043e-04 -8.627e-05 -1.563e-05  8.535e-05  5.995e-04

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.183e-04  2.430e-05  -8.983  < 2e-16 ***
R$BSV        1.154e-02  5.202e-04  22.188  < 2e-16 ***
R$ETC       -5.378e-04  1.127e-04  -4.774 2.05e-06 ***
R$ETH        2.815e-02  1.524e-03  18.475  < 2e-16 ***
R$DOGE      -8.011e-08  2.020e-08  -3.966 7.76e-05 ***
R$LTC        2.351e-03  4.805e-04   4.892 1.14e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Call:
lm(formula = R$BSV ~ R$BTC + R$ETC + R$ETH + R$DOGE + R$LTC,
    data = R)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0224797 -0.0064245  0.0007227  0.0047604  0.0316217

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.182e-02  1.012e-03  21.559  < 2e-16 ***
R$BTC        2.642e+01  1.191e+00  22.188  < 2e-16 ***
R$ETC        2.604e-02  5.389e-03   4.832 1.54e-06 ***
R$ETH        3.384e-01  8.265e-02   4.094 4.54e-05 ***
R$DOGE       1.153e-05  9.102e-07  12.671  < 2e-16 ***
R$LTC       -4.439e-02  2.319e-02  -1.914   0.0559 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Call:
lm(formula = R$ETC ~ R$BTC + R$BSV + R$ETH + R$DOGE + R$LTC,
    data = R)

Residuals:
      Min        1Q    Median        3Q       Max
-0.131872 -0.039627 -0.004527  0.028235  0.168799

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.968e-02  5.891e-03  16.920  < 2e-16 ***
R$BTC       -3.701e+01  7.753e+00  -4.774 2.05e-06 ***
R$BSV        7.828e-01  1.620e-01   4.832 1.54e-06 ***
R$ETH       -2.231e-02  4.565e-01  -0.049   0.9610
R$DOGE      -8.325e-05  4.721e-06 -17.633  < 2e-16 ***
R$LTC        2.191e-01  1.272e-01   1.723   0.0852 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Call:
lm(formula = R$ETH ~ R$BTC + R$BSV + R$ETC + R$DOGE + R$LTC,
    data = R)

Residuals:
      Min        1Q    Median        3Q       Max
-0.007533 -0.001252 -0.000070  0.001575  0.009284

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.498e-04  4.316e-04   0.579 0.562846
R$BTC        8.285e+00  4.485e-01  18.475  < 2e-16 ***
R$BSV        4.350e-02  1.062e-02   4.094 4.54e-05 ***
R$ETC       -9.542e-05  1.952e-03  -0.049 0.961028
R$DOGE      -7.143e-06  2.762e-07 -25.866  < 2e-16 ***
R$LTC       -3.007e-02  8.281e-03  -3.631 0.000296 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Call:
lm(formula = R$DOGE ~ R$BTC + R$BSV + R$ETC + R$ETH + R$LTC,
    data = R)

Residuals:
    Min      1Q  Median      3Q     Max
-860.41 -157.79  -28.96  132.43 1147.31

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     362.13      35.31  10.255  < 2e-16 ***
R$BTC       -172603.62   43517.97  -3.966 7.76e-05 ***
R$BSV         10854.13     856.64  12.671  < 2e-16 ***
R$ETC         -2606.39     147.82 -17.633  < 2e-16 ***
R$ETH        -52296.17    2021.80 -25.866  < 2e-16 ***
R$LTC        -10407.67     641.44 -16.225  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

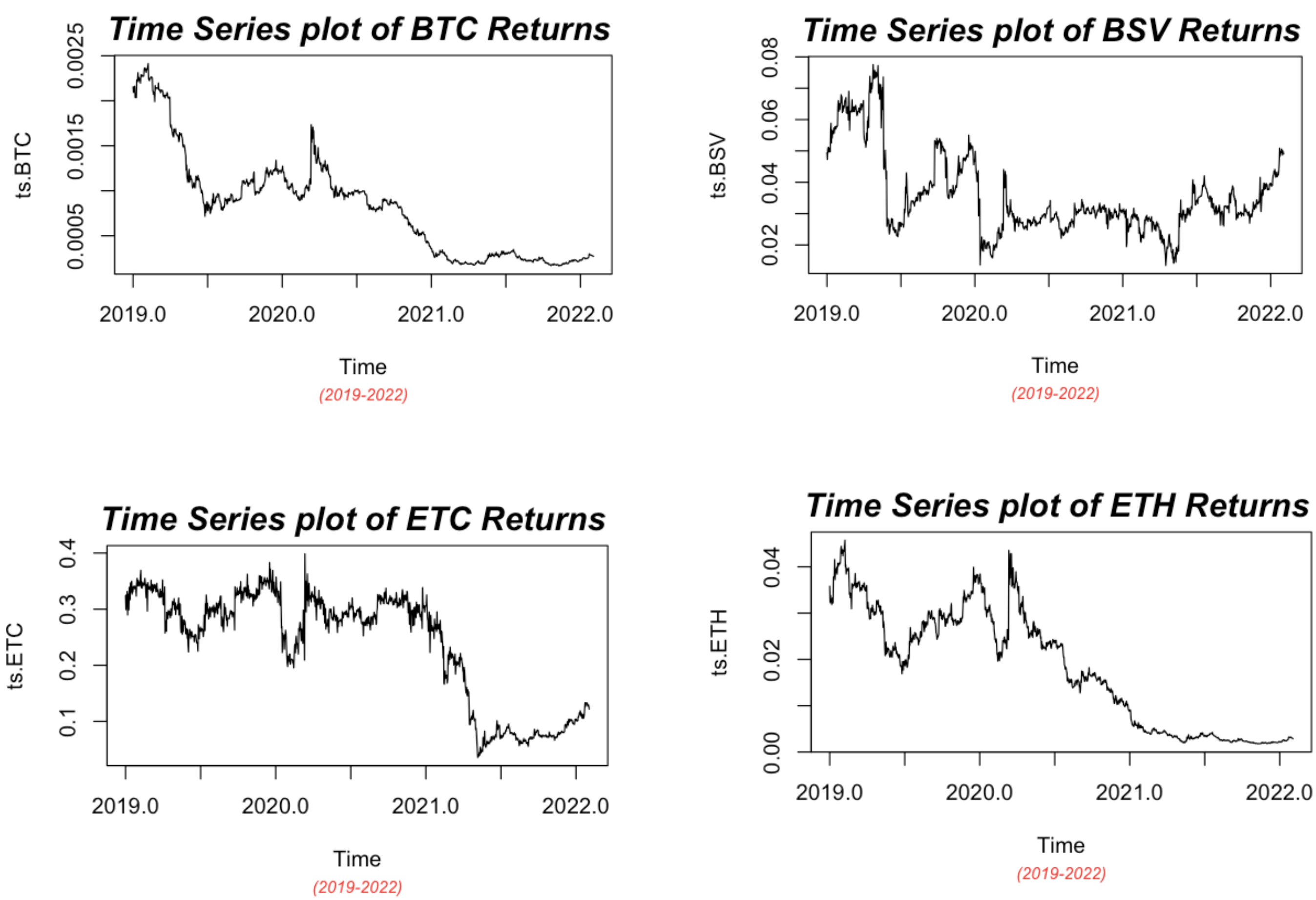
```
Call:
lm(formula = R$LTC ~ R$BTC + R$BSV + R$ETC + R$ETH + R$DOGE,
    data = R)

Residuals:
      Min        1Q    Median        3Q       Max
-0.029018 -0.007484  0.000281  0.007820  0.033734

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.797e-02  1.302e-03  21.476  < 2e-16 ***
R$BTC        8.886e+00  1.816e+00   4.892 1.14e-06 ***
R$BSV       -7.330e-02  3.830e-02  -1.914 0.055888 .
R$ETC        1.204e-02  6.988e-03   1.723 0.085249 .
R$ETH       -3.862e-01  1.064e-01  -3.631 0.000296 ***
R$DOGE      -1.826e-05  1.125e-06 -16.225  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

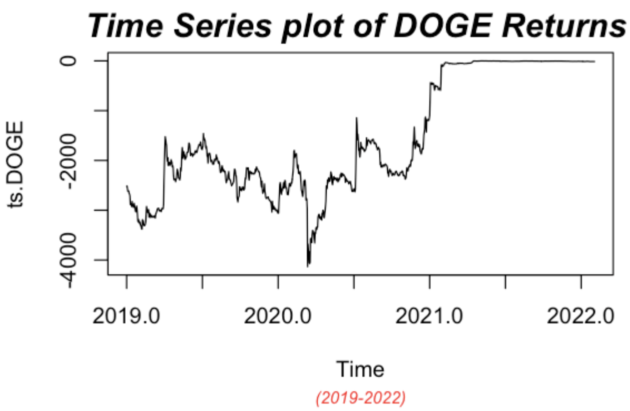
3. Methodology I: Time-Series Analysis

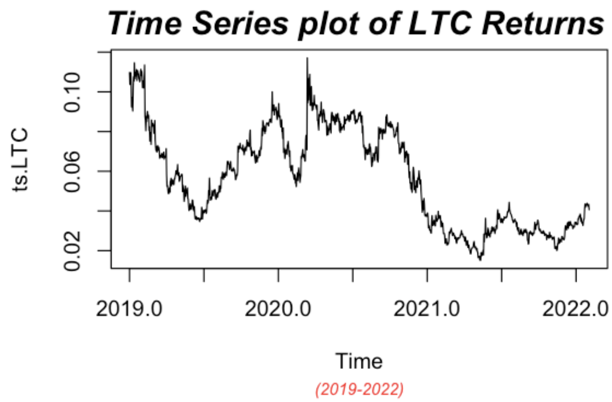
3.1 Preliminary assessments and stationarity

The first methodology applied in this study is Time-Series Analysis. This analysis aims to analyze the sequence of data collected over certain periods. A time series is defined as a sequence of observations in chronological order and represent the partial realization of the stochastic process's trajectory for the time considered. Time-series plot can be used to assess the presence of stationary behaviors. Generally, stationary time series show oscillations around a fixed level. Therefore, any different behavior can cause the non-stationarity of the process.
Figure 7 shows the plots of the time series of returns for each cryptocurrency. The variables show many fluctuations but not around a certain fixed level.

Figure 7: Plots of time-series returns







Stationarity plays a relevant role in time-series analysis. Stationary processes are defined as processes in which when observing a time series, the fluctuations appear random but often with the same type of stochastic behavior from one period to the next. Generally, stationary processes show time-invariant behavior, and therefore, the probability distribution of a sequence of observations does not depend on the time origin.

To confirm the initial assumption of non-stationarity, the first approach was performing the Augmented Dickey-Fuller test (ADF). The ADF test is used to check for the presence of unit roots in the process. The test has under the null hypothesis the presence of unit roots, therefore, a non-stationary process. The test statistics are usually given by $\frac{\gamma}{S.E.\ \gamma}$ where γ is the coefficient of the process at time $t-1$ divided by its standard error. Generally, if the value of the test statistics is lower than the critical value, the test rejects the null hypothesis. If the null hypothesis is rejected then the process is stationary. Otherwise, if the test fails to reject the null hypothesis, the process has unit roots and therefore is non-stationary.

The ADF test was performed for each cryptocurrency. The $p-value$ resulting from the test is summarized in Table 3. The confidence interval assumed for this test is 95%. Since the $p-value$ resulting from the ADF test for each cryptocurrency is greater than the value of α which is equal to 5%, the test fails to reject the null hypothesis, and therefore, it confirms the initial assumption of non-stationarity. For this reason, the initial process assumption for these time series is that they represent Random Walk processes.

Table 3. P-values obtained from the ADF test for each cryptocurrency

Cryptocurrency	BTC	BSV	ETC	ETH	DOGE	LTC
p-value	0.4534	0.3044	0.6295	0.327	0.5415	0.5735

The results obtained from the ADF tests are shown below in Figure 8.

Figure 8: ADF test for each cryptocurrency

```
        Augmented Dickey-Fuller Test

data:  R.BTC
Dickey-Fuller = -2.2951, Lag order = 10, p-value =
0.4534
alternative hypothesis: stationary
```

```
        Augmented Dickey-Fuller Test

data:  R.BSV
Dickey-Fuller = -2.6471, Lag order = 10, p-value =
0.3044
alternative hypothesis: stationary
```

```
        Augmented Dickey-Fuller Test

data:  R.ETC
Dickey-Fuller = -1.8791, Lag order = 10, p-value =
0.6295
alternative hypothesis: stationary
```

```
        Augmented Dickey-Fuller Test

data:  R.ETH
Dickey-Fuller = -2.4874, Lag order = 10, p-value =
0.372
alternative hypothesis: stationary
```

```
        Augmented Dickey-Fuller Test

data:  R.DOGE
Dickey-Fuller = -2.0869, Lag order = 10, p-value =
0.5415
alternative hypothesis: stationary
```

```
        Augmented Dickey-Fuller Test

data:  R.LTC
Dickey-Fuller = -2.0115, Lag order = 10, p-value =
0.5735
alternative hypothesis: stationary
```

Due to the absence of the stationarity condition, the second approach was to establish whether by taking the first difference of the preliminary data, the processes would become stationary. After taking the first difference, the ADF was performed obtaining the p-values summarized in Table 4. Since the p-values are lower than the value α, the processes are stationary and integrated on order 1.

Table 4. P-values obtained from the ADF test for each cryptocurrency after taking the first difference

Cryptocurrency	BTC	BSV	ETC	ETH	DOGE	LTC
p-value	0.01	0.01	0.01	0.01	0.01	0.01

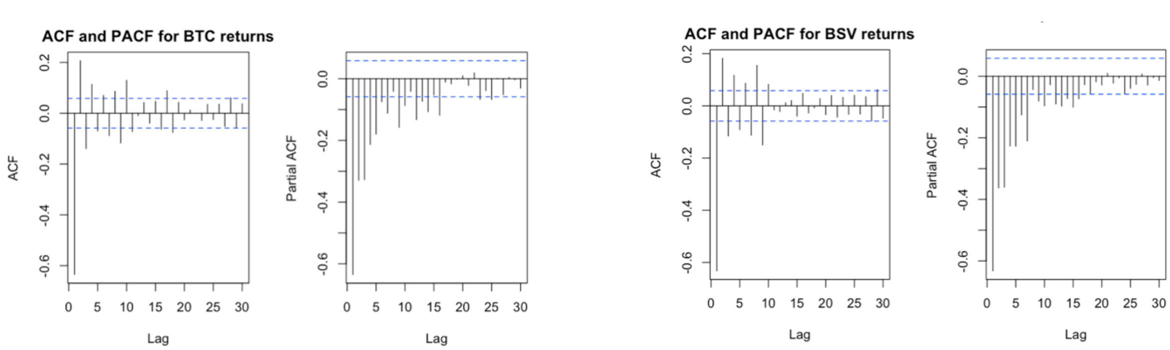
3.2 ARIMA model

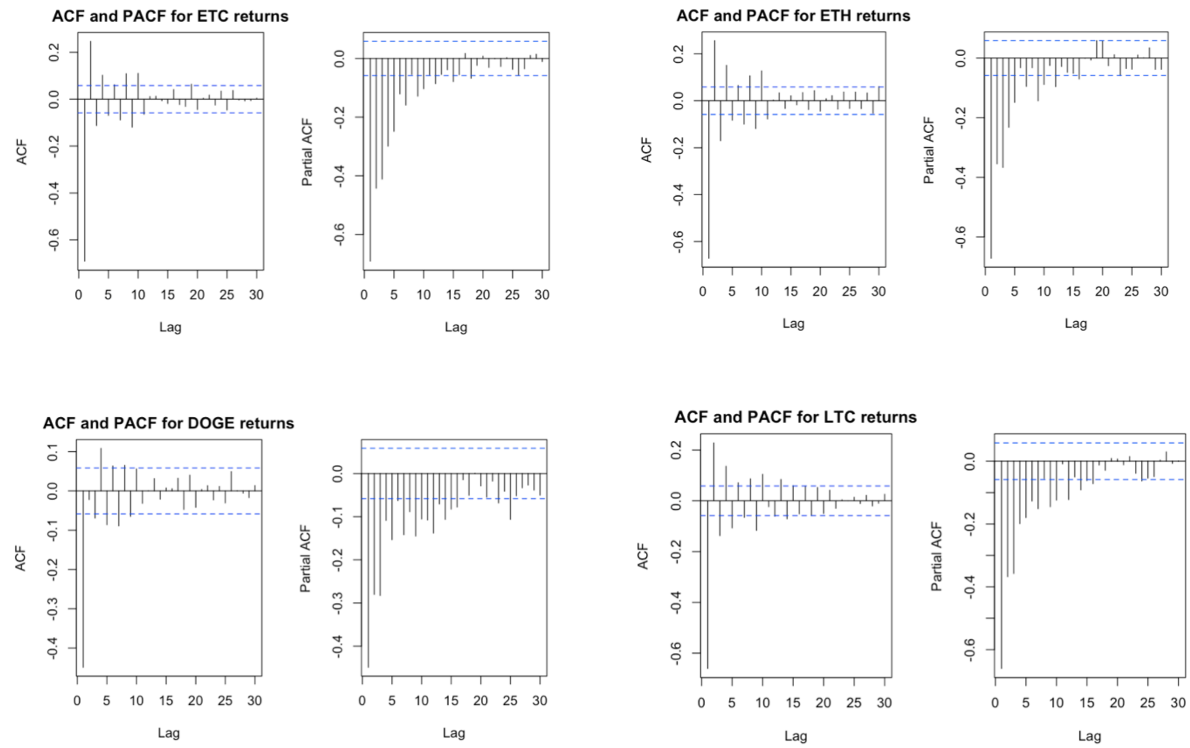
The first difference of a Random Walk process is usually a white noise that is always stationary, and therefore, independent of time. Based on this assumption, the log-returns seemed to follow an ARMA model. Taking into account the integration of order 1, the first plausible model tested to forecast cryptocurrency returns was an Autoregressive Integrated Moving Average (ARIMA). The model is used in regression analysis to forecast values by analyzing time series data. The model combines the Autoregressive (AR) and the Moving Average (MA) components. Autoregression refers to the regression of the process on its own lagged values and aims to forecast the returns based on past values and periods. The Moving Average component instead represents the weighted average of the past values of the white noise process.

3.2.1 ARIMA model order identification and forecasts

The first approach applied to identify the most appropriate order for the time series was the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF).
The ACF provides information regarding how the time series are related to lagged values, on average. RStudio shows the ACF with test bounds which aim to test whether the autocorrelation coefficient is 0. The null hypothesis is that there is no correlation which is rejected when the autocorrelation crosses the bounds. The PACF refers to the conditional correlation between observations taking into account shorter lags. The ACF and PACF plots of the time series show many spikes on the first lags and are represented in Figure 9.

Figure 9: ACF and PACF for each cryptocurrency





Preliminary order identification has been conducted using the Extended Autocorrelation Function (EACF[19]). The results obtained from running the EACF using RStudio are shown in Figure 10. Based on these results, different values for the Autoregressive and the Moving Average components were tested to identify the most appropriate order.

Figure 10: EACF for each cryptocurrency

```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x x x x x x x x x  o  o  o
1 x x o o o o o o o x x  o  o  o
2 x x x o o o o o o x o  o  o  o
3 x x x o x o o o o x x  o  o  o
4 x x x x x o o o x o o  o  o  o
5 x x x x o x o o x o o  o  o  o
6 x x x x o x o o x x o  o  o  o
7 x x x x x x x x x o o  o  o  o
```

BTC

```
AR/MA
  0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x x x x x x x x o  o  o  o
1 x x x o o o o x x x o  o  o  o
2 x x x o o o o o x o o  o  o  o
3 x x x o x o o o x o o  o  o  o
4 x x x x x o o o o o o  o  o  o
5 x x x o x x o o o o o  o  o  o
6 x x o x x x x x o o o  o  o  o
7 x x x x x x x o o o o  o  o  o
```

BSV

[19] Proposed by Tsay and Tiao IN 1984, the EACF is used to identify the order of ARMA models based on iterated least square estimates of the Autoregressive component.

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	x	x	x	o	o	o
1	x	x	x	o	o	o	o	o	o	x	o	o	o	o
2	x	x	x	o	o	o	o	o	o	o	o	o	o	o
3	x	x	x	o	x	o	o	o	o	x	o	o	o	o
4	x	x	x	x	x	x	o	o	o	x	o	o	o	o
5	x	x	x	x	x	x	x	o	o	o	o	o	o	o
6	x	x	x	o	x	x	x	o	o	o	o	o	o	o
7	x	x	x	o	x	x	x	o	o	o	o	o	o	o

ETC

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	x	x	x	o	o	o
1	x	x	x	x	x	o	o	o	o	x	x	o	o	o
2	x	x	x	o	o	x	o	o	o	x	o	o	o	o
3	x	o	x	x	x	x	o	o	o	x	x	o	o	o
4	x	o	x	x	x	x	o	o	x	x	o	o	o	o
5	x	x	x	x	x	x	o	o	o	x	o	o	o	o
6	x	x	x	x	x	x	o	o	o	o	o	o	o	o
7	x	x	x	o	x	x	x	x	o	o	o	o	o	o

ETH

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	o	x	x	x	x	x	x	x	o	o	o	o	o
1	x	x	o	o	o	o	o	o	o	o	o	o	o	o
2	x	x	x	o	o	o	o	o	o	o	o	o	o	o
3	x	x	x	x	o	o	o	o	o	o	o	o	o	o
4	x	x	x	x	o	o	o	o	o	o	o	o	o	o
5	x	x	x	x	x	o	o	o	o	o	o	o	o	o
6	x	x	o	x	x	o	o	o	o	o	o	o	o	o
7	x	x	o	o	x	o	o	o	o	o	o	o	o	o

DOGE

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	x	x	x	x	x	x	x	x	x	o	x	x	x
1	x	x	x	o	o	o	o	o	o	x	o	x	o	o
2	x	x	x	x	o	o	o	o	o	x	o	x	o	o
3	x	x	o	x	x	o	o	o	o	x	o	o	o	o
4	x	x	o	x	o	x	o	o	o	o	o	o	o	o
5	x	x	o	o	o	x	o	o	o	o	o	o	o	o
6	x	x	x	o	o	x	o	o	o	o	o	o	o	o
7	x	x	o	o	x	x	x	x	x	o	o	o	o	o

LTC

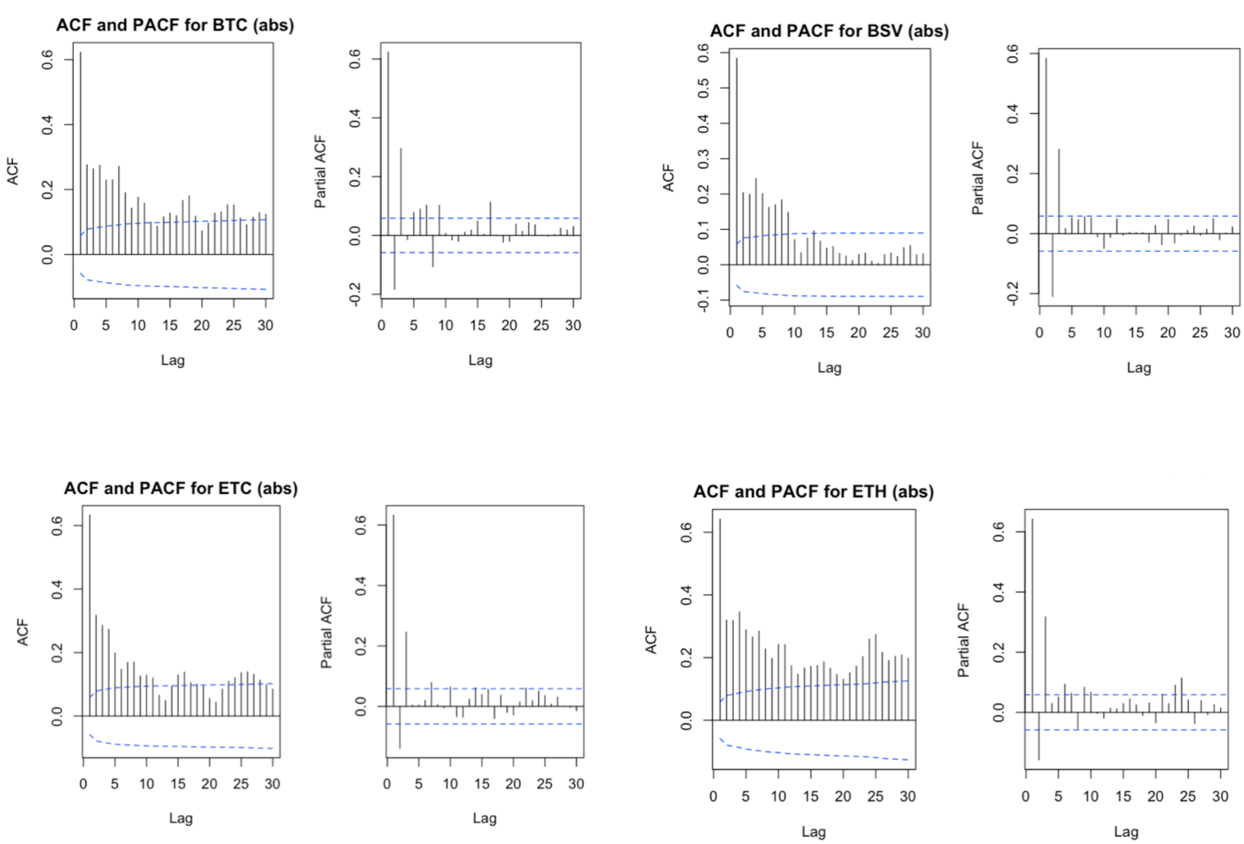
The selection among the different models has been done using the Akaike Information Criterion (AIC) as a statistical criterion to discriminate models resulting from the preliminary approach and obtain the optimal lag order. The best fit is represented by the model that provides the lower AIC. The result obtained suggested the most appropriate lag order for each variable, subsequently employed to forecast the returns for the next month for each variable. The results obtained by the forecast model are not entirely precise in terms of accuracy, mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). The residual analysis shows no relevant correlation in the residuals besides a few spikes for each cryptocurrency. Concerning the Goodness of Fit, determined by the Adjusted Person Goodness of fit test, assuming a 95% confidence interval, the p-value is greater than alpha, which in this case is 5%. Therefore, I cannot conclude that the observed proportions are significantly different from the specified proportions. The same conclusion is obtained for all cryptocurrencies.

3.3 GARCH Model

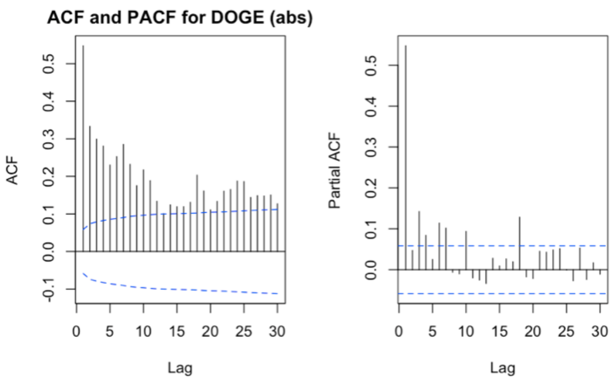
The second approach for Time-Series analysis was the fitting of a Generalized Autoregressive Conditional Heteroskedasticity (GARCH[20]) model. GARCH models represent a preferred choice as they provide a more realistic context for forecasting returns and estimating the volatility of the variables included in this study. In a statistical model, heteroskedasticity refers to an irregular pattern of variability in one or more variables. In the case of heteroskedasticity, observations tend to cluster rather than conform to a linear pattern.

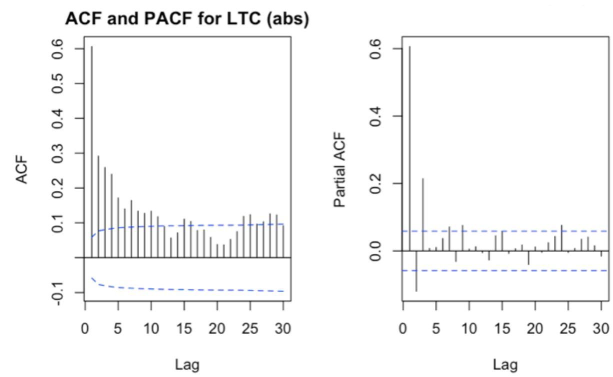
For this approach, this study considered the log-returns absolute value for each variable. The ACF and PACF of absolute log returns are shown below in Figure 11.

Figure 11: ACF and PACF of absolute log returns for each cryptocurrency



[20] Developed by Robert F. Engle in 1982

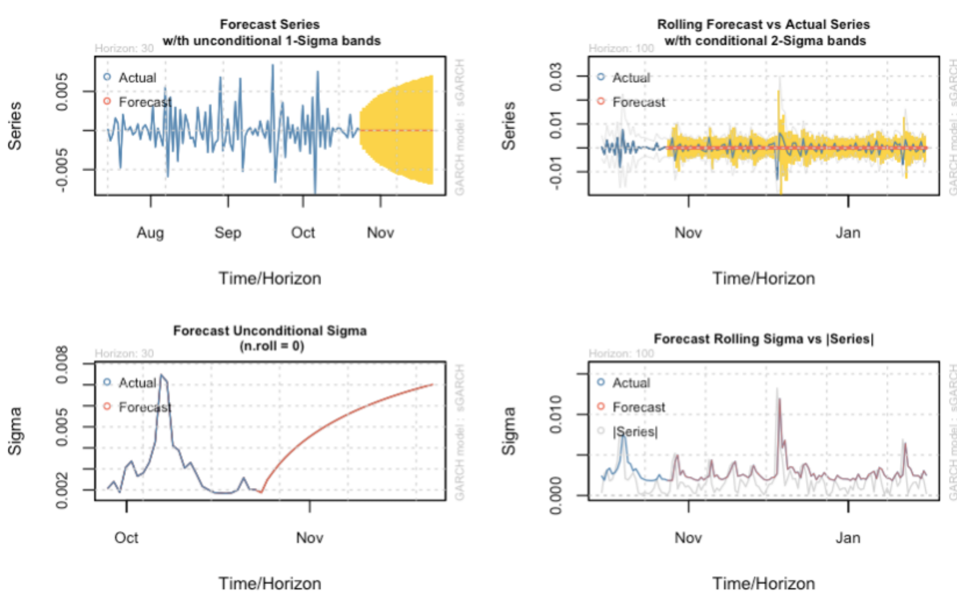




3.3.1 GARCH model order identification and forecasts

The preliminary order identification was conducted using EACF. The Jarque-Bera test performed on the preliminary order confirmed that skewness and kurtosis are not significantly different from their expected value for normal distribution. Thus, the distribution of model residuals is not statistically different from a normal distribution. Based on the AIC criterion I selected the optimal order to run the model integrating it with the distributions identified in the data analysis for each cryptocurrency. The GARCH model was employed to forecast the returns using a 30 days horizon. Figure 12 shows the forecasted results obtained by the GARCH model for BTC, the remaining variables' plots are listed in the appendix. From the plots, the forecasts provide relevant information regarding the rolling sigma forecast.

Figure 12: BTC return forecasts using GARCH



4. Methodology II: Technical Analysis

By analyzing price movement and volumes trends from trading activity, technical analysis determines the investment opportunities and evaluates investments[21]. Technical analysis focuses on prices and employs charts as primary tolls for analyzing investment opportunities. This analysis employs mathematical indicators used to forecast future trends derived from the previous trading and is based on the Dow Theory[22] developed by Charles Dow in 1900. Technical analysis is characterized by qualitative and quantitative elements. Respectively, the qualitative element refers to the use of charts and plots to identify whether the data follow a certain pattern. The quantitative part, instead, focuses on past information and concerns time-series analysis used to identify trading signals. The main difference between the two attributes refers to the level of objectiveness. The quantitative part is absolute, which means that every individual analyzing a time series would obtain the same conclusion. On the contrary, the qualitative attribute, leaves room for individual interpretation, creating a high level of subjectivity. Trading rules employed in technical analysis can be decomposed into five different classes. Moving averages rules are employed to check for breaks and are associated with the momentum effect; support-resistance rules are used to generate support or resistance bounds to analyze price movements; filter rules create trends able to tell the investor whether to buy or sell respectively when the price increases or decreases; oscillator rules used to check for overbought or oversold conditions and predict price reversion; finally, breakout rules refer also to the support-resistance rule except for taking into account changes in time.

For the technical analysis methodology, a Bollinger Band model was applied to trace price volatility using momentum indicators. As a second approach, this study included the Klinger Oscillator to check whether the market has experienced an oversold or overbought condition by measuring the trend of the flows.

[21] Lo, A.W., Mamaysky, H. and Wang, J., 2000. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation *The journal of finance*, *55*(4), pp.1705-1765.

[22] Dow Theory: considers the trending nature of prices and stems from the confirmation and divergence principles along with taking into account changes in prices. The market index related to the Dow Theory is the Dow Jones Industrial Average (DJIA).

4.1. Bollinger Bands Model, RSI, and Momentum

The Bollinger Bands is used in technical analysis to visualize volatility in prices and to generate buy or sell signals. Among the major benefits deriving from the implementation of the Bollinger Bands model, there is its capability to capture the stochastic nature of volatility. Specifically, through the model implementation, the bands' widths are automatically adapted to the respective volatility changes. Moreover, the model provides relevant information to investors regarding whether to take a long or short position when technically analyzing a stock. Specifically, if the stock's price fluctuates close to the lower band, the trader would be taking a long position. On the other hand, if the stock's price fluctuates close to the upper band, the investor would be taking a short position. The price's fluctuations close to the bands indicate an overbought or oversold condition. As a result of these conditions, the expectation is that, eventually, the price will start reverting to an average price. An additional advantage deriving from the model refers to the capability of capturing extreme events that could cause changes in volatility. However, this characteristic could also represent a limit because the model would capture only events with a high probability of happening (generally 95% of observations) and won't take into account events with a low probability of execution. The model can also be integrated with the Relative Strength Index[23] (RSI), momentum indicators, and Moving Averages to provide deeper knowledge to investors in decision making.

This approach employed the closing prices for each variable considered for the previous analysis with the date range stemming from January 1, 2019, to March 21, 2022. The outputs indicate many oscillations above and below 0, indicating the location of the price. The Momentum for BTC indicates an increase in fluctuations starting at the end of 2020 when the Pandemic shut down the economy and increased the fear of inflationary pressure on the USD. From this moment, the fluctuations kept increasing for the following periods reaching values above 5000 and -5000. The model identified an upper and lower bound respectively equal to 43,393.64 and 37,126.68. According to the model, a close price's fluctuation to the upper bound reflects the overbought market's condition. Otherwise, a close price's fluctuation to the lower bound reflects the oversold market's condition. For BTC, the overbought condition was recorded for the first months of 2021

[23] An indicator used in technical analysis to measure the magnitude of recent price changes to evaluate overbought or oversold conditions in the price of the cryptocurrency.

following the increase in momentum fluctuations while an oversold condition was recorded from May 2021 until July 2021 and at the end of 2021. The results obtained from the model are shown in Figure 13. The Bollinger Bands model appears along with the Relative Strength Index (RSI).

Figure 13: Bollinger Bands, RSI, and Momentum on BTC

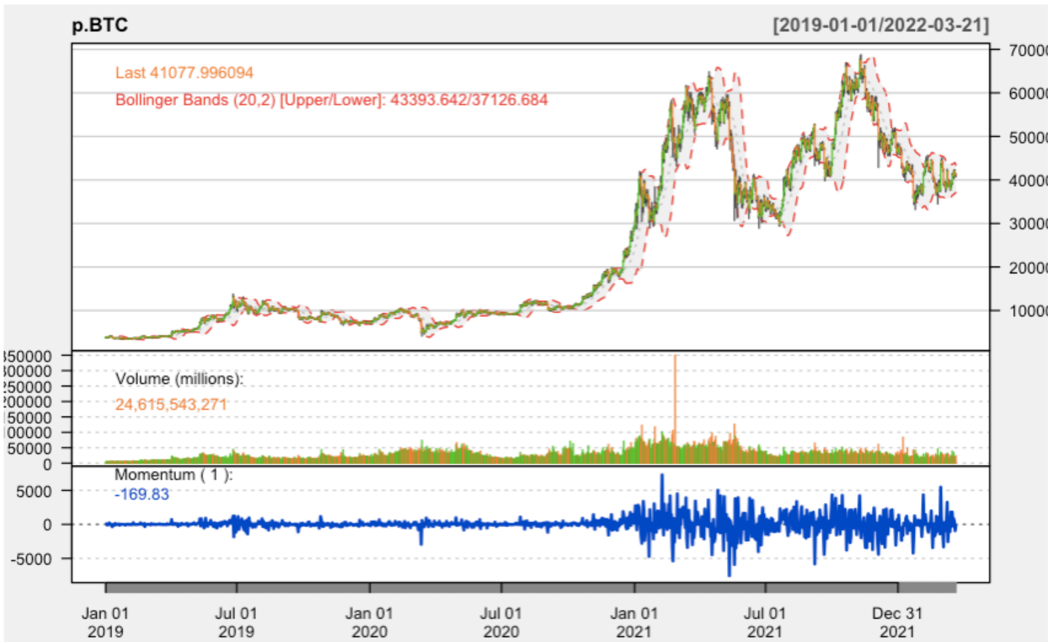


Figure 2 in the appendix shows Bollinger Band, RSI, and Momentum for all variables included in this study. From this figure, it is clear how the model provided different results for each variable. The model applied to BSV provides an upper and lower bound respectively of 86.33 and 73.63. The model identified major spikes in January 2020, January 2021, may 2021, and June 2021 indicating an extreme overbought condition. A spike beyond the lower bound was recorded in October 2021 reflecting an extreme oversold condition. The momentum indicates major fluctuations starting from January 2020 and for the first months of 2020 reaching -100 and 300. Regarding ETC, the model identified upper and lower bounds respectively equal to 35.42 and 21.62. A major overbought condition was recorded in June 2021. The momentum analysis found fluctuations starting from June 2021 ranging between -30 and 40. For ETC the upper and lower bounds are respectively 3,001.25 and 2,417.09, from the graph it is evident that the model identifies an overbought condition on May 2021 followed by an oversold condition on June 2021. The fluctuations started in January 2021 ranging between 500 and -1000 which reflects a major negative fluctuation recorded in June 2021. Concerning DOGE, the upper and lower bounds are extremely lower than the previous cryptocurrencies, reflecting the difference in prices of this variable. The bounds are respectively equal to 0.128 and 0.11. DOGE didn't provide any fluctuations before February 2021. Starting from that moment, the cryptocurrency alternated moments of overbought and oversold conditions. The momentum analysis found oscillations of 0.2 and below -0.1 mostly concentrated in the second trimester of 2021. Lastly, LTC presents upper and lower bounds equal to 118.51 and 96.72. The model provides insights regarding small fluctuations which started to increase from January 2021. Starting from this period, LTC alternated overbought and oversold conditions. The momentum analysis found oscillations ranging from 50 to -100.

4.2. Klinger Oscillator and Exponential Moving Average

As a second approach for technical analysis, this study employs the Klinger Oscillator[24] (KO). The KO analyzes flows to forecast trends for both short-term and long-term fluctuations. The indicator compares volumes and prices and is primarily used to predict price reversals. The indicator oscillates at a point of 0 and works on crossover and divergence. When using the Klinger Oscillator, two time periods are used in the analysis: the fast Exponential Moving Average (EMA) and the slow EMA. The EMA represents a type of Moving Average (MA) that gives, respectively, weight and significance to the most recent and less recent data points.

In this study, the Klinger Oscillator has been employed to check whether the market has experienced an overbought or oversold condition by measuring the trend of the flows for Bitcoin (BTC) and Ethereum (ETH)[25]. The data employed in this analysis are Inflows, Outflows, and Netflow.

The Fast EMA and Slow EMA are shown in Figure 14 which for BTC indicates a major spike above 0 in July 2019 and a major spike below 0 around November 2021.

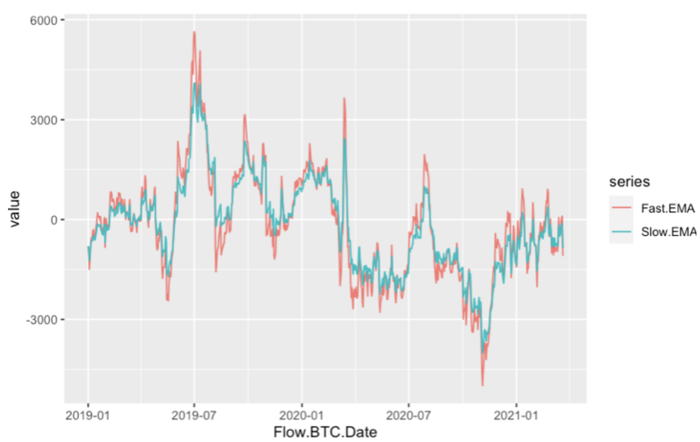
From this approach, is interesting to notice how the pattern changed starting from March 2020. Before this period, the majority of oscillations happened above 0, while after that period majority of oscillations happened below 0.

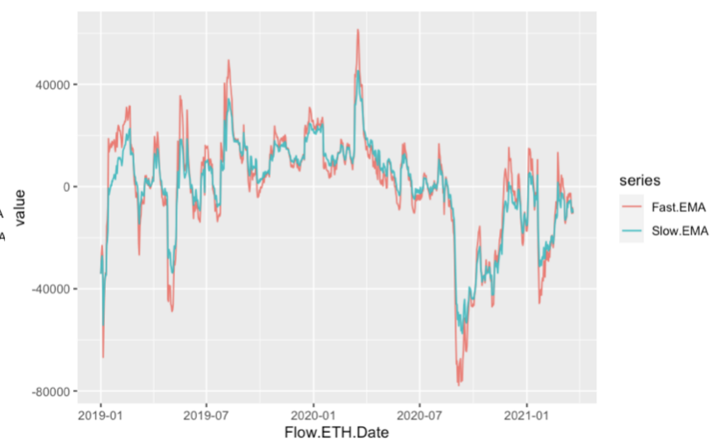
[24] Designed by Stephen Klinger in 1977.

[25] Data was downloaded from CryptoQuant.com only for BTC and ETH due to the absence of data for the remaining cryptocurrencies employed in this study.

For ETH, the fast and slow ema plot show a period of alternating oscillations above and below 0 starting from the first trimester of 2019 followed by a period of only values above 0 until June 2020. The opposite happened after that period where a major spike below 0 was recorded followed by alternating fluctuations mainly below 0.

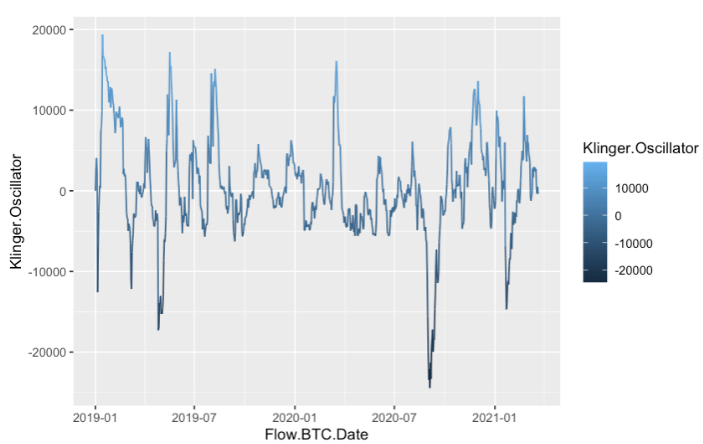
Figure 14: Fast and slow EMA for BTC and ETH

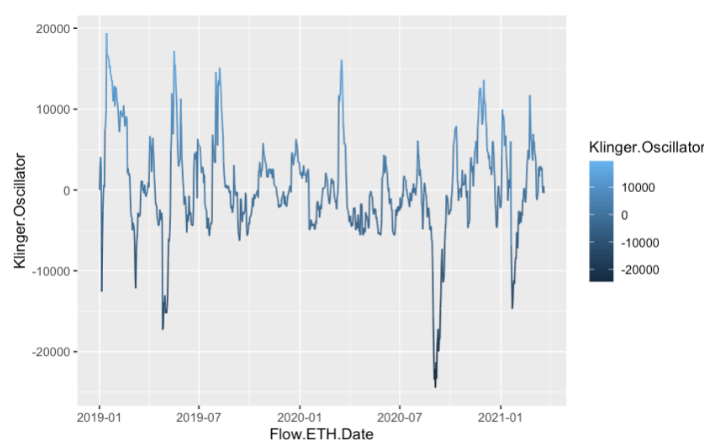




The Klinger Oscillator is derived from the difference between the fast EMA and the slow EMA. The overall objective of this indicator is to provide insights regarding the moment to buy and sell these cryptocurrencies. Traders could buy when the Klinger Oscillator moves above zero and sell when it moves below zero. For BTC a particularly positive period to buy happened around July 2019 followed by a less favorable period in the next few months. For ETH, an appropriate period to sell was September 2020.

Figure 15: Klinger Oscillator for BTC and ETH





5. Methodology III: Fundamental Analysis

Fundamental analysis is a type of analysis commonly used to measure the intrinsic value of a security. Fundamental analysis is usually performed on equities to determine their intrinsic value. However, the purpose of the fundamental analysis on cryptocurrencies is to reduce investor risks and evaluate the potential profit of the asset. This analysis employs the examination of financial and economic factors to obtain an objective measure of the asset's worth and assess whether the project is overvalued or undervalued.

Additionally, Blockchain metrics and financial metrics are the two key factors to consider when analyzing cryptocurrencies fundamentally. Specifically, the Hash rate and addresses analysis are the most important Blockchain metrics. The hash rate refers to the amount of computational power required to mine on the Proof-of-Work blockchain and is estimated from public data. However, it only represents an estimation since the actual hash rate is impossible to accurately know. Cryptocurrency hash rates serve as indicators of the health of the network since they motivate miners to mine for a profit. A high hash rate reflects a higher level of security in the network. On the contrary, miners may begin to find that mining cryptocurrency is unprofitable after the hash rate begins to decline, resulting in miner capitulation. This often occurs when the cryptocurrency market spirals lower, creating pressure for miners to sell off their hardware. Generally, a loss of interest by investors is indicated by a lower hash rate. The second metric refers to address analysis, in particular taking into consideration active, sending, and receiving addresses which measure the number of each address activity in the blockchain. By analyzing factors like market capitalization and volumes, financial metrics aim to understand the trading conditions of the assets. In this sense, fundamental analysis aims to produce quantitative value for investors.

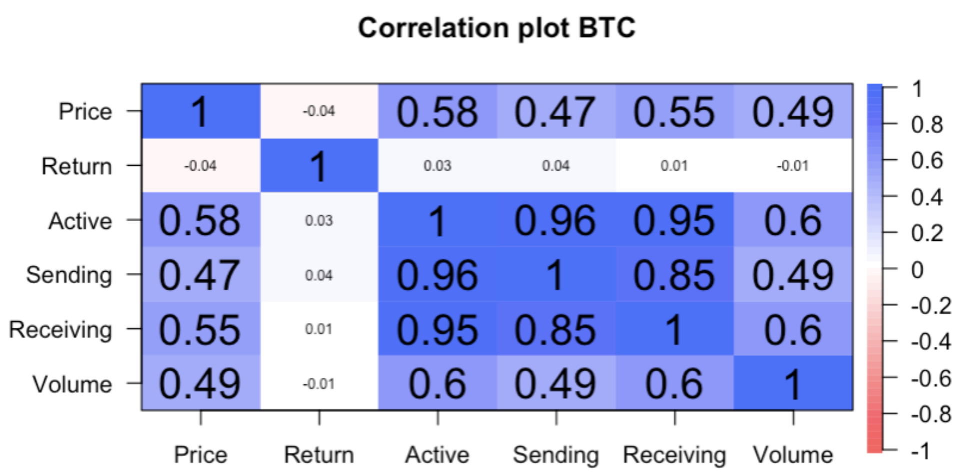
5.1 Correlation

As for Fundamental Analysis, the data[26] employed in this methodology are prices, returns, addresses[27], and volumes. The first approach was the correlation matrix among the data for each cryptocurrency. The correlation coefficient is a statistical measure used to identify how variables

[26] Downloaded from CryptoQuant.com for BTC, ETH, and LTC
[27] Addresses: Sending, Receiving, Active (New and Total for LTC)

move with one another. The correlation coefficient ranges between -1 and 1 which indicates respectively perfectly negative and perfectly positive correlation. In the case of perfect positive correlation, the variables will tend to move in the same direction and therefore, an increase (or decrease) in one variable will also cause an increase (or decrease) in the other variable taken into account. The opposite effect will result in the case of perfect negative correlation, which will result in an inversely proportional relationship among the variables. As opposed, a correlation coefficient equal to zero reflects the absence of a relationship among the variable. In this case, a change in one variable would not involve a change in the other variable. The correlation matrix for BTC is represented by Figure 16, which identifies that the correlation among addresses is high and positive, while the correlation among prices, addresses, along with volumes seems still positive but smaller. Considering instead the returns, the conclusion reflects a very low correlation not significantly different from zero.

Figure 16: Correlation matric for BTC

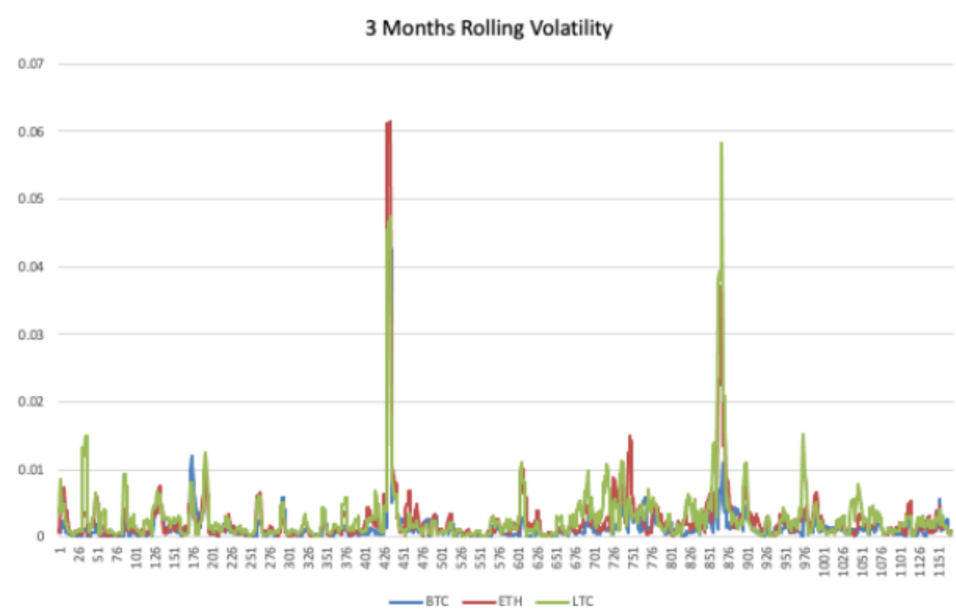


The same pattern happens for ETH with a difference concerning the correlation of prices and addresses which appear to be higher. For LTC, the pattern is also similar, with a difference regarding volumes which is negatively correlated with all variables. The negative correlation is due to the different nature of LTC compared to the other cryptocurrencies. The correlation matrices of ETH and LTC are shown in Figure 3 in the Appendix.

5.2 Rolling Volatility

The second approach this study employed for Fundamental Analysis is the rolling volatility calculation of returns through rolling standard deviations considering 3 months and 6 months windows. The windows represent rolling volatility periods of returns which reflect a standard deviation value for each period. The volatility obtained for each period was used to create a graph to better understand the volatility trend. The results considering the three months window provide evidence regarding the volatile nature of the variables underlying periods of high volatility. From the graph, the variables seem to follow the same behavior, however, ETH and LTC are characterized by higher rolling volatility compared to BTC. The results are shown below in Figure 17. The same conclusion is obtained considering the six months window. In fact, in this case, periods of higher volatility were detected, identifying the same conclusions regarding the level of rolling volatility obtained for each cryptocurrency.

Figure 17: Rolling Volatility with 3 months window for BTC, ETH, and LTC



5.3 Rolling Correlation

Concerning the last approach for Fundamental Analysis on cryptocurrency, this study employed the correlation between time series over time. Specifically, this approach concerns the calculation of the rolling correlation among prices and volumes along with returns and volumes using the same windows as for rolling volatility. For BTC the correlation between prices and volumes during months 1 through month 3 was 0.96. This result reflects a high positive correlation. Despite this result, the correlation during months 7 through month 9 became negative and remained negative also during the next periods taken into account, specifically, during months 8 through 10 and during months 9 through 11. Concerning the correlation of returns and volumes, during months 1 through 3, there has been a high negative correlation which remained negative also during months 2 and 4. Overall, the majority of rolling correlations among returns and volumes have been negative for BTC.

Concerning ETH, the time series is characterized by periods of positive and negative correlation with an overall majority of positive rolling correlation between prices and volumes. The same pattern regards the rolling correlation between returns and volumes. However, in this case, the overall majority reflects a negative correlation also confirmed by the correlation matrix computed in the previous approach.

Lastly, regarding LTC, the rolling correlation among prices and volumes and between prices and returns follows the same pattern as for ETH.

Conclusion

As a result of implementing these analyses on cryptocurrencies, it is evident how different analyses provide different useful results to various types of investors to make informed decisions. Time-series analysis is useful for both fundamental and technical analysis. Therefore, it represents a valid analysis. Even well-established cryptocurrencies such as Bitcoin and Ethereum can experience sudden volatility due to the high volatility of the cryptocurrency market. When non-technical investors perform fundamental analysis, they can trade market movements confidently and develop profitable strategies. However, traders and investors who are interested in forecasting the direction of the financial market using technical indicators will find the technical analysis useful. Even though fundamental analysis, despite being a useful tool to analyze the market, can be applied to cryptocurrencies, it provides insufficient information for investors and traders to develop trading strategies and make informed decisions. It is possible to determine favorable and unfavorable conditions to buy or sell cryptocurrencies using technical indicators and momentum strategies. By combining technical indicators and time-series analysis, this approach is best suited to analyzing specific assets such as cryptocurrencies. To predict a cryptocurrency price's low and high within different periods, technical analysis provides the investor with more information, allowing them to make a better-informed decision about buying at a favorable price and selling at a profit. In conclusion, concerning cryptocurrencies, technical analysis is generally more appropriate, however, the combination with fundamental analysis can provide deeper insights into the world of cryptocurrencies.

References

Bovaird, C. (2016, June 24). Bitcoin Rollercoaster Rides Brexit As Ether Price Holds Amid DAO Debacle. Retrieved June 2016, from CoinDesk Website: http://www.coindesk.com/bitcoin-brexit-ether-price- rollercoaster/

Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

Price, R. (2016, June 17). Digital currency Ethereum is cratering because of a $50 million hack. Retrieved from Business Insider Website: http://www.businessinsider.com/dao-hacked-ethereum-crashing-in-value-tens- of-millions-allegedly-stolen-2016-6.

Baur, D. G., K. Hong, and A. D. Lee (2018). Bitcoin: Medium of exchange or speculative assets? Journal of International Financial Markets, Institutions and Money 54, 177–189.

King, R. S. (2013, December 17). By reading this article, you're mining bitcoins. Retrieved from Quartz.com Website: http://qz.com/154877/by-reading-this-page-you-are-mining-bitcoins/.

Corbet, S., A. Meegan, C. Larkin, B. Lucey, and L. Yarovaya (2018). Exploring the dynamic relationships between cryptocurrencies and other financial assets. Economics Letters 165, 28– 34.

Shi, N. A new proof-of-work mechanism for bitcoin. *Financ Innov* 2, 31 (2016)

Magro, P. (2016, July 16). What Greece can learn from bitcoin adoption in Latin America. Retrieved July 2016, from International Business Times Website: http://www.ibtimes.co.uk/what-greece-can-learn-bitcoin- adoption-latin-america-1511183

Desjardins, J. (2016, January 5). It's Official: Bitcoin was the Top Performing Currency of 2015. Retrieved from The Money Project Website: http://money.visualcapitalist.com/its-official-bitcoin-was-the-top- performing-currency-of-2015/

DeVries, P.D., 2016. An analysis of cryptocurrency, bitcoin, and the future. *International Journal of Business Management and Commerce*, *1*(2), pp.1-9

Chen, S., Min, W. and Chen, R., 2013. Model identification for time series with dependent innovations. *Statistica Sinica*, pp.873-899.

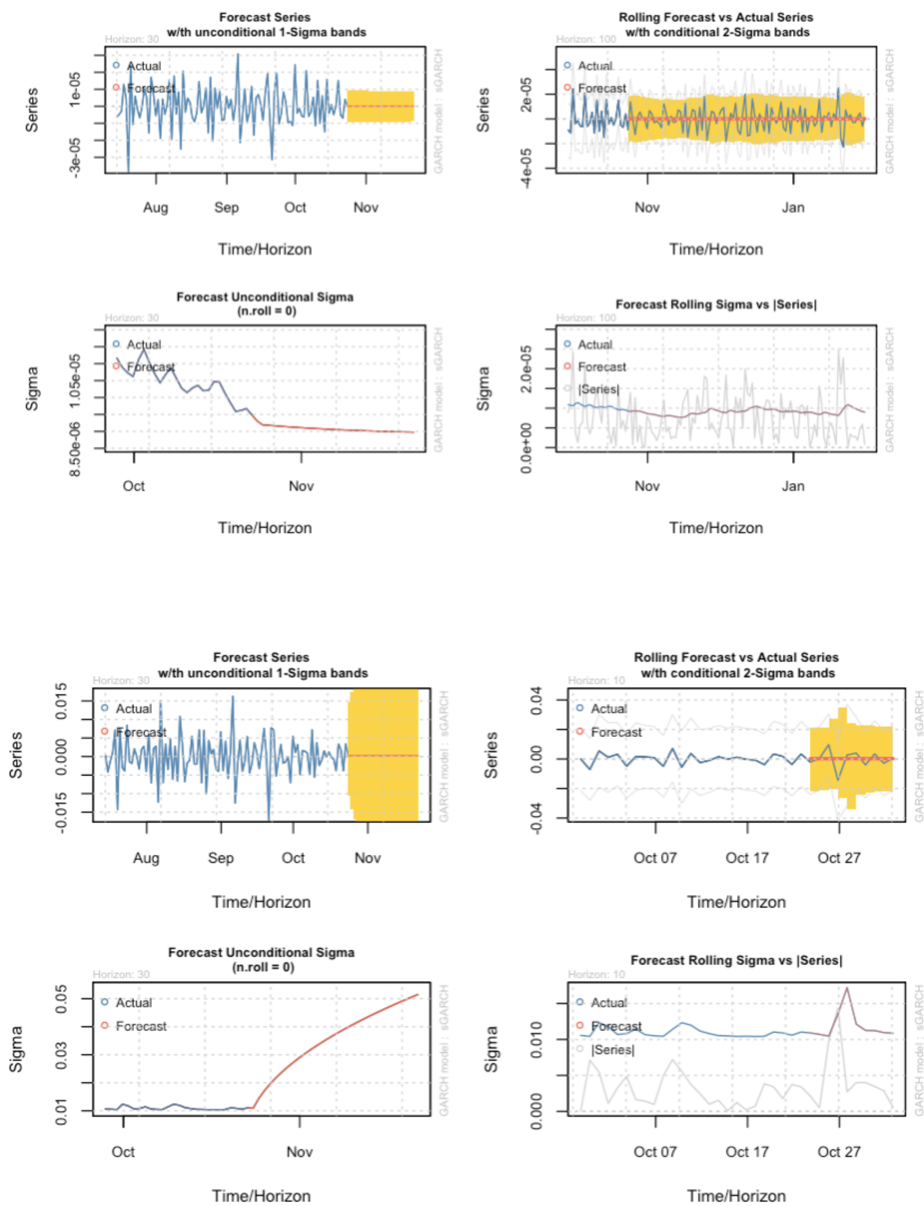
Lo, A.W., Mamaysky, H. and Wang, J., 2000. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation *The journal of finance*, *55*(4), pp.1705-1765

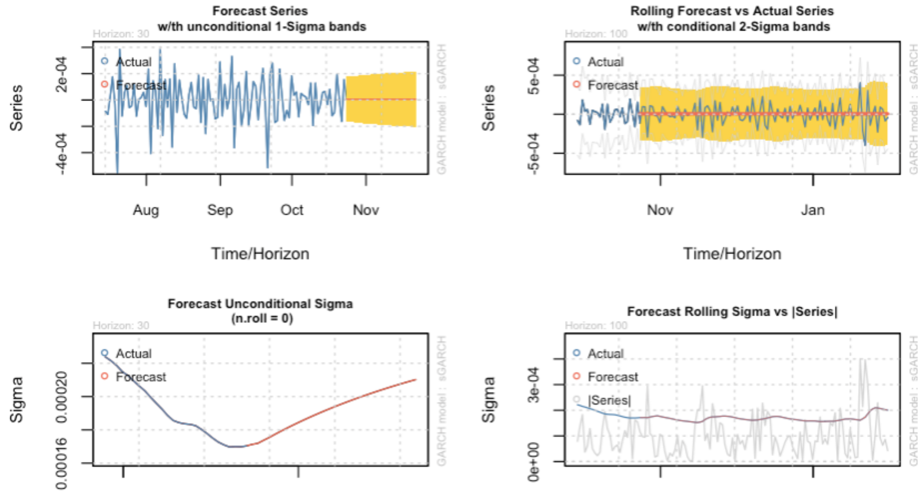
Pinakin, S.N. and Manubhai, P.T., 2015. A comparative study on technical analysis by Bollinger Band and RSI. *International Journal in Management & Social Science*, *3*(6), pp.234-251.

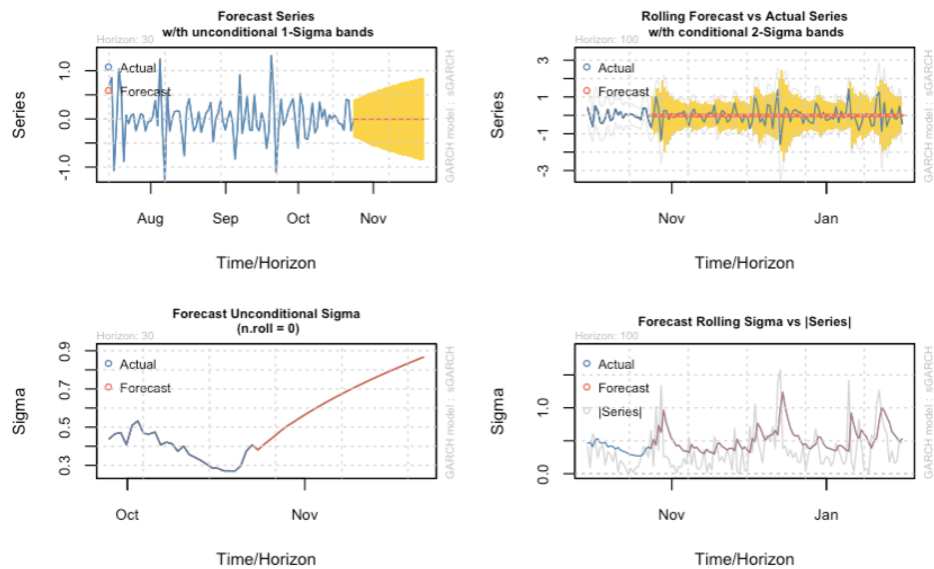
Lehtoalho, J., 2018. Optimizing Bollinger band parameters: Individual stock and portfolio approach.

Appendix

Figure 1: Returns forecasts using GARCH for BSV, ETC, ETH, DOGE, and LTC




Forecast Series
w/th unconditional 1-Sigma bands
Horizon: 30
Actual
Forecast
Series
2e-04
-4e-04
Aug
Sep
Oct
Nov
Time/Horizon
GARCH model : sGARCH
Rolling Forecast vs Actual Series
w/th conditional 2-Sigma bands
Horizon: 100
Actual
Forecast
Series
5e-04
-5e-04
Nov
Jan
Time/Horizon
GARCH model : sGARCH
Forecast Unconditional Sigma
(n.roll = 0)
Horizon: 30
Actual
Forecast
Sigma
0.00020
0.00016
Oct
Nov
GARCH model : sGARCH
Forecast Rolling Sigma vs |Series|
Horizon: 100
Actual
Forecast
|Series|
Sigma
3e-04
0e+00
Nov
Jan
GARCH model : sGARCH


Forecast Series
w/th unconditional 1-Sigma bands
Horizon: 30
Actual
Forecast
Series
1.0
0.0
-1.0
Aug
Sep
Oct
Nov
Time/Horizon
GARCH model : sGARCH
Rolling Forecast vs Actual Series
w/th conditional 2-Sigma bands
Horizon: 100
Actual
Forecast
Series
3
1
-1
-3
Nov
Jan
Time/Horizon
GARCH model : sGARCH
Forecast Unconditional Sigma
(n.roll = 0)
Horizon: 30
Actual
Forecast
Sigma
0.9
0.7
0.5
0.3
Oct
Nov
Time/Horizon
GARCH model : sGARCH
Forecast Rolling Sigma vs |Series|
Horizon: 100
Actual
Forecast
|Series|
Sigma
1.0
0.0
Nov
Jan
Time/Horizon
GARCH model : sGARCH

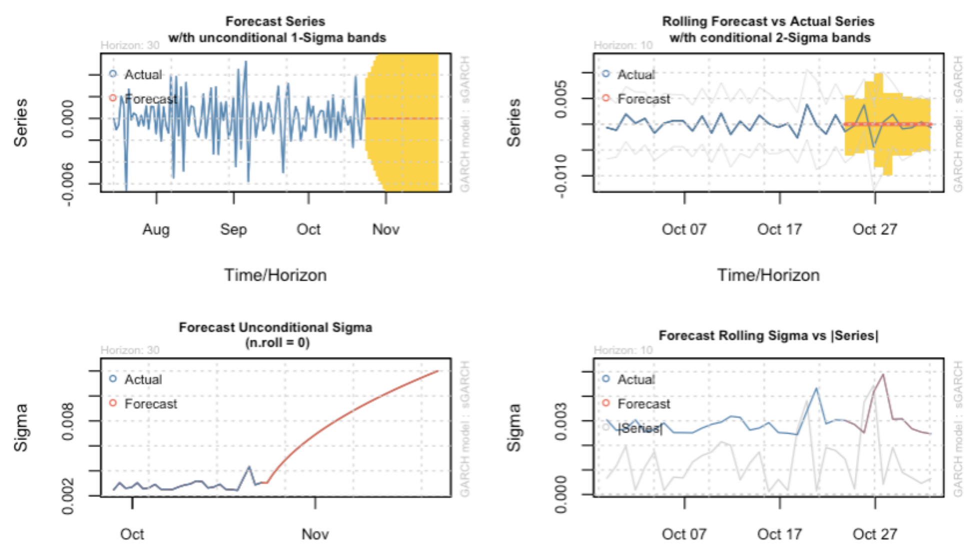
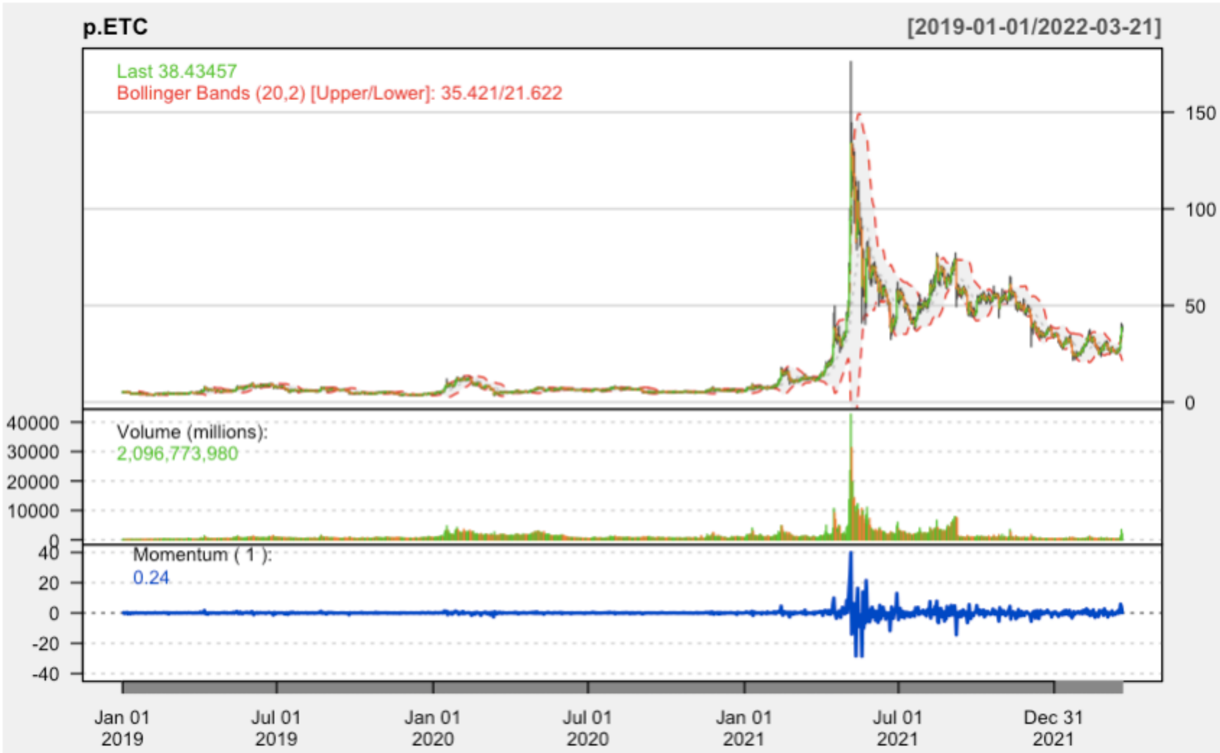
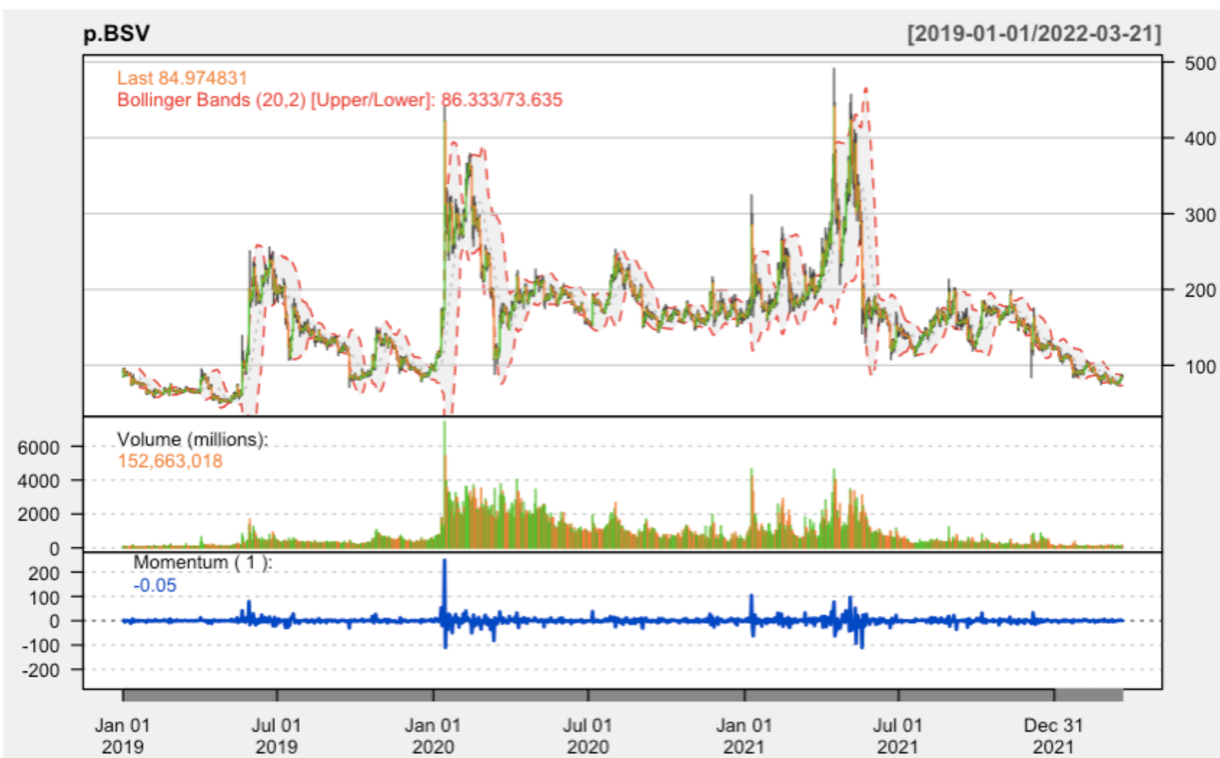
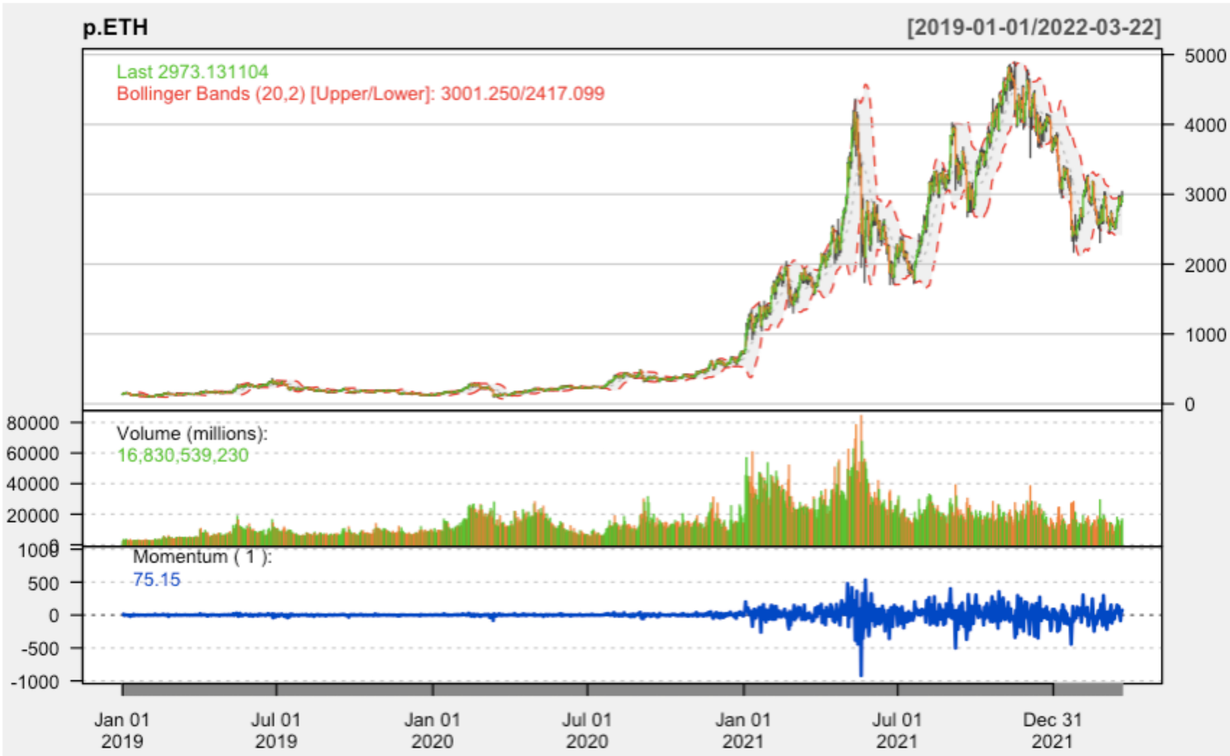

Forecast Series
w/th unconditional 1-Sigma bands
Horizon: 30
Actual
Forecast
Series
0.000
-0.006
Aug
Sep
Oct
Nov
Time/Horizon
GARCH model : sGARCH
Rolling Forecast vs Actual Series
w/th conditional 2-Sigma bands
Horizon: 10
Actual
Forecast
Series
0.005
-0.010
Oct 07
Oct 17
Oct 27
Time/Horizon
GARCH model : sGARCH
Forecast Unconditional Sigma
(n.roll = 0)
Horizon: 30
Actual
Forecast
Sigma
0.008
0.002
Oct
Nov
GARCH model : sGARCH
Forecast Rolling Sigma vs |Series|
Horizon: 10
Actual
Forecast
|Series|
Sigma
0.003
0.000
Oct 07
Oct 17
Oct 27
GARCH model : sGARCH

Figure 2: Bollinger Bands, RSI, and Momentum for each cryptocurrency



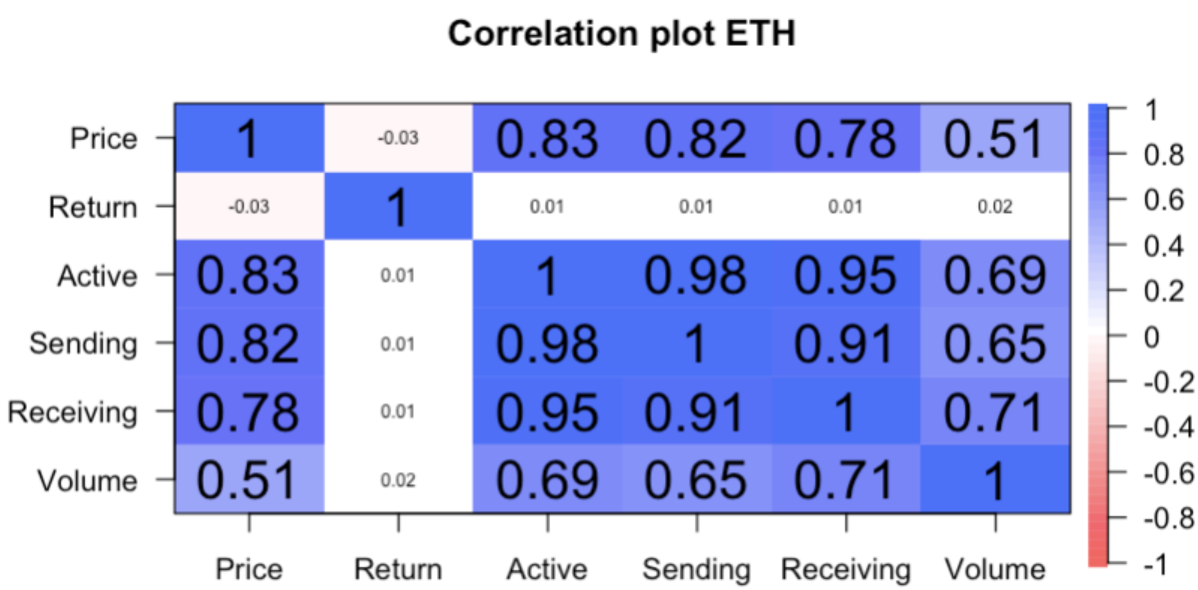


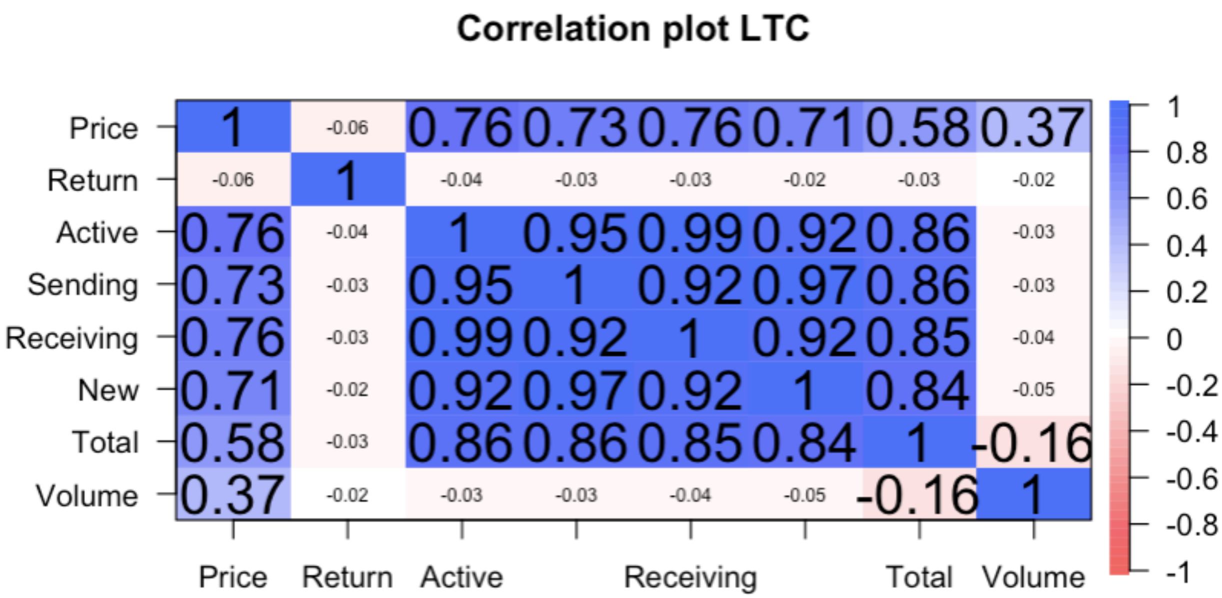


p.ETH
[2019-01-01/2022-03-22]
Last 2973.131104
Bollinger Bands (20,2) [Upper/Lower]: 3001.250/2417.099
Volume (millions):
16,830,539,230
Momentum (1):
75.15

p.DOGE
[2019-01-01/2022-03-22]
Last 0.122481
Bollinger Bands (20,2) [Upper/Lower]: 0.128/0.111
Volume (millions):
610,507,111
Momentum (1):
0.00

Figure 3: Correlation matrix for ETH and LTC




Correlation plot LTC
Price
Return
Active
Sending
Receiving
New
Total
Volume
1 -0.06 0.76 0.73 0.76 0.71 0.58 0.37
-0.06 1 -0.04 -0.03 -0.03 -0.02 -0.03 -0.02
0.76 -0.04 1 0.95 0.99 0.92 0.86 -0.03
0.73 -0.03 0.95 1 0.92 0.97 0.86 -0.03
0.76 -0.03 0.99 0.92 1 0.92 0.85 -0.04
0.71 -0.02 0.92 0.97 0.92 1 0.84 -0.05
0.58 -0.03 0.86 0.86 0.85 0.84 1 -0.16
0.37 -0.02 -0.03 -0.03 -0.04 -0.05 -0.16 1
Price
Return
Active
Receiving
Total
Volume
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

Final Project Beatrice Busato

Initial Data

```
library(lubridate)
library(quantmod)
library(dplyr)
library(readxl)
library(writexl)
library(ggplot2)
library(magrittr)
library(broom)

P <- read_excel("/Users/beatricebusato/Desktop/FA 800/Data/crypto.xlsx")
head(P)
```

```
## # A tibble: 6 x 7
##   Date                  BTC   BSV   ETC   ETH    DOGE   LTC
##   <dttm>              <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl>
## 1 2019-01-01 00:00:00 3844.  92.1  5.23  141. 0.00239  32.0
## 2 2019-01-02 00:00:00 3943.  94.9  5.44  155. 0.00241  33.4
## 3 2019-01-03 00:00:00 3837.  88.6  5.09  149. 0.00236  32.0
## 4 2019-01-04 00:00:00 3858.  87.6  5.25  155. 0.00231  32.4
## 5 2019-01-05 00:00:00 3845.  88.5  5.14  156. 0.00232  34.9
## 6 2019-01-06 00:00:00 4077.  90.4  5.5   158. 0.00233  39.3
```

Returns and descriptive statistics

```
R.BTC <- na.omit(log(P$BTC)/lag(P$BTC))
R.BSV <- na.omit(log(P$BSV)/lag(P$BSV))
R.ETC <- na.omit(log(P$ETC)/lag(P$ETC))
R.ETH <- na.omit(log(P$ETH)/lag(P$ETH))
R.DOGE <- na.omit(log(P$DOGE)/lag(P$DOGE))
R.LTC <- na.omit(log(P$LTC)/lag(P$LTC))
Date <- P$Date[2:1129]
R <- data.frame(Date, BTC = R.BTC, BSV = R.BSV, ETC = R.ETC, ETH = R.ETH,
                DOGE = R.DOGE, LTC = R.LTC)
head(R)
```

```
##         Date         BTC        BSV       ETC        ETH      DOGE       LTC
## 1 2019-01-02 0.002154224 0.04940973 0.3238583 0.03581698 -2520.641 0.1097390
## 2 2019-01-03 0.002092701 0.04726584 0.2991320 0.03227917 -2513.126 0.1036995
## 3 2019-01-04 0.002152304 0.05046651 0.3257815 0.03379852 -2571.333 0.1085906
## 4 2019-01-05 0.002139756 0.05114921 0.3118196 0.03265329 -2620.570 0.1096800
## 5 2019-01-06 0.002161929 0.05088142 0.3316631 0.03251742 -2614.379 0.1050723
## 6 2019-01-07 0.002036079 0.04963666 0.2896925 0.03183458 -2612.281 0.0924319
```

Mean, Standard Deviation, Variance, Skewness, and Kurtosis of Returns

```
library(moments)
```

```
mean.BTC <- mean(R.BTC)
mean.BSV <- mean(R.BSV)
mean.ETC <- mean(R.ETC)
mean.ETH <- mean(R.ETH)
mean.DOGE <- mean(R.DOGE)
mean.LTC <- mean(R.LTC)
means <- c(mean.BTC,mean.BSV, mean.ETC, mean.ETH, mean.DOGE, mean.LTC)

sd.BTC <- sd(R.BTC)
sd.BSV<- sd(R.BSV)
sd.ETC<- sd(R.ETC)
sd.ETH<- sd(R.ETH)
sd.DOGE<- sd(R.DOGE)
sd.LTC<- sd(R.LTC)
sd <- c(sd.BTC,sd.BSV, sd.ETC, sd.ETH, sd.DOGE, sd.LTC)

var.BTC <- var(R.BTC)
var.BSV <- var(R.BSV)
var.ETC <- var(R.ETC)
var.ETH <- var(R.ETH)
var.DOGE <- var(R.DOGE)
var.LTC <- var(R.LTC)
var <- c(var.BTC,var.BSV, var.ETC, var.ETH, var.DOGE, var.LTC)

s.BTC <- skewness(R.BTC)
s.BSV <- skewness(R.BSV)
s.ETC <- skewness(R.ETC)
s.ETH <- skewness(R.ETH)
s.DOGE <- skewness(R.DOGE)
s.LTC <- skewness(R.LTC)

k.BTC <- kurtosis(R.BTC)
k.BSV <- kurtosis(R.BSV)
k.ETC <- kurtosis(R.ETC)
k.ETH <- kurtosis(R.ETH)
k.DOGE <- kurtosis(R.DOGE)
k.LTC <- kurtosis(R.LTC)

D <- data.frame(Mean = c(mean.BTC, mean.BSV, mean.ETC,mean.ETH, mean.DOGE,
                         mean.LTC), St.Dev = c(sd.BTC,sd.BSV,sd.ETC,sd.ETH,
                                               sd.DOGE, sd.LTC),
                Variance = c(var.BTC, var.BSV, var.ETC, var.ETH, var.DOGE,
                             var.LTC), Skewness = c(s.BTC, s.BSV, s.ETC, s.ETH,
                                                    s.DOGE, s.LTC),
                Kurtosis = c(k.BTC, k.BSV, k.ETC, k.ETH, k.DOGE, k.LTC))
rownames(D) <- c("BTC","BSV","ETC","ETH", "DOGE", "LTC")
```

Two-Sample Kolmogorov-Smirnov test

Normal Distribution

```
test.normal.BTC <- rnorm(R.BTC)
tBTC <- ks.test(R.BTC, test.normal.BTC)

test.normal.BSV <- rnorm(R.BSV)
```

```
tBSV <- ks.test(R.BSV, test.normal.BSV)

test.normal.ETC <- rnorm(R.ETC)
tETC <- ks.test(R.ETC, test.normal.ETC)

test.normal.ETH <- rnorm(R.ETH)
tETH <- ks.test(R.ETH, test.normal.ETH)

test.normal.DOGE <- rnorm(R.DOGE)
tDOGE <- ks.test(R.DOGE, test.normal.DOGE)

test.normal.LTC <- rnorm(R.LTC)
tLTC <- ks.test(R.LTC, test.normal.LTC)

normal.distribution <- data.frame(BTC = 0.51596, BSV = 0.49379, ETC = 0.51684,
                                  ETH = 0.49025, DOGE = 0.97606, LTC = 0.52128)
rownames(normal.distribution) <- "Normal Distribution"
```

T-Student Distribution

```
test.student <- rt(length(R.BTC), length(R.BTC) -1)
tBTC <- ks.test(R.BTC, test.student)

test.student.BSV <- rt(length(R.BSV), length(R.BSV) -1)
tBSV <- ks.test(R.BSV, test.student.BSV)

test.student.ETC <- rt(length(R.ETC), length(R.ETC) -1)
tETC <- ks.test(R.ETC, test.student.ETC)

test.student.ETH <- rt(length(R.ETH), length(R.ETH) -1)
tETH <- ks.test(R.ETH, test.student.ETH)

test.student.DOGE <- rt(length(R.DOGE), length(R.DOGE) -1)
tDOGE <- ks.test(R.DOGE, test.student.DOGE)

test.student.LTC <- rt(length(R.LTC), length(R.LTC) -1)
tLTC <- ks.test(R.LTC, test.student.LTC)

tstudent.distribution <- data.frame(BTC = 0.53546, BSV = 0.52394, ETC = 0.48493,
                                    ETH = 0.50532, DOGE = 0.97606,
                                    LTC = 0.50266)
rownames(tstudent.distribution) <- "T-Student Distribution"
```

Laplace Distribution

```
library(jmuOutlier)

test.laplace <- rlaplace(R.BTC)
tBTC <- ks.test(R.BTC, test.laplace)

test.laplace.BSV <- rlaplace(R.BSV)
tBSV <- ks.test(R.BSV, test.laplace.BSV)

test.laplace.ETC <- rlaplace(R.ETC)
tETC <- ks.test(R.ETC, test.laplace.ETC)
```

```
test.laplace.ETH <- rlaplace(R.ETH)
tETH <- ks.test(R.ETH, test.laplace.ETH)

test.laplace.DOGE <- rlaplace(R.DOGE)
tDOGE <- ks.test(R.DOGE, test.laplace.DOGE)

test.laplace.LTC <- rlaplace(R.LTC)
tLTC <- ks.test(R.LTC, test.laplace.LTC)

laplace.distribution <- data.frame(BTC = 0.50887, BSV = 0.50975, ETC = 0.5328,
                                   ETH = 0.50841, DOGE = 0.96277, LTC = 0.5328)
rownames(laplace.distribution) <- "Laplace Distribution"
```

T-sallis Distribution

```
library(tsallisqexp)

test.ts <- rtsal(R.BTC, mean.BTC)
tBTC <- ks.test(R.BTC, test.ts)

test.ts.BSV <- rtsal(R.BSV, mean.BSV)
tBSV <- ks.test(R.BSV, test.ts.BSV)

test.ts.ETC <- rtsal(R.ETC, mean.ETC)
tETC <- ks.test(R.ETC, test.ts.ETC)

test.ts.ETH <- rtsal(R.ETH, mean.ETH)
tETH <- ks.test(R.ETH, test.ts.ETH)

test.ts.DOGE <- rtsal(R.DOGE, mean.DOGE)
tDOGE <- ks.test(R.DOGE, test.ts.DOGE)

test.ts.LTC <- rtsal(R.LTC, mean.LTC)
tLTC <- ks.test(R.LTC, test.ts.LTC)

tsallis.distribution <- data.frame(BTC = 1, BSV = 0.99645, ETC = 0.9344,
                                   ETH = 0.99911, DOGE = 1, LTC = 0.99557)
rownames(tsallis.distribution) <- "T-sallis Distribution"
```

Distribution Comparison

```
D <- data.frame(BTC = c(normal.distribution$BTC,
                        tstudent.distribution$BTC, laplace.distribution$BTC,
tsallis.distribution$BTC),
                BSV = c(normal.distribution$BSV, tstudent.distribution$BSV,
laplace.distribution$BSV, tsallis.distribution$BSV),
                ETC = c(normal.distribution$ETC, tstudent.distribution$ETC,
laplace.distribution$ETC, tsallis.distribution$ETC),
                ETH = c(normal.distribution$ETH, tstudent.distribution$ETH,
laplace.distribution$ETH, tsallis.distribution$ETH),
                DOGE = c(normal.distribution$DOGE, tstudent.distribution$DOGE,
laplace.distribution$DOGE, tsallis.distribution$DOGE),
                LTC = c(normal.distribution$LTC, tstudent.distribution$LTC,
laplace.distribution$LTC, tsallis.distribution$LTC))
```

```r
rownames(D) <- c("Normal Distribution","T-Student Distribution",
                 "Laplace Distribution","T-sallis Distribution")
```

Linear Regression

```r
lr <- lm(R$BTC ~ R$BSV + R$ETC + R$ETH + R$DOGE + R$LTC, data = R)

lr.1 <- lm(R$BSV ~ R$BTC + R$ETC + R$ETH + R$DOGE + R$LTC, data = R)

lr.2 <- lm(R$ETC ~ R$BTC + R$BSV + R$ETH + R$DOGE + R$LTC, data = R)

lr.3 <- lm(R$ETH ~ R$BTC + R$BSV + R$ETC + R$DOGE + R$LTC, data = R)
summary(lr.3)
```

```
##
## Call:
## lm(formula = R$ETH ~ R$BTC + R$BSV + R$ETC + R$DOGE + R$LTC,
##     data = R)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.007533 -0.001252 -0.000070  0.001575  0.009284
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.498e-04  4.316e-04   0.579 0.562846
## R$BTC        8.285e+00  4.485e-01  18.475  < 2e-16 ***
## R$BSV        4.350e-02  1.062e-02   4.094 4.54e-05 ***
## R$ETC       -9.542e-05  1.952e-03  -0.049 0.961028
## R$DOGE      -7.143e-06  2.762e-07 -25.866  < 2e-16 ***
## R$LTC       -3.007e-02  8.281e-03  -3.631 0.000296 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003043 on 1122 degrees of freedom
## Multiple R-squared:  0.9432, Adjusted R-squared:  0.9429
## F-statistic:  3725 on 5 and 1122 DF,  p-value: < 2.2e-16
```

```r
lr.4 <- lm(R$DOGE ~ R$BTC + R$BSV + R$ETC + R$ETH + R$LTC, data = R)

lr.5 <- lm(R$LTC ~ R$BTC + R$BSV + R$ETC + R$ETH + R$DOGE, data = R)
```

Time Series Analysis

Time Series Returns plots

```r
ts.BTC <- ts(as.vector(R.BTC), start= 2019,frequency = 365)

ts.BSV <- ts(as.vector(R.BSV), start= 2019,frequency = 365)

ts.ETC <- ts(as.vector(R.ETC), start= 2019,frequency = 365)

ts.ETH <- ts(as.vector(R.ETH), start= 2019,frequency = 365)

ts.DOGE <- ts(as.vector(R.DOGE), start= 2019,frequency = 365)
```

```
ts.LTC <- ts(as.vector(R.LTC), start= 2019,frequency = 365)
```

ARIMA Model

Augmented Dickey Fuller Test H0: the time series is non-stationary H1: the time series is stationary

```
library(kableExtra)
library(tidyverse)
library(tseries)
library(forecast)

R <- ts(R)

adf.BTC <- adf.test(R.BTC)
adf.BSV <- adf.test(R.BSV)
adf.ETC <- adf.test(R.ETC)
adf.ETH <- adf.test(R.ETH)
adf.DOGE <- adf.test(R.DOGE)
adf.LTC <- adf.test(R.LTC)

R.BTC <- diff(R.BTC)
R.BSV <- diff(R.BSV)
R.ETC <- diff(R.ETC)
R.ETH <- diff(R.ETH)
R.DOGE <- diff(R.DOGE)
R.LTC <- diff(R.LTC)

adf.BTC <- adf.test(R.BTC)
adf.BSV <- adf.test(R.BSV)
adf.ETC <- adf.test(R.ETC)
adf.ETH <- adf.test(R.ETH)
adf.DOGE <- adf.test(R.DOGE)
adf.LTC <- adf.test(R.LTC)
```

ARIMA Model identification for BTC

```
library(lmtest)
library(TSA)

m1.BTC <- arima(R.BTC, order =c(1,1,2), method='ML')
aic.BTC.1 <- m1.BTC$aic

m2.BTC <- arima(R.BTC, order =c(1,1,3), method='ML')
aic.BTC.2 <- m2.BTC$aic

m3.BTC <- arima(R.BTC, order =c(1,1,4), method='ML')
aic.BTC.3 <- m3.BTC$aic

m4.BTC <- arima(R.BTC, order =c(1,1,5), method='ML')
aic.BTC.4 <- m4.BTC$aic

m5.BTC <- arima(R.BTC, order = c(1,1,6), method='ML')
aic.BTC.5 <- m5.BTC$aic

m6.BTC <- arima(R.BTC, order = c(2,1,3), method='ML')
```

```
aic.BTC.6 <- m6.BTC$aic
```

ARIMA Model Identification for BSV

```
m1.BSV <- arima(R.BSV, order =c(1,1,3), method='ML')
aic.BSV.1 <- m1.BSV$aic

m2.BSV <- arima(R.BSV, order =c(1,1,4), method='ML')
aic.BSV.2 <- m2.BSV$aic

m3.BSV <- arima(R.BSV, order = c(1,1,5), method='ML')
aic.BSV.3 <- m3.BSV$aic

m4.BSV <- arima(R.BSV, order = c(1,1,6), method='ML')
aic.BSV.4 <- m4.BSV$aic

m5.BSV <- arima(R.BSV, order = c(2,1,3), method='ML')
aic.BSV.5 <- m5.BSV$aic
```

ARIMA Model Identification for ETC

```
m1.ETC <- arima(R.ETC, order =c(1,1,3), method='ML')
aic.ETC.1 <- m1.ETC$aic

m2.ETC <- arima(R.ETC, order =c(1,1,4), method='ML')
aic.ETC.2 <- m2.ETC$aic

m3.ETC <- arima(R.ETC, order = c(1,1,5), method='ML')
aic.ETC.3 <- m3.ETC$aic

m4.ETC <- arima(R.ETC, order = c(1,1,6), method='ML')
aic.ETC.4 <- m4.ETC$aic

m5.ETC <- arima(R.ETC, order = c(1,1,7), method='ML')
aic.ETC.5 <- m5.ETC$aic
```

ARIMA Model Identification for ETH

```
m1.ETH <- arima(R.ETH, order =c(0,1,11), method='ML')
aic.ETH.1 <- m1.ETH$aic

m2.ETH <- arima(R.ETH, order =c(1,1,5), method='ML')
aic.ETH.2 <-m2.ETH$aic

m3.ETH <- arima(R.ETH, order = c(1,1,6), method='ML')
aic.ETH.3 <-m3.ETH$aic

m4.ETH <- arima(R.ETH, order = c(1,1,7), method='ML')
aic.ETH.4 <-m4.ETH$aic

m5.ETH <- arima(R.ETH, order = c(1,1,8), method='ML')
aic.ETH.5 <-m5.ETH$aic
```

ARIMA Model Identification for DOGE

```
m1.DOGE <- arima(R.DOGE, order =c(0,1,1), method='ML')
aic.DOGE.1 <- m1.DOGE$aic
```

```r
m2.DOGE <- arima(R.DOGE, order =c(0,1,9), method='ML')
aic.DOGE.2 <- m2.DOGE$aic

m3.DOGE <- arima(R.DOGE, order = c(1,1,2), method='ML')
aic.DOGE.3 <- m3.DOGE$aic

m4.DOGE <- arima(R.DOGE, order = c(1,1,3), method='ML')
aic.DOGE.4 <- m4.DOGE$aic

m5.DOGE <- arima(R.DOGE, order = c(1,1,4), method='ML')
aic.DOGE.5 <- m5.DOGE$aic
```

ARIMA Model Identification for LTC

```r
m1.LTC <- arima(R.LTC, order =c(0,1,10), method='ML')
aic.LTC.1 <- m1.LTC$aic

m2.LTC <- arima(R.LTC, order =c(1,1,3), method='ML')
aic.LTC.2 <- m2.LTC$aic

m3.LTC <- arima(R.LTC, order = c(1,1,4), method='ML')
aic.LTC.3 <- m3.LTC$aic

m4.LTC <- arima(R.LTC, order = c(1,1,5), method='ML')
aic.LTC.4 <- m4.LTC$aic

m5.LTC <- arima(R.LTC, order = c(1,1,6), method='ML')
aic.LTC.5 <- m5.LTC$aic
```

ARIMA Candidate Model AIC and Model Selection

```r
AIC <- data.frame(BTC = c(aic.BTC.1, aic.BTC.2, aic.BTC.3, aic.BTC.4,
                          aic.BTC.5), BSV = c(aic.BSV.1, aic.BSV.2, aic.BSV.3,
                                              aic.BSV.4, aic.BSV.5),
                  ETC = c(aic.ETC.1, aic.ETC.2, aic.ETC.3, aic.ETC.4,
                          aic.ETC.5), ETH = c(aic.ETH.1, aic.ETH.2,
                                              aic.ETH.3, aic.ETH.4, aic.ETH.5),
                  DOGE = c(aic.DOGE.1, aic.DOGE.2, aic.DOGE.3, aic.DOGE.4,
                           aic.DOGE.5), LTC = c(aic.LTC.1, aic.LTC.2, aic.LTC.3,
                                                aic.LTC.4, aic.LTC.5))

selection <- c(min(AIC$BTC), min(AIC$BSV), min(AIC$ETC), min(AIC$ETH),
               min(AIC$DOGE), min(AIC$LTC))
```

BTC: ARIMA(1,1,3) BSV: ARIMA(1,1,4) ETC: ARIMA(1,1,4) ETH: ARIMA(1,1,7) DOGE: ARIMA(0,1,9) LTC: ARIMA(1,1,4)

Residual Analysis

```r
residual.analysis <- function(model, std = TRUE,start = 2,
class = c("ARIMA","GARCH","ARMA-GARCH")[1]){
library(TSA)
  library(FitAR)
  if (class == "ARIMA"){
    if (std == TRUE){
      res.model = rstandard(model)
```

```
    }else{
      res.model = residuals(model)
    }
  }else if (class == "GARCH"){
    res.model = model$residuals[start:model$n.used]
  }else if (class == "ARMA-GARCH"){
      res.model = model@fit$residuals
  }else {
    stop("The argument 'class' must be either 'ARIMA' or 'GARCH' ")
  }
}
res.BTC <- residual.analysis(m3.BTC, std = TRUE,start = 1)
res.BSV <- residual.analysis(m4.BSV, std = TRUE,start = 1)
res.ETC <- residual.analysis(m2.ETC, std = TRUE,start = 1)
res.ETH <- residual.analysis(m1.ETH, std = TRUE,start = 1)
res.DOGE <- residual.analysis(m4.DOGE, std = TRUE,start = 1)
res.LTC <- residual.analysis(m5.LTC, std = TRUE,start = 1)
```

Forecast returns for BTC through ARIMA(1,1,3)

```
library(forecast)
fit.BTC <- predict(arima(R.BTC,order = c(1,1,3)), n.ahead = 30)
y.BTC <- fit.BTC$pred
f.BTC <- data.frame(y.BTC)
library(quantmod)
h <- last(R.BTC)
for (i in 2:nrow(f.BTC))
{
dy1 <- f.BTC$y[1]+h
dy2 <- f.BTC$y[2]+dy1
dy3 <- f.BTC$y[3]+dy2
dy4 <- f.BTC$y[4]+dy3
dy5 <- f.BTC$y[5]+dy4
dy6 <- f.BTC$y[6]+dy5
dy7 <- f.BTC$y[7]+dy6
dy8 <- f.BTC$y[8]+dy7
dy9 <- f.BTC$y[9]+dy8
dy10 <- f.BTC$y[10]+dy9
dy11 <- f.BTC$y[11]+dy10
dy12 <- f.BTC$y[12]+dy11
dy13 <- f.BTC$y[13]+dy12
dy14 <- f.BTC$y[14]+dy13
dy15 <- f.BTC$y[15]+dy14
dy16 <- f.BTC$y[16]+dy15
dy11 <- f.BTC$y[11]+dy10
dy12 <- f.BTC$y[12]+dy11
dy13 <- f.BTC$y[13]+dy12
dy14 <- f.BTC$y[14]+dy13
dy15 <- f.BTC$y[15]+dy14
dy16 <- f.BTC$y[16]+dy15
dy17 <- f.BTC$y[17]+dy16
dy18 <- f.BTC$y[18]+dy17
dy19 <- f.BTC$y[19]+dy18
dy20 <- f.BTC$y[20]+dy19
```

```
dy21 <- f.BTC$y[21]+dy20
dy22 <- f.BTC$y[22]+dy21
dy23 <- f.BTC$y[23]+dy22
dy24 <- f.BTC$y[24]+dy23
dy25 <- f.BTC$y[25]+dy24
dy26 <- f.BTC$y[26]+dy25
dy27 <- f.BTC$y[27]+dy26
dy28 <- f.BTC$y[28]+dy27
dy29 <- f.BTC$y[29]+dy28
dy30 <- f.BTC$y[30]+dy29
}
Day_1 <- dy1
Day_2 <- dy2
Day_3 <- dy3
Day_4 <- dy4
Day_5 <- dy5
Day_6 <- dy6
Day_7 <- dy7
Day_8 <- dy8
Day_9 <- dy9
Day_10 <- dy10
Day_11 <- dy11
Day_12 <- dy12
Day_13 <- dy13
Day_14 <- dy14
Day_15 <- dy15
Day_16 <- dy16
Day_17 <- dy17
Day_18 <- dy18
Day_19 <- dy19
Day_20 <- dy20
Day_21 <- dy21
Day_22 <- dy22
Day_23 <- dy23
Day_24 <- dy24
Day_25 <- dy25
Day_26 <- dy26
Day_27 <- dy27
Day_28 <- dy28
Day_29 <- dy29
Day_30 <- dy30

Predict <- rbind(Day_1,Day_2, Day_3, Day_4, Day_5,Day_6,Day_7,Day_8,Day_9,Day_10,Day_11,
Day_12,Day_13,Day_14,Day_15,Day_16,Day_17,
Day_18,Day_19,Day_20,Day_21,Day_22,
               Day_23,Day_24,Day_25,Day_26,
Day_27,Day_28,Day_29,Day_30)
colnames(Predict) <- c("USD")
```

Forecast returns for BSV through ARIMA(1,1,4)

```
fit.BSV <- predict(arima(R.BSV,order =c(1,1,4)), n.ahead = 30)
y.BSV <- fit.BSV$pred
f.BSV <- data.frame(y.BSV)
```

```r
h <- last(R.BSV)
for (i in 2:nrow(f.BSV))
{
dy1 <- f.BSV$y[1]+h
dy2 <- f.BSV$y[2]+dy1
dy3 <- f.BSV$y[3]+dy2
dy4 <- f.BSV$y[4]+dy3
dy5 <- f.BSV$y[5]+dy4
dy6 <- f.BSV$y[1]+dy5
dy7 <- f.BSV$y[2]+dy6
dy8 <- f.BSV$y[3]+dy7
dy9 <- f.BSV$y[4]+dy8
dy10 <- f.BSV$y[5]+dy9
dy11 <- f.BSV$y[1]+dy10
dy12 <- f.BSV$y[2]+dy11
dy13 <- f.BSV$y[3]+dy12
dy14 <- f.BSV$y[4]+dy13
dy15 <- f.BSV$y[5]+dy14
dy16 <- f.BSV$y[1]+dy15
dy17 <- f.BSV$y[2]+dy16
dy18 <- f.BSV$y[3]+dy17
dy19 <- f.BSV$y[4]+dy18
dy20 <- f.BSV$y[5]+dy19
dy21 <- f.BSV$y[1]+dy20
dy22 <- f.BSV$y[2]+dy21
dy23 <- f.BSV$y[3]+dy22
dy24 <- f.BSV$y[4]+dy23
dy25 <- f.BSV$y[5]+dy24
dy26 <- f.BSV$y[1]+dy25
dy27 <- f.BSV$y[2]+dy26
dy28 <- f.BSV$y[3]+dy27
dy29 <- f.BSV$y[4]+dy28
dy30 <- f.BSV$y[5]+dy29
}
Day_1 <- dy1
Day_2 <- dy2
Day_3 <- dy3
Day_4 <- dy4
Day_5 <- dy5
Day_6 <- dy6
Day_7 <- dy7
Day_8 <- dy8
Day_9 <- dy9
Day_10 <- dy10
Day_11 <- dy11
Day_12 <- dy12
Day_13 <- dy13
Day_14 <- dy14
Day_15 <- dy15
Day_16 <- dy16
Day_17 <- dy17
Day_18 <- dy18
Day_19 <- dy19
```

```
Day_20 <- dy20
Day_21 <- dy21
Day_22 <- dy22
Day_23 <- dy23
Day_24 <- dy24
Day_25 <- dy25
Day_26 <- dy26
Day_27 <- dy27
Day_28 <- dy28
Day_29 <- dy29
Day_30 <- dy30
Predict.BSV <- rbind(Day_1,Day_2, Day_3, Day_4, Day_5,Day_6,Day_7,Day_8,Day_9,Day_10,Day_11,
Day_12,Day_13,Day_14,Day_15,Day_16,Day_17,
Day_18,Day_19,Day_20,Day_21,Day_22,Day_23,Day_24,
Day_25,Day_26,Day_27,Day_28,Day_29,Day_30)
colnames(Predict.BSV) <- c("USD")
```

Forecast returns for ETC through ARIMA(1,1,4)

```
fit.ETC <- predict(arima(R.ETC,order = c(1,1,4)), n.ahead = 30)
y.ETC <- fit.ETC$pred
f.ETC <- data.frame(y.ETC)

h <- last(R.ETC)
for (i in 2:nrow(f.ETC))
{
dy1 <- f.ETC$y[1]+h
dy2 <- f.ETC$y[2]+dy1
dy3 <- f.ETC$y[3]+dy2
dy4 <- f.ETC$y[4]+dy3
dy5 <- f.ETC$y[5]+dy4
dy6 <- f.ETC$y[1]+dy5
dy7 <- f.ETC$y[2]+dy6
dy8 <- f.ETC$y[3]+dy7
dy9 <- f.ETC$y[4]+dy8
dy10 <- f.ETC$y[5]+dy9
dy11 <- f.ETC$y[1]+dy10
dy12 <- f.ETC$y[2]+dy11
dy13 <- f.ETC$y[3]+dy12
dy14 <- f.ETC$y[4]+dy13
dy15 <- f.ETC$y[5]+dy14
dy16 <- f.ETC$y[1]+dy15
dy17 <- f.ETC$y[2]+dy16
dy18 <- f.ETC$y[3]+dy17
dy19 <- f.ETC$y[4]+dy18
dy20 <- f.ETC$y[5]+dy19
dy21 <- f.ETC$y[1]+dy20
dy22 <- f.ETC$y[2]+dy21
dy23 <- f.ETC$y[3]+dy22
dy24 <- f.ETC$y[4]+dy23
dy25 <- f.ETC$y[5]+dy24
dy26 <- f.ETC$y[1]+dy25
dy27 <- f.ETC$y[2]+dy26
dy28 <- f.ETC$y[3]+dy27
```

```
dy29 <- f.ETC$y[4]+dy28
dy30 <- f.ETC$y[5]+dy29
}
Day_1 <- dy1
Day_2 <- dy2
Day_3 <- dy3
Day_4 <- dy4
Day_5 <- dy5
Day_6 <- dy6
Day_7 <- dy7
Day_8 <- dy8
Day_9 <- dy9
Day_10 <- dy10
Day_11 <- dy11
Day_12 <- dy12
Day_13 <- dy13
Day_14 <- dy14
Day_15 <- dy15
Day_16 <- dy16
Day_17 <- dy17
Day_18 <- dy18
Day_19 <- dy19
Day_20 <- dy20
Day_21 <- dy21
Day_22 <- dy22
Day_23 <- dy23
Day_24 <- dy24
Day_25 <- dy25
Day_26 <- dy26
Day_27 <- dy27
Day_28 <- dy28
Day_29 <- dy29
Day_30 <- dy30
Predict.ETC <- rbind(Day_1,Day_2, Day_3, Day_4, Day_5,Day_6,Day_7,Day_8,Day_9,Day_10,Day_11,
Day_12,Day_13,Day_14,Day_15,Day_16,Day_17,
Day_18,Day_19,Day_20,Day_21,Day_22,Day_23,Day_24,
Day_25, Day_26,Day_27,Day_28,Day_29,Day_30)
colnames(Predict.ETC) <- c("USD")
```

Forecast returns for ETH through ARIMA(1,1,7)

```
fit.ETH <- predict(arima(R.ETH,order = c(1,1,7)), n.ahead = 30)
y.ETH <- fit.ETH$pred
f.ETH <- data.frame(y.ETH)

h <- last(R.ETH)
for (i in 2:nrow(f.ETH))
{
dy1 <- f.ETH$y[1]+h
dy2 <- f.ETH$y[2]+dy1
dy3 <- f.ETH$y[3]+dy2
dy4 <- f.ETH$y[4]+dy3
dy5 <- f.ETH$y[5]+dy4
dy6 <- f.ETH$y[1]+dy5
```

```
dy7 <- f.ETH$y[2]+dy6
dy8 <- f.ETH$y[3]+dy7
dy9 <- f.ETH$y[4]+dy8
dy10 <- f.ETH$y[5]+dy9
dy11 <- f.ETH$y[1]+dy10
dy12 <- f.ETH$y[2]+dy11
dy13 <- f.ETH$y[3]+dy12
dy14 <- f.ETH$y[4]+dy13
dy15 <- f.ETH$y[5]+dy14
dy16 <- f.ETH$y[1]+dy15
dy17 <- f.ETH$y[2]+dy16
dy18 <- f.ETH$y[3]+dy17
dy19 <- f.ETH$y[4]+dy18
dy20 <- f.ETH$y[5]+dy19
dy21 <- f.ETH$y[1]+dy20
dy22 <- f.ETH$y[2]+dy21
dy23 <- f.ETH$y[3]+dy22
dy24 <- f.ETH$y[4]+dy23
dy25 <- f.ETH$y[5]+dy24
dy26 <- f.ETH$y[1]+dy25
dy27 <- f.ETH$y[2]+dy26
dy28 <- f.ETH$y[3]+dy27
dy29 <- f.ETH$y[4]+dy28
dy30 <- f.ETC$y[5]+dy29
}
Day_1 <- dy1
Day_2 <- dy2
Day_3 <- dy3
Day_4 <- dy4
Day_5 <- dy5
Day_6 <- dy6
Day_7 <- dy7
Day_8 <- dy8
Day_9 <- dy9
Day_10 <- dy10
Day_11 <- dy11
Day_12 <- dy12
Day_13 <- dy13
Day_14 <- dy14
Day_15 <- dy15
Day_16 <- dy16
Day_17 <- dy17
Day_18 <- dy18
Day_19 <- dy19
Day_20 <- dy20
Day_21 <- dy21
Day_22 <- dy22
Day_23 <- dy23
Day_24 <- dy24
Day_25 <- dy25
Day_26 <- dy26
Day_27 <- dy27
Day_28 <- dy28
```

```
Day_29 <- dy29
Day_30 <- dy30
Predict.ETH <- rbind(Day_1,Day_2, Day_3, Day_4, Day_5,Day_6,Day_7,Day_8,Day_9,Day_10,Day_11,
Day_12,Day_13,Day_14,Day_15,Day_16,Day_17,
Day_18,Day_19,Day_20,Day_21,Day_22,Day_23,Day_24,
Day_25,Day_26,Day_27,Day_28,Day_29,Day_30)
colnames(Predict.ETH) <- c("USD")
```

Forecast returns for DOGE through ARIMA(0,1,9)

```
fit.DOGE <- predict(arima(R.DOGE,order = c(0,1,9)), n.ahead = 30)
y.DOGE <- fit.DOGE$pred
f.DOGE <- data.frame(y.DOGE)

h = last(R.DOGE)
for (i in 2:nrow(f.DOGE))
{
dy1 <- f.DOGE$y[1]+h
dy2 <- f.DOGE$y[2]+dy1
dy3 <- f.DOGE$y[3]+dy2
dy4 <- f.DOGE$y[4]+dy3
dy5 <- f.DOGE$y[5]+dy4
dy6 <- f.DOGE$y[1]+dy5
dy7 <- f.DOGE$y[2]+dy6
dy8 <- f.DOGE$y[3]+dy7
dy9 <- f.DOGE$y[4]+dy8
dy10 <- f.DOGE$y[5]+dy9
dy11 <- f.DOGE$y[1]+dy10
dy12 <- f.DOGE$y[2]+dy11
dy13  <- f.DOGE$y[3]+dy12
dy14 <- f.DOGE$y[4]+dy13
dy15 <- f.DOGE$y[5]+dy14
dy16 <- f.DOGE$y[1]+dy15
dy17 <- f.DOGE$y[2]+dy16
dy18 <- f.DOGE$y[3]+dy17
dy19 <- f.DOGE$y[4]+dy18
dy20 <- f.DOGE$y[5]+dy19
dy21 <- f.DOGE$y[1]+dy20
dy22 <- f.DOGE$y[2]+dy21
dy23 <- f.DOGE$y[3]+dy22
dy24 <- f.DOGE$y[4]+dy23
dy25 <- f.DOGE$y[5]+dy24
dy26 <- f.DOGE$y[1]+dy25
dy27 <- f.DOGE$y[2]+dy26
dy28 <- f.DOGE$y[3]+dy27
dy29 <- f.DOGE$y[4]+dy28
dy30 <- f.DOGE$y[5]+dy29
}
Day_1 <- dy1
Day_2 <- dy2
Day_3 <- dy3
Day_4 <- dy4
Day_5 <- dy5
Day_6 <- dy6
```

```
Day_7 <- dy7
Day_8 <- dy8
Day_9 <- dy9
Day_10 <- dy10
Day_11 <- dy11
Day_12 <- dy12
Day_13 <- dy13
Day_14 <- dy14
Day_15 <- dy15
Day_16 <- dy16
Day_17 <- dy17
Day_18 <- dy18
Day_19 <- dy19
Day_20 <- dy20
Day_21 <- dy21
Day_22 <- dy22
Day_23 <- dy23
Day_24 <- dy24
Day_25 <- dy25
Day_26 <- dy26
Day_27 <- dy27
Day_28 <- dy28
Day_29 <- dy29
Day_30 <- dy30
Predict.DOGE<-rbind(Day_1,Day_2, Day_3, Day_4, Day_5,Day_6,Day_7,Day_8,Day_9,Day_10,Day_11,
Day_12,Day_13,Day_14,Day_15,Day_16,Day_17,
Day_18,Day_19,Day_20,Day_21,Day_22,Day_23,Day_24,
Day_25,Day_26,Day_27,Day_28,Day_29,Day_30)
colnames(Predict.DOGE) <- c("USD")
```

Forecast returns for LTC through ARIMA(1,1,4)

```
fit.LTC <- predict(arima(R.LTC,order = c(1,1,4)), n.ahead = 30)
y.LTC <- fit.LTC$pred
f.LTC <- data.frame(y.LTC)

h = last(R.LTC)
for (i in 2:nrow(f.LTC))
{
dy1 <- f.LTC$y[1]+h
dy2 <- f.LTC$y[2]+dy1
dy3 <- f.LTC$y[3]+dy2
dy4 <- f.LTC$y[4]+dy3
dy5 <- f.LTC$y[5]+dy4
dy6 <- f.LTC$y[1]+dy5
dy7 <- f.LTC$y[2]+dy6
dy8 <- f.LTC$y[3]+dy7
dy9 <- f.LTC$y[4]+dy8
dy10 <- f.LTC$y[5]+dy9
dy11 <- f.LTC$y[1]+dy10
dy12 <- f.LTC$y[2]+dy11
dy13 <- f.LTC$y[3]+dy12
dy14 <- f.LTC$y[4]+dy13
dy15 <- f.LTC$y[5]+dy14
```

```
dy16 <- f.LTC$y[1]+dy15
dy17 <- f.LTC$y[2]+dy16
dy18 <- f.LTC$y[3]+dy17
dy19 <- f.LTC$y[4]+dy18
dy20 <- f.LTC$y[5]+dy19
dy21 <- f.LTC$y[1]+dy20
dy22 <- f.LTC$y[2]+dy21
dy23 <- f.LTC$y[3]+dy22
dy24 <- f.LTC$y[4]+dy23
dy25 <- f.LTC$y[5]+dy24
dy26 <- f.LTC$y[1]+dy25
dy27 <- f.LTC$y[2]+dy26
dy28 <- f.LTC$y[3]+dy27
dy29 <- f.LTC$y[4]+dy28
dy30 <- f.LTC$y[5]+dy29
}
Day_1 <- dy1
Day_2 <- dy2
Day_3 <- dy3
Day_4 <- dy4
Day_5 <- dy5
Day_6 <- dy6
Day_7 <- dy7
Day_8 <- dy8
Day_9 <- dy9
Day_10 <- dy10
Day_11 <- dy11
Day_12 <- dy12
Day_13 <- dy13
Day_14 <- dy14
Day_15 <- dy15
Day_16 <- dy16
Day_17 <- dy17
Day_18 <- dy18
Day_19 <- dy19
Day_20 <- dy20
Day_21 <- dy21
Day_22 <- dy22
Day_23 <- dy23
Day_24 <- dy24
Day_25 <- dy25
Day_26 <- dy26
Day_27 <- dy27
Day_28 <- dy28
Day_29 <- dy29
Day_30 <- dy30
Predict.LTC<-rbind(Day_1,Day_2, Day_3, Day_4, Day_5,Day_6,Day_7,Day_8,Day_9,Day_10,Day_11,
Day_12,Day_13,Day_14,Day_15,Day_16,Day_17,
Day_18,Day_19,Day_20,Day_21,Day_22,
Day_23,Day_24,Day_25, Day_26,Day_27,Day_28,
Day_29,Day_30)
colnames(Predict.LTC) <- c("USD")
```

GARCH Model

Absolute Returns, ACF, and PACF

```
abs.BTC <- abs(R.BTC)
abs.BSV <- abs(R.BSV)
abs.ETC <- abs(R.ETC)
abs.ETH <- abs(R.ETH)
abs.DOGE <- abs(R.DOGE)
abs.LTC <- abs(R.LTC)
```

GARCH Model Estimation for BTC

```
library(tseries)

m.21 <- garch(R.BTC, order=c(2,1), trace = F)

m.41 <- garch(R.BTC, order=c(4,1),trace =FALSE)

m.51 <- garch(R.BTC, order=c(5,1),trace =FALSE)

m.71 <- garch(R.BTC, order=c(7,1),trace =FALSE)

m.32 <- garch(R.BTC, order=c(3,2),trace =FALSE)

m.43 <- garch(R.BTC, order=c(4,3),trace =FALSE)

m.53 <- garch(R.BTC, order=c(5,3),trace =FALSE)

m.03 <- garch(R.BTC, order=c(0,3),trace =FALSE)
```

GARCH Model Estimation for BSV

```
m.41.BSV <- garch(R.BSV, order=c(4,1),trace =FALSE)

m.21.BSV <- garch(R.BSV, order=c(2,1),trace =FALSE)

m.51.BSV <- garch(R.BTC, order=c(5,1),trace =FALSE)

m.61.BSV <- garch(R.BTC, order=c(6,1),trace =FALSE)

m.42.BSV <- garch(R.BSV, order=c(4,2),trace =FALSE)

m.52.BSV <- garch(R.BTC, order=c(5,2),trace =FALSE)

m.62.BSV <- garch(R.BTC, order=c(6,2),trace =FALSE)

m.72.BSV <- garch(R.BTC, order=c(7,2),trace =FALSE)

m.92.BSV <- garch(R.BTC, order=c(9,2),trace =FALSE)
```

GARCH Model Estimation for ETC

```
m.10.ETC <- garch(R.ETC, order=c(1,0),trace =FALSE)

m.20.ETC <- garch(R.ETC, order=c(2,0),trace =FALSE)

m.21.ETC <- garch(R.ETC, order=c(2,1),trace =FALSE)
```

```r
m.23.ETC <- garch(R.ETC, order=c(2,3),trace =FALSE)

m.30.ETC <- garch(R.ETC, order=c(3,0),trace =FALSE)

m.31.ETC <- garch(R.ETC, order=c(3,1),trace =FALSE)

m.33.ETC <- garch(R.ETC, order=c(3,3),trace =FALSE)

m.40.ETC <- garch(R.ETC, order=c(4,0),trace =FALSE)

m.41.ETC <- garch(R.ETC, order=c(4,1),trace =FALSE)

m.43.ETC <- garch(R.ETC, order=c(4,3),trace =FALSE)
```

GARCH Model Estimation for ETH

```r
m.21.ETH <- garch(R.ETH, order=c(2,1),trace =FALSE)

m.25.ETH <- garch(R.ETH, order=c(2,5),trace =FALSE)

m.41.ETH <- garch(R.ETH, order=c(4,1),trace =FALSE)

m.51.ETH <- garch(R.ETH, order=c(5,1),trace =FALSE)

m.61.ETH <- garch(R.ETH, order=c(6,1),trace =FALSE)

m.70.ETH <- garch(R.ETH, order=c(7,0),trace =FALSE)

m.71.ETH <- garch(R.ETH, order=c(7,1),trace =FALSE)
```

GARCH Model Estimation for DOGE

```r
m.51.DOGE <- garch(R.DOGE, order=c(5,1),trace =FALSE)

m.21.DOGE <- garch(R.DOGE, order=c(2,1),trace =FALSE)

m.31.DOGE <- garch(R.DOGE, order=c(3,1),trace =FALSE)

m.32.DOGE <- garch(R.DOGE, order=c(3,2),trace =FALSE)

m.71.DOGE <- garch(R.DOGE, order=c(7,1),trace =FALSE)

m.52.DOGE <- garch(R.DOGE, order=c(5,2),trace =FALSE)
```

GARCH Model Estimation for LTC

```r
m.21.LTC <- garch(R.LTC, order=c(2,1),trace =FALSE)

m.22.LTC <- garch(R.LTC, order=c(2,2),trace =FALSE)

m.41.LTC <- garch(R.LTC, order=c(4,1),trace =FALSE)

m.42.LTC <- garch(R.LTC, order=c(4,2),trace =FALSE)

m.51.LTC <- garch(R.LTC, order=c(5,1),trace =FALSE)
```

```
m.52.LTC <- garch(R.LTC, order=c(5,2),trace =FALSE)
```

GARCH Model Selection

```
library(dLagM)
library(stats)

BTC.AIC <- AIC(m.21, m.41, m.51, m.71, m.32, m.43, m.53, m.03)

BSV.AIC <- AIC(m.41.BSV, m.21.BSV, m.51.BSV, m.61.BSV, m.42.BSV, m.52.BSV,
               m.62.BSV, m.72.BSV, m.92.BSV)

ETC.AIC <- AIC(m.10.ETC, m.20.ETC, m.21.ETC, m.23.ETC, m.30.ETC, m.31.ETC,
               m.33.ETC, m.40.ETC, m.41.ETC, m.43.ETC)

ETH.AIC <- AIC(m.21.ETH, m.25.ETH, m.41.ETH, m.51.ETH, m.61.ETH, m.70.ETH,
               m.71.ETH)

DOGE.AIC <- AIC(m.51.DOGE, m.21.DOGE, m.31.DOGE, m.32.DOGE, m.71.DOGE,
                m.52.DOGE)

LTC.AIC <- AIC(m.21.LTC, m.22.LTC, m.41.LTC, m.51.LTC, m.52.LTC)
```

Model Fitting BTC GARCH(2,1)

```
library(rugarch)

model.21.BTC <-ugarchspec(variance.model = list(model = "sGARCH",
                                                garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "norm")

model.21.BTC <-ugarchfit(spec = model.21.BTC, data = R.BTC, out.sample = 100)


model.21_t_dis.BTC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "std")
model.21_t_dis.BTC <- ugarchfit(spec = model.21_t_dis.BTC,data = R.BTC,
                                out.sample = 100)

model.21_t_skw.BTC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "sstd")

model.21_t_skw.BTC <- ugarchfit(spec = model.21_t_skw.BTC,data = R.BTC,
                                out.sample = 100)
```

Model Fitting BSV GARCH(5,2)

```
model.52.BSV <- ugarchspec(variance.model = list(model = "sGARCH",
                                                 garchOrder = c(5, 2)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
```

```
                  distribution.model = "norm")

m.52_BSV <- ugarchfit(spec = model.52.BSV, data = R.BSV, out.sample = 100)


model.52_t_dis.BSV <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(5, 2)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "std")

model.52_t_dis.BSV <- ugarchfit(spec = model.52_t_dis.BSV, data = R.BSV,
                                out.sample = 100)

model.52_t_skw.BSV <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(5, 2)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "sstd")

model.52_t_skw.BSV <- ugarchfit(spec = model.52_t_skw.BSV, data = R.BSV,
                                out.sample = 100)
```

Model Fitting ETC GARCH(2,1)

```
model.21.ETC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                 garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "norm")

m.21_ETC <- ugarchfit(spec = model.21.ETC, data = R.ETC, out.sample = 100)


model.21_t_dis.ETC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "std")

model.21_t_dis.ETC <-ugarchfit(spec = model.21_t_dis.ETC, data = R.ETC,
                               out.sample = 100)

model.21_t_skw.ETC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "sstd")

model.21_t_skw.ETC <- ugarchfit(spec = model.21_t_skw.ETC, data = R.ETC,
                                out.sample = 100)
```

Model Fitting ETH GARCH(4,1)

```
library(rugarch)
model.41.ETH <- ugarchspec(variance.model = list(model = "sGARCH",
                                                 garchOrder = c(4, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "norm")
```

```r
m.41_ETH <- ugarchfit(spec = model.41.ETH, data = R.ETH, out.sample = 100)

model.41_t_dis.ETH <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(4, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "std")

model.41_t_dis.ETH <- ugarchfit(spec = model.41_t_dis.ETH, data = R.ETH,
                                out.sample = 100)

model.41_t_skw.ETH <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(4, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "sstd")

model.41_t_skw.ETH <- ugarchfit(spec = model.41_t_skw.ETH, data = R.ETH,
                                out.sample = 100)
```

Model Fitting DOGE GARCH(7,2)

```r
model.71.DOGE <- ugarchspec(variance.model = list(model = "sGARCH",
                                                  garchOrder = c(7, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "norm")

m.71_DOGE <- ugarchfit(spec = model.71.DOGE, data = R.DOGE, out.sample = 100)

model.71_t_dis.DOGE <- ugarchspec(variance.model = list(model = "sGARCH",
                                                        garchOrder = c(7, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "std")

model.71_t_dis.DOGE <- ugarchfit(spec = model.71_t_dis.DOGE, data = R.DOGE,
                                 out.sample = 100)

model.71_t_skw.DOGE <- ugarchspec(variance.model = list(model = "sGARCH",
                                                        garchOrder = c(7, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "sstd")

model.71_t_skw.DOGE <- ugarchfit(spec = model.71_t_skw.DOGE, data = R.DOGE,
                                 out.sample = 100)
```

Model Fitting LTC GARCH(2,1)

```r
model.21.LTC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                 garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                  distribution.model = "norm")

m.21_LTC <- ugarchfit(spec = model.21.LTC, data = R.LTC, out.sample = 100)

model.21_t_dis.LTC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(2, 1)),
                  mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
```

```
                   distribution.model = "std")

model.21_t_dis.LTC <- ugarchfit(spec = model.21_t_dis.LTC, data = R.LTC,
                                 out.sample = 100)

model.21_t_skw.LTC <- ugarchspec(variance.model = list(model = "sGARCH",
                                                       garchOrder = c(2, 1)),
                   mean.model = list(armaOrder = c(0, 0), include.mean = TRUE),
                   distribution.model = "sstd")

model.21_t_skw.LTC <- ugarchfit(spec = model.21_t_skw.LTC, data = R.LTC,
                                 out.sample = 100)
```

Forecast BTC returns through GARCH

```
library(fGarch)

forecast.BTC <- ugarchforecast(model.21_t_skw.BTC, data = R.BTC, n.ahead = 30,
                                n.roll =100)
```

Forecast BSV returns through GARCH

```
forecast.BSV <- ugarchforecast(m.52_BSV, data = R.BSV, n.ahead = 30,
                                n.roll =100)
```

Forecast ETC returns through GARCH

```
forecast.ETC <- ugarchforecast(model.21_t_skw.ETC, data = R.ETC, n.ahead = 30,
                                n.roll =10)
```

Forecast ETH returns through GARCH

```
forecast.ETH <- ugarchforecast(model.41_t_dis.ETH, data = R.ETH, n.ahead = 30,
                                n.roll =100)
```

Forecast DOGE returns through GARCH

```
forecast.DOGE <- ugarchforecast(model.71_t_skw.DOGE, data = R.DOGE,
                                 n.ahead = 30, n.roll =100)
```

Forecast LTC returns through GARCH

```
forecast.LTC <- ugarchforecast(model.21_t_dis.LTC, data = R.LTC, n.ahead = 30,
                                n.roll =10)
```

Technical Analysis

Bollinger Bands BTC

```
library(TTR)
library(quantmod)

ticker <- "BTC-USD"
p.BTC <- get(getSymbols(ticker, from = "2019-01-01", to = "2022-03-21"))
bb.BTC <- BBands(Cl(p.BTC),n=20, sd=2)
M.BTC <- momentum(Cl(p.BTC), n=2)
```

Bollinger Bands BSV

```r
ticker.BSV <- "BSV-USD"
p.BSV <- get(getSymbols(ticker.BSV, from = "2019-01-01", to = "2022-03-21"))
bb.BSV <- BBands(Cl(p.BSV),n=20, sd=2)
M.BSV <- momentum(Cl(p.BSV), n=2)
```

Bollinger Bands ETC

```r
ticker.ETC <- "ETC-USD"
p.ETC <- get(getSymbols(ticker.ETC, from = "2019-01-01", to = "2022-03-21"))
bb.ETC <- BBands(Cl(p.ETC),n=20, sd=2)
M.ETC <- momentum(Cl(p.ETC), n=2)
```

Bollinger Bands ETH

```r
ticker.ETH <- "ETH-USD"
p.ETH <- get(getSymbols(ticker.ETH, from = "2019-01-01", to = "2022-03-22"))
bb.ETH <- BBands(Cl(p.ETH),n=20, sd=2)
M.ETH <- momentum(Cl(p.ETH), n=2)
```

Bollinger Bands DOGE

```r
ticker.DOGE <- "DOGE-USD"
p.DOGE <- get(getSymbols(ticker.DOGE, from = "2019-01-01", to = "2022-03-22"))
bb.DOGE <- BBands(Cl(p.DOGE),n=20, sd=2)
M.DOGE <- momentum(Cl(p.DOGE), n=2)
```

Bollinger Bands LTC

```r
ticker.LTC <- "LTC-USD"
p.LTC <- get(getSymbols(ticker.LTC, from = "2019-01-01", to = "2022-03-22"))
bb.LTC <- BBands(Cl(p.LTC),n=20, sd=2)
M.LTC <- momentum(Cl(p.LTC), n=2)
```

Fast EMA, Slow EMA, and Klinger Oscillator (BTC and ETH)

```r
library(pracma)
library(ggplot2)
library(reshape2)
library(readxl)

Flow.BTC <- read_excel("/Users/beatricebusato/Desktop/FA 800/Data/Flow_BTC.xlsx")
head(Flow.BTC)
```

```
## # A tibble: 6 x 4
##   Date                 Inflow Outflow Netflow
##   <dttm>                <dbl>   <dbl>   <dbl>
## 1 2019-01-01 00:00:00 25176.   25972.   -796.
## 2 2019-01-02 00:00:00 48217.   52338.  -4121.
## 3 2019-01-03 00:00:00 48348.   58257.  -9909.
## 4 2019-01-04 00:00:00 55411.   51810.   3601.
## 5 2019-01-05 00:00:00 36895.   31716.   5180.
## 6 2019-01-06 00:00:00 33018.   29405.   3613.
```

```r
Flow.ETH <- read_excel("/Users/beatricebusato/Desktop/FA 800/Data/Flows_ETH.xlsx")
head(Flow.ETH)
```

```
## # A tibble: 6 x 4
##   Date                  Inflow  Outflow  Netflow
##   <dttm>                 <dbl>    <dbl>    <dbl>
```

```
## 1 2019-01-01 00:00:00 346266.  380453.  -34186.
## 2 2019-01-02 00:00:00 693144.  589937.  103207.
## 3 2019-01-03 00:00:00 567440.  536316.   31124.
## 4 2019-01-04 00:00:00 605772.  694635.  -88863.
## 5 2019-01-05 00:00:00 632765. 1359695. -726930.
## 6 2019-01-06 00:00:00 679580.  505375.  174205.
```

```
Fast.EMA <- movavg(Flow.BTC$Netflow, n=34, type='e')
Slow.EMA <- movavg(Flow.BTC$Netflow, n=55, type='e')
df <- data.frame(Flow.BTC$Date, Fast.EMA, Slow.EMA)
df <- melt(df,  id.vars = 'Flow.BTC.Date', variable.name = 'series')

Klinger.Oscillator <- Fast.EMA - Slow.EMA
df1 <- data.frame(Flow.BTC$Date, Fast.EMA, Slow.EMA, Klinger.Oscillator)

Fast.EMA <- movavg(Flow.ETH$Netflow, n=34, type='e')
Slow.EMA <- movavg(Flow.ETH$Netflow, n=55, type='e')
df <- data.frame(Flow.ETH$Date, Fast.EMA, Slow.EMA)
df <- melt(df,  id.vars = 'Flow.ETH.Date', variable.name = 'series')

Klinger.Oscillator <- Fast.EMA - Slow.EMA
df1 <- data.frame(Flow.ETH$Date, Fast.EMA, Slow.EMA, Klinger.Oscillator)
```

Fundamental Analysis

Correlation using prices, returns, addresses, and volumes

```
library(ggplot2)
library(readxl)
library(psych)

Addresses.BTC <- read_excel("/Users/beatricebusato/Desktop/FA 800/Data/Adresses_BTC.xlsx")

Addresses.ETH <- read_excel("/Users/beatricebusato/Desktop/FA 800/Data/Adresses_ETH.xlsx")

Addresses.LTC <- read_excel("/Users/beatricebusato/Desktop/FA 800/Data/AdLTC.xlsx")

Addresses.BTC <- na.omit(Addresses.BTC)
Addresses.BTC <- Addresses.BTC[-1]
Addresses.BTC$Price <- as.numeric(Addresses.BTC$Price)
Addresses.BTC$Return <- as.numeric(Addresses.BTC$Return)
Addresses.BTC$Active <- as.numeric(Addresses.BTC$Active)
Addresses.BTC$Sending <- as.numeric(Addresses.BTC$Sending)
Addresses.BTC$Receiving <- as.numeric(Addresses.BTC$Receiving)
Addresses.BTC$Volume <- as.numeric(Addresses.BTC$Volume)
cor(Addresses.BTC)
```

```
##                 Price       Return     Active    Sending  Receiving
## Price      1.00000000 -0.044454137 0.57657016 0.47358997 0.54981441
## Return    -0.04445414  1.000000000 0.02641798 0.04274559 0.01013276
## Active     0.57657016  0.026417979 1.00000000 0.95642846 0.94733605
## Sending    0.47358997  0.042745590 0.95642846 1.00000000 0.85451284
## Receiving  0.54981441  0.010132763 0.94733605 0.85451284 1.00000000
## Volume     0.48600724 -0.006735517 0.60285646 0.48626410 0.59947877
##                Volume
## Price      0.486007236
```

```
## Return    -0.006735517
## Active     0.602856465
## Sending    0.486264102
## Receiving  0.599478767
## Volume     1.000000000
```

```
Addresses.ETH <- Addresses.ETH[-1]
Addresses.ETH <- na.omit(Addresses.ETH)
Addresses.ETH$Price <- as.numeric(Addresses.ETH$Price)
Addresses.ETH$Return <- as.numeric(Addresses.ETH$Return)
Addresses.ETH$Active <- as.numeric(Addresses.ETH$Active)
Addresses.ETH$Sending <- as.numeric(Addresses.ETH$Sending)
Addresses.ETH$Receiving <- as.numeric(Addresses.ETH$Receiving)
Addresses.ETH$Volume <- as.numeric(Addresses.ETH$Volume)
cor(Addresses.ETH)
```

```
##                 Price       Return      Active     Sending  Receiving
## Price      1.00000000 -0.033326472 0.829894706 0.821064858 0.77770436
## Return    -0.03332647  1.000000000 0.007907492 0.008722384 0.01408465
## Active     0.82989471  0.007907492 1.000000000 0.976688779 0.95215499
## Sending    0.82106486  0.008722384 0.976688779 1.000000000 0.90914973
## Receiving  0.77770436  0.014084648 0.952154990 0.909149734 1.00000000
## Volume     0.50797340  0.018984373 0.685057059 0.646460917 0.71397672
##               Volume
## Price     0.50797340
## Return    0.01898437
## Active    0.68505706
## Sending   0.64646092
## Receiving 0.71397672
## Volume    1.00000000
```

```
Addresses.LTC <- Addresses.LTC[-1]
Addresses.LTC <- na.omit(Addresses.LTC)
Addresses.LTC$Price <- as.numeric(Addresses.LTC$Price)
Addresses.LTC$Return <- as.numeric(Addresses.LTC$Return)
Addresses.LTC$Active <- as.numeric(Addresses.LTC$Active)
Addresses.LTC$Sending <- as.numeric(Addresses.LTC$Sending)
Addresses.LTC$Receiving <- as.numeric(Addresses.LTC$Receiving)
Addresses.LTC$Volume <- as.numeric(Addresses.LTC$Volume)
cor(Addresses.LTC)
```

```
##                 Price      Return      Active     Sending   Receiving
## Price      1.00000000 -0.06405025  0.76478926  0.72964786  0.76007198
## Return    -0.06405025  1.00000000 -0.03629186 -0.03183649 -0.02898825
## Active     0.76478926 -0.03629186  1.00000000  0.94982846  0.99036830
## Sending    0.72964786 -0.03183649  0.94982846  1.00000000  0.92439297
## Receiving  0.76007198 -0.02898825  0.99036830  0.92439297  1.00000000
## New        0.70840102 -0.02313784  0.91528013  0.97040523  0.91697922
## Total      0.57730974 -0.02672509  0.85590149  0.85940322  0.85210058
## Volume     0.37385527 -0.01731030 -0.03132782 -0.03025074 -0.04225178
##                   New       Total      Volume
## Price      0.70840102  0.57730974  0.37385527
## Return    -0.02313784 -0.02672509 -0.01731030
## Active     0.91528013  0.85590149 -0.03132782
## Sending    0.97040523  0.85940322 -0.03025074
## Receiving  0.91697922  0.85210058 -0.04225178
```

```
## New        1.00000000  0.83660061 -0.05180986
## Total      0.83660061  1.00000000 -0.15849147
## Volume    -0.05180986 -0.15849147  1.00000000
```

Rolling Volatility

```
library(zoo)

window <- 3

BTC_rolling_sd <- rollapply(Addresses.BTC$Return, window,
                            function(x) sd(x))

BTC_rolling_volatility <- BTC_rolling_sd^2

ETH_rolling_sd <- rollapply(Addresses.ETH$Return, window,
                            function(x) sd(x))
ETH_rolling_volatility <- ETH_rolling_sd^2


LTC_rolling_sd <- rollapply(Addresses.LTC$Return, window,
                            function(x) sd(x))

LTC_rolling_volatility <- LTC_rolling_sd^2


window <- 6

BTC_rolling_sd <- rollapply(Addresses.BTC$Return, window,
                            function(x) sd(x))

BTC_rolling_volatility <- BTC_rolling_sd^2


ETH_rolling_sd <- rollapply(Addresses.ETH$Return, window,
                            function(x) sd(x))

ETH_rolling_volatility <- ETH_rolling_sd^2

LTC_rolling_sd <- rollapply(Addresses.LTC$Return, window,
                            function(x) sd(x))
LTC_rolling_volatility <- LTC_rolling_sd^2
```

Rolling Correlation

```
library(zoo)
library(PerformanceAnalytics)

BTC.roll.corr.price_volume <- rollapply(Addresses.BTC, width = 3, function(x)
  cor(x[,1],x[,6]), by.column = FALSE)

BTC.roll.corr.return_volume <- rollapply(Addresses.BTC, width = 3, function(x)
  cor(x[,2],x[,6]), by.column = FALSE)

ETH.roll_corr.price_volume <- rollapply(Addresses.ETH, width = 3, function(x)
  cor(x[,1],x[,6]), by.column = FALSE)
```

```r
ETH.roll.corr.return_volume <- rollapply(Addresses.ETH, width = 3, function(x)
  cor(x[,2],x[,6]), by.column = FALSE)

LTC.roll.corr.price_volume <- rollapply(Addresses.LTC, width = 3, function(x)
  cor(x[,1],x[,6]), by.column = FALSE)

LTC.roll.corr.return_volume <- rollapply(Addresses.LTC, width = 3, function(x)
  cor(x[,2],x[,6]), by.column = FALSE)

BTC.roll.corr.price_volume <- rollapply(Addresses.BTC, width = 6, function(x)
  cor(x[,1],x[,6]), by.column = FALSE)

BTC.roll.corr.return_volume <- rollapply(Addresses.BTC, width = 6, function(x)
  cor(x[,2],x[,6]), by.column = FALSE)

ETH.roll_corr.price_volume <- rollapply(Addresses.ETH, width = 6, function(x)
  cor(x[,1],x[,6]), by.column = FALSE)

ETH.roll.corr.return_volume <- rollapply(Addresses.ETH, width = 6, function(x)
  cor(x[,2],x[,6]), by.column = FALSE)

LTC.roll.corr.price_volume <- rollapply(Addresses.LTC, width = 6, function(x)
  cor(x[,1],x[,6]), by.column = FALSE)

LTC.roll.corr.return_volume <- rollapply(Addresses.LTC, width = 6, function(x)
  cor(x[,2],x[,6]), by.column = FALSE)
```