Fall 1-2021

Forecasting the Prices of Cryptocurrencies using a Novel Parameter Optimization of VARIMA Models

Alexander Barrett
Chapman University, barre110@mail.chapman.edu

Forecasting the Prices of Cryptocurrencies using a Novel Parameter Optimization of VARIMA Models

A Dissertation by

Alexander Barrett

Chapman University

Orange, CA

Schmid College of Science and Technology

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computational and Data Sciences

January 2021

Committee in charge:

Cyril Rakovski, Ph.D., Chair

Mihaela Vajiac, Ph.D.

Mohamed Allali, Ph.D.

CHAPMAN UNIVERSITY
SCHMID COLLEGE OF SCIENCE AND TECHNOLOGY
Computational and Data Sciences

The dissertation of Alexander Barrett is approved.

Cyril Rakovski

Cyril Rakovski, Ph.D., Chair

Mihaela Vajiac, Ph.D.

Mohamed Allali, Ph.D.

December 2020

Forecasting the Prices of Cryptocurrencies using a Novel Parameter Optimization of VARIMA Models

by Alexander Barrett

ACKNOWLEDGEMENTS

I would like to thank Dr. Sidy Danioko, Dr. Kyle Anderson, Dr. Chris Watkins, Ehsan Yaghmaei, Jianwei Zheng, my advisor, Dr. Cyril Rakovski, and other committee members Dr. Mohamed Allali and Dr. Mihaela Vajiac, and everyone who offered thoughts and advice that were useful in completing this research.

LIST OF PUBLICATIONS

Zheng J, Chu H, Struppa D, Zhang J, Yacoub M, El-Askary H, Chang A, Ehwerhemuepha L, Abudayyeh I, Barrett A, Fu G. *Optimal Multi-Stage Arrhythmia Classification Approach*. Scientific reports. 2020 Feb 19;10(1):1-7.

Alsheikh MY, Seoane-Vazquez E, Barrett A, Rakovski C, Brown LM, Fawaz S, Tomaszewski DM, Rodriguez-Monguio R. *PRM99-TRENDS IN OPIOID PRESCRIBING IN PRIMARY CARE PHYSICIAN OFFICES IN THE UNITED STATES: 2006-2015*. Value in Health. 2018 Oct 1;21:S372.

ABSTRACT

Forecasting the Prices of Cryptocurrencies using a Novel Parameter Optimization of VARIMA Models

by Alexander Barrett

This work is a comparative study of different univariate and multivariate time series predictive models as applied to Bitcoin, other cryptocurrencies, and other related financial time series data. ARIMA models, long regarded as the gold standard of univariate financial time series prediction due to both its flexibility and simplicity, are used a baseline for prediction. Given the highly correlative nature amongst different cryptocurrencies, this work aims to show the benefit of forecasting with multivariate time series models—primarily focusing on a novel parameter optimization of VARIMA models outlined in this paper.

These models are trained on 3 years of historical data, aggregated from different cryptocurrency exchanges by Coinmarketcap.com, which includes: daily average prices and trading volume. Historical time series data of traditional market data, including the stock Nvidia, the de facto leading manufacture of gaming GPU's, is also analyzed in conjunction with cryptocurrency prices, as gaming GPU's have played a significant role in solving the profitable SHA256 hashing problems associated with cryptocurrency mining and have seen equivalently correlated investor attention as a result. Models are trained on this historical data using moving window subsets, with window lengths of 100, 200, and 300 days and forecasting 1 day into the future. Validation of this prediction against the actually price from that day are done with following metrics: Directional Forecasting (DF), Mean Absolute Error (MAE), and Mean Squared Error (MSE).

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

Abbreviation	**Meaning**
CI	Confidence Interval
RV	Relative Variance
BIT	Bitcoin
ARMA	Auto Regressive Moving Average
ARIMA	Auto Regressive Integrated Moving Average
VARIMA	Vector Autoregressive Integrated Moving Average
pc1	First Principal Component Time Series

LIST OF SYMBOLS

Symbol	Meaning
$\hat{A}$	The operator corresponding to the observable, A
σ	Standard Deviation

1 Bitcoin and Other Cryptocurrencies

1.1 Introduction to Bitcoin

A truly remarkable thing is that Bitcoin has both inextricably taken the financial and tech worlds by storm while simultaneously remaining entrenched in an air mystery to most people as to what it actually is. At its core, Bitcoin is nothing more than a digital ledger for a currency that bears its same name. Simple enough. However, the mathematical and cryptographic technology behind securely maintaining this ledger is what makes Bitcoin novel, particularly, the ability to do so without the seeming requirement of a trusted third party. Much like the U.S. Dollar and other fiat currencies digitally maintained by banks, Bitcoin is not backed by anything in the physical world—it only has value because people believe it has value. This allows Bitcoin share some commonalities between both currencies and commodities, which we will explore more in this chapter.

1.2 Protocol Overview

How does Bitcoin work? There exists a network of computers all running the same Bitcoin protocol software throughout the world. All of these computers "agree" on which Bitcoin account has how much of the currency according to its underlying algorithm.

The algorithm behind this agreement is what is referred to as the "Blockchain." Each metaphorical block in this chain, contains the digital transcript of transactions between individual Bitcoin accounts. Here, individual transactions between accounts employee the traditional usage of public-private key encryption (although here, is actualized through elliptic curves rather than the more commonly used RSA). These blocks are published serially, one after another—on average, every 10 minutes. This time delay is actually by design, and provides adequate time for the publication of this block to disseminate throughout this network work of computers and officially "cement" as the agreed upon block of transactions before the next one is published.

The time delay of publishing this block is accomplished by requiring a published block to contain a solution to computationally difficult to solve math problem based off encoded information uniquely tied to the previous block—specifically this is done by brute forcing inputs to the SHA256 hashing function so that the output has a minimum leading number of 0's.

Finally, to determine which account has how much of the digital coin, every computer in the network, simply traverses through all of the transactions of this blockchain since the beginning of time to see which account each Bitcoin (or portion of a Bitcoin) points to.

Since solving the aforementioned hashing problems is both computationally expensive (literally as well, in terms of the electricity used) and is a necessary role in the blockchain protocol, its execution must be incentivized. This is accomplished by granting Bitcoins themselves as a reward to those in the network who successfully solve the problem associated with a block—this is referred to as "mining a block". This is the mechanism behind how all bitcoins are created. It should be noted that the work involved in successfully mining a block is so computationally difficult that it would be infeasible for a single individual in the network to mine the block alone in a reasonable amount of time; therefore, the work (and Bitcoin reward) is distributed over a collection of users in a "mining pool." There are many mining pools competing against one another in hopes of being the pool who successfully solves the block first to reap the reward.

The logistics involved in mining these blocks is of particular interest to this work because gaming GPU's have proven to be the most ubiquitous piece of hardware for performing the computations involved. Nvidia, the de facto leading manufacture of gaming GPU's, is also analyzed in conjunction with cryptocurrency prices, as gaming GPU's have played a significant role in solving the profitable SHA256 hashing problems associated with cryptocurrency mining and have seen equivalently correlated investor attention as a result.

1.3 Other Cryptocurrencies

With the success and popularity of the first cryptocurrency, Bitcoin, there are now well over a hundred other cryptocurrencies, many built on top the same technology, some using slight variations, and others using completely different protocols. This section will list several of the leading cryptocurrencies ordered by their market capitalization (marketcap—how much total value of the currency is in existence) and provide a brief description of their protocol/relationship to Bitcoin (descriptions and marketcaps sourced from coinmarketcap.com). The time series data of the daily prices and trading volumes for these cryptocurrencies will be discussed and used in modeling in later sections. Although there are differences between the machinery of how these cryptocurrencies are maintained, their time series of daily prices can all be modeled with same methods.

1.3.1 Ethereum (ETH)

"Ethereum is a decentralized open-source blockchain system that features its own cryptocurrency, Ether. ETH works as a platform for numerous other cryptocurrencies, as well as for the execution of decentralized smart contracts. Smart contracts are computer programs that automatically execute the actions necessary to fulfill an agreement between several parties on the internet. The average time it takes to mine an Ethereum block is around 13-15 seconds." Marketcap in December 2020: ~$67 Billion (coinmarketcap.com).

1.3.2 RIPPLE (XRP)

"XRP is the currency that runs on a digital payment platform called RippleNet, which is on top of an open source distributed ledger database called XRP Ledger (not based on blockchain). The RippleNet payment platform is a real-time gross settlement system that aims to enable instant monetary transactions globally—you can actually use any currency to transact on the platform. The XRP Ledger processes transactions roughly every 3-5 seconds." Marketcap in December 2020: ~$22 Billion (coinmarketcap.com).

1.3.3 Tether (USDT)

"USDT is a stable-value cryptocurrency that mirrors the price of the U.S. dollar, issued by a Hong Kong-based company Tether. The token's peg to the USD is achieved via maintaining a sum of dollars in reserves that is equal to the number of USDT in circulation. Originally launched in 2014 as a Realcoin, a second-layer cryptocurrency token build on top of Bitcoin's blockchain through the use of the Omni platform. It was later updated to work on the Ethereum, EOS, Tron, Algorand, and OMG blockchains. The stated purpose of ISDT is to combine the unrestricted nature of cryptocurrencies –which can be sent between users without a trusted third-party intermediary –with the stable value of the US dollar." Marketcap in December 2020: ~$20 Billion (coinmarketcap.com).

1.3.4 Litecoin (LTC)

"The Litecoin currency was based on the Bitcoin protocol, but it differs in terms of the hashing algorithm used, hard cap, block transaction times and a few other factors. Litecoin Differs from Bitcoin in its prioritization of transaction confirmation speed, which is about 2.5 minutes per block." Marketcap in December 2020: ~$5 Billion (coinmarketcap.com).

1.3.5 Bitcoin Cash (BCH)

"In 2017, the Bitcoin project and its community split in two over concerns about Bitcoin's scalability. The result was a hark fork which created Bitcoin Cash, a new cryptocurrency considered by supporters to be the legitimate continuation of the Bitcoin. All Bitcoin holders at the time of the fork (block 478,558) automatically became owners of Bitcoin Cash. Unlike Bitcoin, the Bitcoin Cash block size was increased from 1MB to 8MB, which means Bitcoin Cash can now handle significantly more transactions per second." Marketcap in December 2020: ~$5 Billion (coinmarketcap.com).

1.3.6 Chainlink (LINK)

"Chainlink is a decentralized oracle network which aims to connect smart contracts with data from the real world. It held an ICO in September 2017, raising $32 million, with a total supply of 1 billion LINK tokens. LINK, the cryptocurrency native to the Chainlink decentralized oracle network, is used to pay node operators. Since the Chainlink network has a reputation system (not proof of work), node providers that have a large amount of LINK can bet rewarded with larger contracts, while a failure to deliver accurate information results in a deduction of tokens." Marketcap in December 2020: ~$5 Billion (coinmarketcap.com).

1.3.7 Cardano (ADA)

"Cardano is a proof-of-stake blockchain platform. The ADA token is designed to ensure that owners can participate in the operation of the network. Because of this, those who hold the cryptocurrency have the right to vote on any proposed changes to the software. Cardano is one of the biggest blockchains to successfully use a proof-of-stake consensus mechanism, which is less energy intensive that the proof-of-work algorithm relied upon by Bitcoin." Marketcap in December of 2020: ~ $4.7 Billion (coinmarketcap.com).

1.3.8 Binance Coin (BNB)

"BNB was initially launched on the Ethereum network with a total supply capped at 200 million coins, and 100 million BNBs offered in the ICO. However, in April 2019, with the launch of the Binance Chain Mainnet, are now no longer hosted on Ethereum. You cannot mine BNB as you would a proof-of-work cryptocurrency, since the Binance Blockchain uses BFT consensus mechanism. Instead, there are validators that earn from securing the network by validating blocks." Marketcap in December 2020: ~ $4.1Billion (coinmarketcap.com).

1.3.9 Monero (XMR)

"Monero's goal is to allow transactions to take place privately and with anonymity. Even though it's commonly thought that BTC can conceal a person's identity, it's often easy to trace payments back to their original source because blockchains are transparent. However, XMR is designed to obscure senders and recipients alike through the use of cryptography." Marketcap in December 2020: ~$2.7 Billion (coinmarketcap.com).

1.3.10 Dogecoin (DOGE)

"Dogecoin is based on the popular "doge" internet meme and features a Shiba Inu on its logo. Was Forked from Litecoin in 2013. Dogecoin's creators envisaged it as a fun, light-hearted cryptocurrency that would have greater appeal beyond the core Bitcoin audience, since it was based on a dog meme. The cryptocurrency has a block time of 1 minute, and unlike Bitcoin, the supply is uncapped, which means that there is no limit to the number of Dogecoins that can be mined." Marketcap in December 2020: ~ $410 Million (coinmarketcap.com).

1.4 Similarities and Differences Between Other Financial Data

Unlike tradition stocks, cryptocurrencies are traded at all hours of the day and the exchanges they are traded on do not close for holidays or weekends. Cryptocurrencies in general also behavior more similarly to traditional fiat currencies (like the USD) as they are primarily designed to purchase goods and services.

2 Univariate Time Series Analysis

In this section, we not only introduce some fundamental concepts on time series, but we also uncover few core ideas that seem to be crucial in understanding of modeling time series. Among many other interesting concepts, we will take a tour over stochastic processes and their related moments, the auto-correlation and partial auto-correlation. We then investigate financial time series and the stylized facts they exhibit. Finally, we will conduct an empirical study of a financial stock to shed light onto few of their behaviors.

2.1 Introduction

Loosely speaking, a time series is identified to an ordered set of observation that have been collected through equally spaced time intervals. In this era of data deluge, time series are ubiquitous and touch at every domain. Almost every human activity is in one way or another confronted the extensive use of time series. From pure science to engineering, from medicine to social sciences, from industry to academia; it is almost difficult, if not impossible, to not find at least one variable that is not measured sequentially through time. For example, in physics time series arises quite often when studying very dynamical complex systems. In engineering, electricians are always engaged to better understand time-dependent aspects of power flow for long time in a fixed interval of time period. In medicine, doctors daily or weekly conduct interval measurements of some patients' blood. In social sciences, the population growth rates is yearly measure to be able to prescribe some recommendations for the future. In finance and economics, the daily, weekly and monthly prices of stocks are constantly under investigations for better investment plans. In the industry world, some scientists tirelessly observe the time evolution of the densities of plasma. Due to the very intense presence of time series in our daily life, understanding and modeling has been attracting various communities of researchers and practitioners,

Depending on the data collection method, time series are divided between two grand classes; some are discrete and others are continuous. In discrete case, the data is recorded in evenly spaced time intervals, meanwhile in the continuous case where the measurement is randomly conducted through time.

Without loss of generality, the principal motivation behind studying time series are to: 1-) have a well-defined understanding of the mechanism that generates the series, 2-) understand the underlying structure of the series, 3-) be able to foresee the future dynamics of the series via the computation of possible future values of the series, and 4-) enable us control the system that produce the series, Wei (2006).

To respond to the first two motives as cited above, great efforts have been brought to effective actions, which led to the results that "It is said that time series exhibit very complex structures made up of four major components: trend, cyclic, seasonal and irregular".

2.1.1 Trend Component

The trend displayed by a time series is the long-term general direction or movement taken by the series in a duration longer than a year. It often refers to historical changes in the observed historical data of a series. Also, it strongly suggests a view about the future dynamics of a time series. A trend can be linear, quadratic or exponential. A series with a linearly increasing long term trend is said to have a positive trend, whereas a linearly decreasing long term trend indicates a negative trend.

In the case neither of these two cases is apparent, the series is generally called to be stationary around the mean. More details about stationary series will be given later on.

2.1.2 Cyclic Component

The cyclic component of time series is defined as the long-term oscillations around the mean of the series. In general, such features are prominent in financial and economic data. They usually describe the successive expansion and contraction of a given business or the economy as a whole. Precisely, they inform about the well be of the business or the economic system under investigation. One should be aware that the cycles and their lengths are business dependent. Nevertheless, empirical observations have demonstrated that, cycles generally take place every over two years.

2.1.3 Seasonal Component

With time series, the seasonal variations are regular events of same lengths that occur constantly during the period of time (year, month, semester, quarter). They are short terms movements that are related to some seasonal factors. For example, it is a lapalissade that most of gift shops live a remarkable increase in sell around Christmas time. seasonal variations in a series are generally caused are: climate and weather conditions, customs, traditional habits, etc Adhikari and Agrawal (2013).

2.1.4 Irregular Component

This component of a series is obtained after removing its cyclic and seasonal components. This allows to say that the irregular component of a series is assumed to be captured neither by the seasonal nor the cyclic component of the series. In point of fact, it corresponds to the unexpected shocks that occur in a time series. It occurs randomly and gives to time series their unpredictable characteristic. Furthermore, due to the irregular component of time series data, they are mathematically modeled as random variables.

Without prejudice a time series is described to be an additive or a multiplicative combination of the four components mentioned above. In mathematical notation, this is usually written as

$$TS(t) = T + C + S + I \qquad (1)$$

or

$$TS(t) = T * C * S * I \qquad (2)$$

wherein eq(1) describes the additive relationship between a given time series and its various components; and eq(2) represents the multiplicative one. In each of the two equations TS(t), indicates the observed value of the series at the instant t , and T, C, S, and I respectively stand for the trend, cyclic, seasonal and irregular components of the series under study. The key assumption that holds the additive model is the four components that comprise a time series are independent from each other. This could be supported by the mathematical equation that formulates the model as expressed above. Such a model may be appropriate if an increase or a decrease in the values of the series does not lead to a change of the amplitude of the seasonal effects. In other terms, if the magnitude of the seasonal effects of the series does not depend on the magnitude of the values obtained by the series. Oppositely, the multiplicative model assumes that the four components of the series are not imperatively independent. In a certain sense, one component can affect another. Thus, it common to hear that a multiplicative model may be well suited if the seasonal effects depend on the magnitude of the values achieved by the series.

In addition to these components, the complex structure of time series has to do with the fact that any of the observed value of the series at time t is affected by its values at time t - 1. This results in the concept of auto-correlation or serial correlation in time series data.

The concept of auto-correlation in time series data dictates the necessity of considerable attention while dealing with time series data. Hence, over the past decennials consequential efforts have been spent by researchers for the development of efficient approaches to improve our understanding of time series. As a result, the birth of “Time series analysis took place”.

Time series analysis is the harmonious use of a set of statistical techniques on time series data in order to improve our knowledge and strengthen our understanding on a given data. The purpose is to achieve four principal objectives as follows:

- Data compression - aims at providing a reduced representation of the data
- Explanatory - aims at getting insights to the data. This can be done by using many techniques such as conducting univariate (for the presence of any seasonal and cyclic components) and bivariate studies of the variables present in the data in order to detect any significant relationship between them.

- Signal Processing - refers to extracting the signal from impurities(noise). This enhances us to communicate on the data. It frequently involves some mathematical transformations or manipulations on the collected data. In the process of extracting signals, some do it in time domain and others in frequency domain. The preeminent examples of such techniques are Wavelets and Fourier Analyses.
- Prediction or forecasting - consists at building a reliable model that can be used to tell something about the dynamics of the series in a future horizon considering what happened in the past. Precisely stated, it is the use of a mathematical or computer model to predict future values of the series based on some historical data.
- Meanwhile, the main goal of time series analysis is not only to develop, but to also provide mathematical models that allow to achieve plausible representations for sample data Shumway and Stoffer (2017). A fundamental part of time series analysis is the prediction, which tries to learn from the past in order to take a jump in the future. In time series Later on, a great amount of time will be spent on in the context of the market of stocks.

More well formed information about techniques to analyze time series can be found in Anderson (1976), Li and McLeod (1986).

2.2 Mathematical Background

In what follows, let us consider a stochastic process, $\{X_t: t = \pm 1, \pm 2, ...\}$ to model a given time series. For simplicity's sake, let us assume that we are dealing with one dimensional and discrete-time stochastic process whose probability distribution is determined by the set of distribution of all finite collected values of all the Xs. As learned from undergraduate probability class, the joint distribution of all those random variables is generally.

2.2.1 Descriptive Statistic

For such a process, let's define the mean function as

$$\mu_x(t) = E(X_t), t = \pm 1, \pm 2, \ldots \tag{3}$$

Where at every single time step t, μ_X represents the expected value of our process which takes different values at different moments. One routinely comes across the expected value of a stochastic process being called as the first moment when it is referred a random variable. Also known as the first moment, it measures the central location the data under investigation. The second moment or variance of any stochastic process (or any other random variable) is expressed as

$$\sigma_X^2(t) = Var(X_t) = E\left[\left(X_t - \mu_x(t)\right)^2\right] \tag{4}$$

This quantity gives information on the spread or the variability of the data. In practice, the first and second moments are enough to describe any normal distribution. For complex data with complicated distributions, one should think of using higher order moments to summarize the shape of distribution of the data.

The skewness of the process, known as the third moment, is one of the higher order moments that are extensively used for complex data distributions. It is specified by

$$skew(X_t) = \frac{E\left[\left(X_t - \mu_x(t)\right)^3\right]}{\sigma^3} \tag{5}$$

It measures the symmetry of the distribution of the process or the random variable with respect to its mean. Symmetric distributions have zero skewness, while distributions with a long tail in the positive x-axis direction have positive skewness and those with a long tail in the negative x-axis direction have negative skewness.

Following is the Kurtosis, known as the fourth moment

$$kurt(X_t) = \frac{E\left[\left(X_t - \mu_X(t)\right)^4\right]}{\sigma^4} \tag{6}$$

It tells us about the behavior of the tail of the distribution of a given process or random variable X. It measures the flatness of a distribution. If Kurt(X)=3, we say that the process is normally distributed. Whenever Kurt(X) > 3 or Kurt(X) - 3 > 0, we talk of the existence of an excess kurtosis or positive kurtosis. In this case, we say that the distribution of X has a tail fatter than that of the normal distribution. Such a distribution is referred to as "leptokurtic". On the other hand, a distribution that has negative kurtosis displays short tail. In such a case, the distribution is referred to as "platykurtic".

In toto, the third and fourth moments are used to measure the extent of asymmetry and tail thickness.

Of course, as mentioned above, the mean and variance are important quantities in the study of time series. However, it remains true that other quantities are needed for complex distributions. Other concepts that plays key role in time series study is those of the auto − covariance and auto − correlation. Before we dive into those concepts, let us review some mathematical notions in probability theories and statistics. Consider that we have two random variables X, and Y.

2.2.2 Covariance

The covariance between X, and Y is a quantitative measure of the joint variability between X, and Y defined by .

$$\gamma(X,Y)=\operatorname{cov}(X,Y)=E\left[\left(X-E\left[X\right]\right)\left(Y-E\left[Y\right]\right)\right]=E\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right] \qquad (7)$$

The above mathematical equation can be interpreted as the measure of the interaction effects between the deviation of the two variables from their respective expected values. In other terms, it can be seen as a quantitative measure of how much each of the variable deviates from their own mean and from the mean of the other variable as well. This is to say that the two variables simultaneously go above or below their expected values, then the covariance assumes positive values. On the other hand, the covariance is said to be negative if one the variables jumps up above its expected mean while the other variable jumps down its expected value. Thus, it can be stated that the behavior of the two variables with respect their means, guides the tendency of the linear relationship that relates them.

2.2.3 Correlation

The covariance, as just defined, is quite cumbersome to be interpreted since the magnitude of the covariance depends with fidelity on the variables. To palliate to this drawback, it is common to encounter the use of correlation coefficient.

Correlation is the normalized version of the covariance. It is obtained by dividing the covariance between two variables X, and Y by the product of the standard deviations of the two variables. Mathematically, it is expressed by

$$\rho(X,Y) = Corr(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(x)Var(Y)}} = \frac{\gamma(X,Y)}{\sqrt{\gamma(X,X)\gamma(Y,Y)}}$$

where $\gamma(X,X) = Var(X)$, and $\gamma(Y,Y) = Var(Y)$

The values of the correlation lay in the interval [-1,1]. When $\rho(X,Y)$ is 1 or -1, we say that that there is a strong dependence between the two variables. In contrast, we say that there is no linear relationship between them, if ρ assumes the zero value. In such a scenario, we formally say that X and X are uncorrelated.

A discussion about the mathematics surrounding time series will not be complete by omitting or disregarding stationarity of time series data since it is an extremely useful concept to time series analysis.

2.2.4 Stationarity

Now that we have covered some basic aspects of stochastic process or random variables, we are in a better shape to talk about stationarity in time series.

In the most intuitive sense, a process $\{X_t\}$ is said to be stationary if the probabilistic laws that governs its behaviors do not change over time Maindonald (2009). As, at the basic level, it is a common practice to describe a stochastic process in terms of the first four moments, a process could be said to be stationary if its stationary statistical properties such as mean, variance, autocorrelation, etc. do not change over time. Mathematically speaking, this can be expressed as $E[X_t] = E[X_{t-k}]$, $Var[X_t] = Var[X_{t-k}]$,and $\gamma_{t,t-k} = \gamma_{0,k}$. Such a stationary is known under the name of weak stationarity. Thus, it can be stated that for a weakly stationary process X_t the following conditions Brockwell and Davis (2016):

(i) $\mu_X(t) = \mu$, a constant that is independent of t

(ii) $\sigma_X^2(t) = \sigma$, a constant that is independent of t

(iii) $\gamma_X(t + k) = \gamma_X(k) := \gamma_X(k,0)$, independent of t for each lag k

It results from eq(iii) that the covariance function, γ_X, of a stationary time series X_t solemnly depends on the

lag k Brockwell and Davis (2016).

At the mercy of eq(ii) and eq(13), we can express the autocovariance function (ACVF) of a given process $\{X_t\}$ at a certain lag k as

$$\gamma_X(k) = \gamma(k) = Cov(X_t, X_{t+k}) = E\left[(X_t - \mu)(X_{t+k} - \mu)\right] = E(X_t X_{t+k}) - \mu^2 \quad (9)$$

From eq(9), one can see that if k = 0, we have $\gamma(0) = E(X^2_t) - \mu = Var(X_t)$.

From eq(8) and eq(9) with the setting X = Y, we have to the autocorrelation function (ACF) of $\{X_t\}$ at lag k defined as

$$\rho_X(k) = \rho(k) = \frac{\gamma_X(k)}{\gamma_X(0)} = \frac{\gamma(k)}{\gamma(0)} = Corr(X_t, X_{t+k}) \quad (10)$$

The auto-correlation function (ACF) at lag k overflows with great proprieties Garg and Wang (2005): 1-) It allows to measure the correlation between a series with a shifted copy of itself as a function of the lag k. Sated otherwise, it aims at measuring the dependence of values of the series that are k time points apart from each other. 2-) It is an even function, which means that measuring the similarity between X_t and X_{t+k} is the same as between X_{t-k} and X_t. 3-) The auto-correlation function has its maximum magnitude lag k = 0, that is $abs(\rho(k)) \leq \rho(0)\ \forall\ k$.

Later, we will learn more details about auto-correlation, also known as serial correlation.

On the probabilistic ground, the equivalent definition of a stationary process refers to a process whose joint distribution of $X_{t_1}, X_{t_2}, X_{t_3}, \ldots, X_{t_n}$ is the same as the one for $X_{t_{1-k}}, X_{t_{2-k}}, X_{t_{3-k}}, \ldots, X_{t_{n-k}}$ for any time choices t_1, t_1,...,t_n and for any time choice in the past denoted by lag k. In this scenario, we talk about strict stationarity.

On the statistical ground, Hipel and co-researchers introduce a stationary process as a form of statistical equilibrium since the statistical properties do not depend on time, Hipel and McLeod (1994).

For a better understanding of those complex mathematical concepts as developed earlier, let us take a look at their sample analogues. To do so, let us assume that we have a set of observed data sample $\{x_t\}$ where t is between 1 and some integer T, of a stochastic process X_t. The corresponding statistical measures defined above are called sample statistics. They are recognized as :

$$\mu_x = \bar{x} = \frac{1}{T}\sum_{t=1}^{T} x_i \tag{11}$$

$$\sigma_x = s_x^2 = \frac{1}{T-1}\sum_{t=1}^{T}\left(x_t - \bar{x}\right)^2 \tag{12}$$

$$skew_x = \frac{\frac{1}{T-1}\sum_{t=1}^{T}\left(x_t - \overline{x}\right)}{s_x^3} \tag{13}$$

$$kurt_x = \frac{\frac{1}{T-1}\sum_{t=1}^{T}\left(x_t - \overline{x}\right)}{s_x^4} \tag{14}$$

One should note that the statistical quantities in eqs(11,12,13,14) with the hat sign on top are nothing more than the values in eq(3,4,5,6) but within a smaller sample size setting. They represent the sample estimates of the corresponding population quantity. In precise terms, eq(11) computes the sample mean, which is the sample estimate of the expected value of the population. It aims to measure the distribution of the sample data. Eq(12) shows the sample estimate of the population variance. It indicates how spread the sample data are around the sample mean. In eq(13), we have the sample estimate of the population skweness. This helps investigate how symmetric the sample distribution is. Lastly, eq(14) demonstrates how to compute the sample estimate of the population kurtosis. With this quantity, we try to measure the fatness of the sample distribution.

Obtaining the quantities computed in eq-11,12,13,14 is of great importance as they equip one with necessary tools, at least at the basic level, to make some statistical inference about the structure of a random variable with the utilization of some observations of the given variable.

The sample analogue of the auto-covariance function, known as the sample autocovariance function (SACF), is defined by

$$\gamma(k) = n^{-1}\sum_{t=1}^{n-|k|}\left(x_{t+|k|}-\bar{x}\right)\left(x_t-\bar{x}\right), -n<k<n \tag{15}$$

The associated sample autocorrelation function (SACF) is expressed as:

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)}, -n<k<n \tag{16}$$

SACF, in time series, plays a capital role be it in assessing the dependence in the data or in selecting a model an adequate model that best fits a given set of observations. For For a stationary process, SACF will provide us with an estimate of the ACF. This estimates guide one to the choice of the adequate model among many others.

While analyzing or modeling time series data, it is common practice to assume the processes generating time series being stationary. Such a simplification comes with multiple advantages. First of all, stationary processes are much easier to deal with since the complexity of the mathematics that is involved is considerably reduced. Second, with stationary processes, it easier to predict that their statistical properties in future time will not be much different from the ones they had in the past time. Third, with some sample records, it is untroublesome to achieve reliable inferences about the structure of the process. All in all, in order to design an adequate model that aspires to achieve better forecasting tasks, it is almost indispensable to have an underlying time structure that is stationary Adhikari and Agrawal (2013).

However, one should be aware that, in nature, time series data such as in finance and economics, are found to be non-stationary. They either exhibit either trend or seasonality. To detect those non-stationary features in time series data, one can rely on visual and mathematical techniques.

The first technique aims at searching for any non-stationary behavior such as trends and cycles via graphical representation and visualization tools, while the second one uses rigorous tests. Among many others, the most prominent is the one proposed Duckey and Fuller Cochrane (2005).

Face to non-stationary time series, steps must be taken to make them stationary. In actuality, this is achieved through mathematical transformations. To achieve stationary series, researchers and practitioners routinely use the differencing method. In some situations, the logarithmic transformation is found to be useful in making time series stationary. This transformation is generally used whenever the series displays a highly skewed distribution. In general, the logarithmic transformation is also used whenever the series displays higher levels. Since more variation is seen with series with higher levels, applying the logarithmic transformation to the data provides us with a new series with constant variance over time Shumway and Stoffer (2017). It is quite frequent also to see a two-step approach to make a series stationary. The logarithmic is applied first, followed by the differencing operation. Such a technique is generally used whenever we are dealing with series that is evolving through time in the exponential fashion Shumway and Stoffer (2017).

In addition, parametric families of data transformation are used to remedy to the non-stationarity issues present in time series. Among those families, the one that has received popularity is the Box-Cox power transformations that are defined as:

$$h(x,\lambda)=\begin{cases}\dfrac{x^{\lambda}-1}{\lambda}, & \text{for } \lambda\neq 0\\ \log x, & \text{for } \lambda=0\end{cases}$$

The first equation of the system operates in three steps. First, each value of the data is taken to the power λ. Second, each of the transformed value is shifted by subtracting 1. Third, the obtained values are rescaled by divided them by λ. The rescaling aims at making h(x,λ) converges to log x when λ goes to 0.

Different values for λ lead to different transformations. More details about the effects of λ can be found in the research paper produced by Ruppert Ruppert (2004). It shall be noted that one of the major conditions for using Box-Cox's power transformation is to have positive data values. In the case negative or zero values are present in the data, they should be shifted by adding a positive constant value before applying Box-Cox's power transformation.

2.3 Time Series Modeling and Forecasting

Now that we have we got the grip on time series, we can start with the forecasting stage, the heart of time series and time series analysis.

2.3.1 Time Series Modeling

Time series modeling is one of the key steps in the process of analyzing time series. It is a pervasive practice that consists of fitting a suitable model to a given time series in order to understand the underlying structure of a time series. Constructing time series models traces its roots back Yule, who in 1927 wrote one of his famous papers that analyses the sunspot Yule (1927). For many, this was likely the starting point of time series modeling, which furthered to the takeoff of the whole theory of linear auto-regressive models. In its turn, linear auto-regressive models in its concepts and structure facilitated the rapid bloom of non-linear time series models.

In one way or another, modeling time series comprises steps that require selecting a suitable model for the data with the intent of giving a better explanation of the series based on theoretical foundations and mathematical reasons. This is done by fitting either a single model or a class of models to a given time series data, which leads to the estimation of the parameters that constitute the model(s). In the case of one single model, the estimated parameters are then used to presage the behavior of the series in the future. In the case we have a class of models that are fitted into the data, the one that outputs estimated parameters with less errors is chosen for predicting the next values of the series.

In general, the resulting method should support few rules and principles. A suitable model must fulfill some criteria Frohn (1995):

- capable of well capturing the generating process
- theoretical soundness
- reliable parameter estimation
- simplicity

Lately, the principle of simplicity(model parsimony) of time series model has received an increasing awareness. In the process of building a time series model, the principal of parsimony supports that one should keep the model as simple as possible Chatfield (1996), Zhang (2007), Zhang (2003), Adhikari and Agrawal (2013). By making the model simple, we mean to reduce the number of possible parameters that are needed to capture the underlying structure of the data. In doing so, one should be very careful as a model that is too simple might not have the full capacity to decrypt the most fundamental look of the series. Such scenario is knwon as underfitting. The opposite of underfitting overfitting, which occurs in the presence of very complex models. Complex models have the tendency to easily learn from noises which can have serious impacts on prediction accuracy of the model. Consequently, one should always aim at finding a middle ground between parameters that result in underfitted and those that give overfitted models.

2.3.2 Time Series Forecasting

Forecasting is a step with a major importance in time series analysis. It is a practice that involves discovering a solid pattern in a given historical data(past values of a series) that aims to be used for extrapolating the values of the series in the future. Put in an other way, suppose that we have a given observed sample data $\{x_1,x_1,...,x_T\}$ of a stochastic process $\{X_t\}$, and we are tasked with knowing the future values of X_t at a certain time point j. Let us look at few examples before drawing a general formula that responds to all the the cases.

Given $\{x_1,x_1,...,x_T\}$, i.e based on all the available information up to time T, we aim to know the value x_{T+1}. This is denoted by

$$E(x_{T+1}|x_T,x_{T-1},...,x_1)$$

Similarly, after knowing x_{T+1}, the next value of the series, x_{T+2} is found via

$$E(x_{T+2}|x_{T+1},x_T,...,x_1)$$

Iteratively, the value at time j is determined by

$$E\left(x_{T+j} \mid x_{T+j-1}, x_{T+j-2}, \ldots, x_1\right)$$

We learn from the above expressions that a new foretasted value of a series not only depend on the past values of the series, but also on past forecast errors. The use of the past observations of the series is to be able to develop a mathematical model that optimally captures the underlying structure of the generating process Adhikari and Agrawal (2013), Zhang (2007), Park (1999) . Once an optimal model is found, it can be used for prediction of future events.

Forecast methods can come into three types:

- Judgmental forecasting - is a forecasting type that mainly depends on the judgment, intuition, emotion, apprehension, comprehension and the degree of anticipation of the forecasters. Judgmental forecasting is a widely used when the forecasters do not hold much information about the process, i.e. enough historical data, or when the forecaster has record on conducting such an activity, or the environment is relatively stable Kavanagh and Williams (2014), Lawrence et al. (2006). With this forecasting method, it is quite a difficult task to best address the human component of judgmental forecasting. This comes as major weakness of this method since it involves substantial risks to the accuracy of the prediction. However, researchers have provided that the accuracy of can be improved if the forecasters have considerable amount of domain knowledge and more historical data are available Lawrence et al. (2006).
- Univariate forecasting - refers to the forecasting of a univariate time series, which is given as a single column of observed numbers. With this type forecasting method, the forecasted values depend on the actual and some past values of the actual series itself that is being investigated, the one being predicted. This method relies on some past values of the series, one or more mathematical or statistical models that well capture the dynamics of the data generating process. The adopted model is then used to predict reasonable. Univariate forecast methods are more reliable if sufficient amount of observed data is available. Unlike Judgmental forecast methods which are rooted is judgment, one of the advantages of univariate forecasting methods is that the resulting predicted values are more or less accurate since they results from a rational study or rigorous analysis of pertinent patterns in the data. However, univariate time series analysis has also been panned as it comprises of studying a single variable while not taking into consideration some interesting aspects found in real world data such as causes or relationships. For example, in finance and

investment portfolio construction, it will be always wise to look at relationships or correlation among different stocks before putting them together. This leads us to the concept of portfolio diversification, which will, later, be discussed.

- Multivariate forecasting- is used in situations where there is more than one-time dependent variables. In such a case, forecasters not only care about the past values of each variable but also on the dependence between the variables present in the study. That is, the patterns and relationships between the existing variables in a data set are used to predict the future values of each of the variables. It provides a tool to predict the effects of a small change in one variable on the others. In general, univariate time series analysis is first conducted. Then, knowledge and understanding gathered from univariate serves as a precursor to multivariate study. Since taking into consideration the dependency of all the variables is at the center of multivariate time series forecasting, this gives a considerable advantage over univariate time series forecasting. However, the latter forecasting method is exposed to more complication such as the increasing number of parameters and the identifiability problem Tsay (2013). Tsay in his multivariate time series analysis book proposed a cascade of techniques and models in order to address some eventual undesirable outcomes while dealing with data with multivariate time series Tsay (2013).
- Regardless of the forecasting method that is used, many researchers consider that the main objective of forecasting is to study the dynamical structure of processes. A group of people achieve this objective by using emotion. Another group does it by studying the variables individually.The third group of people proceed in studying the relationship between variables, but also in understanding whether or not they are dynamically interconnected.

3 ARIMA

3.1 Introduction

As mentioned earlier, financial stocks and the predictions of their prices play an important role in today's economy. Due to asymmetrical behaviors and numerous uncertainties, stock price prediction remains one of the most challenging activities in financial forecasting Dong et al. (2017). Thus, constructing efficiently functioning and predictive models for stock prices has been of great interest to researchers. Especially, investors have been interested in such models for better investment decisions, effective strategies, higher profits and minimal risks Ariyo et al. (2014). As a result, we are witnessing an unprecedented proliferation of financial models among which autoregressive moving average (ARMA) has gained notoriety in financial forecasting Siami-Namini et al. (2018) due to its flexibility and capability in representing different types of time series Chen et al. (2014). As reported earlier and in literature, time-series data exhibit two major problems: linearity and nonstationarity. These represent serious challenges to the utilization of ARMA across areas in while dealing with real-word time series. Despite its popularity and success, ARMA has faced the same critiques Chen et al. (2014). In this regard, autoregressive integrated moving average(ARIMA) models have been proposed to overcome some major weaknesses and flaws of ARMA models.

ARIMA models constitute a class of mathematical models for analyzing autocorrelation in temporal data and making predictions based on past behaviors of a given process. Conceptually, an ARIMA model is the combination of an autoregressive(AR) process and a moving average(MA) process. The AR process assumes that the value achieved by a variable at a given time t is achieved via a linear combination of its lag values, whereas the MA process takes an observation as the linear regression of the residual errors. Such a concept was first introduced by Yule, Slutsky, Walker and Yaglom Chen et al. (2014). In ARIMA models, making non-stationary time series stationary is considered. As stated earlier, stationarity can be induced in a non-stationary data via a variety of mathematical transformations, of which differencing is accepted to be the most common technique. The main advantage of differencing is to eliminate the influences of the trend from the data Chen et al. (2014).

3.2 Mathematical Background

Let us assume that we are dealing with a time series defined by a stochastic process $\{X_t\}$. ARIMA models are presented as a combination of Autoregressive (AR) process and Moving Average (MA) to achieve a mixed models that captures the dynamics of the series X_t.

- Autoregressive (AR) - A regression model that supposes that the value of a process at a given t might be expressed as a linear combination of the previous values of the process up to time t and some random errors. To this end, it shows the the dependencies between an observed value and previously observed values. A process that is expressed $X_t = \alpha X_{t-1} + \varepsilon_t$ assumes that today's value of the process is its yesterday's value multiplied by a certain constant plus 1 the stochastic error at that time. Such process is named as an AR model of order 1,AR(1) . More generally, an AR model of p is given by :

$$
\begin{aligned}
X_t &= c + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \cdots + \alpha_p X_{t-p} + \varepsilon_t \\
&= c + \sum_{i=1}^{p} X_{t-i} + \alpha_t \\
&= c + \sum_{i=1}^{p} \alpha_i B^i X_t
\end{aligned}
$$

wherein X_t represents the stationary process under study, c is constant, a specific α_i denotes autocorrelation coefficient at lag i, and ε_t goes as $N(0,\sigma_\varepsilon^2)$. The first equality of eq(1) gives the extended form of the formula, the second equality is a compacted version, and the third equation is an equivalent expression that uses the backshift operator B. The simplest autoregressive process is AR(0). With such a model, there is no dependence between the terms.

- Integrated (I) - Indicates the differencing of the original time series data. It is to replace an observed value by the difference between that value and the immediate data value. The motivation behind this is to transform a nonstationary process to a stationary one. A non-stationary data can be differenced more than once to achieve stationarity. For example, first-differencing a time series conduces to the removal of the trend components of the series; twice differencing will remove the quadratic trend. Alternatively, the first difference can be written as $\nabla X_t = X_t - X_{t-1}$. The second difference is $\nabla(\nabla X_t) = \nabla^2 X_t = (X_t - X_{t-1}) - (X_{t-1} - X_{t-2}) = X_t - 2X_{t-1} + X_{t-2}$. With ARIMA models, over-differencing and under-differencing can lead to serious problems.

- Moving Average (MA) - A modeling technique that consider the dependency between the observed data and the residuals errors obtained from fitting a moving average model to previous observations. A first-order moving-average process, written as MA(1), has the general equation $X_t = \mu + \theta_0\varepsilon_t + \theta\varepsilon_{t-1}$. The q^{th} order moving average model, denoted by MA(q) is :

$$
\begin{aligned}
X_t &= \mu + \varepsilon_t + \theta_1\varepsilon_{t-1} + \cdots + \theta_q X_q + \varepsilon_{t-q} \\
&= \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i\, \varepsilon_{t-1}
\end{aligned}
$$

$$= \mu + \mu_t + \sum_{i=1}^{q} \theta_i B^i \varepsilon_t$$

In eq(2), μ indicates the expectation of the X_t. In general, μ is assumed to be equal to zero. For any i, θ_i denotes the coefficients of the model, ε_t is assumed to be $N(0,\sigma_\varepsilon^2)$.

The combination of AR(p) and MA(q) builds an ARIMA model of order (p,q), which models the time series as :

$$X_t = c + \sum_{i=1}^{p} \alpha_i B^i X_t + \varepsilon_t + \sum_{i=1}^{q} \theta_i B^i \varepsilon_t$$

Where for any i, α_i and θ_i are non-zero coefficients.

The ARIMA modeling approach, commonly known as Box and Jenkins methods, involves three main stages. The first step is for model identification, which consists of guessing an eventual model that is assumed to be the model that fits the data the best. The chosen model can be an AR, MA, or ARIMA model. One way to choose the appropriate model is by examining the graphs of ACF and PACF functions Mishra and Desai (2005). Without of loss of generality, ACF function identifies the appropriate lag q, whereas PACF function helps find the appropriate lag p in an ARIMA model. For an AR(p) model, the ACF function progressively declines in from its highest value at lag p while its PACF function abruptly after lag p. On the other hand, the ACF function of an MA(q) abruptly cuts off after lag q, whereas its PCAF function declines in geometric progression from its highest value at lag q.

Another way is to construct an array of ARIMA models, where each model is defined by a specific (p, d, q) model. Iteratively, each of the models is fitted to the series. Finally, the model with that gives the minimum Akaike Information Criterion (AIC) is selected as the best-fit model Faruk (2010).

The second stage of the ARIMA modeling is the parameter estimation. This consists of estimating the coefficients present in the model. In recent times, various estimations techniques have been proposed including method of moments, least squares estimation, and maximum likelihood methods Cryer and Kellet (1991). Introduced in 1887 by Pafnuty Chebyshev, the method of moments is an estimation technique that relies on the translation of well known information about a given population to a sample of the population. Fundamentally, the method moment is an estimation method that is associated with the law of large numbers. The main idea is to express the sample moments and equate them to their theoretical corresponding Cryer and Kellet (1991). A common use of the method of moments is to estimate a stationary process mean by a sample mean Cryer and Kellet (1991). The main advantage of such an estimation method lies in its simplicity. However, with the method of moments, the chance to get inaccurate estimations is very high mostly with small sized-samples. In addition, the method does not consider all the information in the data, which can lead to the violation of the sufficiency principle.

The Least Square Method (LSM) method is a technique that attempts to estimate the parameters of a model by minimizing the squared discrepancies between observed data and their expected values Time et al.. LSM is considered as the oldest and most popular technique in modern statistics. One of its main advantages is that it is an optimization that allows us to achieve optimal estimators. Another of its advantages is that it deals well with complex models, scenarios where it is generally difficult to obtain very optimal estimators. However, LSM is highly criticized for being sensitive to outliers and having the tendency to often overfit. Also, LSM suffers from working only around the first and second moments. In other terms, LSM also does not take into consideration all the information in the data.

The Maximum Likelihood Estimation (MLE) is a statistical method for estimating unknown parameters of a given modern based with respect to some given observations. MLE attempts to find the parameter values that maximize the likelihood of attaining the available observed data given the parameters. Unlike the above-cited methods, MLE takes all of the information in the data in how it operates. Other advantages of MLE are its ability to have lower variance than other methods and the fact that it is a method that is statistically well understood. One major disadvantage of MLE is that it is computationally expensive.

The last stage of the Box-Jenkins methodology is the diagnostic checking, which aims at checking the adequacy of a statistical model. In the diagnostic checking, the central motive is to find a superior model based on some fixed purposes. To obtain a dream model, notable mathematical, computational and empirical techniques have been proposed by academicians and practitioners. For this purpose, numerous kinds of Diagnostic of Goodness fit exist according to the motive behind the analysis of the residuals, purpose, and need of having it. Regardless of the objectives, with appropriate models, the residuals should not display any traceable patterns. As such, there should not be any correlation structure in the residuals. Thus, to check on the independence on the independence of the noise in a given model, many techniques have been deployed. Some rely on the graphical representation of the residuals and other focus on the plot of the residual autocorrelation and partial autocorrelation at a certain number of lags. The first method checks on the plot of the residuals over time to see whether or not the appropriate model has been found. For an appropriate model, the residuals are expected to be quasi-evenly distributed around the horizontal axis fixed at zero.

With the second method the residuals of the fitted model are investigated with the ACF plots. In this, the ACF versus lags is plotted. Such a plot displays some dotted horizontal lines which denote the interval $1/_T$ +/- $1.96\, 1/_{\sqrt{T}}$ where 1.96 is the critical value at 5% level. Any autocorrelation value that falls outside this interval is considered to be statistically significantly different from zero at the 5% fixed level.

This technique reveals if there is any autocorrelation structure remaining in the residuals. For an adequate model, all the autocorrelations for the residuals are expected to be insignificant. In the case the residual autocorrelations are significant, one needs to try a different model.

Though these techniques have gained some popularity due to their simplicity, it is important to stretch on the fact they can deliver misleading results. The major downside of using them is that they only display the significance at the individual level but not jointly. Stated otherwise, though investigating residual correlations at individual lags, it is wiser to have means that allow us to take into consideration their magnitude as a group. The interest of this thinking resides in the fact it is always possible to have residual autocorrelations that are moderate, even more close their critical values, but as a group they might high chance to explode Cryer and Kellet (1991)

To address this issue, the desire of developing more robust diagnostics tests based on the auto-correlation with theoretical justification has been continuously fulfilled.

3.3 Literature Review

Over the years, ARIMA models have been applied in a wide range of disciplines and areas. An intensive review of the literature shows that:

In Dong et al. (2017), Yichen et al. present an extensive process of building a financial predictive model by using the ARIMA model on Apple data. The authors found that in the short-run forecast, ARIMA has all the abilities to compete with many other predictive models. Thus, it can be used as a good tool for making investment decisions.

In Edward (2016), Aloysius et al. forecast stock prices in the automobile sector. For, the historical prices of 4 different companies were collected for 8 years. The forecasting models ARMAs were deployed to forecast the stock prices. The collected data were partitioned 70-40, where 70% of the data were used as teacher data and 40% of them as student data. They found that ARIMA(1,1, 0) fits the 8 time series data the best. Also, they found a prediction accuracy that is higher than 75%.

In Mondal et al. (2014), a study of the effectiveness of the ARIMA models in foresting stock prices is conducted. In this study fifty-six (56) stocks from 6 sectors in the Indian market of stocks were used. The chosen sectors are information technology (IT), infrastructure, bank, automobile, power, fast-moving consumer goods (FMCG) and steel. Twenty-three months of historical price data were collected to conduct the empirical study. Numerous ARIMA models were constructed by choosing an array of values for the parameters that define an ARIMA model. The best model that fits the most each of the series was performed by using the Akaike Information Criterion(AIC).

As result, the authors found that a high performance of the ARIMA model no matter what the sector is. An intersector performance comparison indicated that the ARIMA model attained high accuracy in FMCG than any other sector, whereas it performed poorly in the automobile sector.

In Ozturk and Ozturk (2018), Suat et al. foretasted the energy consumption of Turkey via the deployment of the ARIMA models. In this study, 45 years (1975-2015) of coil, oil, natural gas, renewable, and total enery consumption data were collected. It was found that a single absolute model does not exist and the ARIMA models are purely and simply data-driven models. They indicated that ARIMA(1, 1, 1), ARIMA(0, 1, 0), ARIMA(0, 0, 0), ARIMA(1, 1, 0), and ARIMA(0, 1, 2) best fit coal consumption, oil consumption, natural gas consumption, renewable energy consumption and total energy consumption data respectively. They further predicted an increase of the consumption of coal, oil, natural gas, renewable energy and total energy by 4.87%, 3.92 %, 4.39 %, 1.64 % and 4.20 % respectively in the next 25 years.

ARIMA models have been involved in some comparative studies:
In Siami-Namini et al. (2018), a comparison of the accuracy of ARIMA and Long Short-Term Memory (LSTM)-based algorithm in forecasting time series data. It was empirically noticed that the LSTM-based model outperformed the ARIMA statistical model. The authors reported an improvement of the accuracy by 85% by LSTM compared to ARIMA. This result prompted the authors to advocate the superiority of deep learning based-model algorithms over statistical models in the forecasting of economic and financial data.

In Karakoyun and Cibikdiken (2018), the accuracy and the power of the statistical ARIMA models are compared to those of the LSTM, in finding suitable time series models for Bitcoin prices in predicting the Bitcoin for the next 30 days. The chosen models and algorithms were deployed and tested on the Bitcoin prices from April 28, 2013, and October 29, 2017. Of these data, 1646 daily Bitcoin prices were used as teacher data. In their modeling process, it was found that ARIMA(4, 2, 1) is the best ARIMA model that optimally fits the data. With the utilization of different accuracy test results such as MAPE, RMSE, MAE, and MPE, it was empirically found that LSTM outperformed the well the widely used ARIMA models in the prediction of time series data.

In Navares et al. (2018), Navares et al. examined the predicting daily hospital admissions in Madrid. The goal of the project was to provide the best technique, in terms of accuracy, in predicting the admissions to hospitals due to circulatory and respiratory diseases one day-ahead. For, an array of predictive models including a statistical model (ARIMA), two machine learning models such as Random Forest and (RF), and Gradient Boosting Machines (GBM), and Artificial Neural Networks (ANN) were trained and tested. Concerning the accuracy, the authors found that we found that ARIMA models and the ANN over-perform random forests and gradient boosting machines.

Window Size	Accuracy
25	48.90%
50	50.33%
75	54.04%
100	48.84%
125	48.86%
175	51.75 %
200	53.765%
300	54.731

Table 3-1. Mean of the Directional Forecast

Notwithstanding the great success of ARIMA models, they still have some limitations. The model assumes constant variance through real-world financial data exhibits non-constant volatility Petric˘a et al. (2016). Another assumption that considerably goes against ARIMA models is the fact they rely on the white noise terms to be IID or t-distributed Hamilton (1994) Damsleth and El-Shaarawi (1989).

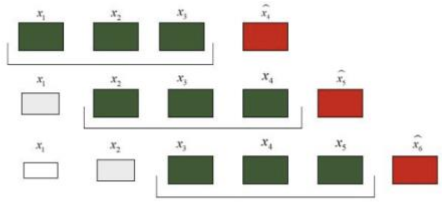
3.4 Motivating Example

In the hope of examining the performance of ARIMA model, we conducted a series of experiments on financial data. For the sake of coherence, let us consider the same AMAZON time series data as used above.

The historical daily price of the stock was extracted from March 9th 2010 to March 9th 2020, having 2514 observations. The forecasting is based on rolling window or look back days methods Hyndman (2014). The rolling window forecasting method takes place in three main steps. Firstly, a window size should be chosen and fixed. Secondly, the chosen window is used to extract the subset of data that will serve for training the model. is continuously side by a unit index. This window size helps to choose the subset of the data that is going to be used for training the model. Thirdly, the next data point is predicted, i.e the one-step forecast is computed on the rest of the data. Fourthly, this sliding window continues till reach the end of the data.

As illustrated graphically below, consider a one-dimensional array where each cell contains a datum of the time series and window of size 3. With rolling window forecasting method, the far we go down in the series, the less impact the first few observations will have on the predicted values. Hence, one can notice the reduction in size of the first few observations in our graphical illustration, Figure 3-1.

Figure 3-1. Rolling Window Forecasting



Several variations of rolling forecast methods have been proposed including one-step forecasts without re-estimation, multi-step forecasts without re-estimation and multi-step forecasts with re-estimation Hyndman (2014), Siami-Namini et al. (2018). The popularity of the rolling window techniques resides in their ability to assess the stability of statistical models over time Zivot and Wang (2007).

To analyze the accuracy of the deployed ARIMA models, several trials were conducted to see how they are able to capture thy dynamics of the chosen stocks. In this process, 8 window (25, 50, 75, 100, 125, 175, 200 and 300) sizes were chosen for the performance analysis. The accuracy of the prediction was measured by using the notion of Directional Forecasting (DF) instead of some of the conventional techniques such as Root Mean Square(RMSE), Mean Absolute Error (MAE), and Mean Square Error (MSE), which have been extensively used in evaluating the forecasting ability of modern predictive models. The core pitfalls of these methods is that they just allow to measure how far off the predicted price is from the actual one. In other terms, they just inform about the size of the forecasting error. In Leitch and Tanner (1991), the authors strongly believe support that the conventional metrics, which rely on the size of the forecast errors, do not have any systematic relationship to profits Leitch and Tanner (1991). They furthermore support that these metrics might be inappropriate since they only refer to point forecast, which is linked to the notion of how closeness of the forecasted value to the real value of the series under investigation at a given time. These result in the increasing use of DF since it mostly captures the future movements of a financial instrument or a market of stocks. St´adn´ık et al. (2013) indicate that DF has recently been the focus of interest of investment companies, individual investors, banks and other financial market participants. In Leitch and Tanner (1991), the authors, via an explicit discussion and an in-depth analysis of the profitability of economic factors empirically, demonstrated the strong relationship between profit of an investment with DF. They also showed a more significant relationship between profit and DF than between profit and traditional accuracy metrics. This being said, it should be noted that the use of DF in literature remains limited. The reduced number of publications using DF in financial forecasting is explained by the challenges that professionals encounter when they try to implement appropriate techniques to test for directional forecasting. However, the Pesaran and Timmermann has been recently gained popularity for being one of the most commonly used techniques in testing for directional forecasting Pesaran and Timmermann (1992).

Table 3-1 displays the effects of the window size on the directional forecasting. Computing the rate of being in the same direction as the market, it appears to find an optimal window size for a better performance of the ARIMA models on financial time series. In addition, the overall performances seem to not be favorable to the simple deployment of the ARIMA models in forecasting financial stocks. For Naylor et al. (1972), the poor performance of ARIMA models in predicting stock price may presumably due to the fact that they : 1- do not have any explanatory power; 2- are rooted into any economic theory; 3- are more smoothing techniques than economic models. Lately, it has also advocated that prediction accuracy of numerous models solely depends on the nature of the historical data under investigation Mendes et al. (2009). To overcome the poor performance of ARIMA models, it has been recently suggested to either use exogenous or explanatory variables, or their cointegration of the two types of variables Mendes et al. (2009), Naylor et al. (1972). Also, it has been proposed to combine ARIMA and dynamic models to better capture the dynamics of data generating process that produces a given time series Naylor et al. (1972).

To get deeper understanding of the dynamic and the structure of the historical price of the AMAZON stock, we decided to conduct a rolling window analysis of the chosen time series, where the window size equal to 100. Such a size, arguably considered as a larger size in term of rolling window, was chosen for the sake of better estimates. This, since it is known that longer rolling window sizes tend to yield smoother values than the ones obtained via the utilization of shorter sizes.

For each of the sub-sampled data continuously obtained from rolling the window, the iterative Box-Jenkins methodology is employed. The optimal model is obtained with the help of Akaike information criterion (AIC). For each of the rolling window, the parameters of the optimal model is then extracted. Table 3-2 shows the frequency distribution of the ARIMA models based on rolling window of size 100 on the historical price of AMAZON stock. The table indicates that the top 5 preeminent models are (0,1,0), (1,0,0), (0,0,0), (1,1,0) and (2,0,0). One surprising point is to see the appearance of the ARIMA(0,0,0) models among the most dominating models. Such a feature appeals to the unpredictability of the stock price, a theory that has been advocated by many researchers as discussed earlier. Another surprising point is the non appearance of the ARIMA(1,0,1) and ARIMA(1,1,1), models that have been lately discussed in ARIMA modeling.

p	d	q	counts	rates
0	1	0	1067548	26.3 %
1	0	0	71683	17.7 %
0	0	0	47665	11.8 %
1	1	0	28628	7.1 %
2	0	0	24034	5.9 %
0	2	0	22506	5.6 %
2	2	0	16240	4.0 %
0	3	0	12941	3.2 %
2	1	0	10003	2.5 %
3	0	0	9387	2.3 %
0	4	0	7878	1.9 %
0	5	0	7757	1.9 %
1	2	0	7407	1.8 %
4	0	0	5728	1.4 %
3	2	0	5368	1.3 %
2	3	0	5432	1.3 %
5	0	0	4215	1.0 %
1	3	0	3793	0.9 %
1	4	0	3086	0.8 %
3	1	0	2829	0.7 %
4	1	0	1934	0.5 %

Table 3-2. Frequency Distribution of ARIMA(p,d,q) Models

4 Multivariate Time Series Analysis and VARMA

4.1 Introduction

Regularly, financial and economic decisions are taken by simultaneously considering many interrelated factors or variables. As a result, the desire to properly study such a complex scenario has grown among financial deciders, economic advisors and policy-makers. In this process, multivariate Statistical Analysis (MVSA) came into the picture as the champion of statistically analyzing of many variables at once Long (2013). Roughly speaking, MVSA is an umbrella term representing a set of statistical theory and methods for analyzing these multivariate or vector of variables. More often than not, some variables are simultaneously collected in discrete and equally spaced time fashion over a long period of time. Variables of this kind are named as multivariate time series. Multivariate time series are important to numerous fields Wan et al. (2019), e.g areology Lajevardi and Minaei-Bidgoli (2008), meteorology Simmonds et al. (2017), finance Wu et al. (2013) and transportation Yu et al. (2017). In jointly studying variables or phenomena of this nature, multivariate time series analysis (MVTSA) has been the most solicited approach. Implicitly to its name, MVTSA can be seen as a branch of MVSA that specifically deals with dependent data Tsay (2013).

Typically, a multivariate time series is of high dimension and exhibit various types of dependence of including temporal and cross-sectional ones Zhao et al. (2018). Temporal dependence concerns each individual series of the multivariate time series while cross-sectional dependence across all the univariate series Zhao et al. (2018). With the large number of involved unknown parameters, the correlation and statistical dependence between different univariate series, MVTSA introduces more complexity and challenges than its univariate counterpart Beukelman and Brunner (2015), Tsay (2013). Nevertheless, MVTSA remains a very interesting subject since it aims to Tsay (2013): 1-) understand the relationships between the different variables that are in the study, 2-) study the dynamic relationships between the variables, 3-) provide prediction of the variables and 4-) improve the accuracy of prediction as this could be valuable in decision-making.

4.2 Mathematical Background

Let us consider $X_t = (X_{1t}, X_{2t}, ..., X_{kt})'$ be a multivariate time series of dimension k or a vector of k time series . In this vector representation, each X_{it} represents a single time series which was lengthy studied earlier. At each point time δ, the observed value of the multivariate time series X_δ is defined by the values realized by each of the variables

$X_{i\delta}$, i = 1, 2,..., k. For the sake of simplicity and mathematical purposes, it is often convenient to agree that each of the component series is on their own a random variable, where the subscript i denotes the position occupied by a specific series in the vector of series and t represents the time at which the measurement has been made.

As with the univariate case, stationarity is a vital condition in modeling multivariate time series simultaneously. In such a case, the notion of stationarity was earlier defined to with respect to some of the statistical features (mean, variance, auto-correlation and partial auto-correlation) of the process under investigation. Extending this approach to multivariate time series allows one to say that the stationarity of a vector X_t of time series data requires its mean vector, correlation matrix function and partial autocorrelation function be time invariant Beukelman and Brunner

(2015). Thus, the k- dimensional time series X_t is said to be weakly stationary if its unconditional mean and variance finite and constant through time, and that the cross-covariance between two component series at different time steps only depends on the difference between two time steps. Mathematically speaking, this could be expressed as follows :

$$\bullet E(X_t) = \mu < \infty \; \forall \, t$$
$$\bullet \Gamma_0 = Cov(X_t) = E[(X_t - \mu)(X_t - \mu)'] < \infty \; \forall \, t$$
$$\bullet \, \Gamma_h = Cov(X_h) = E[(X_t - \mu)(X_{t-h} - \mu)'] < \infty \; \forall \, t, h$$

Where μ denotes a vector of length k composed of the mean of each of the component series, $Cov(X_t)$ a kxk covariance matrix, and $Cov(X_h)$ the cross-covariance at a certain lag h.

Cross-Covariance is another fundamental concept in multivariate time series analysis. Just as with univariate time series, the linear dynamic dependence is also measured by the cross-covariance matrix of a multivariate time series. With high-dimensional time series, the autocovariances of lag h does not obey the symmetry property unlike in the univariate case Tsay (2013). Thus, to find the cross-covariance of lag -h, it suffices to take the transpose of the cross-covariance at lag h. This could be succinctly expressed by $\Gamma_h = \Gamma'_{-h}$. Hence, one can state that the cross-covariance is not symmetric at a lag h > 0.

Last but not least, Cross-Correlation is also an important concept with high-dimensional time series. In MVTSA, the most commonly used method for checking whether two or more time series much up with each other is cross-correlation. This technique is generally used to measure the correlation between a given univariate time series and the lagged copies of others composing a high dimensional time series. For a stationary multivariate time series X_t, the lag-0 correlation matrix is expressed as follows:

$$\rho_0 = D^{-1}\Gamma_0 D^{-1}$$

where D represents the kxk diagonal matrix whose entries represent the standard deviation of $X_{i,t}$, for i = 1,2,... k. For series i and j, we have

$$\rho_{i,j}(0) = \frac{Cov(X_{i,t}, X_{j,t})}{\sigma_{i,t}\sigma_{j,t}}$$

From the above equation, it is obvious that $\rho_{i,j(0)}$ is the same as $\rho_{j,i(0)}$. For any i, $\rho_{i,i} = 1$. This results in ρ_0 being a symmetric matrix with entries in the principal diagonal being equal to 1 with off-diagonal elements describing the instantaneous correlations between all the univariate time series that constitute the high-dimensional series X_t.

At any other lag h > 0, the cross-correlation matrix can be defined as :

$$\rho_h = D^{-1}\Gamma_h D^{-1}$$

Thus, at a given time t the correlation coefficient between two series $X_{i,t}$ and $X_{j,t-h}$ is defined as:

$$\rho_{i,j}(0) = \frac{Cov(X_{i,t} - X_{j,t-h})}{\sigma_{i,t}\sigma_{j,t}}$$

Since $\rho_{i,j}(h)$ and $\rho_{j,i}(h)$ represent different linear dependence measurements and $\Gamma_h = \Gamma'_{-h}$, it can be deduce that $\rho_h = \rho'_{-h}$. In short, one can say that the cross-correlation does not hold the symmetry property at a lag $h > 0$.

Two more important aspects of multivariate time series analysis are linearity and invertibility. More details about these aspects are in given in Tsay (2013), Hosking (1980), and Hosking (1981).

4.3 Literature Review

As much the rapid advent of supercomputers has opened a corridor to a significant data explosion and the eventual related "curse of dimensionality", as much they offered specialized technical skills and equipment such as multivariate analysis to better forecast financial aspects Kumar and Ganesalingam (2001). This attempt to predict a multivariate time series comes down to enhancing the forecasting accuracy for each univariate time series component of the high-dimensional dataset. More formally, suppose that we have a multi-dimensional series $X_t = (X_{1t}, X_{2t}, ..., X_{kt})'$. For a given horizon h, the aim is to predict the values that could be attained by the process in the future, X_{t+h}. For example, suppose that some historical data that up to a certain time point T, said $X_1, X_2, ... X_T$ were generated and we would like to predict X_{T+1}. Once X_{T+1}, we use all the information up to T+1 to predict X_{T+2}. This process keeps rolling till reaching the end of the multivariate time series. Tsay mathematically expressed this concept as Tsay (2013) :

$$\hat{X}_{t+h} = g(X_{T+h-1}, X_{T+h-2}, \dots, X_1)$$

In recent literature, various statistical models and machine learning algorithms have been deployed and utilized for multivariate time series modeling and forecasting. In Kumar and Ganesalingam (2001), a detection of financial distress via the use of multivariate analysis was conducted. For that matter, the stocks(from 1986 to 1991) of seventy-one(71) firms were extracted from the Autralian stock exchange and 10 financial ratios such as return on equity after extraordinary and abnormal(ROE), Debt to Asset before revaluation after Intangibles, long term debt to asset after intangibles and before revaluation, Current Ratio, Acide test Ratio, Return on assets after intangibles and before revaluation, Net profit margin, Earnings before Interest and Tax assets after intangibles and before revaluation, Operation Income to Operating assets before revaluations and after intangibles and Liquid ratio were used in order to determine which companies should be considered for investments. Multivariate techniques such as principal component analysis, factor analysis, discriminant analysis and cluster analysis were utilized. As results, factors that could be potentially used as measures of profitable companies were found. In addition, some operational indexes and techniques for grouping companies were indicated for future decision-making processes.

In medicine, Spencer et al. adopted a multivariate time series approach to modeling and forecasting demand in the emergency department (ED) Jones et al. (2009). The aim of this study was to study the existence of eventual relationships between key resources in ED and the inpatient hospital in order to develop multivariate forecasting models. As such, 2006 hourly data were collected from three different hospitals. In this process, numerous multivariate models were constructed and compared to a univariate benchmark model. As results, the authors found that vector autoregressive (VAR) models capture more the dynamics of demand in the ED and the inpatient hospital at different locations. They also noticed that for larger forecast horizons, VAR models, compared with classic univariate time series models, provide better forecasting accuracies of ED census. Similar results were demonstrated in forecasting the demands for diagnostic resources. Despite these encouraging results, the authors remain skeptical about adopting these forecasts in real clinical settings. As such, the authors advocate more robust analytical methods such as queuing theory, optimization and simulation modeling for the implementation of better decision support avenues when it comes down to clinical staffing, real-time monitoring and forecasting.

Andreoni and Postorino (2006) conducted a multivariate approach in forecasting air transport demand. In the study, the authors provided a comparison between univariate and ARIMAX models. It was found that univariate models do a better job in fitting the data than the ARIMAX model mostly in peak times. The authors also found that the forecasting performance of univariate models depends on the stability of the boundary conditions. They further noticed that those shortcomings from univariate models can be corrected by multivariate models. However, the authors support that it is not possible to assert that univariate models are better than multivariate models and vice versa.

In Kanchymalay et al. (2017) a multivariate time series analysis on nine time series data was carried out. The historical monthly closing price from January 1987 to February 2017 of crude palm oil (CPO) price, sunflower oil price, olive oil price, rapeseed oil price, coconut oil price, peanut oil price, soybean oil price and West Texas Intermediate(WTI) crude oil spot price as well as the Exchange rates of US dollar to Malaysian Ringgit were used. Strong correlation between CPO price and the prices of some vegetable oils such as soybean oil, sunflower oil, rapeseed oil, coconut oil and peanut oil was reported. In contrast a negative correlation coefficient between CPO and exchange rate was found. Such a result shows that an increase in the exchange rate causes a decrease in the price of CPO. Whereas a high correlation between crude oil was obtained. Thus, it was agreed that the above mentioned highly correlated vegetable oils with CPO as well as crude oil price to be the best predictors for CPO price. In addition, Multi-Layer-Perceptron (MLP) with two hidden layers, Support vector regression (SVR) with Sequential minimal optimization (SMO), and the Holt Winters exponential smoothing algorithm were deployed to forecast the monthly CPO price. The forecasting horizon was five months. On one hand, MLP and SMO achieved better forecasting accuracy than Holt Winters. On the other hand, SMO achieved higher predictive accuracy than MLP.

4.4 Multivariate Models

In recent times, developing methodologies that could enable researchers and finance practitioners to learn interconnected relations among multiple time series has been an open question Hu et al. (2019). Therefore, a myriad of multivariate time series forecasting models and associated variations have been proposed in literature. Due to the plethora of models proposed since the wake of multivariate time series analysis, it would be unreasonable to try to investigate all the extant varieties. As a result, we have decided to examine two models such as Vector Autoregressive Moving Average (VARMA) and Multivariate Singular Spectrum Analysis (MSSA). The choice of the first class of models mainly resides in their popularity among the most commonly used linear models in finance literature. The rationale behind the choice of the second models stemmed from the rarity the so-called techniques in literature.

Below is a short description of the models that have selected for this study.

4.4.1 The VARMA Model

Before any discussion about the vector AutoRegressive Model(VARMA), it seems opportune to first glance at the Vector AutoRegressive(VAR) model. A good understanding of the VAR models will enable us gaze upon VARMA models, which is noting more than the combination of VAR and VMA models equipped with some mathematical concepts and constraints. The operating mode of VARMA models is to leverage strengths of VAR and VMA models.

Virtually, VAR models have been accepted as linear multivariate time-series models that have the ability and the ease to capture the joint dynamics of multiple time series Miranda-Agrippino and Ricco (2019). VAR models operate as an extension of the univariate autoregressive models to multivariate time series settings. VAR models were first Introduced by the macro-econometrician Christopher Sims Sims (1980) to jointly analyze, model, and understand

the causal relations between multivariate macroeconomic variables. At an elementary level, the theory with VAR is just an extension of the theory of univariate time series data. In the basic structural form of the VAR models, each of the variable is expressed as a linear combination of not only their own previous values, but also of the previous values present in a k-dimensional vector of time series. As part of offering an illustrative example in order to support this description of VAR models, suppose that we have a multivariate time series made of three (k = 3) univariate time series such the weekly average weekly house prices in California, New York, and Hawaii. Let us denote those prices at a given time t as $PrCa_{t,1}$, $PrNy_{t,2}$, and $PrHa_{t,3}$. A VAR model of order 1 in the three defined variables can be written as follows:

$$\begin{cases} P_r Ca_{t,1} = C_1 + \alpha_{1,1} P_r Ca_{t-1,1} + \alpha_{1,2} P_r Ny_{t-1,2} + \alpha_{1,3} P_r Ha_{t-1,3} + \varepsilon_{t,1} \\ P_r Ny_{t,2} = C_2 + \alpha_{2,1} P_r Ca_{t-1,1} + \alpha_{2,2} P_r Ny_{t-1,2} + \alpha_{2,3} P_r Ha_{t-1,3} + \varepsilon_{t,2} \\ P_r Ha_{t,3} = C_3 + \alpha_{3,1} P_r Ca_{t-1,1} + \alpha_{3,2} P_r Ny_{t-1,2} + \alpha_{1,3} P_r Ha_{t-1,3} + \varepsilon_{t,3} \end{cases} \quad (4)$$

The above system of equations is a system of three equations where the average weekly house prices in Ca, NY and Hawaii are respectively explained by the first, second and third equations. The concerned system of equations teach us that the instantaneous observed value of each variable explicitly depends on its own previous values as well as the previous values of the other variables. An interesting feature of this system of equation is that all of the explained variables share the same explanatory variables.

Using matrix multiplication one can express an equivalent of the system of equations as :

$$\begin{bmatrix} P_r Ca_{t,1} \\ P_r Ny_{t,2} \\ P_r Ha_{t,3} \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} \end{bmatrix} \begin{bmatrix} P_r Ca_{t-1,1} \\ P_r Ny_{t-1,2} \\ P_r Ha_{t-1,3} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} \tag{5}$$

A generalized expression of a VAR model with lag p on a multivariate time series $\boldsymbol{X_t} = (X_{1t}, X_{2t}, \ldots, X_{kt})'$ can be obtained as :

$$\boldsymbol{X}_t = c + A_1 \boldsymbol{X}_{t-1} + A_2 \boldsymbol{X}_{t-2} + \cdots + A_p \boldsymbol{X}_{t-p} + \boldsymbol{\varepsilon}_t \tag{6}$$

Where for any τ ,X_τ, c and ε_t are k x 1 matrices, and for any I, A_i represents a k x k matrix. Note that ε_t is independently and identically distributed.

This scenario is described as k-variable VAR model of order p, where the first p lags of each variable composing the high-dimensional multivariate time series are used as predictors. By extension of the VAR(1) case, one can explicitly say that we are dealing with a system of k equations, where each of the equations describes a variable as a linear function of not only its past previous p values, but also with the p lagged values of the other variables. In VAR modeling, selecting the optimal lag p is of major importance. A lag p value larger than what is needed in the model can result in serious consequences on the computational aspects of the model, and the decision-making process as well. On one hand, any unit increment of the lag p leads to the reduction of the degree of freedom by the square of the total number of variables present in the system Vayej (2012), Fackler and Krieger (1986). Also, a p value larger than what is needed can lead to the deployment of an over-fitted model. On the other hand, a p value smaller than the required value can be source of producing an under-fitted model. Thus, a reasonable choice for the order of VAR models is crucial. In general, such a choice is made based on some defined selection criteria. In this process, numerous lag p values are tested and the one that minimizes the chosen selection criteria is maintained. According to Tia et. al, in this matter, the ultimate goal is to be able find a model with few possible estimated parameters while having the power to capture the dynamic interactions between different economic variables Tiao and Tsay (1983). Yet, it is widely accepted that inference in VAR models strongly depends on the choice of the lag-length Karlssony (1997), Gredenhoff and Karlsson (1999). In recent years, various lag-length selections procedures, including Akaike (AIC), Schwarz-Bayesian (BIC) and Hannan-Quinn (HQ), have been proposed and tested Gredenhoff and Karlsson (1999), Akaike (1969), Hatemi-J and S. Hacker (2009). For example, in the presence seasonal time series data, Brandt et al. proposed some simple techniques for selecting the lag val Brandt and Williams (2006). Detailed information on the use of model selection criteria in VAR models can be found in Lu ̈tkepohl (2013).

As a forecasting algorithm that is mostly utilized when two or more time series influence interactively, VAR models have gained popularity by delivering good forecasting results. Given some past observations up to a t - 1, one can then forecast the next value that is likely to be attained by the process. This can be formulated as

$$\boldsymbol{X}_{t|t-1} = E(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}, \boldsymbol{X}_{t-2}, \dots) = A_1\boldsymbol{X}_{t-1} + A_2\boldsymbol{X}_{t-2} + \cdots + A_p\boldsymbol{X}_{t-p} \qquad (7)$$

Eq(7) can be accepted as a linear regression over the lags of the multivariate process. As with any normal regression, the involved matrix-coefficients have to be estimated. Given this one day (h = 1) ahead forecasting approach, one can recursively forecasts the possible value that could be obtained by the process for larger horizons (h > 1). It is widely believed that it is eq(6) that constitutes the starting point in deriving more variants of the VARMA models.

VAR models are full of advantages. One of their main advantages over many univariate statistical models is their easy implementation, which in fact, explains their wide applications while studying economic variables. Another advantage of VAR models is the facility of its specification stage, which solely depends only on one lag order. Taking in a vector of variables, which interact linearly not only with their own current and lagged values, and the current and the lagged values of the remaining time series in the k-dimensional data set, is another advantage of VAR models . The easy estimation of the coefficients involved in the modeling where every variable is explained by the same explanatory variables is also a key primary advantage to VAR models.

Despite their success and wild deployment in finance and economic related studies, VAR models undergo varieties of criticisms. At the very early stage of their acceptance, critics were quick to note that VAR models suffer from not being able to capture the underlying structure of the economy. Hence, its unpopularity in the development and making of economic policy prescriptions. Another pitfall of the VAR models is that they are not suitable for establishing the relationship between a group of variables and their shocks at different time periods Vayej (2012). Lu ̈tkepohl and Poskitt (1996) report that due to some theoretical reasons, VAR models are not the best models to rely on. In Lu ̈tkepohl and Poskitt (1991), the authors reported that VAR models are not generally closed under under marginalization and temporal aggregation. Also, it is conventionally accepted that VAR models are not closed under linear mathematical transformations. For example, suppose that a vector of variables that follow a VAR process. Nothing reassures that a subvector of the original vector will also follow a VAR process. As a consequence, the necessity of having other models that are potentially capable of overcoming some of shortcomings associated with the use of VAR models seemed to be essential.

VARMA models are such models! VARMA models originated from early seminal work by Quenouille, Quenouille and Quenouille (1957). The point of VARMA models is to propose another way of representing the data generator process(GDP) of a set of time series as parsimonious as possible. The principal objectives of VARMA models is to leverage past history via combining past observations and previous errors in a given system. In this context, a multidimensional series $X_t = (X_{1t}, X_{2t}, ..., X_{kt})'$ is said to have a VARMA(p,q) representation if it follows the mathematical specification :

$$\boldsymbol{X}_t = \sum_{i=1}^{p} \Phi_i \boldsymbol{X}_{t-i} + \varepsilon_t - \sum_{j=1}^{q} \Theta_j \varepsilon_{t-j}\,, t \ in\ \mathbb{Z} \qquad (8)$$

where X_t is k x 1, ∀ i, Φ_i and Θ_j denote k x k matrices , ε_t is as before a zero mean white noise containing $(\varepsilon_{1t}, \varepsilon_{2t}, ..., \varepsilon_{kt})'$,

In the multivariate stationary time series case, the preceding equation could be rewritten as

$$\Phi(L) = \boldsymbol{\Theta}(L)\boldsymbol{\varepsilon_t} \tag{9}$$

where $\Phi(L) = \Phi_0 - \Phi_1 L - \ldots - \Phi_p L^p$ and $\Theta(L) = \Theta_0 - \Theta_1 L - \ldots - \Theta_p L^p$ represent the VAR and MA operators and for any i, Φ_i, and Θ_i respectively denote the auto-regressive and moving-average parameter matrices. The involved operator matrices are not identified or not unique Elliott and Timmermann (2013).

Since being deployed by the pioneers, VARMA models have been studied, revised or criticized by many researchers Hannan (1969), Wilson (1973), Box et al. (1979), Tiao and Box (1981). Athanasopoulos et al. (2012) supported that VARMA models can be seen as a corollary to the well known notion of Wold's decomposition for multivariate time series. Sargent et. al concluded the existence of a strong link between linearized dynamics stochastic general equilibrium (LSDE) models and VARMA ones Sargent et al. (2005). With time and the devotional literature of the period, a public perception that VARMA models appears to be preferable in obtaining a parsimonious representation of some types of data has been vastly forged. Wilson et al. (2001) utilized a VARMA(1,1) model on seven daily US dollar term rates to show that it is possible to extend methods based on conditional independence graphs may be to structural VARMA models . Kascha and Mertens (2009) reported a comparative study between structural vector autoregression and VARMA models. They noticed a better performance of the VARMA face to VAR models. They furthered compared VARMA-based models and state space representations of the input data. They found a poor performance of the two classes of models. Model identification, which is an important step in modeling multivariate time series with VARMA, was extensively studied by Boubacar Mainassara (2012), Levitt et al. (2011), Hurvich and Tsai (1989). Kascha (2012).

After the great success of Arima models, many believed that an adequate representation of the data generated process of a vector series could be obtained via a simple extension of the univariate ARIMA models and its time series goodness-of-fit diagnostic test to multivariate cases. This actually came out to be a very naive approach that ended up facing numerous challenges. The first challenge resides in the identification stage of the VARMA models. As mentioned earlier, the standard representation of the VARMA models presented in eq(7) is not unique. It appears that this makes the determination of the orders of VARMA models a little bite cumbersome. In addition, this non-uniqueness leads to complication in the specification and estimation stages since cosntant statistical estimation require impose on a unique representation of a GDP Elliott and Timmermann (2013). Additional requirements and restrictions on the AR ad MA operators are then needed in order to ensure a unique representation Dufour and Pelletier (2008). In response to this situation, procedures have been proposed to identify VARMA models Dufour and Pelletier (2008), Athanasopoulos and Vahid (2008), Athanasopoulos et al. (2012), Tiao and Tsay (1989), Deistler and Hannan (1988). Based on this development, it has been concluded that the straight generalization of the ARMA models might not lead to an identified representation Dufour and Pelletier (2008), Lütkepohl (2013).

As in the univariate case, adopting procedures for specifying and estimating cointegrated VARMA models is also a vital step in the multivariate case. However, it shall be noted that there is not a universally accepted strategy to specify the orders of VARMA Models. In reflecting on the success of the corresponding stage with Ljung-Box approach in univariate case, the first attempt to find the orders of the composing models was to use the autocorrelation, partial autocorrelation and cross-correlation functions. In record time, it turned out this procedure was dotted with difficulties in the multivariate settings. For instance, it becomes more difficult to detect the values of p and q when we are dealing with more than two time series Vayej (2012), Lu ̈tkepohl and Poskitt (1996). With time numerous specification techniques have been adopted and deployed depending on different representations of Vayej (2012), Yap and Reinsel (1995), Dufour and Pelletier (2008), Kascha and Trenkler (2011). Rich discussions about different specification procedures could also be found in Hanan (1970), Hanan (1976), Akaike (1974). Recently, a significant amount of efforts has been devoted to structural specification of VARMA models. The technique of structural specification aims at finding the underlying structure of a vector of time series in order to find a well-defined VARMA model that can be easily identified Tsay (2013). As of now, two main approaches exist for structural specification of multivariate time series. The first method, known as Kronecker index approach, seeks to find the maximum order of the AR and MA for each of the univariate time series. The second method or the concept of canonical correlation analysis infers information from cross-covariance matrices. They both try to overcome the difficulty of identifiability as mentioned earlier Tsay (2013). More information about structural specification of VARMA models could be found in Tsay (2013), Elliott and Timmermann (2013).

Upon the estimation of the coefficients of VARMA models, the exact and the conditional likelihood methods have been solicited Tsay (2013). However, in recent times, the use of maximum likelihood has been much more advocated mostly with small sample sizes. With times numerous variations of maximum likelihood techniques have been proposed to better suit the idea around VARMA models Dias and Kapetanios (2014), Hannan and Rissanen (1982), Doukhan et al. (1995), Dufour and Pelletier (2008). Nevertheless, it is worth noting that the maximum likelihood estimation still suffers from various numerical challenges Kascha and Ravazzolo (2010).

As with the univariate case, model checking is also an essential step of the VARMA modeling. In order to check for the adequacy of fitted a vector of ARMA models, numerous techniques have been recently proposed. Nevertheless, the multivariate extension of the portmanteau tests has been the most commonly used. The popular adoption of the multivariate version of the portmanteau tests was first introduced in 1974 by Chiturri Chitturi (1974). In 1980, Hosking generalized the procedure by using a relaxed definition of the concept of multivariate autocorrelations Hosking (1980). In 1981, Li et al, via a simulation study proposed a modified version of the portmanteau tests that performs better than the original test in small sample settings Li and McLeod (1981). In the same year, Hosking (1981) demonstrated that there is an equivalence relation between the different forms of the portmanteau statistic. In 1995, Ling et al, developed a new portmanteau statistic based of the sum of squared residual autocorrelations instead of the residual or the squared residual as usually done Li and McLeod (1981). In 2005, a close similarity between the portmanteau tests for VARMA and VAR models Lu¨tkepohl (2005). In 2007, Francq et al proposed a multivariate portmanteau diagnostic test which only requires the innovations to be uncorrelated instead of independently and identically distributed Francq and Ra¨issi (2007).

Just like in the univariate case, model checking is a very important stage in modeling vectors of time series. In checking VARMA models, the aim is to find a technique that enables one have an adequate, or at least, a very acceptable representation of the DGP that generates the vector of series. For the purposes, many tools have been suggested including the well-known t- and F-tests. Afterwards, much attention was given to the techniques that emphasize on measuring the magnitude of residuals obtained from fitting a chosen model. In the same carping vain as univariate time series, many model adequacy checking methodologies that mostly focus on vector of residuals have been proposed in literature. The null hypothesis of such tests is the autocorrelation functions of all the series composing the multivariate series must not have values that are statistically significant after a certain lag. The associated alternative hypothesis is that at least one of the series is not white noise. These lags could either be chosen by the user or via a criterion selection as discussed earlier. In the multivariate context, the well known Ljung-Box statistic test at a fixed lag s could be expressed as Johansen (1995):

$$LB(s) = T(T+2)\sum_{j=1}^{n}\frac{1}{T-J}tr[\hat{C}_{0j}\hat{C}_{00}^{-1}\hat{C}_{0j}'\hat{C}_{00}^{-1}] \quad (10)$$

where $\hat{C}_{0j} = T^{-1}\sum_{t=j+1}^{T}\hat{\varepsilon}_t\,\hat{\varepsilon}_{t-j}'$ denotes the sample size and tr() is the trace operator . Conventionally, It is has been proven and accepted that under the null hypothesis, such a test statistic follows a chi-square probability distribution with $k^2(m-p-q)$ degrees of freedom where k, m, p, q indicate the how many univariate time series composes the set of variables, the number of lags, the order of vector autoregressive part of the model and that of the moving average, respectively.

In high-dimensional time series analysis, the generalized versions of the portmanteau statistics have gained considerable appeal and attention. Nevertheless, its utilization should be carefully done mostly in the context of cointegration Sperling and Baum (2001). Another pitfall of the multivariate portmanteau statistics, mostly the one defined in eq(10) is that the fitted model turned to be inadequate with very large LB values Vayej (2012). Box et al. (2015) proposed that such a shortcoming could be fixed by imposing some extra constraints on the parameters.

Even with a vector of time series, forecasting remains an integral part of the modeling efforts and also of great benefits. Without loss of generality, the principal objective of multivariate time studies is to determine the possible future values that could be assumed by the DGP that is under investigation. Since VARMA contains a moving average part, and under the condition of white noise process, the future forecasted values could easily be obtained from the pure VAR process. For illustrative purposes, suppose that all the parameters of the VARMA models are known and that we possess some available and important information(I) up to time a certain time h denoted I_h, the one day prediction is obtained via Tsay (2013) :

$$\boldsymbol{X}_h(1) = E(\boldsymbol{X}_t|I_h) = \Phi_0 + \sum_{i=1}^{P} \Phi_i \boldsymbol{X}_{h+i-1} - \sum_{j=1}^{q} \Theta_j \varepsilon_{h+1-j} \qquad (11)$$

where the forecasting error is defined by

$$e_h(1) = \boldsymbol{X}_{h+1} - \boldsymbol{X}_h(1) = \varepsilon_{h+1} \qquad (12)$$

Iteratively, the τ-step ahead prediction could be formulated as proposed by Tsay (2013):

$$E(\boldsymbol{X}_{h+\tau}|I_h) = \begin{cases} \boldsymbol{X}_{h+\tau}, \text{if } \tau \leq 0 \\ \boldsymbol{X}_h(\tau), \text{if } \tau > 0 \end{cases} \qquad (13)$$

with the additional assumption that

$$E(\varepsilon_{h+\tau}|I_h) = \begin{cases} \varepsilon_{h+\tau}, \text{ if } \tau \leq 0 \\ 0, \text{ if } \tau > 0 \end{cases} \qquad (14)$$

There has been a growing interest in examining the forecasting power VARMA models in many areas, particularly in financial time series analysis. However, few research papers that outline the deployment of VARMA models on vectors that contains more than 3 univariate variables are available. In a mathematically involved discussion, Tsay demonstrated that stationary VARMA models are suited for short term-predictions while in their long term predictions are just the sample mean Tsay (2013). Dias and Kapetanios (2014) by adopting the Iterative Least Squares(IOLS) methodology as estimation procedure discovered a better prediction accuracy than VAR and AR(1).

4.4.2 VARIMA

As with ARIMA models, the 'I' in VARIMA stands for *Integrated*—where the integrated value, I, indicates the number of times the given time series the VARMA model is trained on needs to be differenced in order for the time series to become stationary. However, since for VARMA models the time series its trained on is multivariate, I is a 1xk dimensional vector where k is the dimensionality of the multivariate time series, containing the number of times each individual univariate time series needs to be differenced for the multivariate time series to be stationary. Although there seems to be good intuitive reasons to have each of the values of I be equal to the max of the each of the individual time series necessary differences, since this would keep each individual univariate time series on the same scale, this is not necessarily a required course of action—VARMA models capitalize off cross correlation between each time series, so if uneven differencing results in higher cross correlation this would be preferential to uniform over differencing of all time series. Although when dealing with similar univariate time series data, this is likely a moot point in that the most likely differencing across each univariate time series that would result in both multivariate stationarity and maximized cross correlation would be uniform differencing.

5 Parameter optimization and prediction of VARIMA

In this section I will propose a fully automated VARIMA model section algorithm. I have also implemented this algorithm in an R package I have created, *"auto_varima"*, that I plan to publish to the R repository CRAN. The package is also written to take advantage of multithreading to support working with large, time intensive training sets. I will later apply this proposed algorithm to forecasting Bitcoin and other cryptocurrency data using this package. The functions in this package currently only support bivariate VARIMA models, but this will be updated to include higher dimensionality VARIMA models in the future.

5.1 Data Preprocessing

- First each of the individual k univariate time series are each scaled by subtracting their mean and dividing by their variance (this is done to avoid numerical non-convergence of implementation of parameter estimation of the VARMA model).
- Next the 1xk dimensional vector, I, is determined, and each of the k univariate time series is differenced according to I_i . In the bivariate case, we will difference each of the univariate time series X_{it} will first be differenced by I_i and then separately differenced by max(I), if corr(diff(X_{1t},I_1) , diff(X_{2t},I_2)) < corr(diff(X_{1t},max(I)) , diff(X_{2t},max(I))) then both I_1,I_2=: max(I) .

5.2 Model Parameter Estimation

The AR and MA order, p and q, of the VARIMA model are determined by iteratively training VARMA models on all combinations of p and q (within a fixed range) and selecting the one the particular pair that results in lowest AIC.

5.3 Model Forecasting

Forecasting is done 1 time unit into the future with maximum likelihood estimation. Standard Errors of the forecast error are also used to forecast 80% and 95% prediction intervals around the forecast.

5.4 Data Postprocessing

Each univariate time series prediction is first undifferenced I_i times (values from the original undifferenced time series are required to do this). Then, unscaled by first multiplying by the original variance divided by during preprocessing and then adding the original mean subtracted during preprocessing. The standard errors for predicting one time point into the future remain the same if the time series data is differenced or undifferenced, so standard errors only need to be multiplied by the original scaling variance. From here, the prediction intervals can be created from these new standard errors in the typical way.

6 Principal Component Analysis (PCA)

Often, in this period of deluge of data we are inundated with datasets composed of thousands even millions of features. Unfortunately, having training datasets comes with numerous serious consequences. First, it makes some tasks include exploration and visualization more difficult to be completed. Second, it makes the training process extremely slow. Third, with many features, statistical models and machine learning algorithms become more complex. As a result, they could overfit the training dataset and poorly perform on unseen data. This scenario is generally referred to as “the curse of dimensionality”.

To palliate to this disagreeable scenario, a battery of solutions has been proposed. One easier solution that has been adopted to overcome the curse of dimensionality is considerably increase the size of the training set to reach a sufficient density of training instances Geron (2017). This is problematic the number of instances required to reach a given density grows exceptionally with the number of dimensions Geron (2017). Another solution to the curse of dimensionality, which is in fact the most widely used, is the ***dimensionality reduction***. Dimensionality reduction is a mathematical transformation that is applied to a higher dimensional data for the purpose of obtaining a reduced representation of the original data, while the statistical information remains preserved. Without loss of generality, the dimension of the reduced representation indicates the required minimum number of parameters to account for the observed properties of the data Maaten (2009), Fukunaga (1990). Such a technique comes with a plethoric number of advantages. The top benefit of dimensionality reduction is its ability to mitigates the curse of dimensionality and other undesired properties of high-dimensional space Maaten et al. (2009). Dimensionality reduction techniques applied to a dataset takes of the multicollinearity present in the dataset by removing redundant features. Not only does dimensionality reduction speeds up the training portion of the learning process, but it also helps gets insights into the dataset via visualization. For example, reducing a very high dimensional dataset to two features allows one to have a graphical representation in 2-D plane.

In short, Dimensionality reduction consists of reducing the number of features present in the dataset, while preserving as much variability as possible. It can be done by a) selecting features, or b) creating a new set of features, where each of the new feature represents a combination of the original features in the dataset.

In a) a transformation of the original features is performed. The goal of this technique is to transform the original features to a small set of features. It results in a smaller but richer set of features that keeps much of the underlying information. For example, one can merge a set of highly correlated of features into one feature that can represent a characteristic of a given observation. Feature extraction has proven records to be useful in numerous research areas including image processing Yang et al. (2008), Kumar et al. (2014), Choras (2007), in time series analysis Olszewski (2001), Zhang (2005), Fulcher and Jones (2017), and in medicine Karpagachelvi et al. (2010), Acharya et al. (2019).

In b) a low-dimensional representation of the original data, previously in higher dimension, is constructed. The obtained representation provides as much of the variance in the data as possible. This is done by finding a linear basis of reduced dimensionality for the data, in which the amount of variance in the data is maximal Maaten et al. (2009).

b) can be achieved by either using linear or nonlinear transformations. If the data is linearly separable, the linear transformations work well. In this case, different classes are obtained by just drawing a simple line or a hyperplane to subdivide a dataset. In contrary, if the data display complex structures or is not separable by a line or a hyperplane, one is better off with a nonlinear transformation.

In recent times, numerous linear and nonlinear transformation for reduction of dimensionality have been proposed Maaten et al. (2009), Burges (2005), Lee and Verleysen (2007), Johnson et al. (2007). For the sake of this work, let us focus on the ***Principal Component Analysis*** (PCA), by far the most popular technique for dimensionality reduction.

PCA is one of the oldest and most widely used dimensionality reduction techniques Maaten et al. (2009). Standard PCA reduces the number of features by using a linear progression while preserving as much variability as possible. The elements of the reduced feature set are called principal component. By "preserving as much variability as possible", one means to find a smaller set of features where each of the newly constructed feature is a linear combination of those in the original set Maaten et al. (2009). In addition, the reduced set of features successively maximize the variance and while remaining uncorrelated with each other Maaten et al. (2009). This tells two things: a. PCA tends to explain the variance-covariance structure of a set of variables in terms of a linear combination of these variables, 2. the total variability in a dataset can be approximately reproduced by a small set of principal components. If so, suppose that or dataset is $\boldsymbol{X}_{n,p}$, where n represents the number of observations and p the number of features. The principal component analysis reduces the dataset to $\boldsymbol{X}_{n,k}$, where still represents the number of observations, and k designates the number of principal components with $k < p$. The obtained principal components often reveal details and relationships that were not previously seen or obvious.

Let us see what principal component analysis means in the framework of mathematics. For this, we will literally refer to Johnson et al. (2007).

Suppose that our above mentioned $\mathbf{X}_{n,p}$ has $X_1, X_2, \ldots, X_p$ features. Algebraically, the principal components are linear combination of the p variables present in the original dataset. Geometrically, these principal components form a new coordinate system that is obtained by a series of rotations of the p variables. In the new coordinate system, $X_1, X_2, \ldots, X_p$ will not only be the coordinate axes, but they will represent the axes that account for the largest amounts of variability.

Let us define $\mathbf{X}' = [X_1, X_2, \ldots, X_p]$, with covariance matrix $\boldsymbol{\Sigma}$, and eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$. Consider the following equations that represent a set of linear combinations

$$Y_1 = \mathbf{a}_1'\mathbf{X} = a_{11}X_1 + a_{12}X_{12} + \ldots + a_{1p}X_1$$

$$Y_2 = \mathbf{a}_2'\mathbf{X} = a_{21}X_1 + a_{22}X_{12} + \ldots + a_{2p}X_1$$

$$\vdots$$

$$Y_p = \mathbf{a}_p'\mathbf{X} = a_{p1}X_1 + a_{p2}X_{12} + \ldots + a_{pp}X_p$$

From undergraduate statistics classes, one can express the variance of each of the Y_i as:

$$var(Y_i) = \mathbf{a}_i'\boldsymbol{\Sigma}\mathbf{a}_{\boldsymbol{i}}, \boldsymbol{i} = 1,2,\ldots, p$$

$$\boldsymbol{Cov}(\boldsymbol{Y}_{\boldsymbol{i}}, \boldsymbol{Y}_{\boldsymbol{k}}) = \mathbf{a}_i'\boldsymbol{\Sigma}\mathbf{a}_{\boldsymbol{i},\boldsymbol{k}}, k = 1,2,\ldots,p$$

The Y_{is} represent the uncorrelated principal components, where the first principal component is the linear combination $\mathbf{a}_1'\mathbf{X}$ that maximizes $var(\mathbf{a}_1'\mathbf{X})$ subject to the constraint $\mathbf{a}_1'\mathbf{a}_1 = \mathbf{1}$, and the second principal component is the linear combination $\mathbf{a}_2'\mathbf{X}$ that maximizes $var(\mathbf{a}_2'\mathbf{X})$ subject to the constraint $\mathbf{a}_2'\mathbf{a}_2 = \mathbf{1}$ and $Cov(\mathbf{a}_1'\mathbf{X}, \mathbf{a}_2'\mathbf{X}) = 0$, ….. This continues up the *ith* principal component that is the linear combination $\mathbf{a}_i'\mathbf{X}$ that maximizes $var(\mathbf{a}_i'\mathbf{X})$ subject to $\mathbf{a}_i'\mathbf{a}_\mathrm{i} = \mathbf{1}$ and $Cov(\mathbf{a}_i'\mathbf{X}, \mathbf{a}_\mathbf{k}'\mathbf{X}) = 0$ for $k < i$.

For any principal component *i,* the contribution of the associated principal component is found by

$$\frac{\lambda_i}{\sum_{i=1}^{p} \lambda_i}$$

7 Forecasting Cryptocurrencies with auto_varima R Package

7.1 Training Data

I looked at 3 years of data for both financial time series and cryptocurrencies from November 1st, 2017 to November 1st, 2020. Trading volumes and daily prices were used for the following cryptocurrencies: Bitcoin, Ethereum, Ripple, Tether, Litecoin, Bitcoin Cash, Chainlink, Cardano, Binance Coin, Monero, Dogecoin. Since cryptocurrency exchanges are open 24/7, closing prices are defined as the last price of the day. Data was collected from the site coinmarketcap.com, which aggregates prices over multiple exchanges.

Trading volumes and daily prices were also acquired for the for: the S&P 500 (GSPC), Dow Jones Industrial Average (DJI), NASDAQ Composite (IXIC), and NVIDA (NVDA) stock. This data was obtained from Yahoo Finance. However, since the stock market does not trade on holidays or weekends, there are missing data point during the given 3-year period. To resolve this, I put the stock market data on the same time scale as cryptocurrencies by interpolating the missing values with cubic splines.

While this work does make predictions on several cryptocurrencies, it is primarily concerned with the ability to forecast Bitcoin. To aid in this effort, I performed PCA on a matrix containing columns of all other time series besides Bitcoin, and took the first principal component (pc1) of the result to be used as an additional time series (specifically for bivariate forecasting of Bitcoin).

Bitcoin daily price and daily volume time series were paired with the remaining time series (and each other)—resulting in a total of 59 unique bivariate pairs.

7.2 Moving Window Forecasting

A moving window was used to validate each of the bivariate time series pairs. Both auto parameter tuning ARIMA and VARIMA models were applied on a subset of data points (for each univariate time series and bivariate pair respectively), collecting several measures for later use, as well as predicting one day into the future. This subset or window, is then moved forward by 1 day, training another model. This process is repeated until you predict the last value in the dataset (note, since these are auto tuning models, each window might have different parameter values that the previous window). Due to the computation time constraint, only combinations of p and q each in the 0-2 range were for the VARIMA models.

The values collected per window were as follows: forecasted mean, Ljung-Box test p-values for lags 10, 24, and n/2, AR order p, MA order q, difference order d, standard error of the forecast error, correlation of differenced training data, correlation of undifferenced training data. The correlation value collected for bivariate pairs was taken to be the cross-correlation, whereas in the univariate time series windows for ARIMA, it was taken to be the autocorrelation).

After each model is trained and forecasted on each window, the forecasted values are compared to actual values and captured the following validation metrics/relevant statistics: Directional Forecasting Accuracy (dir_acc), Mean Absolute Error (MAE), Mean Squared Error (MSE), proportion of absolute value of undifferenced correlation of training data above .3 (cor), proportion of absolute value of differenced correlation of training data above .3 (corD), proportion of Ljung-Box p-values of residuals above .05 for lags 10, 24, and training length halved (pv10, pv24, pvH), proportion of original training data points within forecasted confidence interval (CI).

8 Results

Validation metrics and statistics were collected for ARIMA and VARIMA models on multiple training window lengths of 100, 200, and 300 days. For tables 8-2 through 8-10, the rows with the best DF accuracy and MAE are highlighted in blue and yellow respectively (and in pink if one row has the best of both). In each of these tables, the relative variance (RV) which we are taking to be the variance of the mean scaled time series, is provided to denote the amount of variation in each time series. Within the results tables, the time series name mentioned on the right-hand side of the title is the predicted time series in question for both ARIMA and VARIMA, where in the case of VARIMA, the other time series name on the left-hand side is a part of the bivariate pair. RV refers to the relative variance, which I define as the variance performed on the mean scaled timeseries. The time series graphs in this section are only for the last 200 days of the moving window for the 3-year period so that we can see the results visually in finer detail.

8.1 Total Model Runtimes

All model variations were run on 35 i9 Virtual Cores in parallel; the runtimes and total number of trained models are denoted in table 8-1.

Table 8-1. Model Runtimes

Model/Training Length	Runtime	Number of Trained Models
arima 100	.6 hours	14,955
arima 200	1.3 hours	13,455
arima 300	2.1 hours	11,955
varima 100	14 hours	58,823
varima 200	24 hours	52,923
varima 300	33 hours	47,023

8.2 Bitcoin-Cryptocurrency Bivariate Pairs

This section will provide analysis and explanation for the results of each of the model types and training lengths run, exploring the relationships between initial cross correlations of bivariate time series pairs and their resulting validation metrics. As there are hundreds of graphs and tables and there is significant overlap in the result findings, detailed analysis will be provided for few bivariate pairs of interest, and summarized for the remaining results. The remainder of the graphs and tables not examined in this section will be included in the appendix.

Figure 8-1. Undifferenced CCF, BIT-BNB Vol.

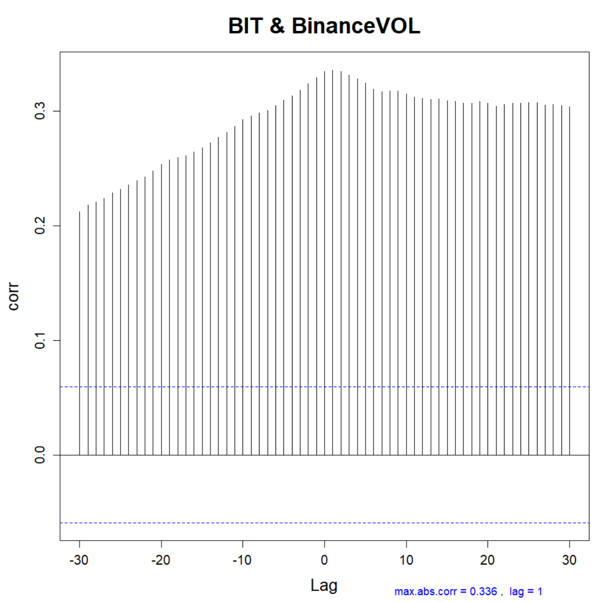
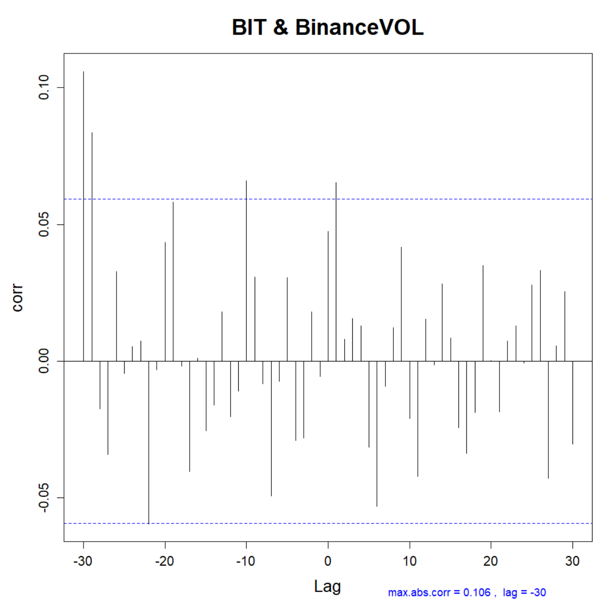


Figure 8-2. Differenced CCF, BIT-BNB Vol.



Figures, 8-1 and 8-2 show the cross correlations of undifferenced and differenced time series pairs respectively across different lags, where negative lags indicate the first mentioned time series leads the second by the given lag, and positive lags indicate the second time series mentioned leading the first by the given lag indicated. Each time series' difference order is the necessary number of times differenced to make the time series stationary.

While the undifferenced time series of Bitcoin and Bitcoin Trading Volume in Figure 8-1 initially shows promise of having moderate cross correlation above .3 at lag 1, this result is not necessarily indicative of a meaningful relationship between the two time series, as cross correlations for non-stationary time series are spurious. After differencing to achieve stationarity, we see in Figure 8-2 that the resulting cross correlation is fairly low, consistently hovering around .05. This can further be seen when looking at the validation metrics for the prediction of Bitcoin using Bitcoin Trading Volume in Table 8-2 below, as the proportion of differenced cross correlations for each individual window above .3 is also low. This low cross correlation results in the overall best performing model in terms of both directional accuracy and mean absolute error to be an ARIMA model with a training length of 200, highlighting that there is no benefit in using the Bitcoin Volume time series as a predictor with VARIMA.

Table 8-2. Validation Metrics, BIT Vol. Predicts BIT

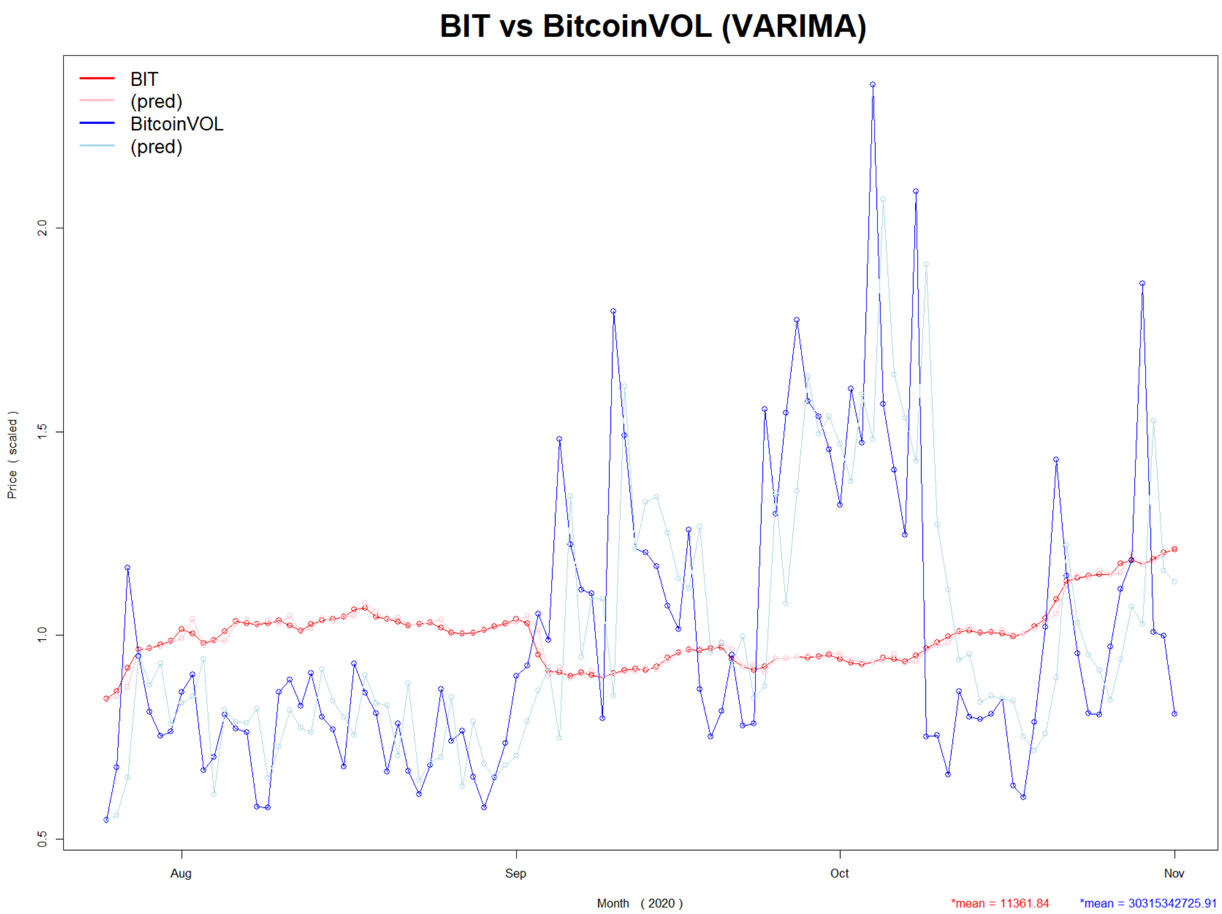
BitcoinVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.707	113.741	1947617.035	0.643	0.241	0.795	0.891	0.955	0.924
varima200	0.715	99.846	1715751.049	0.596	0.108	0.642	0.734	0.887	0.94
varima300	0.715	97.955	1722322.487	0.553	0.03	0.598	0.661	0.762	0.947

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Figure 8-3 below shows the last 200 days of prediction of Bitcoin and Bitcoin Volume using VARIMA model of training length 200. The darker points are the actual values and the lighter colored points are the predicted values (values are mean scaled to allow both time series to appear on the same graph). You can also visually see the much higher RV of Bitcoin Volume over Bitcoin.

Figure 8-3 VARIMA-200 Prediction, BIT-BIT Vol.



Unlike the previous example however, when looking at the cross correlations of Bitcoin and Litecoin shown below in Figures 8-4 and 8-5, we can see that both the undifferenced and differenced pairs have strong correlation above .5 . This higher cross correlation can also be seen in Table 8-3, showing 100 percent of differenced cross correlations for individual windows above

.03. These high cross correlation's impact on the models' results can further be seen in table 8-3, where the forecast accuracy of the best model is now VARIMA.

Figure 8-4. Undifferenced CCF, BIT-LTC.

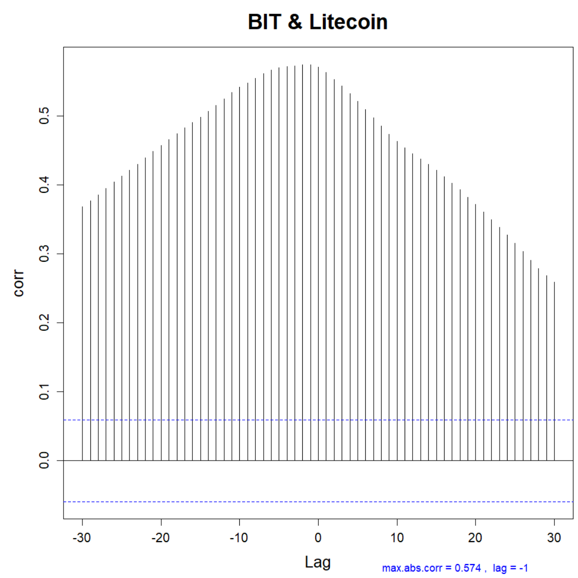


Figure 8-5. Differenced CCF, BIT-LTC

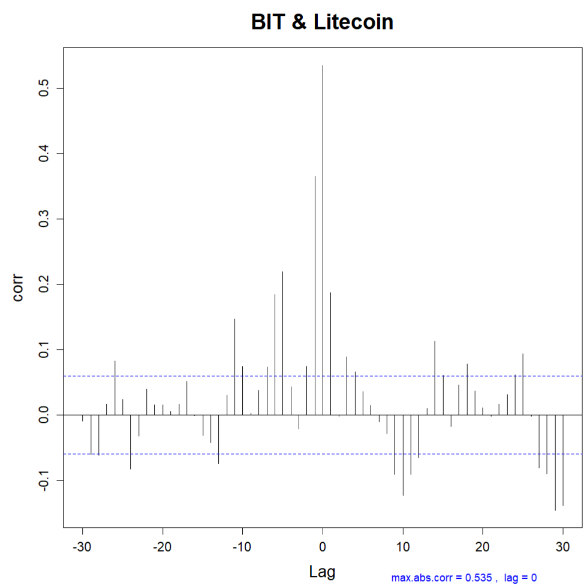


Table 8-3. Validation Metrics, LTC Predicts BIT

Litecoin >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.704	111.774	1932196.976	0.956	1	0.696	0.809	0.932	0.921
varima200	0.723	95.963	1643264.703	0.963	1	0.433	0.561	0.928	0.941
**varima300*	0.729	94.598	1663088.229	1	1	0.344	0.384	0.878	0.942

*Litecoin =(Mean ~ 80.857 , RV ~ 0.443) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Although there is a slight advantage with VARIMA over ARIMA when predicting Bitcoin with the bivariate pair, we can see in Table 8-4 this is not the case when trying to predict Litecoin with the same pair.

Table 8-4. Validation Metrics, BIT Predicts LTC

BIT >>> Litecoin

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.724	1.465	285.466	0.072	0	0.915	0.899	0.899	0.934
* *arima200*	0.727	1.089	158.264	0.711	0	0.747	0.64	0.833	0.951
+ *arima300*	0.714	1.031	141.214	0.832	0	0.689	0.477	0.836	0.95
varima100	0.687	1.466	292.768	0.956	1	0.696	0.809	0.932	0.92
varima200	0.71	1.124	163.324	0.963	1	0.433	0.561	0.928	0.938
varima300	0.715	1.044	144.687	1	1	0.344	0.384	0.878	0.942

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Litecoin =(Mean ~ 80.857 , RV ~ 0.443)

In Table 8-4 and Figure 8-6, we note that the RV is higher for Litecoin than Bitcoin, an observation that we will see is a common theme among the other non-Bitcoin cryptocurrencies as well. This observation helps underline the stability that Bitcoin has relative to other cryptocurrencies. It should be noted that even though Bitcoin's time series shown in Figure 8-6 and 8-7 is of the same time period, they will look different as the scales are different due to the time series they are each paired with.

Figure 8-6. VARIMA-300 Prediction, BIT-LTC

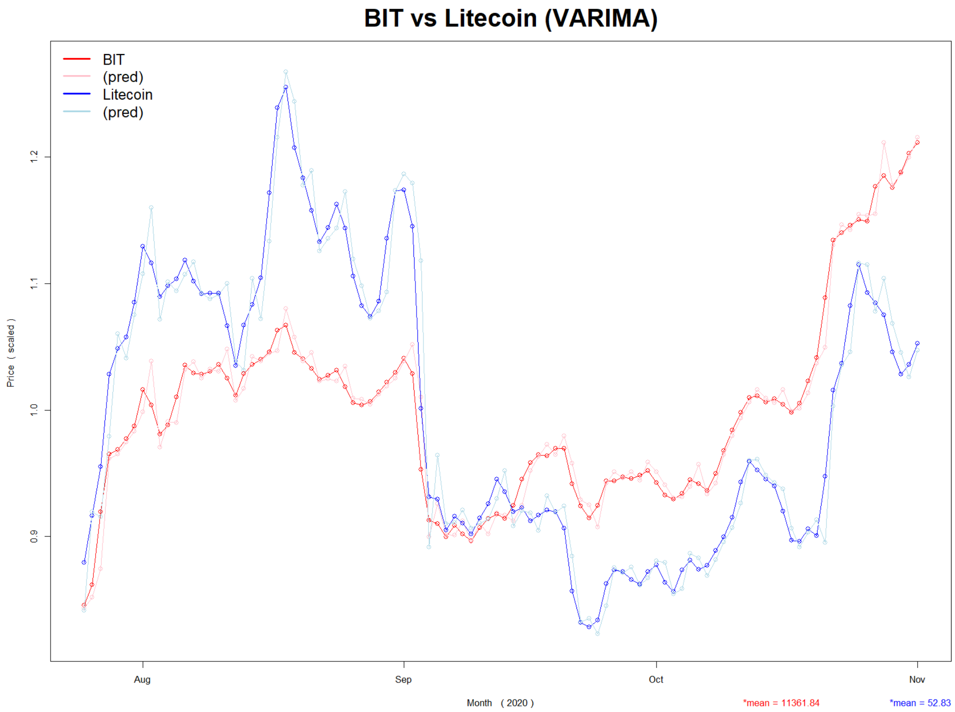
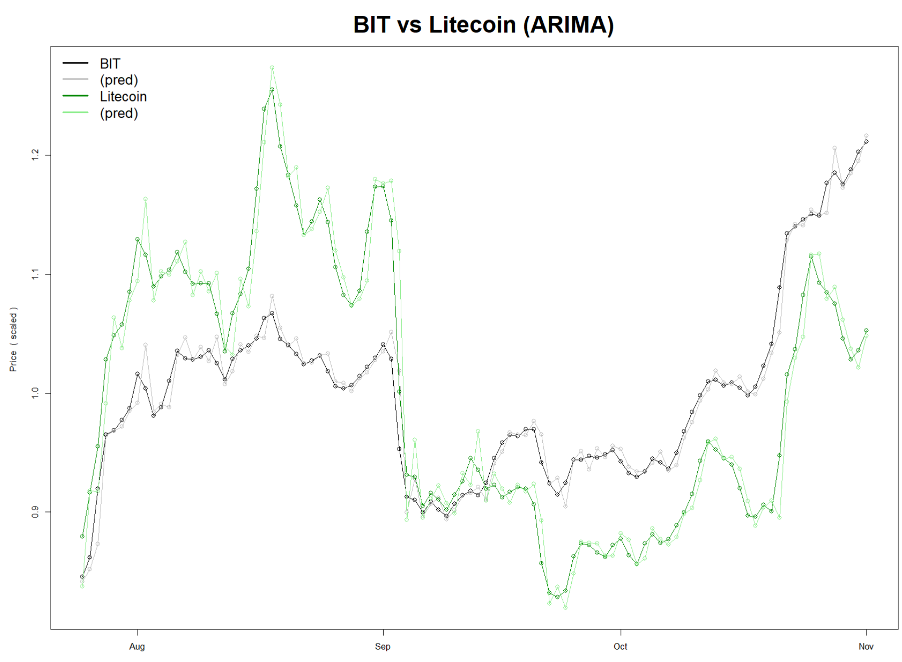


Figure 8-7, shows the period and pairs forecasted with ARIMA-200 models for a comparison. Here, we can observe subtle differences in the forecasted values for each time series.

Figure 8-7. ARIMA-200 Prediction, BIT-LTC



Similar to the Bitcoin-Litecoin bivariate pair, the Bitcoin-Ethereum bivariate pair presented in Tables 8-5 and 8-6 also has significant cross correlations of the differenced and undifferenced training data, with the best model for directional forecast accuracy being VARIMA for forecasting Bitcoin and ARIMA for forecasting Ethereum. This is another common theme for other cryptocurrencies paired with Bitcoin—predicting Bitcoin sees advantages when using VARIMA but predicting the cryptocurrency it is paired with does not.

Table 8-5. Validation Metrics, ETH Predicts BIT

Ethereum >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ arima200	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.703	110.837	1916865.963	0.929	1	0.682	0.788	0.878	0.923
varima200	0.727	96.96	1662934.343	1	1	0.466	0.614	0.926	0.938
*varima300	0.734	94.621	1660783.695	1	1	0.21	0.319	0.947	0.945

*Ethereum =(Mean ~ 321.314 , RV ~ 0.512) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Table 8-6. Validation Metrics, BIT Predicts ETH

BIT >>> Ethereum

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
* arima100	0.74	5.192	4280.824	0.131	0.013	0.887	0.846	0.943	0.939
arima200	0.738	4.091	2427.901	0.67	0	0.654	0.584	0.854	0.949
+ arima300	0.732	3.511	1667.395	0.833	0	0.629	0.384	0.864	0.944
varima100	0.725	5.394	4355.107	0.929	1	0.682	0.788	0.878	0.915
varima200	0.724	4.131	2426.761	1	1	0.466	0.614	0.926	0.939
varima300	0.726	3.54	1693.604	1	1	0.21	0.319	0.947	0.944

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Ethereum =(Mean ~ 321.314 , RV ~ 0.512)

Note that most of the proportion of p-values above .05 in these and previous tables denoted in the pv10, pv24, pvH columns, fall below the 95 percent threshold we would normally expect of strong models, indicating that the majority of the windows have low autocorrelation among the residuals. While this could be interpreted as poor model fitting, it is also likely due to the high type-I error associated medium to larger lags of the Ljung-Box test. The proportion of the original data within the forecasted 95 percent confidence interval (ci) hovers near .95 for these and the previous tables, as they should. This is also typical of all the bivariate pairs. We can visually see this in Figures 8-8 and 8-9, which show the 95 percent confidence intervals of Bitcoin and Ethereum plotted against their original values for the last 200 hundred days.

Figure 8-8. Predicting Bitcoin 95% CI with VARIMA

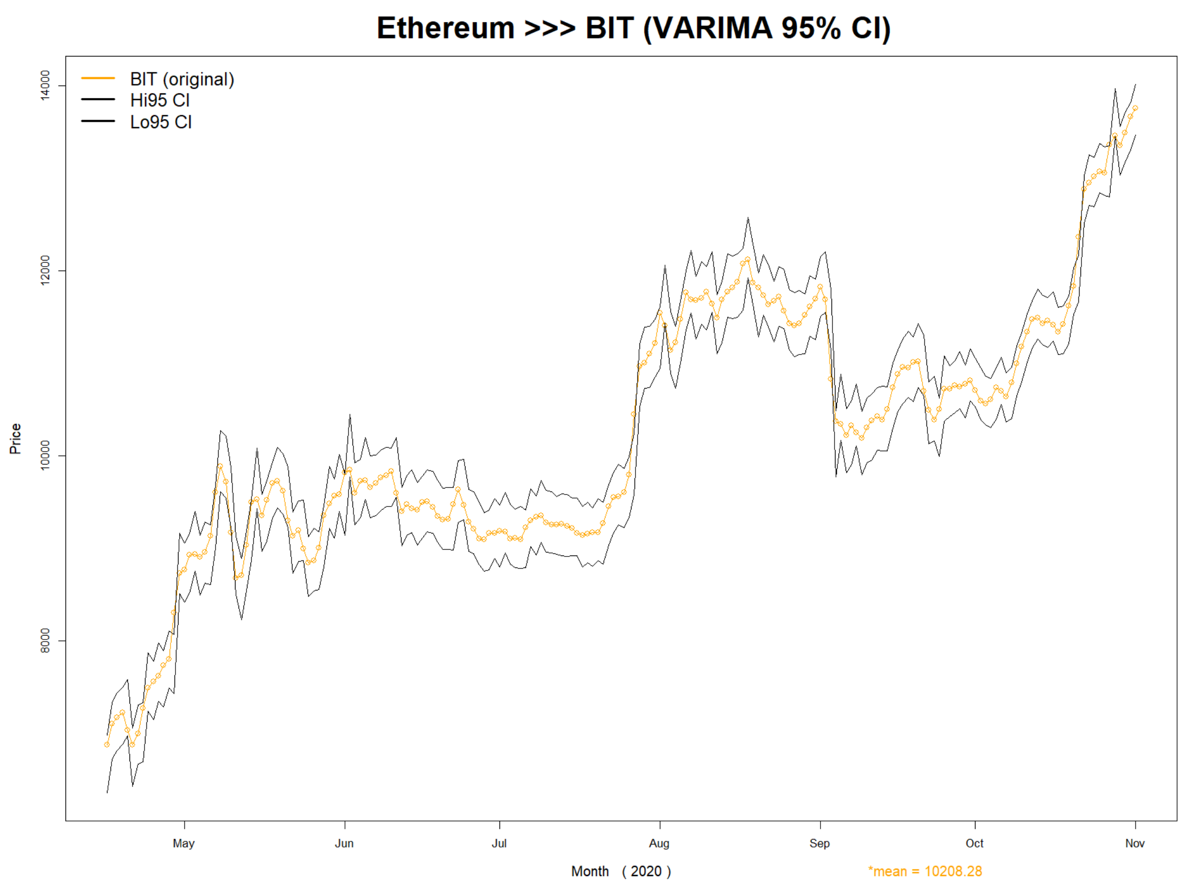
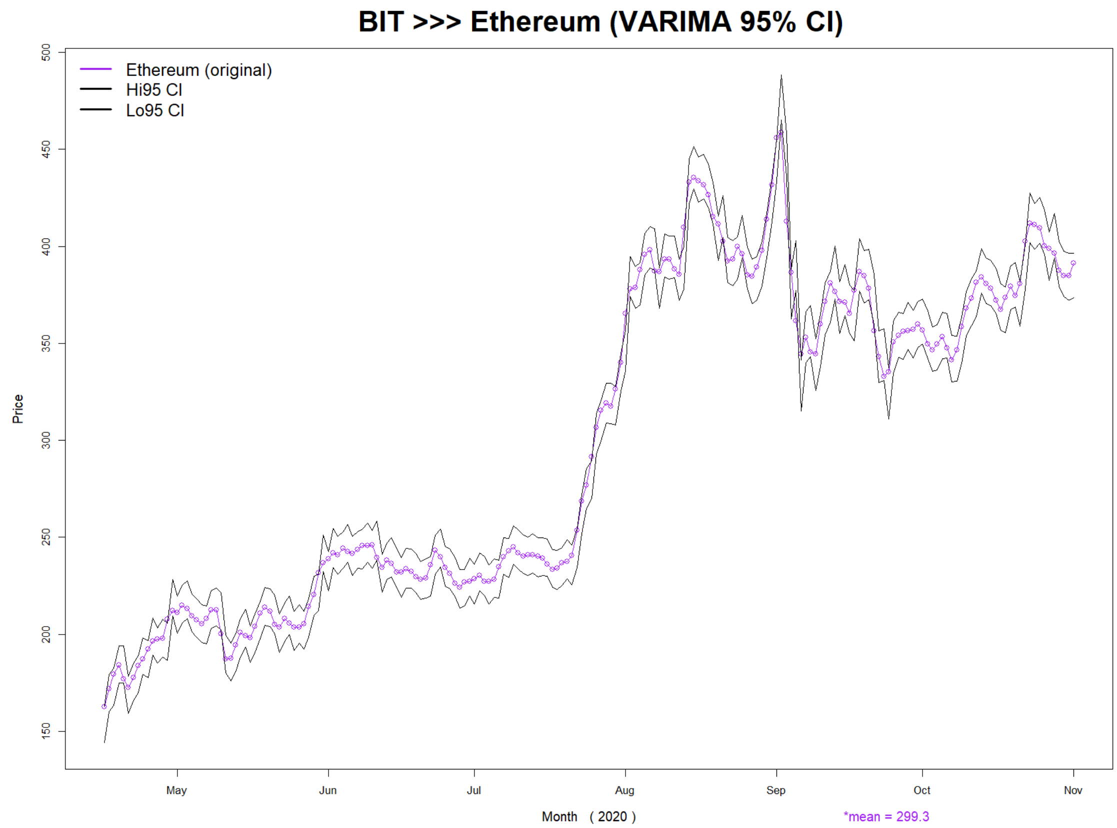


Figure 8-9. Predicting Ethereum 95% CI with VARIMA



Below in Figure 8-10, we can see the cross correlations of Bitcoin and XRP shows promise in the undifferenced data peaking at lag -19, which is further supported by the cross correlations of the differenced data in Figure 8-11.

Figure 8-10. Undifferenced CCF, BIT-XRP

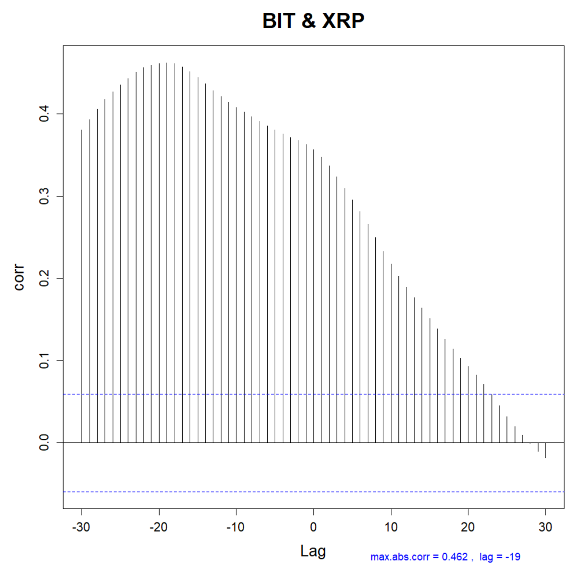
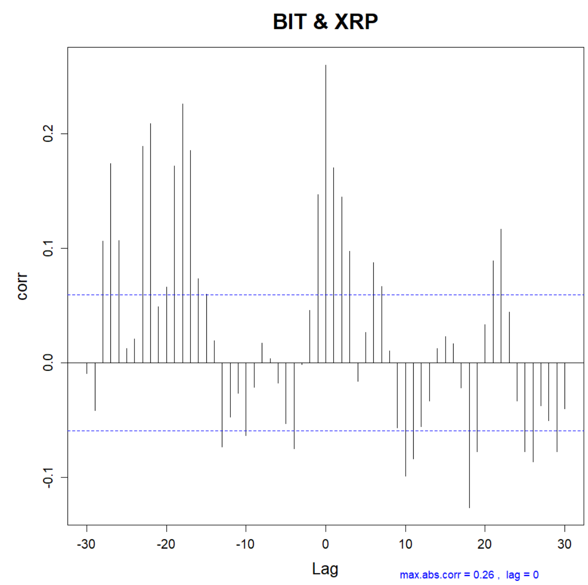
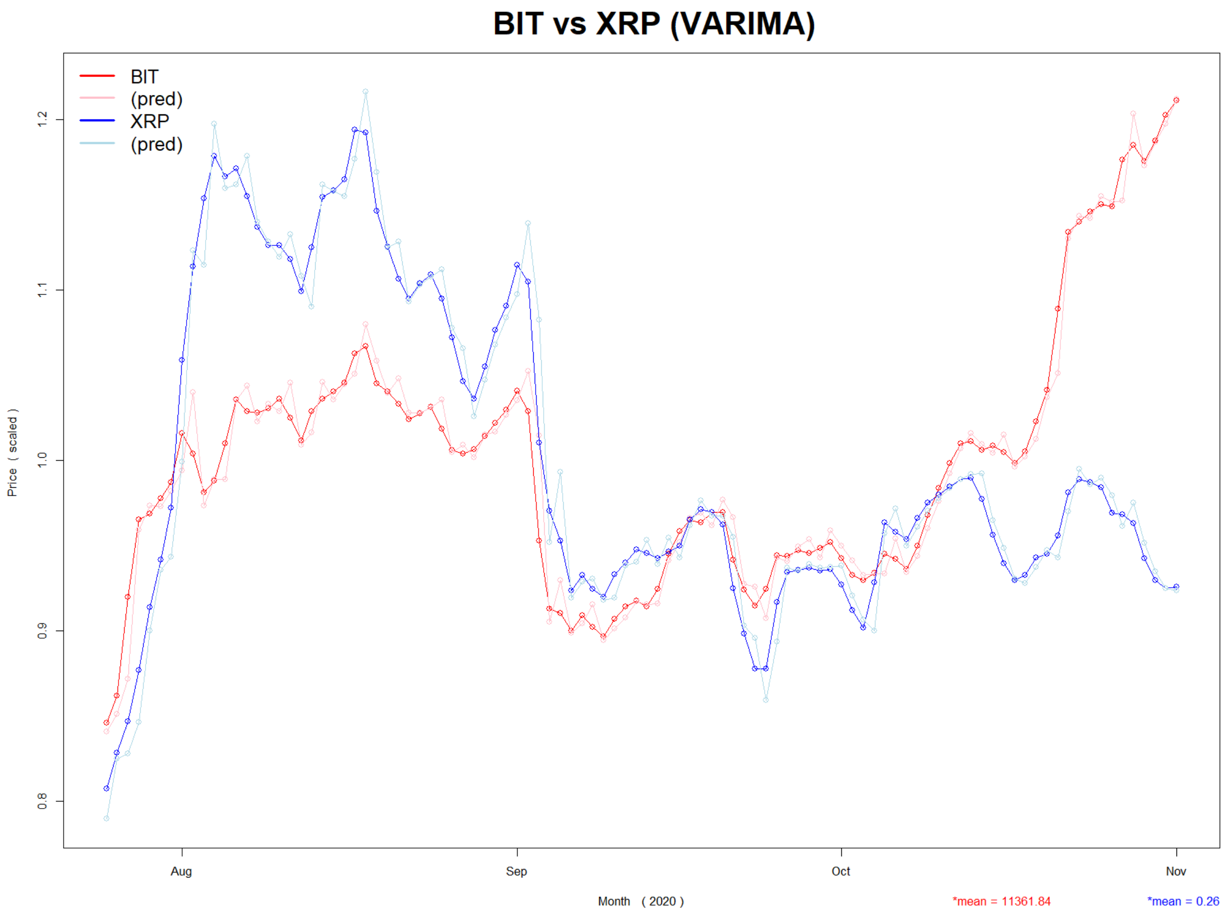


Figure 8-11. Differenced CCF, BIT-XRP



Having Bitcoin *lead* XRP by a lag of 19 above can prove useful for forecasting into the future of XRP. Stated another way, this implies Bitcoin's price on a given day would have moderate correlation with XRP's price 19 days into the future. However, when we examine Figure 8-11 more closely, we see that the highest correlation after differencing is at lag 0. This is something that is common among most of the cryptocurrencies when paired with Bitcoin. This point is further illustrated in Figure 8-12 below, showing that the strongest correlation clearly hovers around lag 0, as both time series' peaks and valley, despite being of different magnitudes, overlap almost perfectly.

Figure 8-12. VARIMA-300 Prediction, BIT-XRP



We can see the success of using VARIMA forecasting Bitcoin with this pair in Table 8-7 below.

Table 8-7. Validation Metrics, XRP Predicts BIT

XRP >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.707	110.347	1905560.913	0.799	0.955	0.742	0.872	0.99	0.928
**varima200*	0.735	94.783	1628816.339	0.887	0.958	0.785	0.843	0.993	0.943
varima300	0.715	94.19	1651918.303	0.77	0.952	0.767	0.832	0.997	0.944

*XRP =(Mean ~ 0.417 , RV ~ 0.71) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

8.3 Bitcoin-Financial Data/Other Bivariate Pairs

Predicting bivariate time series of Bitcoin paired with other financial data has proven less amenable to using VARIMA models than pairing Bitcoin with other cryptocurrencies. In Chapter 1, we had initially hypothesized about a potential strong relationship between Bitcoin prices and Nvidia Stock due to the intertwined nature of blockchain technology with the GPU manufacturer; however, as with some of cryptocurrencies pairs, after differencing, cross correlations no longer exhibit the same significant magnitude and the resulting VARIMA predictions perform worse than ARIMA, as shown in Tables 8-8 and 8-9 below.

Figure 8-13. Undifferenced CCF, BIT-NVDA

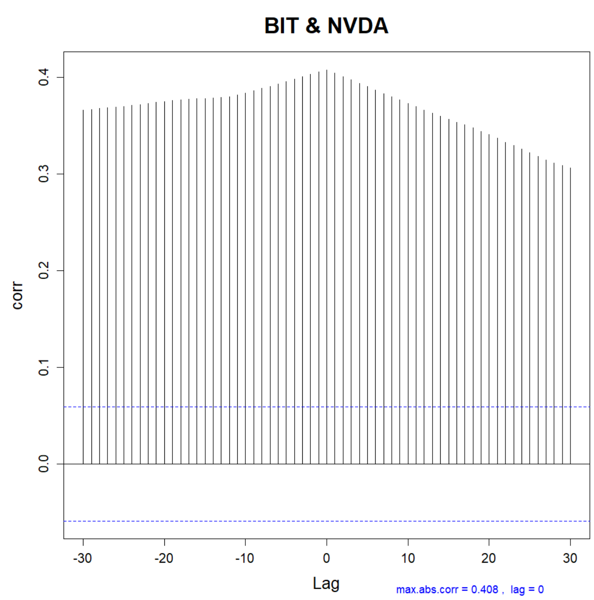


Figure 8-14. Differenced CCF, BIT-NVDA

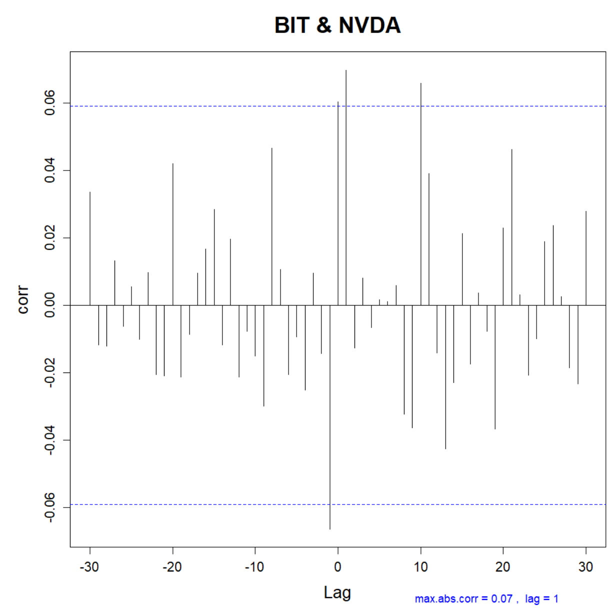


Table 8-8. Validation Metrics, NVDA Predicts BIT

NVDA >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.697	112.781	1951650.597	0.741	0.139	0.825	0.849	0.943	0.925
varima200	0.723	98.971	1687326.023	0.706	0.039	0.745	0.653	0.936	0.939
varima300	0.707	97.364	1697360.904	0.635	0.006	0.647	0.533	0.898	0.94

*NVDA =(Mean ~ 248.571 , RV ~ 0.161) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Table 8-9. Validation Metrics, BIT Predicts NVDA

BIT >>> NVDA

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.628	3.418	2092.229	0.256	0.044	0.988	0.955	0.974	0.911
+ *arima200*	0.646	3.393	2107.286	0.706	0	0.971	0.865	0.886	0.915
* *arima300*	0.663	3.528	2219.773	0.942	0	0.942	0.749	0.888	0.895
varima100	0.597	3.586	2196.879	0.741	0.139	0.825	0.849	0.943	0.893
varima200	0.644	3.432	2130.536	0.706	0.039	0.745	0.653	0.936	0.907
varima300	0.644	3.6	2254.068	0.635	0.006	0.647	0.533	0.898	0.895

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *NVDA =(Mean ~ 248.571 , RV ~ 0.161)

This lackluster performance of VARIMA compared to ARIMA is also observable for Bitcoin paired with the engineered principal component, as shown in Table 8-10 below.

Table 8-10. Validation Metrics, pc1 Predicts BIT

pc1 >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.702	110.328	1887276.03	0.607	0.176	0.737	0.859	0.957	0.928
varima200	0.706	98.818	1690683.524	0.614	0.082	0.644	0.77	0.88	0.933
varima300	0.72	97.088	1702637.016	0.706	0.036	0.548	0.666	0.725	0.945

*pc1 =(Mean ~ 0 , RV ~ NaN) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

This trend continues for the other financial related time series (DJI, GSPC, IXIC) when paired with Bitcoin as well.

Graphs and tables containing model results for all bivariate pairs provided in the appendix for additional reading on pg. 100.

9 Conclusions and Future Work

ARIMA models with a 200-length training window performed the best overall, having the most pairs with highest directional forecast accuracy as well as the lowest MAE. However, VARIMA with 200, 300 training length often had higher forecast accuracy when the cross correlations were high. These high correlations were mainly seen when pairing Bitcoin with other cryptocurrencies, whereas bivariate pairs containing trading volumes and other financial data performed worse for VARIMA than ARIMA.

Overall, all 6 of the model variations of training length/model type performed very similarly across the board for all bivariate pairs, with consistently solid results above 70 percent directional forecast accuracy. For directional forecasting to be a worthwhile avenue for investing, the accuracy needs to be above 55 percent in order to have glimmer of profitability, which all 6 model variations handily surpass. Several different ML methods presented in a recent paper for forecasting Bitcoin, have only reported DFs in 50 and 60 percent range by comparison Mudassir et al. (2020).

However, an unfortunate concession of the VARIMA models used is that they take significantly longer to train than the ARIMA models with what appears to be marginal gain. One potential avenue for the future to boost VARIMA's forecasting accuracy would be to iterate throughout all combinations of p and q in range 0 through 3 rather than 0 through 2, which could produce new VARIMA models with a lower AIC. However, this would considerably increase the already substantial runtime of VARIMA. Training on tri-variate time series models from different cryptocurrencies also could improve results. Again, this would significantly increase the runtime of model training.

Another possibility for improvement would be to make use of high lagged cross correlations, where one time series *leads* another. This will give the ability to offset one time series by the number of lagged days with the highest cross-correlation. Not only will this increase the correlation when training models, which VARIMA models capitalize on, but by offsetting it will also make use of data that correlates well with the future. I have included the cross-correlation plots for each time series pair for reference in the appendix (as well as plots for the forecasted CI's and forecasted means).

Another thing to consider when assessing the marginal improvement of VARIMA over ARIMA is that the process of differencing before model training strips away a considerable amount of information of the original time series. This loss of information several limits any improvements that VARMA models can capitalize on. One alternative to differencing to achieve stationarity would be to check for cointegration within the multidimensional time series, choosing only to difference if no cointegration exists. This could be potentially promising as others have recent published that there exists cointegration amongst different cryptocurrencies Göttfert et al. (2019). However, like the previous considerations, this would also considerably increase the training time involved. Using GARCH models moving forward would also allow each model to consider the volatility of each time series to use as an additional variable for prediction. While this work focused solely on one day forecast, a next step moving forward would be to build on these results to do long term forecasting of Bitcoin and other cryptocurrencies.

References

1. Coinmarketcap.com
2. Adhikari, R. and Agrawal, R. K. (2013). An introductory study on time series modeling and forecasting. arXiv preprint arXiv:1302.6613.
3. Bessembinder, H. and Seguin, P. J. (1993). Price volatility, trading volume, and market depth: Evidence from futures markets. Journal of financial and Quantitative Analysis, 28(1):21–39.
4. Blume, L., Easley, D., and O'hara, M. (1994). Market statistics and technical analysis: The role of volume. The Journal of Finance, 49(1):153–181.
5. Bonato, M. (2011). Robust estimation of skewness and kurtosis in distributions with infinite higher moments. Finance Research Letters, 8(2):77–87.
6. Brockwell, P. J. and Davis, R. A. (2016). Introduction to time series and forecasting. springer. Campbell, J. Y., Champbell, J. J., Campbell, J. W., Lo, A. W., Lo, A. W., and MacKinlay, A. C. (1997). The econometrics of financial markets. princeton University press.
7. Chatfield, C. (1996). Model uncertainty and forecast accuracy. Journal of Forecasting, 15(7):495–508.
8. Clark, P. K. (1973). A subordinated stochastic process model with finite variance for speculative prices. Econometrica: journal of the Econometric Society, pages 135–155.
9. Cochrane, J. H. (2005). Time series for macroeconomics and finance. Manuscript, University of Chicago, pages 1–136.
10. Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues.
11. Cont, R. (2005). Long range dependence in financial markets. In Fractals in engineering, pages 159–179. Springer.
12. Cotter, J. (2011). Absolute return volatility. arXiv preprint arXiv:1103.5976.
13. Engle, R. (2004). Risk and volatility: Econometric models and financial practice. American economic review, 94(3):405–420.
14. Epps, T. W. and Epps, M. L. (1976). The stochastic dependence of security price changes and transaction volumes: Implications for the mixture-of-distributions hypothesis. Econometrica: Journal of the Econometric Society, pages 305–321.
15. Fama, E. F. (1965). The behavior of stock-market prices. The journal of Business, 38(1):34–105.
16. Frohn, J. (1995). Grundausbildung in Ökonometrie. Walter de Gruyter.

17. Fryzlewicz, P. (2007). Lecture notes: Financial time series, arch and garch models. University of Bristol.

18. Gallant, A. R., Rossi, P. E., and Tauchen, G. (1992). Stock prices and volume. The Review of Financial Studies, 5(2):199–242.

19. Garg, V. K. and Wang, Y.-C. (2005). Signal types, properties, and processes. In The Electrical Engineering Handbook, pages 951–956. Elsevier.

20. Giot, P., Laurent, S., and Petitjean, M. (2010). Trading activity, realized volatility and jumps. Journal of Empirical Finance, 17(1):168–175.

21. Hipel, K. W. and McLeod, A. I. (1994). Time series modelling of water resources and environmental systems, volume 45. Elsevier.

22. Jensen, M. H., Johansen, A., and Simonsen, I. (2003). Inverse fractal statistics in turbulence and finance. International Journal of Modern Physics B, 17(22n24):4003–4012.

23. Johansen, A., Simonsen, I., and Jensen, M. H. (2006). Optimal investment horizons for stocks and markets. Physica A: Statistical Mechanics and its Applications, 370(1):64–67.

24. Karpio, K., Za□luska-Kotur, M. A., and Or□lowski, A. (2007). Gain–loss asymmetry for emerging stock markets. Physica A: Statistical Mechanics and its Applications, 375(2):599–604.

25. Karpoff, J. M. (1987). The relation between price changes and trading volume: A survey. Journal of Financial and quantitative Analysis, 22(1):109–126.

26. Kavanagh, S. and Williams, D. (2014). Making the best use of judgmental forecasting. GovernmentFinance Review. http://www.gfoa.org/sites/default/files/GFR61508. pdf.

27. Kendall, M. G. and Hill, A. B. (1953). The analysis of economic time-series-part i: Prices. Journal of the Royal Statistical Society. Series A (General), 116(1):11–34.

28. Kim, T.-H. and White, H. (2004). On more robust estimation of skewness and kurtosis. Finance Research Letters, 1(1):56–73.

29. Lawrence,M.,Goodwin,P.,O’Connor,M.,andÖnkal,D.(2006).Judgmentalforecasting:Areviewofpr ogressover the last 25 years. International Journal of forecasting, 22(3):493–518.

30. Li, W. K. and McLeod, A. I. (1986). Fractional time series modelling. Biometrika, 73(1):217–221. Maindonald, J. H. (2009). Time series analysis with applications in r, by jonathan d. cryer, kung-sik chan. International Statistical Review, 77(2):300–301.

31. Mandelbrot, B. (1967). The variation of some other speculative prices. The Journal of Business, 40(4):393–413.

32. Park, H. (1999). Forecasting three-month treasury bills using arima and garch models. Econ930, Department of Economics, Kansas State University, Tech. Rep.

33. Ruppert, D. (2004). Statistical analysis, special problems of: transformations of data. International Encyclopedia of the Social & Behavioral Sciences, pages 15007–15014.

34. Rydberg, T. H. (2000). Realistic statistical modelling of financial data. International Statistical Review, 68(3):233– 258.

35. Shumway, R. H. and Stoffer, D. S. (2017). Time series analysis and its applications: with R examples. Springer.

36. Tsay, R. S. (2013). Multivariate time series analysis: with R and financial applications. John Wiley & Sons.

37. Tsay, R. S. (2014). Financial time series. Wiley StatsRef: Statistics Reference Online, pages 1–23.

38. Wang, T., Huang, Z., et al. (2012). The relationship between volatility and trading volume in the chinese stock market: a volatility decomposition perspective. Annals of economics and finance, 13(1):211–236.

39. Wei, W. W. (2006). Time series analysis. In The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2. Yin, W. (2010). An empirical research on china's stock market's volume-volatility relationship. World Economic Ourlook, 3:66–79.

40. Yule, G. U. (1927). Vii. on a method of investigating periodicities disturbed series, with special reference to wolfer's sunspot numbers. Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 226(636-646):267–298.

41. Zaluska-Kotur, M. A., Karpio, K., and Orlowski, A. (2006). Comparison of gain-loss asymmetry behavior for stocks and indexes. arXiv preprint physics/0608214.

42. Zhang, G. P. (2003). Time series forecasting using a hybrid arima and neural network model. Neurocomputing, 50:159–175.

43. Zhang, G. P. (2007). A neural network ensemble method with jittered training data for time series forecasting. Information Sciences, 177(23):5329–5346.

44. Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. (2014). Stock price prediction using the arima model. In 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pages 106–112. IEEE.

45. Chen, S., Lan, X., Hu, Y., Liu, Q., and Deng, Y. (2014). The time series forecasting: from the aspect of network. arXiv preprint arXiv:1403.1713.

46. Cryer, J. D. and Kellet, N. (1991). Time series analysis. Springer. Damsleth, E. and El-Shaarawi, A. (1989). Arma models with double-exponentially distributed noise. Journal of the Royal Statistical Society: Series B (Methodological), 51(1):61–69.

47. Dong, Y., Li, S., and Gong, X. (2017). Time series analysis: An application of arima model in stock price forecasting. In 2017 International Conference on Innovations in Economic Management and Social Science (IEMSS 2017). Atlantis Press.

48. Edward, A. (2016). Forecast model using arima for stock prices of automobile sector. International Journal of Research in Finance and Marketing, 6(4):1–9.

49. Faruk, D. Ö. (2010). A hybrid neural network and arima model for water quality time series prediction. Engineering applications of artificial intelligence, 23(4):586–594.

50. Hamilton, J. D. (1994). Time series analysis, volume 2. Princeton New Jersey.

51. Hyndman, R. (2014). Variations on rolling forecasts.

52. Karakoyun, E. and Cibikdiken, A. (2018). Comparison of arima time series model and lstm deep learning algorithm for bitcoin price forecasting. In The 13th multidisciplinary academic conference in prague 2018 (the 13th mac 2018), pages 171–180.

53. Leitch, G. and Tanner, J. E. (1991). Economic forecast evaluation: profits versus the conventional error measures. The American Economic Review, pages 580–590.

54. Mendes, D. et al. (2009). Comparative Study Of Artificial Neural Network And Box-Jenkins Arima For Stock Price Indexes. PhD thesis.

55. Mishra, A. and Desai, V. (2005). Drought forecasting using stochastic models. Stochastic Environmental Research and Risk Assessment, 19(5):326–339.

56. Mondal, P., Shit, L., and Goswami, S. (2014). Study of effectiveness of time series modeling (arima) in forecasting stock prices. International Journal of Computer Science, Engineering and Appli- cations, 4(2):13.

57. Navares, R., Díaz, J., Linares, C., and Aznarte, J. L. (2018). Comparing arima and computational intelligence methods to forecast daily hospital admissions due to circulatory and respiratory causes in madrid. Stochastic Environmental Research and Risk Assessment, 32(10):2849–2859. 11

58. Naylor, T. H., Seaks, T. G., and Wichern, D. W. (1972). Box-jenkins methods: An alternative to econometric models. International Statistical Review/Revue Internationale de Statistique, pages 123–137.

59. Ozturk, S. and Ozturk, F. (2018). Forecasting energy consumption of turkey by arima model. Journal of Asian Scientific Research, 8(2):52.

60. Pesaran, M. H. and Timmermann, A. (1992). A simple nonparametric test of predictive performance. Journal of Business & Economic Statistics, 10(4):461–465.

61. Petrică, A.-C., Stancu, S., and Tindeche, A. (2016). Limitation of arima models in financial and monetary economics. Theoretical & Applied Economics, 23(4).

62. Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2018). A comparison of arima and lstm in forecasting time series. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 1394–1401. IEEE.

63. Šťádník, B. et al. (2013). Market price forecasting and profitability–how to tame random walk? Verslas: teorija ir praktika, 14(2):166–176.

64. Time, C. O., Plots, E. Q.-Q., Rates, E., Designs, F., Fechner, G. T., Fisher, S. R. A., Galton, F., and Gauss, J. C. F. Encyclopedia of statistics in behavioral science–volume 2–page 1 of 4.

65. Zivot, E. and Wang, J. (2007). Modeling financial time series with S-PlusR , volume 191. Springer Science & Business Media.

66. Akaike, H. (1969). Fitting autoregressive models for prediction. Annals of the institute of Statistical Mathematics, 21(1):243–247.

67. Akaike, H. (1974). A new look at the statistical model identification. IEEE transactions on automatic control, 19(6):716–723.

68. Andreoni, A. and Postorino, M. N. (2006). A multivariate arima model to forecast air transport demand. Proceedings of the Association for European Transport and Contributors, pages 1–14.

69. Athanasopoulos, G., Poskitt, D. S., and Vahid, F. (2012). Two canonical varma forms: Scalar component models vis-`a-vis the echelon form. Econometric Reviews, 31(1):60–83.

70. Athanasopoulos, G. and Vahid, F. (2008). A complete varma modelling methodology based on scalar components. Journal of Time Series Analysis, 29(3):533–554.

71. Beukelman, T. and Brunner, H. I. (2015). Trial design, measurement, and analysis of clinical investigations. Textbook of Pediatric Rheumatology E-Book, page 54.

72. Boubacar Mainassara, Y. (2012). Selection of weak varma models by modified akaike's information criteria. Journal of Time Series Analysis, 33(1):121–130.

73. Box, G. E., Hillmer, S., and Tiao, G. C. (1979). Analysis and modeling of seasonal time series. In Seasonal analysis of economic time series, pages 309–346. NBER.

74. Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.

75. Brandt, P. T. and Williams, J. T. (2006). Multiple time series models, volume 148. Sage Publications, Incorporated. Chitturi, R. V. (1974). Distribution of residual autocorrelations in multiple autoregressive schemes. Journal of the American Statistical Association, 69(348):928–934.

76. Deistler, M. and Hannan, E. (1988). The statistical theory of linear systems.

77. Dias, G. F. and Kapetanios, G. (2014). Forecasting medium and large datasets with Vector Autoregressive Moving Average (VARMA) models. School of Economics and Management.

78. Doukhan, P., Massart, P., and Rio, E. (1995). Invariance principles for absolutely regular empirical processes. In Annales de l'IHP Probabilit´es et statistiques, volume 31, pages 393–427.

79. Dufour, J.-M. and Pelletier, D. (2008). Practical methods for modelling weak varma processes: Identification, estimation and specification with a macroeconomic application. Manuscript, McGill University.

80. Elliott, G. and Timmermann, A. (2013). Handbook of economic forecasting. Elsevier. Fackler, J. S. and Krieger, S. C. (1986). An application of vector time series techniques to macroeconomic forecasting.

81. Journal of Business & Economic Statistics, 4(1):71–80. Francq, C. and Ra¨issi, H. (2007). Multivariate portmanteau test for autoregressive models with uncorrelated but

82. nonindependent errors. Journal of Time Series Analysis, 28(3):454–470. Gredenhoff, M. and Karlsson, S. (1999). Lag-length selection in var-models using equal and unequal lag-length procedures. Computational Statistics, 14(2):171–187.

83. Hanan, M. (1976). Venturing corporations-think small to stay strong. Harvard Business Review, 54(3):139–148.

84. Hanan, P. (1970). Sung and yu¨an vernacular fiction: A critique of modern methods of dating. Harvard Journal of Asiatic Studies, pages 159–184.

85. Hannan, E. (1969). The estimation of mixed moving average autoregressive systems. Biometrika, 56(3):579–593. Hannan, E. J. and Rissanen, J. (1982). Recursive estimation of mixed autoregressive-moving average order.

86. Biometrika, 69(1):81–94. Hatemi-J, A. and S. Hacker, R. (2009). Can the lr test be helpful in choosing the optimal lag order in the var model when information criteria suggest different lag orders? Applied Economics 41(9):1121-1125.

87. Hosking, J. (1981). Equivalent forms of the multivariate portmanteau statistic. Journal of the Royal Statistical Society: Series B (Methodological), 43(2):261–262.

88. Hosking, J. R. (1980). The multivariate portmanteau statistic. Journal of the American Statistical Association, 75(371):602–608.

89. Hu, C., Hu, Y., and Seo, S. (2019). A deep structural model for analyzing correlated multivariate time series. In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), pages 69–74. IEEE.

90. Hurvich, C. M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples. Biometrika, 76(2):297–307.

91. Johansen, S. (1995). Likelihood-based inference in cointegrated vector autoregressive models. Oxford University Press on Demand.

92. Jones, S. S., Evans, R. S., Allen, T. L., Thomas, A., Haug, P. J., Welch, S. J., and Snow, G. L. (2009). A multivariate time series approach to modeling and forecasting demand in the emergency department. Journal of biomedical informatics, 42(1):123–139.

93. Kanchymalay, K., Salim, N., Sukprasert, A., Krishnan, R., and Hashim, U. R. (2017). Multivariate time series forecasting of crude palm oil price using machine learning techniques. In IOP Conference Series: Materials Science and Engineering, volume 226, page 012117. IOP Publishing.

94. Karlssony, M. G. S. (1997). Lag-length selection in var-models using equal and unequal lag-length procedures. Kascha, C. (2012). A comparison of estimation methods for vector autoregressive moving-average models. Econometric Reviews, 31(3):297–324.

Kascha, C. and Mertens, K. (2009). Business cycle analysis and varma models. Journal of Economic Dynamics and Control, 33(2):267–282.

95. Kascha, C. and Ravazzolo, F. (2010). Combining inflation density forecasts. Journal of forecasting, 29(1-2):231–250.

96. Kascha, C. and Trenkler, C. (2011). Bootstrapping the likelihood ratio cointegration test in error correction models with unknown lag order. Computational statistics & data analysis, 55(2):1008–1017.

97. Kumar, K. and Ganesalingam, S. (2001). Detection of financial distress via multivariate statistical analysis. Detection and Prediction of Financial Distress, 27(4):45–55.

98. Lajevardi, S. B. and Minaei-Bidgoli, B. (2008). Comparison between ann and decision tree in aerology event predic- tion. In 2008 International Conference on Advanced Computer Theory and Engineering, pages 533–537. IEEE.

99. Levitt, E. J., Pelletier, D. L., Dufour, C., and Pell, A. N. (2011). Harmonizing agriculture and health sector actions to improve household nutrition: policy experiences from afghanistan (2002–2007). Food Security, 3(3):363.

100. Li, W. and McLeod, A. (1981). Distribution of the residual autocorrelations in multivariate arma time series models. Journal of the Royal Statistical Society: Series B (Methodological), 43(2):231–239.

101. Long, F. H. (2013). Multivariate analysis for metabolomics and proteomics data. In Proteomic and Metabolomic Approaches to Biomarker Discovery, pages 299–311. Elsevier.

102. Lu ̈tkepohl, H. (2005). New introduction to multiple time series analysis. Springer Science & Business Media.

103. Lu ̈tkepohl, H. (2013). Introduction to multiple time series analysis. Springer Science & Business Media.

104. Lu ̈tkepohl, H. and Poskitt, D. S. (1991). Estimating orthogonal impulse responses via vector autoregressive models. Econometric Theory, 7(4):487–496.

105. Lu ̈tkepohl, H. and Poskitt, D. S. (1996). Specification of echelon-form varma models. Journal of Business & Economic Statistics, 14(1):69–79.

106. Miranda-Agrippino, S. and Ricco, G. (2019). Bayesian vector autoregressions: Estimation. In Oxford Research Encyclopedia of Economics and Finance.

107. Quenouille, M. H. and Quenouille, M. (1957). The analysis of multiple time-series, volume 1. Griffin London.

108. Sargent, T. J., Fernandez-Villaverde, J., and Rubio-Ramirez, J. (2005). A, B, C's (and D)'s for Understanding VARs. National Bureau of Economic Research.

109. Simmonds, J., Gómez, J. A., and Ledezma, A. (2017). Data preprocessing to enhance flow forecasting in a tropical river basin. In International Conference on Engineering Applications of Neural Networks, pages 429–440. Springer.

110. Sims, C. A. (1980). Comparison of interwar and postwar business cycles: Monetarism reconsidered. Technical report, National Bureau of Economic Research.

111. Sperling, R. and Baum, C. F. (2001). sts19: Multivariate portmanteau (q) test for white noise. Stata Technical Bulletin Reprints, 10:373–375.

112. Tiao, G. C. and Box, G. E. (1981). Modeling multiple time series with applications. journal of the American Statistical Association, 76(376):802–816.

113. Tiao, G. C. and Tsay, R. S. (1983). Multiple time series modeling and extended sample cross-correlations. Journal of Business & Economic Statistics, 1(1):43–56.

114. Tiao, G. C. and Tsay, R. S. (1989). Model specification in multivariate time series. Journal of the Royal Statistical Society: Series B (Methodological), 51(2):157–195.

115. Tsay, R. S. (2013). Multivariate time series analysis: with R and financial applications. John Wiley & Sons.

116. Vayej, S. M. (2012). Multivariate Time Series Modelling. PhD thesis, University of KwaZulu-Natal, Westville.

117. Wan, R., Mei, S., Wang, J., Liu, M., and Yang, F. (2019). Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. Electronics, 8(8):876.

118. Wilson, G. T. (1973). The estimation of parameters in multivariate time series models. Journal of the Royal Statistical Society: Series B (Methodological), 35(1):76–85.

119. Wilson, G. T., Reale, M., and Morton, A. S. (2001). Developments in multivariate time series modeling. Department of Mathematics and Statistics, University of Canterbury.

120. Wu, Y., Hernández-Lobato, J. M., and Ghahramani, Z. (2013). Dynamic covariance models for multivariate financial time series. arXiv preprint arXiv:1305.4268.

121. Yap, S. F. and Reinsel, G. C. (1995). Estimation and testing for unit roots in a partially nonstationary vector autoregressive moving average model. Journal of the American Statistical Association, 90(429):253–267.

122. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., and Liu, Y. (2017). Deep learning: A generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM international Conference on Data Mining, pages 777–785. SIAM.

123. Zhao, Z., Shi, P., and Zhang, Z. (2018). Modeling multivariate time series with copula-linked univariate d-vines. arXiv preprint arXiv:1805.0333

124. Geron, A. (2017). Hands-on machine learning with scikit-learn and tensorflow: Concepts, Tools, And Techniques to Build Intelligent Systems.
125. Mingqiang, Y. and Kidiyo, K., Joseph, R. (2008). Pattern recognition A survey of shape feature extraction techniques.
126. Kumar, G. and Bhatia, P.K. (2014). A detailed review of feature extraction in image processing systems.
127. Choras, R.S. (2007). Image feature extraction techniques and their applications for CBIR and biometrics systems.
128. Olszewski, R.T. (2001). Generalized feature extraction for structural pattern recognition in time-series data, by Generalized feature extraction for structural pattern recognition in time-series data.
129. Zhang, H., Ho, T.B., Zhang, Y., and Lin, M-S (2005). Unsupervised feature extraction for time series clustering using orthogonal wavelet transform.
130. Fulcher, B.D. and Jones, N.S. (2017). A computational framework for automated time-series phenotyping using massive feature extraction.
131. Karpagachelvi, S., Arthanari, M., and Sivakumar, M. (2010). ECG feature extraction techniques-a survey approach.
132. Acharya, U.R., Fernandes, S.L., WeiKoh, J.E., Ciaccio, E.J., Fabell, M.K.M., Tanik, U.J., Rajinikanth, V., and Yeong, Chai H. (2019). Automated detection of Alzheimer's disease using brain MRI images--a study with various feature extraction techniques.
133. Maaten, L., Postma, E., and Herik, Jaap. (2009) Dimensionality reduction: a comparative review.
134. Fukunaga, K. (1990). Introduction to Statistical Pattern Recognition. Academic Press Professional, Inc.,San Diego, CA, USA.
135. Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. Philiosophical Magazine, 2:559–572.
136. Spearman, C. (1904). General intelligence objectively determined and measured. American Journal of Psychology, 15:206–22.
137. Torgerson, W.S. (1952). Multidimensional scaling I: Theory and method. Psychometrika, 17:401–419,1952
138. Burges, C.J.C. (2005). Data Mining and Knowledge Discovery Handbook: A Complete Guide for Prac-titioners and Researchers, chapter Geometric Methods for Feature Selection and DimensionalReduction: A Guided Tour. Kluwer Academic Publishers.
139. Saul, L.K., Weinberger, K.Q., Ham, J.H., Sha F., and Lee. D.D. (2006) Spectral methods for dimensional-ity reduction. InSemisupervised Learning, Cambridge, MA, USA. The MIT Press.
140. Venna, J. (2007). Dimensionality reduction for visual exploration of similarity structures. PhD thesis,Helsinki University of Technology.
141. Lee, J.A. and Verleysen, M. (2007). Nonlinear dimensionality reduction. Springer, New York, NY, USA.
142. Johnson, R.A., Wichern, D.W. and others (2017). Applied multivariate statistical analysis.
143. Mudassir, M., Bennbaia, S., Unal, D., and Hammoudeh, M. (2020) Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach Neural Computing and Applications.
144. Göttfert, Jolin Umeå University, Faculty of Social Sciences, Umeå School of Business and Economics (USBE), Economics. (2019). Cointegration among cryptocurrencies: A cointegration analysis of Bitcoin, Bitcoin Cash, EOS, Ethereum, Litecoin and Ripple.

Appendix A. Graphs and Plots

The following sub sections contain model validation tables and plots for cross-correlations, forecasted confidence intervals for VARIMA, and forecasted means for ARIMA and VARIMA.

A.1 Validation Tables

Dogecoin >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.704	111.407	1911297.595	0.878	0.866	0.688	0.852	0.973	0.923
**varima200*	0.742	96.851	1662468.248	0.926	0.906	0.54	0.825	0.894	0.939
varima300	0.722	95.615	1668615.412	1	0.994	0.341	0.767	0.838	0.942

*Dogecoin =(Mean ~ 0.003 , RV ~ 0.317) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Binance >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.715	109.183	1881548.443	0.743	0.959	0.683	0.852	0.94	0.932
varima200	0.714	97.462	1657465.048	0.865	0.994	0.439	0.591	0.862	0.939
varima300	0.712	96.279	1673884.151	0.885	1	0.3	0.39	0.819	0.944

*Binance =(Mean ~ 16.045 , RV ~ 0.234) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

BitcoinCash >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.709	109.588	1884161.239	0.945	0.961	0.743	0.895	0.984	0.929
varima200	0.727	95.296	1632940.485	0.989	0.96	0.749	0.822	0.992	0.942
+*varima300*	0.721	93.781	1636273.056	0.985	0.955	0.78	0.795	0.992	0.941

*BitcoinCash =(Mean ~ 541.36 , RV ~ 0.981) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Cardano >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.712	108.967	1875398.388	0.909	0.984	0.779	0.838	0.935	0.926
varima200	0.725	97.576	1679727.529	0.884	1	0.562	0.64	0.865	0.94
varima300	0.72	95.041	1669808.228	0.831	1	0.562	0.327	0.823	0.944

*Cardano =(Mean ~ 0.119 , RV ~ 1.427) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Chainlink >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.711	110.995	1927078.338	0.801	0.783	0.737	0.779	0.953	0.918
varima200	0.71	100.024	1722966.303	0.747	0.862	0.54	0.528	0.992	0.938
varima300	0.712	97.892	1718069.764	0.793	0.932	0.371	0.35	0.926	0.939

*Chainlink =(Mean ~ 2.439 , RV ~ 1.923) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

DJI >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.699	114.313	1953459.129	0.644	0.124	0.802	0.85	0.942	0.922
varima200	0.725	98.058	1658214.941	0.818	0.191	0.662	0.603	0.926	0.939
varima300	0.722	96.414	1682887.909	0.655	0.095	0.565	0.486	0.9	0.94

*DJI =(Mean ~ 25795.788 , RV ~ 0.004) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Ethereum >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.703	110.837	1916865.963	0.929	1	0.682	0.788	0.878	0.923
varima200	0.727	96.96	1662934.343	1	1	0.466	0.614	0.926	0.938
**varima300*	0.734	94.621	1660783.695	1	1	0.21	0.319	0.947	0.945

*Ethereum =(Mean ~ 321.314 , RV ~ 0.512) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

GSPC >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.698	113.732	1946725.619	0.691	0.15	0.768	0.845	0.959	0.919
**varima200*	0.734	97.931	1660493.24	0.805	0.175	0.663	0.595	0.909	0.943
varima300	0.726	95.564	1668033.264	0.711	0.09	0.597	0.526	0.876	0.945

*GSPC =(Mean ~ 2903.457 , RV ~ 0.007) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

IXIC >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.701	115.13	1977684.382	0.661	0.164	0.76	0.87	0.964	0.918
varima200	0.713	99.193	1688030.925	0.86	0.145	0.657	0.662	0.974	0.935
varima300	0.71	96.956	1690590.191	0.744	0.072	0.622	0.595	0.945	0.944

*IXIC =(Mean ~ 8218.415 , RV ~ 0.023) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Litecoin >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.704	111.774	1932196.976	0.956	1	0.696	0.809	0.932	0.921
varima200	0.723	95.963	1643264.703	0.963	1	0.433	0.561	0.928	0.941
**varima300*	0.729	94.598	1663088.229	1	1	0.344	0.384	0.878	0.942

*Litecoin =(Mean ~ 80.857 , RV ~ 0.443) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Monero >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.709	108.316	1864976.428	0.987	1	0.702	0.864	0.968	0.921
varima200	0.718	96.707	1658124.114	1	1	0.438	0.625	0.986	0.941
varima300	0.726	95.151	1668603.783	1	1	0.348	0.444	0.982	0.937

*Monero =(Mean ~ 111.63 , RV ~ 0.527) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

NVDA >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.697	112.781	1951650.597	0.741	0.139	0.825	0.849	0.943	0.925
varima200	0.723	98.971	1687326.023	0.706	0.039	0.745	0.653	0.936	0.939
varima300	0.707	97.364	1697360.904	0.635	0.006	0.647	0.533	0.898	0.94

*NVDA =(Mean ~ 248.571 , RV ~ 0.161) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

pc1 >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.702	110.328	1887276.03	0.607	0.176	0.737	0.859	0.957	0.928
varima200	0.706	98.818	1690683.524	0.614	0.082	0.644	0.77	0.88	0.933
varima300	0.72	97.088	1702637.016	0.706	0.036	0.548	0.666	0.725	0.945

*pc1 =(Mean ~ 0 , RV ~ NaN) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

GSPC >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.698	113.732	1946725.619	0.691	0.15	0.768	0.845	0.959	0.919
**varima200*	0.734	97.931	1660493.24	0.805	0.175	0.663	0.595	0.909	0.943
varima300	0.726	95.564	1668033.264	0.711	0.09	0.597	0.526	0.876	0.945

*GSPC =(Mean ~ 2903.457 , RV ~ 0.007) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

Tether >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.71	111.61	1909973.615	0.51	0.286	0.766	0.87	0.953	0.925
varima200	0.706	97.497	1660552.959	0.565	0.185	0.564	0.779	0.972	0.94
varima300	0.72	96.6	1676217.844	0.647	0.072	0.414	0.753	0.923	0.939

*Tether =(Mean ~ 1.002 , RV ~ 0) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

XRP >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.707	110.347	1905560.913	0.799	0.955	0.742	0.872	0.99	0.928
**varima200*	0.735	94.783	1628816.339	0.887	0.958	0.785	0.843	0.993	0.943
varima300	0.715	94.19	1651918.303	0.77	0.952	0.767	0.832	0.997	0.944

*XRP =(Mean ~ 0.417 , RV ~ 0.71) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

BIT >>> Dogecoin

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
+ *arima100*	0.704	0	0	0.094	0.032	0.791	0.871	0.959	0.939
arima200	0.704	0	0	0.472	0	0.67	0.767	0.991	0.953
arima300	0.71	0	0	0.721	0	0.674	0.718	0.999	0.967
varima100	0.692	0	0	0.878	0.866	0.688	0.852	0.973	0.929
varima200	0.704	0	0	0.926	0.906	0.54	0.825	0.894	0.946
**varima300*	0.719	0	0	1	0.994	0.341	0.767	0.838	0.966

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Dogecoin =(Mean ~ 0.003 , RV ~ 0.317)

BIT >>> Litecoin

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.724	1.465	285.466	0.072	0	0.915	0.899	0.899	0.934
* *arima200*	0.727	1.089	158.264	0.711	0	0.747	0.64	0.833	0.951
+ *arima300*	0.714	1.031	141.214	0.832	0	0.689	0.477	0.836	0.95
varima100	0.687	1.466	292.768	0.956	1	0.696	0.809	0.932	0.92
varima200	0.71	1.124	163.324	0.963	1	0.433	0.561	0.928	0.938
varima300	0.715	1.044	144.687	1	1	0.344	0.384	0.878	0.942

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Litecoin =(Mean ~ 80.857 , RV ~ 0.443)

BIT >>> Binance

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.7	0.308	11.992	0.219	0.032	0.815	0.862	0.923	0.928
* *arima200*	0.721	0.299	12.139	0.66	0	0.657	0.761	0.87	0.923
+ *arima300*	0.715	0.297	12.415	0.775	0	0.729	0.587	0.818	0.925
varima100	0.718	0.307	11.915	0.743	0.959	0.683	0.852	0.94	0.908
varima200	0.715	0.304	12.262	0.865	0.994	0.439	0.591	0.862	0.916
varima300	0.707	0.301	12.617	0.885	1	0.3	0.39	0.819	0.912

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Binance =(Mean ~ 16.045 , RV ~ 0.234)

BIT >>> BitcoinCash

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
* *arima100*	0.732	9.491	13818.253	0.065	0.005	0.946	0.968	0.98	0.948
arima200	0.729	6.931	6797.744	0.571	0	0.862	0.878	0.992	0.967
+ *arima300*	0.726	5.384	3512.695	0.735	0	0.848	0.834	0.94	0.97
varima100	0.7	10.207	14654.051	0.945	0.961	0.743	0.895	0.984	0.927
varima200	0.724	6.988	6771.459	0.989	0.96	0.749	0.822	0.992	0.957
varima300	0.712	5.674	3651.825	0.985	0.955	0.78	0.795	0.992	0.967

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *BitcoinCash =(Mean ~ 541.36 , RV ~ 0.981)

BIT >>> Cardano

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
* *arima100*	0.718	0.002	0.001	0.105	0.01	0.897	0.919	0.933	0.939
arima200	0.703	0.002	0	0.686	0	0.794	0.629	0.851	0.933
+ *arima300*	0.705	0.001	0	0.915	0	0.744	0.452	0.863	0.931
varima100	0.687	0.002	0.001	0.909	0.984	0.779	0.838	0.935	0.92
varima200	0.71	0.002	0	0.884	1	0.562	0.64	0.865	0.923
varima300	0.7	0.001	0	0.831	1	0.562	0.327	0.823	0.927

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Cardano =(Mean ~ 0.119 , RV ~ 1.427)

BIT >>> Chainlink

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
+ *arima100*	0.728	0.074	1.219	0.105	0.033	0.878	0.919	0.913	0.897
arima200	0.724	0.08	1.326	0.341	0	0.835	0.719	0.986	0.892
* *arima300*	0.729	0.088	1.497	0.69	0	0.777	0.578	0.94	0.878
varima100	0.714	0.074	1.178	0.801	0.783	0.737	0.779	0.953	0.886
varima200	0.722	0.078	1.269	0.747	0.862	0.54	0.528	0.992	0.891
varima300	0.727	0.087	1.453	0.793	0.932	0.371	0.35	0.926	0.859

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Chainlink =(Mean ~ 2.439 , RV ~ 1.923)

BIT >>> Dogecoin

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
+ *arima100*	0.704	0	0	0.094	0.032	0.791	0.871	0.959	0.939
arima200	0.704	0	0	0.472	0	0.67	0.767	0.991	0.953
arima300	0.71	0	0	0.721	0	0.674	0.718	0.999	0.967
varima100	0.692	0	0	0.878	0.866	0.688	0.852	0.973	0.929
varima200	0.704	0	0	0.926	0.906	0.54	0.825	0.894	0.946
**varima300*	0.719	0	0	1	0.994	0.341	0.767	0.838	0.966

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Dogecoin =(Mean ~ 0.003 , RV ~ 0.317)

BIT >>> Ethereum

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
* arima100	0.74	5.192	4280.824	0.131	0.013	0.887	0.846	0.943	0.939
arima200	0.738	4.091	2427.901	0.67	0	0.654	0.584	0.854	0.949
+ arima300	0.732	3.511	1667.395	0.833	0	0.629	0.384	0.864	0.944
varima100	0.725	5.394	4355.107	0.929	1	0.682	0.788	0.878	0.915
varima200	0.724	4.131	2426.761	1	1	0.466	0.614	0.926	0.939
varima300	0.726	3.54	1693.604	1	1	0.21	0.319	0.947	0.944

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Ethereum =(Mean ~ 321.314 , RV ~ 0.512)

BIT >>> Monero

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.717	1.933	568.685	0.108	0.015	0.841	0.931	0.927	0.934
arima200	0.708	1.426	261.588	0.682	0	0.545	0.724	0.795	0.942
+ arima300	0.709	1.245	200.275	0.858	0	0.606	0.745	0.844	0.95
varima100	0.687	2.078	613.447	0.987	1	0.702	0.864	0.968	0.912
varima200	0.7	1.486	276.951	1	1	0.438	0.625	0.986	0.938
*varima300	0.736	1.248	200.599	1	1	0.348	0.444	0.982	0.956

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Monero =(Mean ~ 111.63 , RV ~ 0.527)

BIT >>> NVDA

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.628	3.418	2092.229	0.256	0.044	0.988	0.955	0.974	0.911
+ arima200	0.646	3.393	2107.286	0.706	0	0.971	0.865	0.886	0.915
* arima300	0.663	3.528	2219.773	0.942	0	0.942	0.749	0.888	0.895
varima100	0.597	3.586	2196.879	0.741	0.139	0.825	0.849	0.943	0.893
varima200	0.644	3.432	2130.536	0.706	0.039	0.745	0.653	0.936	0.907
varima300	0.644	3.6	2254.068	0.635	0.006	0.647	0.533	0.898	0.895

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *NVDA =(Mean ~ 248.571 , RV ~ 0.161)

BIT >>> Tether

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
+ *arima100*	0.62	0.002	0.004	0.018	0.007	0.839	0.905	0.947	0.935
arima200	0.604	0.002	0.004	0.103	0	0.787	0.931	0.984	0.943
* *arima300*	0.627	0.002	0.004	0.236	0	0.695	0.906	0.991	0.96
varima100	0.595	0.002	0.004	0.51	0.286	0.766	0.87	0.953	0.916
varima200	0.588	0.002	0.004	0.565	0.185	0.564	0.779	0.972	0.936
varima300	0.601	0.002	0.004	0.647	0.072	0.414	0.753	0.923	0.947

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *Tether =(Mean ~ 1.002 , RV ~ 0)

BIT >>> XRP

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.718	0.007	0.007	0.076	0.023	0.895	0.945	0.972	0.944
+* *arima200*	0.725	0.005	0.004	0.459	0	0.755	0.766	0.876	0.965
arima300	0.725	0.005	0.003	0.684	0	0.824	0.742	0.955	0.975
varima100	0.678	0.007	0.008	0.799	0.955	0.742	0.872	0.99	0.921
varima200	0.706	0.005	0.004	0.887	0.958	0.785	0.843	0.993	0.955
varima300	0.712	0.005	0.003	0.77	0.952	0.767	0.832	0.997	0.972

*BIT =(Mean ~ 8267.592 , RV ~ 0.11) , *XRP =(Mean ~ 0.417 , RV ~ 0.71)

GSPCVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.703	112.806	1943838.642	0.244	0	0.798	0.846	0.929	0.931
**varima200*	0.738	95.619	1638130.475	0.283	0	0.721	0.731	0.872	0.946
varima300	0.724	97.052	1697261.619	0.334	0	0.704	0.666	0.809	0.944

*GSPCVOL =(Mean ~ 3936578671.325 , RV ~ 0.094) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

BinanceVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.708	111.835	1919908.561	0.54	0.025	0.798	0.901	0.935	0.916
varima200	0.715	97.976	1673196.781	0.439	0	0.668	0.644	0.912	0.942
varima300	0.724	96.572	1693002.9	0.487	0	0.474	0.379	0.763	0.941

*BinanceVOL =(Mean ~ 195895188.64 , RV ~ 0.699) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

BitcoinCashVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.707	111.993	1922176.573	0.614	0.221	0.713	0.815	0.962	0.927
**varima200*	0.729	98.091	1680290.055	0.545	0.06	0.523	0.541	0.899	0.936
varima300	0.714	97.284	1693858.461	0.558	0.033	0.403	0.418	0.853	0.935

*BitcoinCashVOL =(Mean ~ 1612542866.108 , RV ~ 1.111) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

BitcoinVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.707	113.741	1947617.035	0.643	0.241	0.795	0.891	0.955	0.924
varima200	0.715	99.846	1715751.049	0.596	0.108	0.642	0.734	0.887	0.94
varima300	0.715	97.955	1722322.487	0.553	0.03	0.598	0.661	0.762	0.947

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

CardanoVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.696	111.302	1900139.58	0.557	0.217	0.793	0.831	0.941	0.921
varima200	0.708	98.32	1683077.765	0.728	0.113	0.588	0.582	0.932	0.944
varima300	0.715	98.219	1728491.711	0.698	0.034	0.506	0.41	0.799	0.944

*CardanoVOL =(Mean ~ 171551688.416 , RV ~ 1.623) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

ChainlinkVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.714	114.191	1954883.866	0.448	0.068	0.639	0.781	0.944	0.92
varima200	0.709	102.728	1757973.649	0.619	0.01	0.425	0.44	0.977	0.93
**varima300*	0.729	100.534	1765873.723	0.733	0.041	0.448	0.371	0.95	0.936

*ChainlinkVOL =(Mean ~ 239257931.156 , RV ~ 4.268) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

DJIVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+ *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.717	110.429	1905801.126	0.178	0.022	0.858	0.917	0.949	0.93
**varima200*	0.743	95.446	1646752.804	0.379	0	0.766	0.803	0.881	0.941
varima300	0.724	97.085	1695574.734	0.084	0	0.731	0.693	0.875	0.939

*DJIVOL =(Mean ~ 358481806.559 , RV ~ 0.237) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

DogecoinVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.712	111.607	1908974.543	0.624	0.034	0.746	0.909	0.987	0.919
varima200	0.728	95.477	1631137.829	0.386	0.011	0.372	0.708	0.992	0.941
varima300	0.705	95.816	1677780.994	0.391	0.001	0.189	0.513	0.992	0.941

*DogecoinVOL =(Mean ~ 62610259.319 , RV ~ 1.374) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

EthereumVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.702	112.698	1932830.397	0.641	0.129	0.802	0.933	0.97	0.928
varima200	0.705	99.27	1708636.39	0.523	0.048	0.785	0.824	0.936	0.941
varima300	0.712	98.392	1726535.413	0.496	0.013	0.656	0.708	0.78	0.94

*EthereumVOL =(Mean ~ 7066441427.27 , RV ~ 0.663) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

IXICVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.693	114.541	1970340.077	0.249	0.006	0.788	0.868	0.965	0.927
varima200	0.723	97.75	1667409.286	0.076	0	0.667	0.698	0.913	0.942
varima300	0.726	96.73	1681241.934	0.109	0	0.517	0.552	0.891	0.941

*IXICVOL =(Mean ~ 2601300401.599 , RV ~ 0.144) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

LitecoinVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.725	111.107	1903161.627	0.729	0.159	0.768	0.879	0.969	0.929
varima200	0.728	96.496	1652548.741	0.689	0.07	0.589	0.763	0.901	0.936
varima300	0.707	97.453	1703260.127	0.566	0.066	0.468	0.607	0.88	0.935

*LitecoinVOL =(Mean ~ 2032518406.671 , RV ~ 0.631) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

MoneroVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.719	115.359	1981703.598	0.494	0.028	0.73	0.903	0.982	0.914
varima200	0.718	108.844	1816411.158	0.437	0	0.708	0.819	0.991	0.939
varima300	0.709	111.883	1802384.132	0.439	0	0.577	0.794	0.986	0.941

*MoneroVOL =(Mean ~ 188441922.167 , RV ~ 51.232) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

NVDAVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.703	110.894	1902334.152	0.2	0.036	0.797	0.909	0.929	0.923
varima200	0.718	95.089	1620537.496	0.193	0	0.701	0.708	0.935	0.944
varima300	0.712	97.493	1692205.847	0.366	0	0.64	0.641	0.946	0.942

*NVDAVOL =(Mean ~ 12915571.383 , RV ~ 0.296) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

TetherVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.7	112.718	1927493.936	0.562	0.152	0.848	0.926	0.976	0.923
varima200	0.71	100.456	1731114.774	0.467	0.076	0.657	0.745	0.842	0.935
varima300	0.71	98.667	1725909.027	0.442	0.03	0.556	0.556	0.711	0.941

*TetherVOL =(Mean ~ 18377974996.902 , RV ~ 0.947) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

XRPVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.71	111.053	1906314.391	0.56	0.146	0.731	0.871	0.943	0.928
varima200	0.718	96.189	1647090.607	0.274	0.066	0.735	0.809	0.983	0.944
varima300	0.72	96.274	1681949.634	0.334	0.004	0.708	0.664	0.92	0.935

*XRPVOL =(Mean ~ 1306935688.234 , RV ~ 0.706) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

LitecoinVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.725	111.107	1903161.627	0.729	0.159	0.768	0.879	0.969	0.929
varima200	0.728	96.496	1652548.741	0.689	0.07	0.589	0.763	0.901	0.936
varima300	0.707	97.453	1703260.127	0.566	0.066	0.468	0.607	0.88	0.935

*LitecoinVOL =(Mean ~ 2032518406.671 , RV ~ 0.631) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

BitcoinVOL >>> Chainlink

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
+ *arima100*	0.728	0.074	1.219	0.105	0.033	0.878	0.919	0.913	0.897
arima200	0.724	0.08	1.326	0.341	0	0.835	0.719	0.986	0.892
* *arima300*	0.729	0.088	1.497	0.69	0	0.777	0.578	0.94	0.878
varima100	0.71	0.078	1.241	0.599	0.018	0.801	0.9	0.934	0.872
varima200	0.72	0.08	1.29	0.747	0.02	0.588	0.605	0.876	0.877
varima300	0.698	0.088	1.447	0.849	0	0.452	0.457	0.765	0.853

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Chainlink =(Mean ~ 2.439 , RV ~ 1.923)

BitcoinVOL >>> Binance

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.7	0.308	11.992	0.219	0.032	0.815	0.862	0.923	0.928
* *arima200*	0.721	0.299	12.139	0.66	0	0.657	0.761	0.87	0.923
+ *arima300*	0.715	0.297	12.415	0.775	0	0.729	0.587	0.818	0.925
varima100	0.675	0.312	12.049	0.463	0.068	0.786	0.806	0.899	0.905
varima200	0.703	0.306	12.35	0.415	0.016	0.553	0.486	0.775	0.914
varima300	0.704	0.306	12.785	0.378	0.009	0.354	0.353	0.735	0.918

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Binance =(Mean ~ 16.045 , RV ~ 0.234)

BitcoinVOL >>> BIT

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.714	109.31	1911403.323	0.173	0.055	0.842	0.878	0.945	0.939
+* *arima200*	0.728	93.796	1610890.771	0.536	0	0.809	0.818	0.971	0.948
arima300	0.726	96.532	1699733.515	0.868	0	0.742	0.636	0.995	0.944
varima100	0.707	113.741	1947617.035	0.643	0.241	0.795	0.891	0.955	0.924
varima200	0.715	99.846	1715751.049	0.596	0.108	0.642	0.734	0.887	0.94
varima300	0.715	97.955	1722322.487	0.553	0.03	0.598	0.661	0.762	0.947

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *BIT =(Mean ~ 8267.592 , RV ~ 0.11)

BitcoinVOL >>> BitcoinCash

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
* arima100	0.732	9.491	13818.253	0.065	0.005	0.946	0.968	0.98	0.948
arima200	0.729	6.931	6797.744	0.571	0	0.862	0.878	0.992	0.967
+ arima300	0.726	5.384	3512.695	0.735	0	0.848	0.834	0.94	0.97
varima100	0.699	10.346	15136.589	0.634	0.072	0.829	0.901	0.928	0.933
varima200	0.703	7.411	7059.504	0.659	0.002	0.768	0.709	0.806	0.96
varima300	0.701	6.065	3986.492	0.593	0	0.709	0.7	0.763	0.967

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *BitcoinCash =(Mean ~ 541.36 , RV ~ 0.981)

BitcoinVOL >>> Cardano

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
* arima100	0.718	0.002	0.001	0.105	0.01	0.897	0.919	0.933	0.939
arima200	0.703	0.002	0	0.686	0	0.794	0.629	0.851	0.933
+ arima300	0.705	0.001	0	0.915	0	0.744	0.452	0.863	0.931
varima100	0.692	0.002	0.001	0.698	0.06	0.75	0.75	0.84	0.921
varima200	0.692	0.002	0	0.717	0	0.612	0.598	0.719	0.921
varima300	0.705	0.001	0	0.593	0	0.541	0.442	0.716	0.926

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Cardano =(Mean ~ 0.119 , RV ~ 1.427)

BitcoinVOL >>> Dogecoin

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
+ arima100	0.704	0	0	0.094	0.032	0.791	0.871	0.959	0.939
arima200	0.704	0	0	0.472	0	0.67	0.767	0.991	0.953
arima300	0.71	0	0	0.721	0	0.674	0.718	0.999	0.967
varima100	0.695	0	0	0.577	0.039	0.788	0.898	0.947	0.934
*varima200	0.715	0	0	0.629	0	0.616	0.849	0.734	0.951
varima300	0.696	0	0	0.478	0	0.568	0.834	0.753	0.964

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Dogecoin =(Mean ~ 0.003 , RV ~ 0.317)

BitcoinVOL >>> Ethereum

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
* *arima100*	0.74	5.192	4280.824	0.131	0.013	0.887	0.846	0.943	0.939
arima200	0.738	4.091	2427.901	0.67	0	0.654	0.584	0.854	0.949
+ *arima300*	0.732	3.511	1667.395	0.833	0	0.629	0.384	0.864	0.944
varima100	0.721	5.534	4509.282	0.665	0.096	0.826	0.881	0.903	0.922
varima200	0.72	4.2	2450.406	0.702	0.033	0.647	0.547	0.783	0.945
varima300	0.731	3.746	1782.297	0.561	0	0.519	0.433	0.789	0.94

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Ethereum =(Mean ~ 321.314 , RV ~ 0.512)

BitcoinVOL >>> Litecoin

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.724	1.465	285.466	0.072	0	0.915	0.899	0.899	0.934
* *arima200*	0.727	1.089	158.264	0.711	0	0.747	0.64	0.833	0.951
+ *arima300*	0.714	1.031	141.214	0.832	0	0.689	0.477	0.836	0.95
varima100	0.691	1.494	296.109	0.633	0.03	0.806	0.849	0.89	0.931
varima200	0.712	1.12	161.017	0.553	0.009	0.688	0.615	0.731	0.942
varima300	0.706	1.101	151.939	0.556	0	0.568	0.34	0.719	0.945

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Litecoin =(Mean ~ 80.857 , RV ~ 0.443)

BitcoinVOL >>> Monero

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.717	1.933	568.685	0.108	0.015	0.841	0.931	0.927	0.934
arima200	0.708	1.426	261.588	0.682	0	0.545	0.724	0.795	0.942
+ *arima300*	0.709	1.245	200.275	0.858	0	0.606	0.745	0.844	0.95
varima100	0.705	2.079	609.811	0.66	0.098	0.779	0.889	0.912	0.919
varima200	0.709	1.505	277.888	0.604	0.008	0.632	0.764	0.735	0.933
**varima300*	0.72	1.297	208.444	0.508	0	0.747	0.772	0.764	0.955

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Monero =(Mean ~ 111.63 , RV ~ 0.527)

BitcoinVOL >>> NVDA

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.628	3.418	2092.229	0.256	0.044	0.988	0.955	0.974	0.911
+ *arima200*	0.646	3.393	2107.286	0.706	0	0.971	0.865	0.886	0.915
* *arima300*	0.663	3.528	2219.773	0.942	0	0.942	0.749	0.888	0.895
varima100	0.579	3.734	2272.362	0.618	0.001	0.922	0.953	0.923	0.888
varima200	0.625	3.511	2186.623	0.67	0	0.901	0.892	0.802	0.901
varima300	0.621	3.606	2269.518	0.463	0	0.851	0.773	0.792	0.887

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *NVDA =(Mean ~ 248.571 , RV ~ 0.161)

BitcoinVOL >>> Tether

	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
+ *arima100*	0.62	0.002	0.004	0.018	0.007	0.839	0.905	0.947	0.935
arima200	0.604	0.002	0.004	0.103	0	0.787	0.931	0.984	0.943
* *arima300*	0.627	0.002	0.004	0.236	0	0.695	0.906	0.991	0.96
varima100	0.592	0.002	0.004	0.235	0.062	0.79	0.87	0.922	0.902
varima200	0.569	0.002	0.004	0.249	0	0.672	0.745	0.885	0.935
varima300	0.597	0.002	0.004	0.255	0	0.598	0.614	0.792	0.956

*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *Tether =(Mean ~ 1.002 , RV ~ 0)

BitcoinVOL >>> XRP

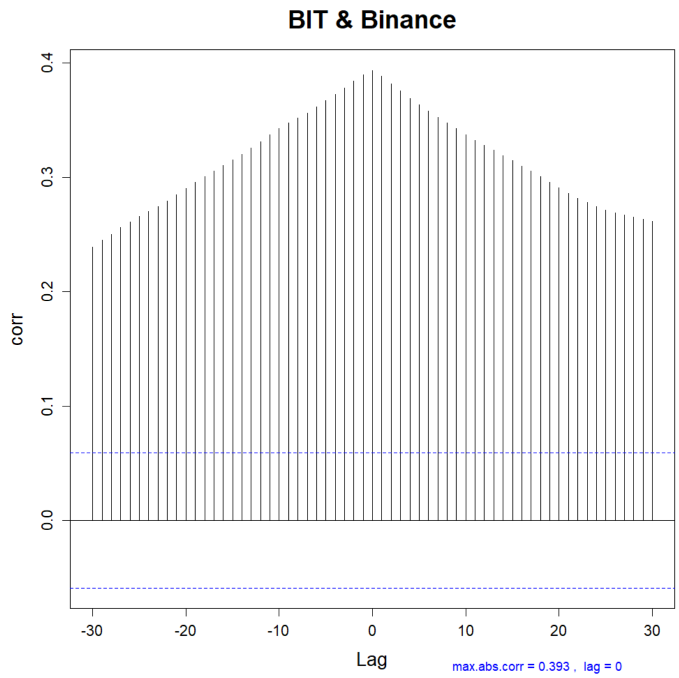
	dir_acc	MAE	MSE	cor	corD	pv10	pv24	pvH	ci
arima100	0.718	0.007	0.007	0.076	0.023	0.895	0.945	0.972	0.944
+ *arima200*	0.725	0.005	0.004	0.459	0	0.755	0.766	0.876	0.965
arima300	0.725	0.005	0.003	0.684	0	0.824	0.742	0.955	0.975
varima100	0.709	0.008	0.008	0.537	0.056	0.776	0.838	0.896	0.924
varima200	0.718	0.005	0.004	0.514	0	0.765	0.788	0.768	0.957
**varima300*	0.73	0.005	0.003	0.504	0	0.689	0.7	0.794	0.976

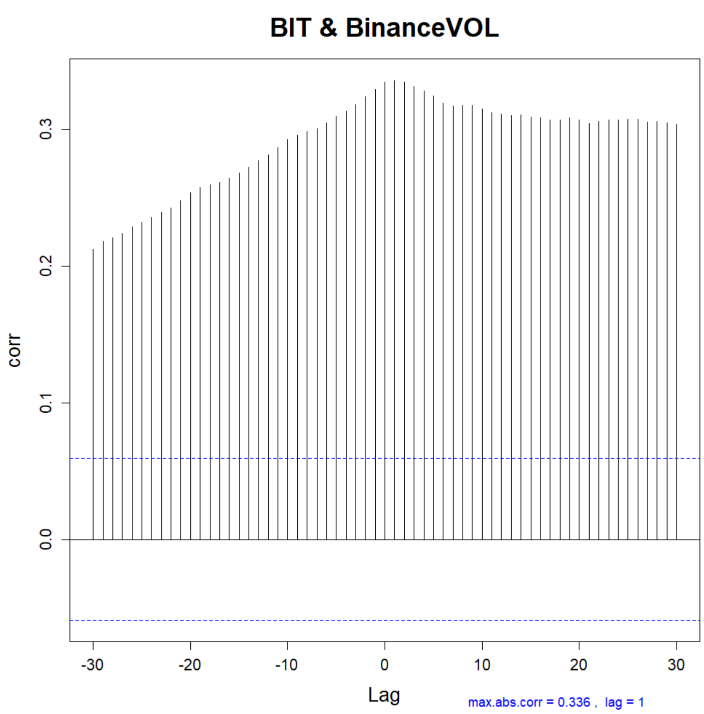
*BitcoinVOL =(Mean ~ 17025035506.785 , RV ~ 0.579) , *XRP =(Mean ~ 0.417 , RV ~ 0.71)

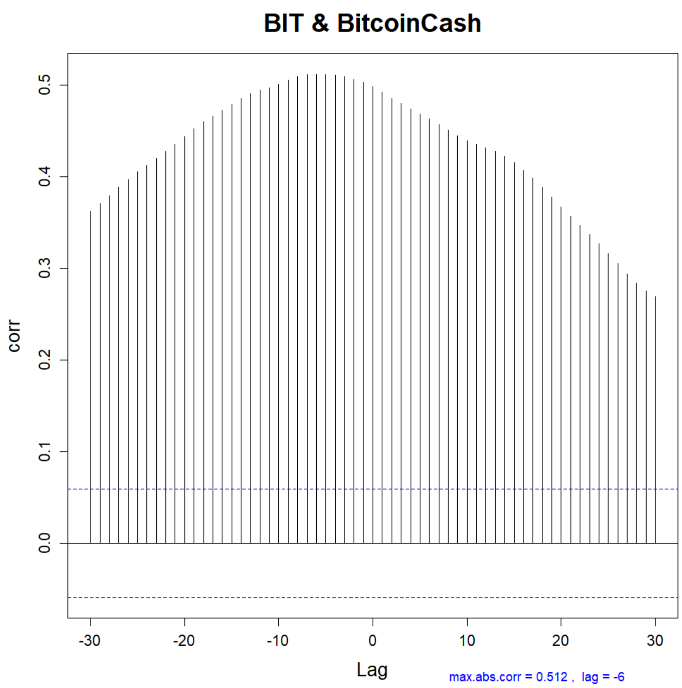
A.2 Cross-Correlation Plots

If the lag is negative, the first mentioned time series leads (predicts) the second one according to the lag. If the lag is positive, the second one leads the first by the given lag. First are the ccf plots for undifferenced data followed by differenced data. For the differenced graphs, each time series is differenced prior by the smallest value that would make it stationary.

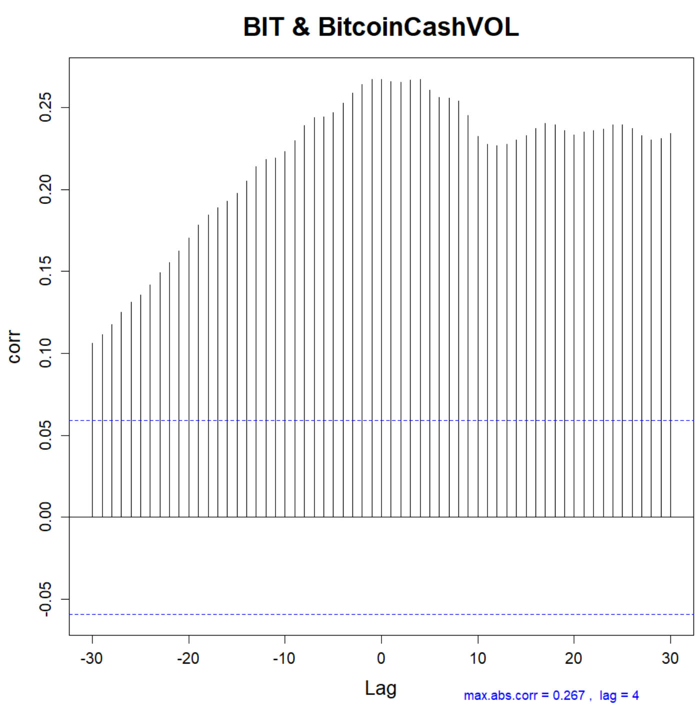
A.2.1 Undifferenced CCF



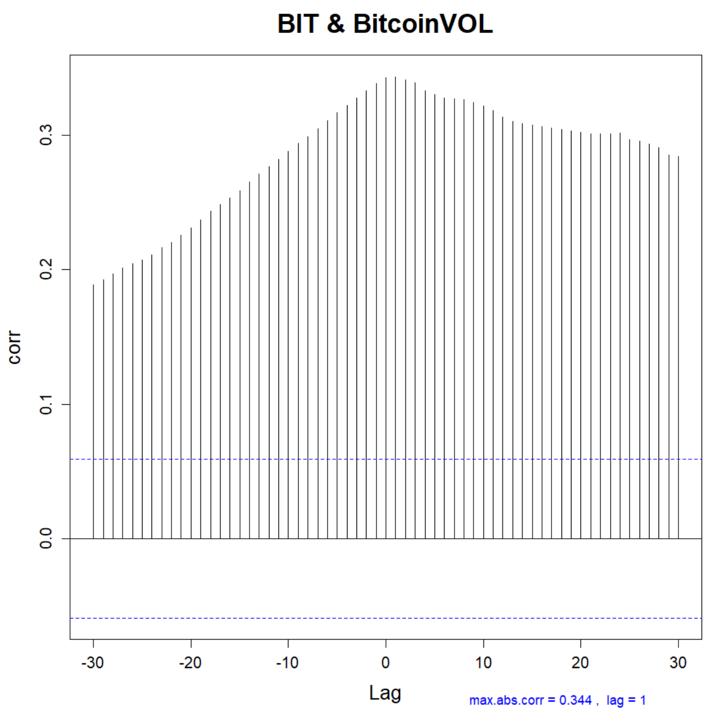


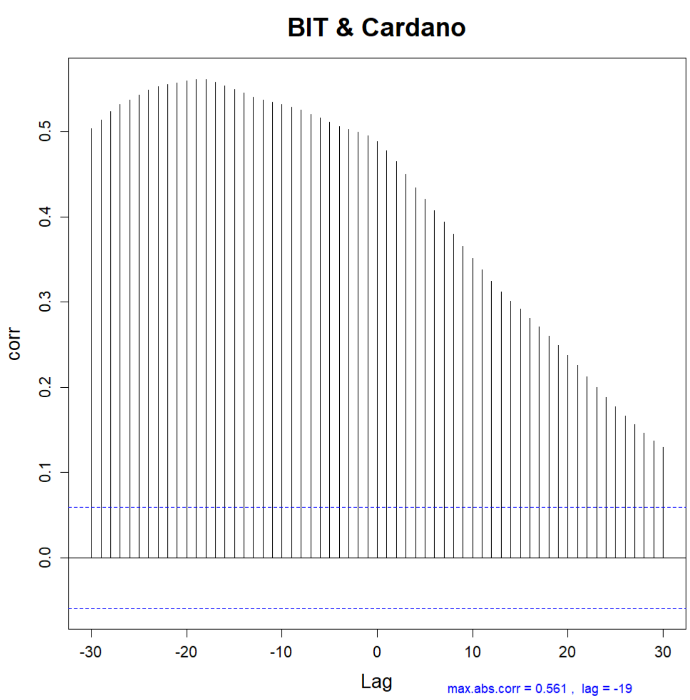


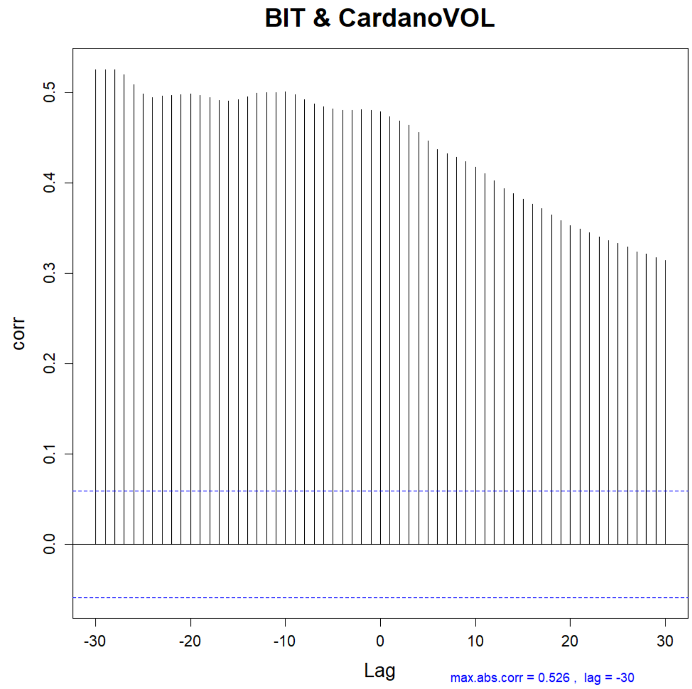
BIT & BitcoinCash
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.512 , lag = -6

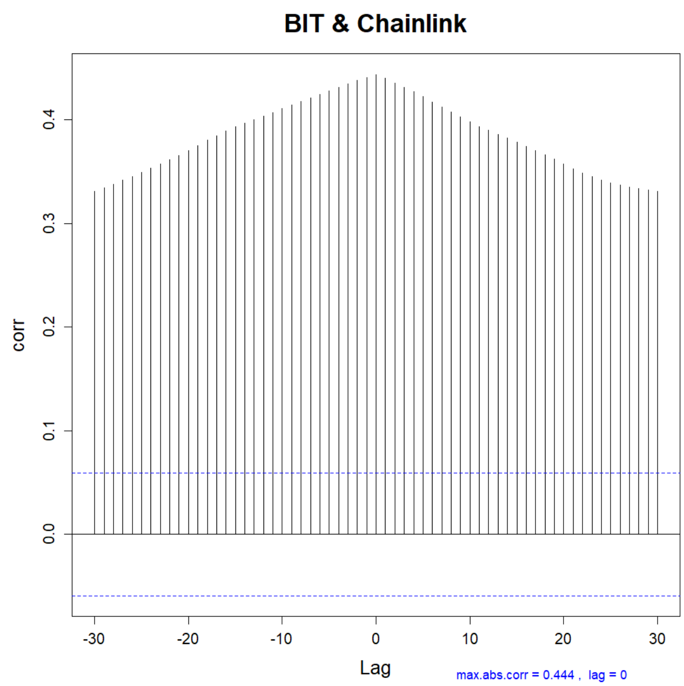


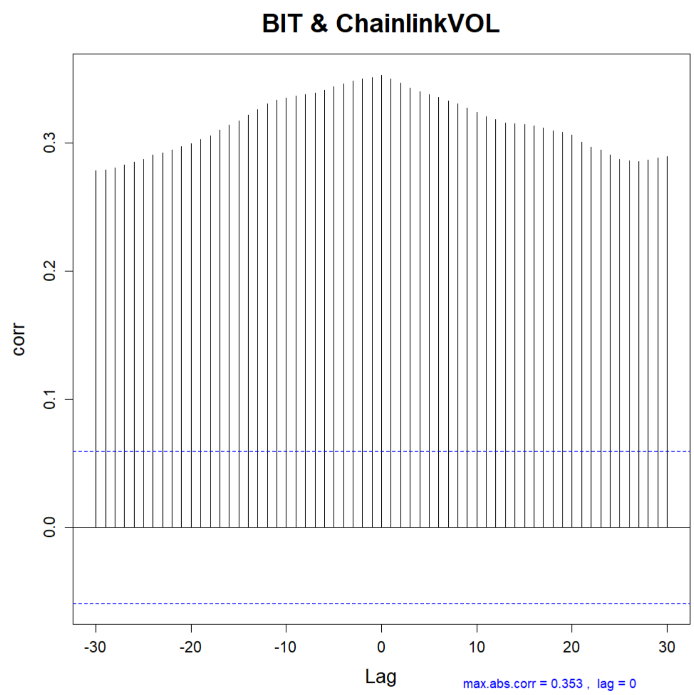
BIT & BitcoinCashVOL
corr
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.267 , lag = 4


BIT & BitcoinVOL
corr
0.0
0.1
0.2
0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.344 , lag = 1

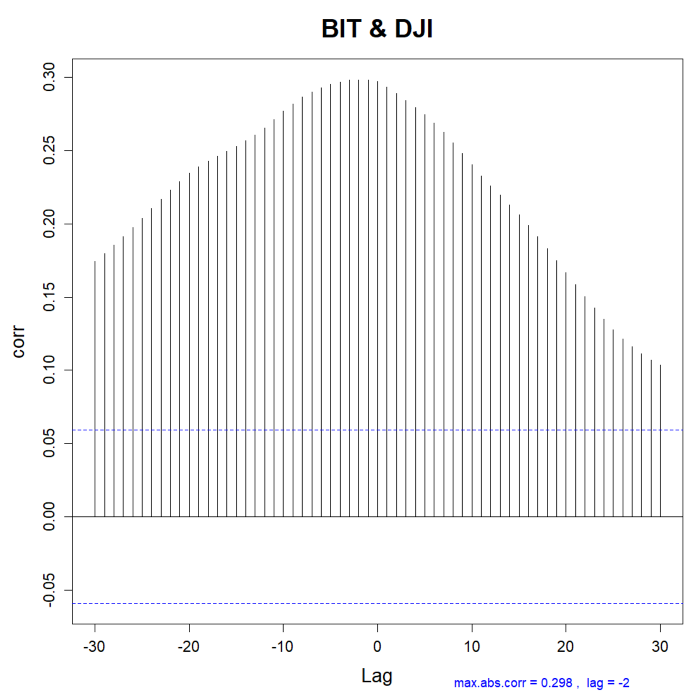

BIT & Cardano
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.561 , lag = -19


BIT & CardanoVOL
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.526 , lag = -30

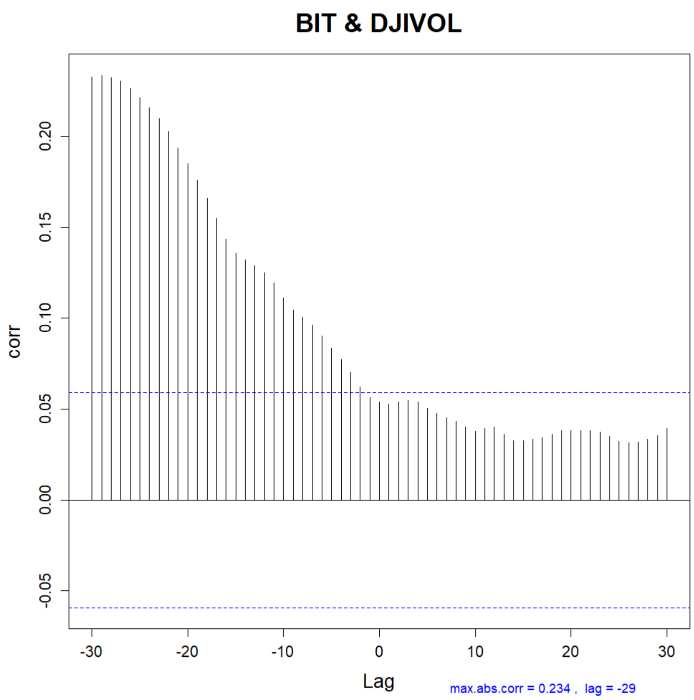

BIT & Chainlink
corr
0.0
0.1
0.2
0.3
0.4
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.444 , lag = 0

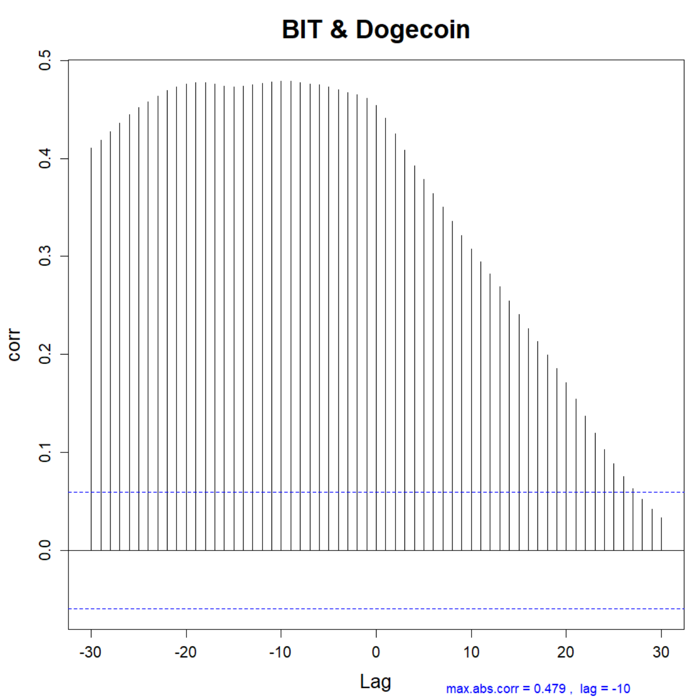


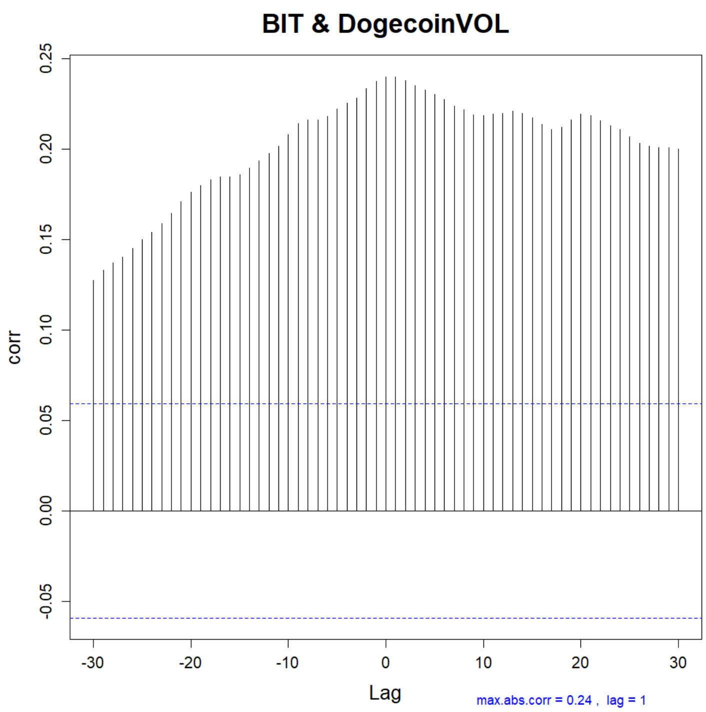
BIT & ChainlinkVOL
corr
0.0
0.1
0.2
0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.353 , lag = 0

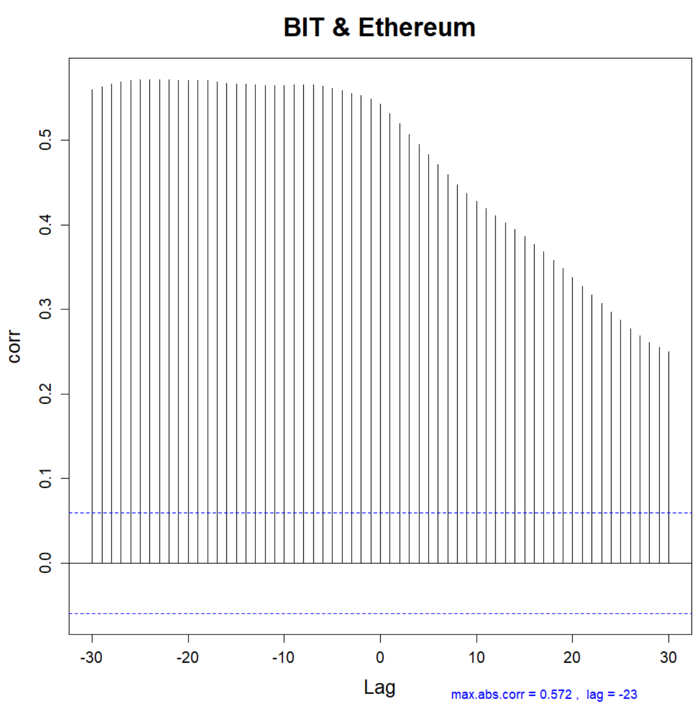


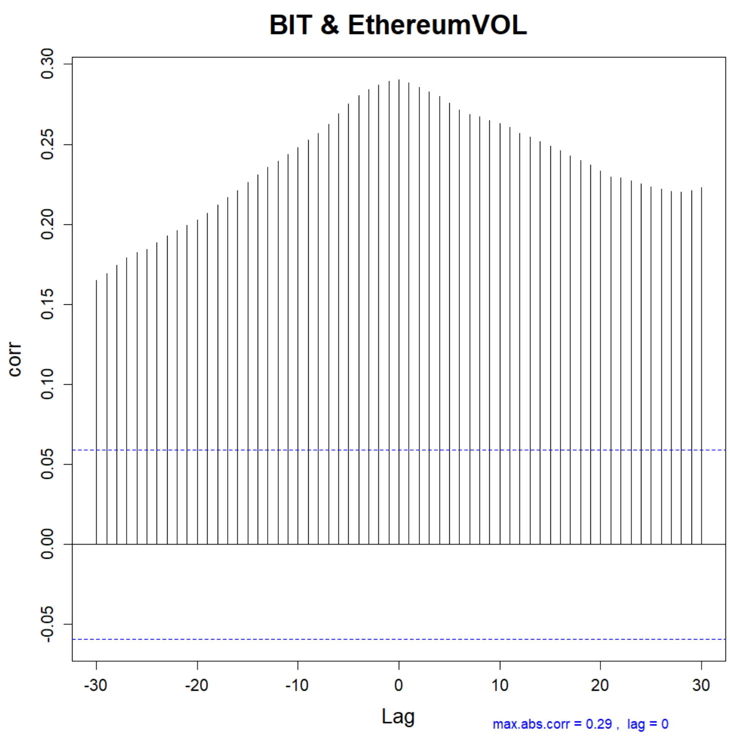
BIT & DJI
corr
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.298 , lag = -2

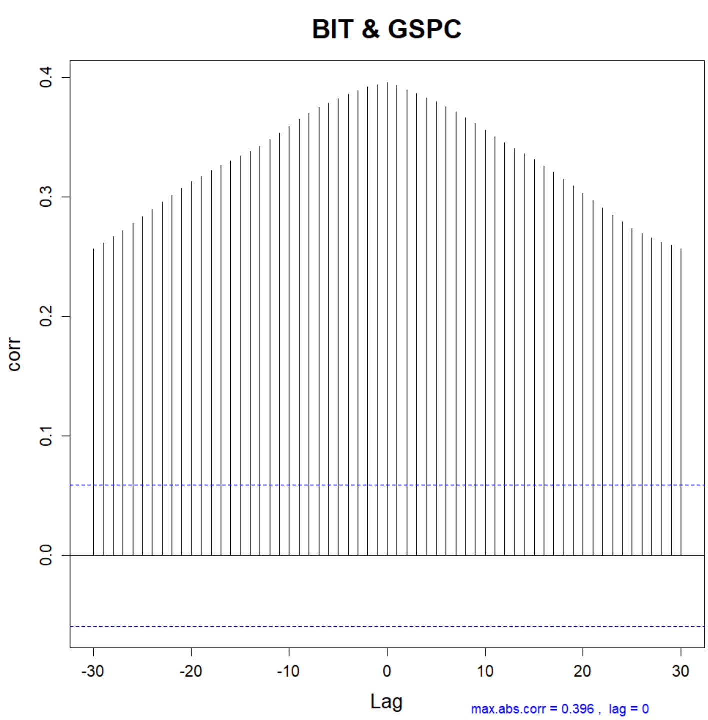

BIT & DJIVOL
corr
0.20
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.234 , lag = -29


BIT & Dogecoin
corr
0.5
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.479 , lag = -10

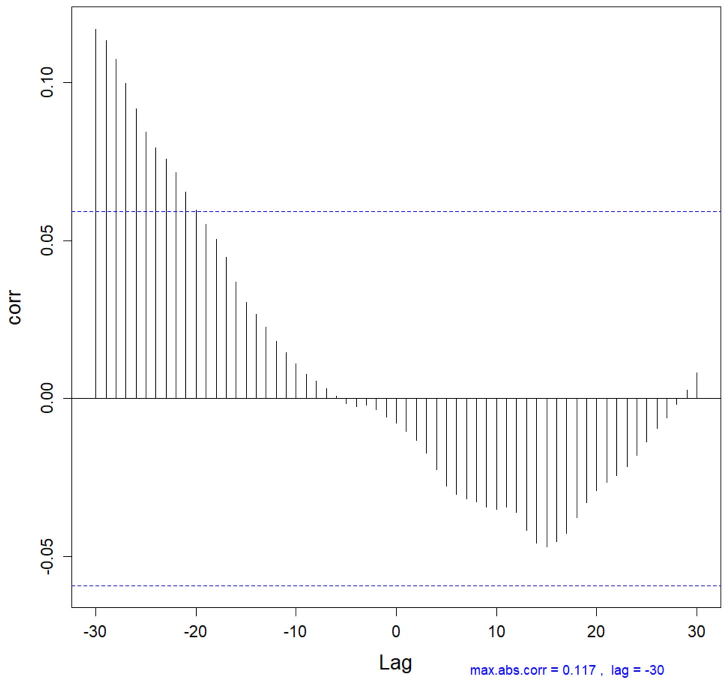

BIT & DogecoinVOL
corr
Lag
max.abs.corr = 0.24 , lag = 1


BIT & Ethereum
corr
Lag
max.abs.corr = 0.572 , lag = -23

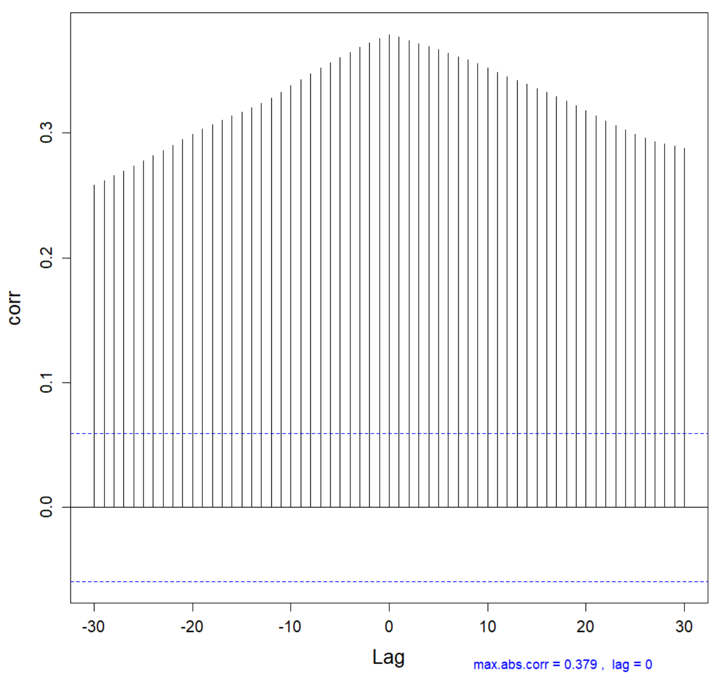

BIT & EthereumVOL
corr
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.29 , lag = 0

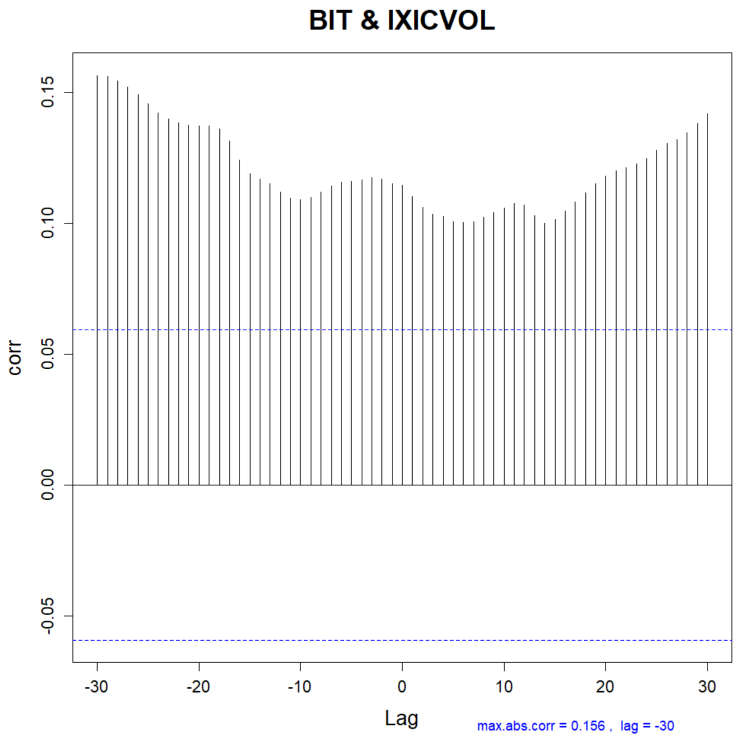

BIT & GSPC
corr
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.396 , lag = 0

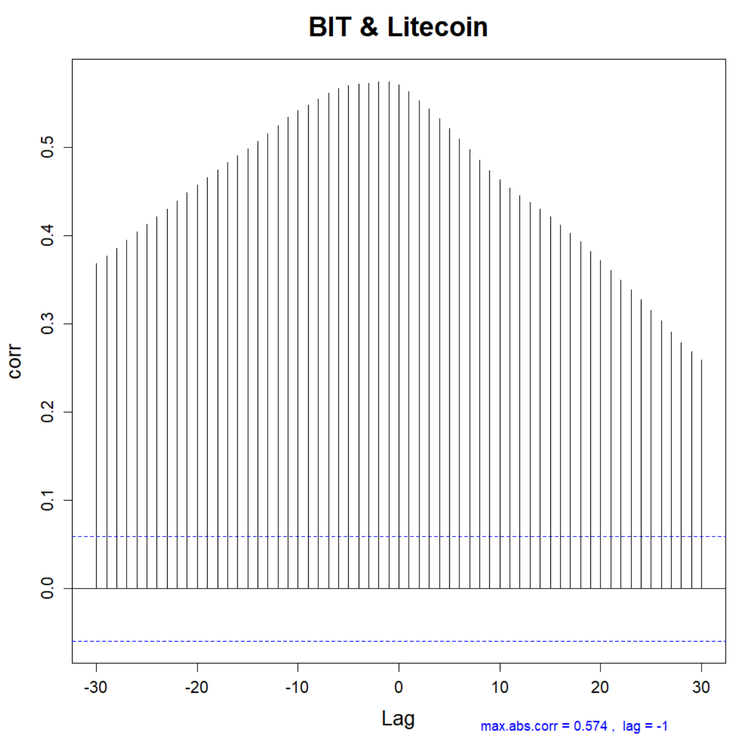
BIT & GSPCVOL

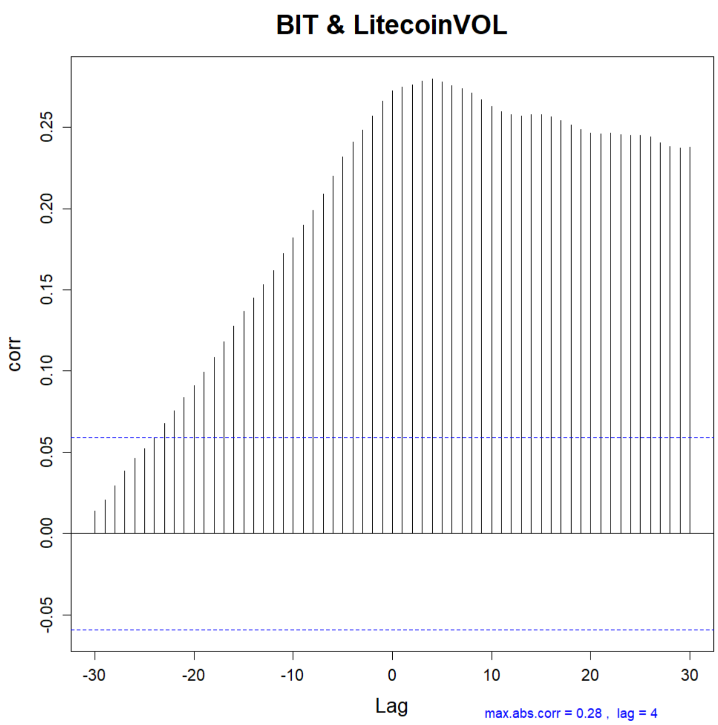


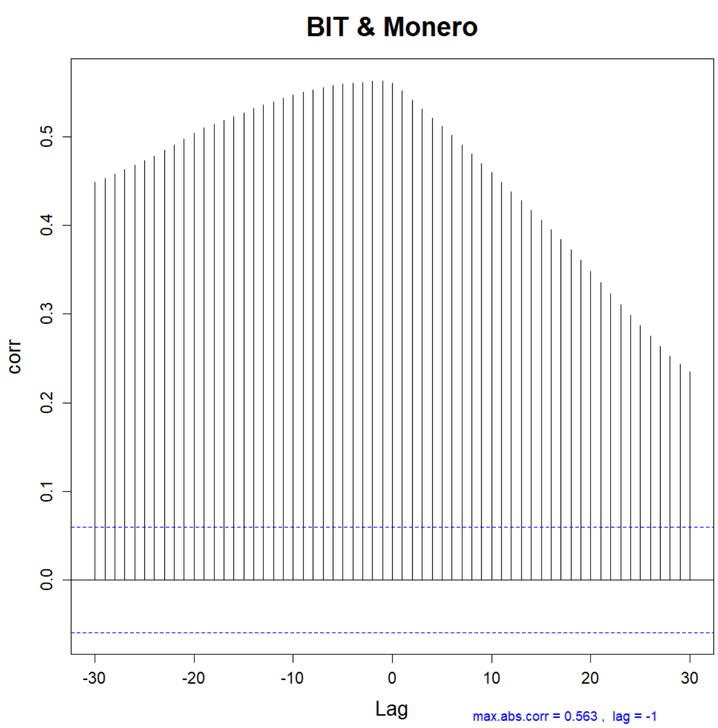
BIT & IXIC

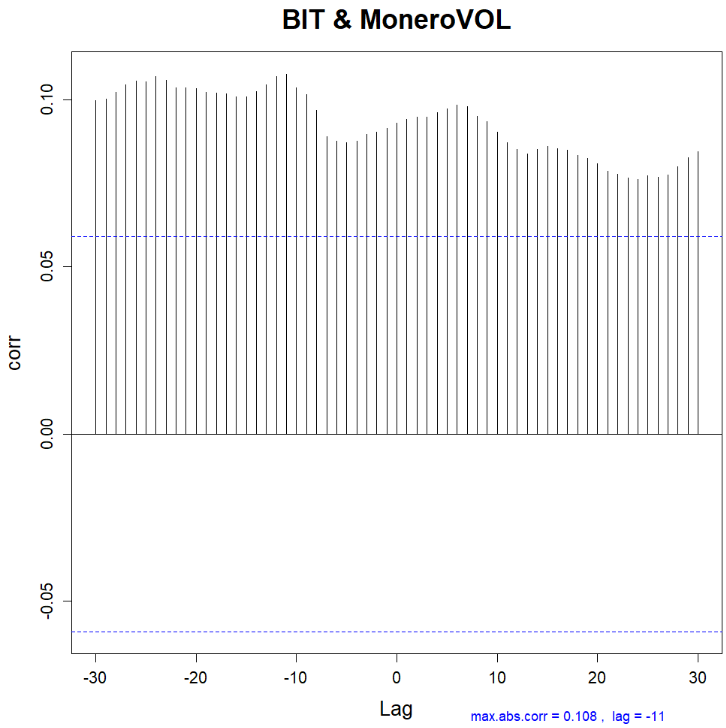


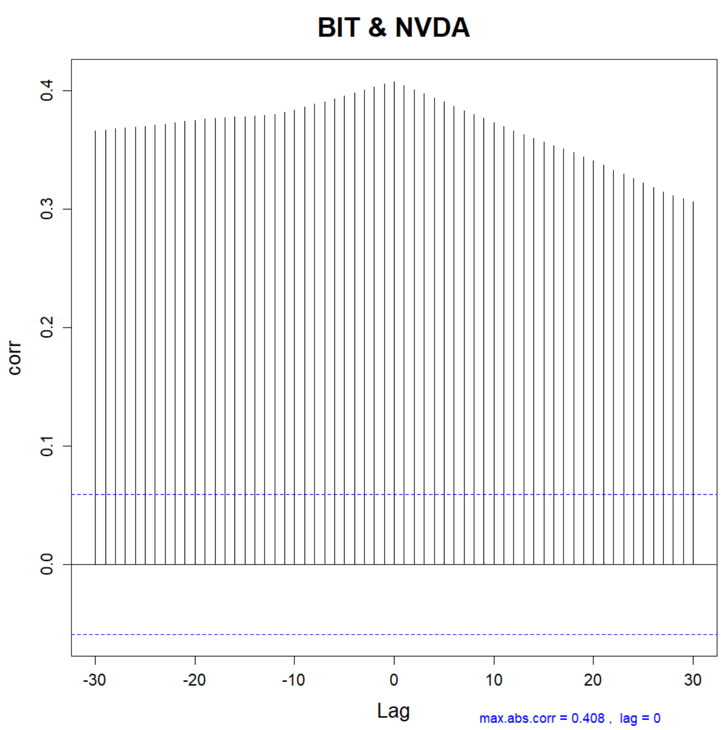

BIT & IXICVOL
corr
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.156 , lag = -30

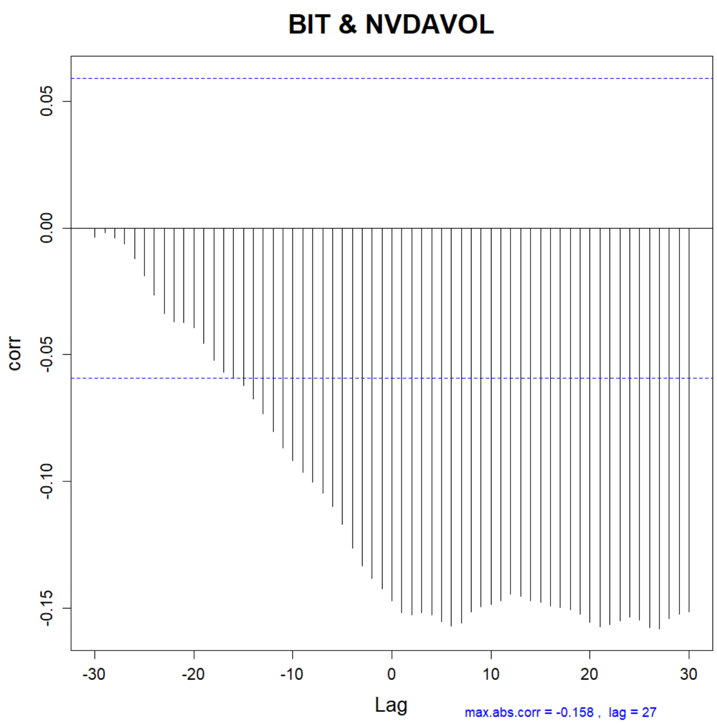

BIT & Litecoin
corr
0.5
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.574 , lag = -1

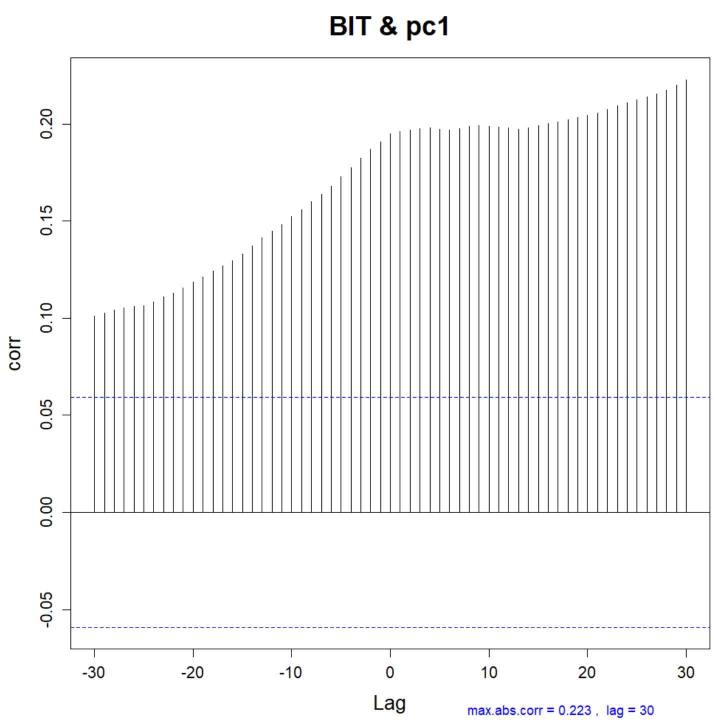

BIT & LitecoinVOL
corr
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.28 , lag = 4

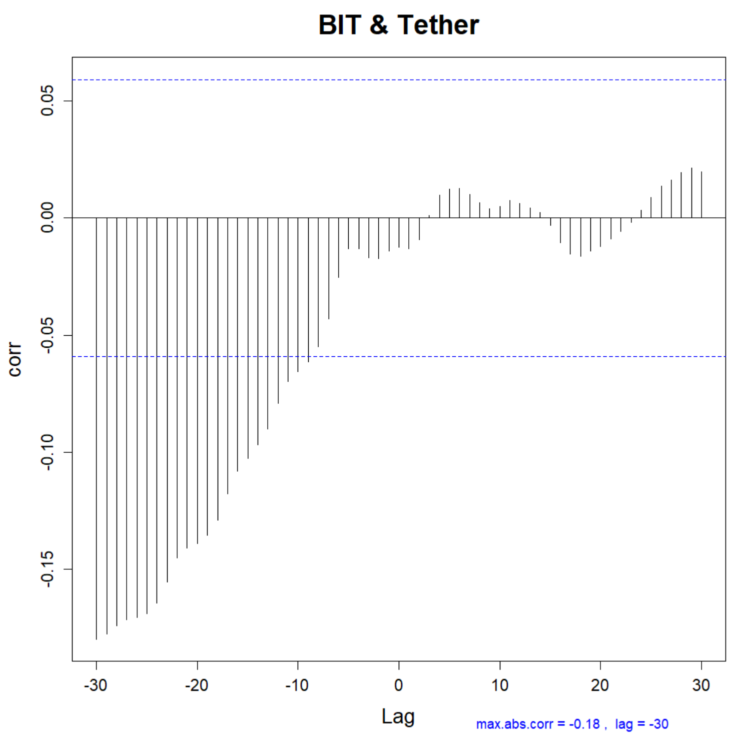

BIT & Monero
corr
0.5
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.563 , lag = -1

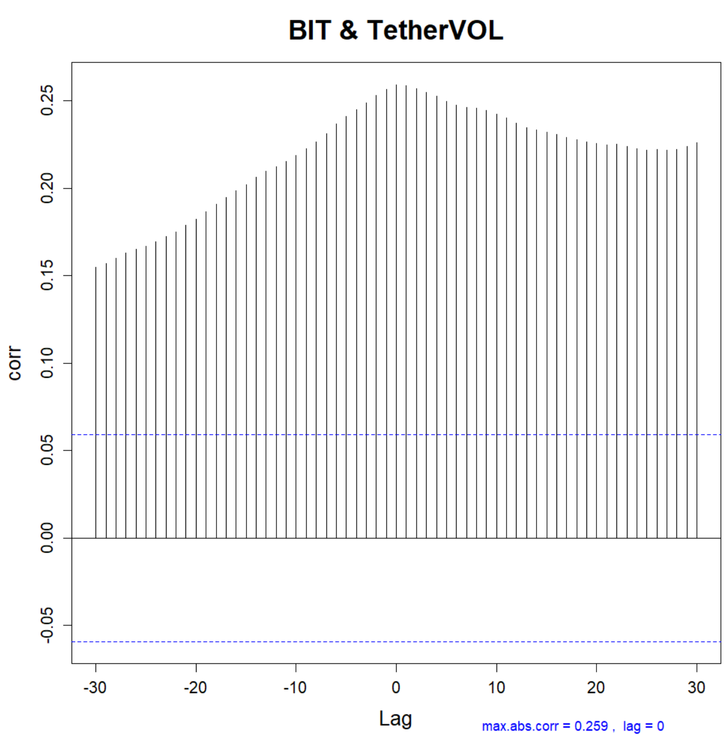

BIT & MoneroVOL
corr
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.108 , lag = -11

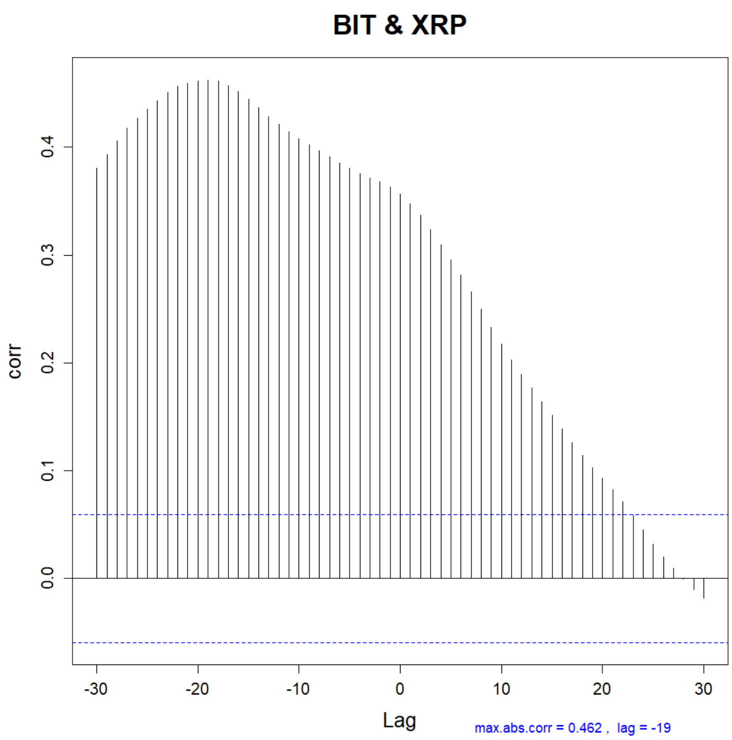

BIT & NVDA
corr
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.408 , lag = 0

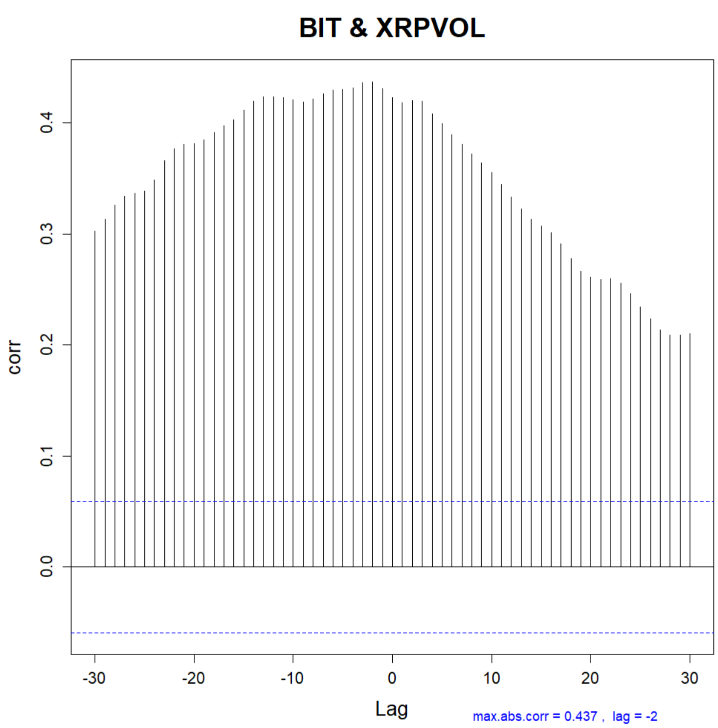

BIT & NVDAVOL
corr
Lag
max.abs.corr = -0.158 , lag = 27

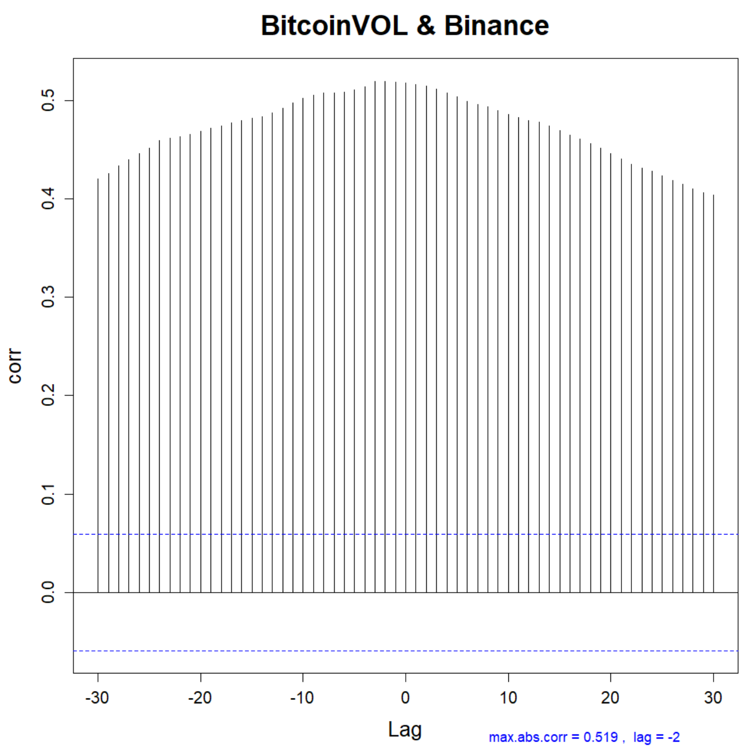

BIT & pc1
corr
Lag
max.abs.corr = 0.223 , lag = 30

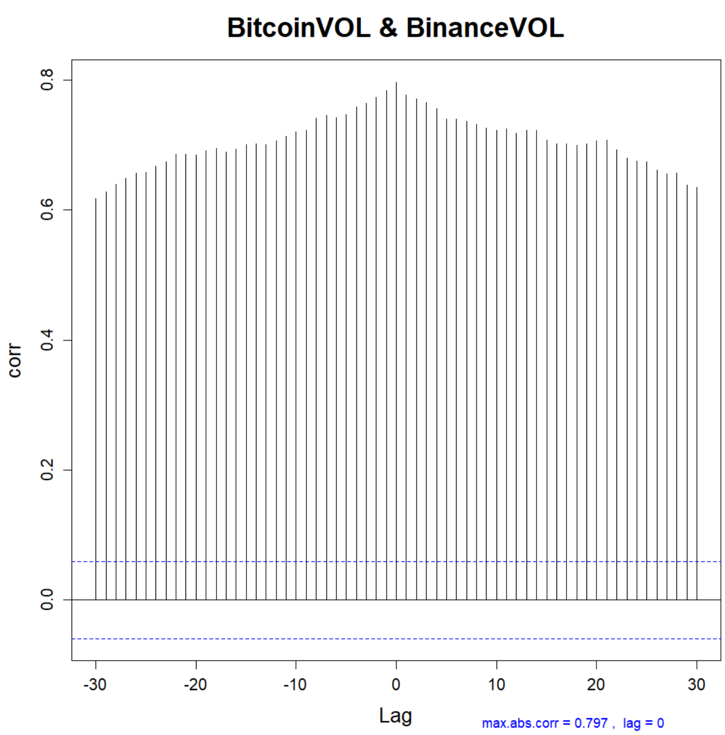

BIT & Tether
corr
0.05
0.00
-0.05
-0.10
-0.15
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.18 , lag = -30

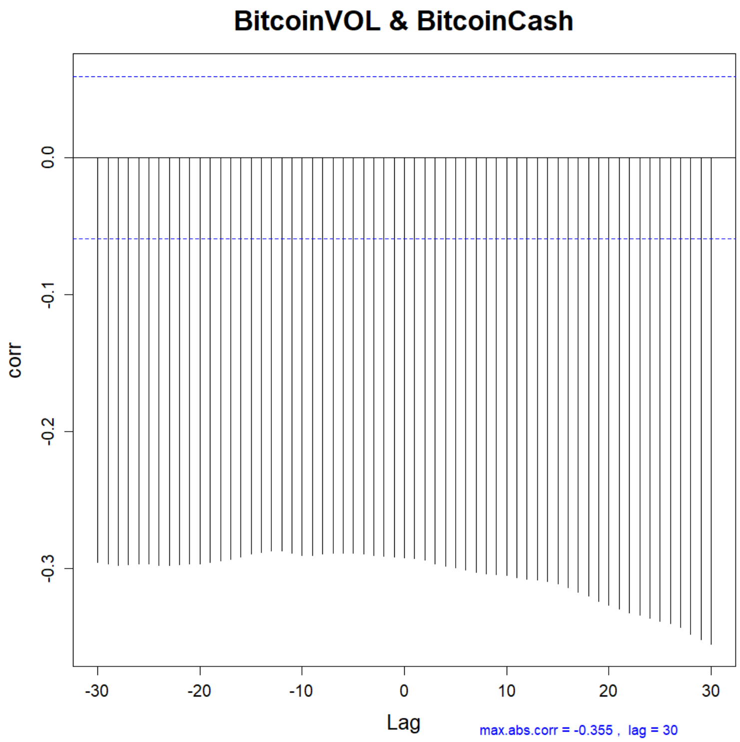

BIT & TetherVOL
corr
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.259 , lag = 0

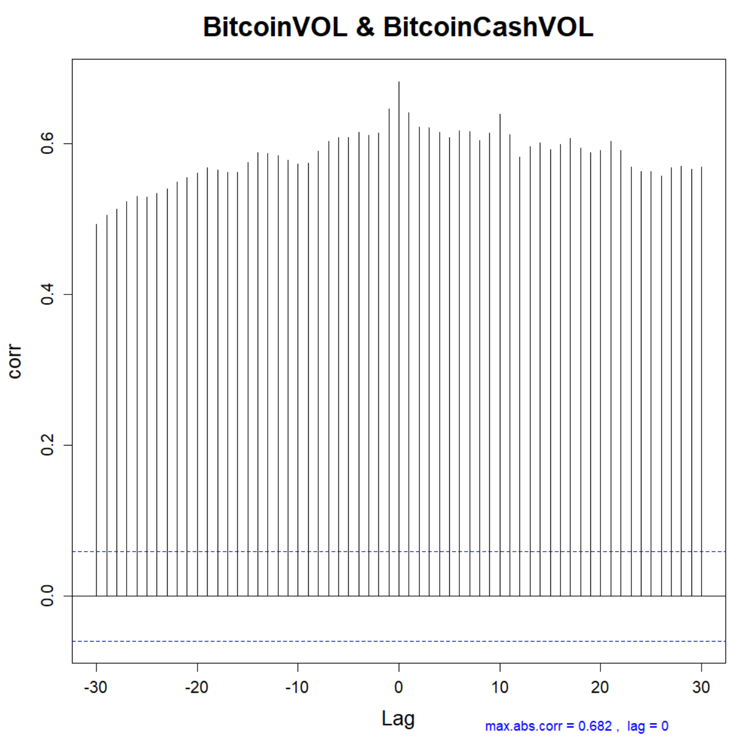

BIT & XRP
corr
0.0
0.1
0.2
0.3
0.4
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.462 , lag = -19

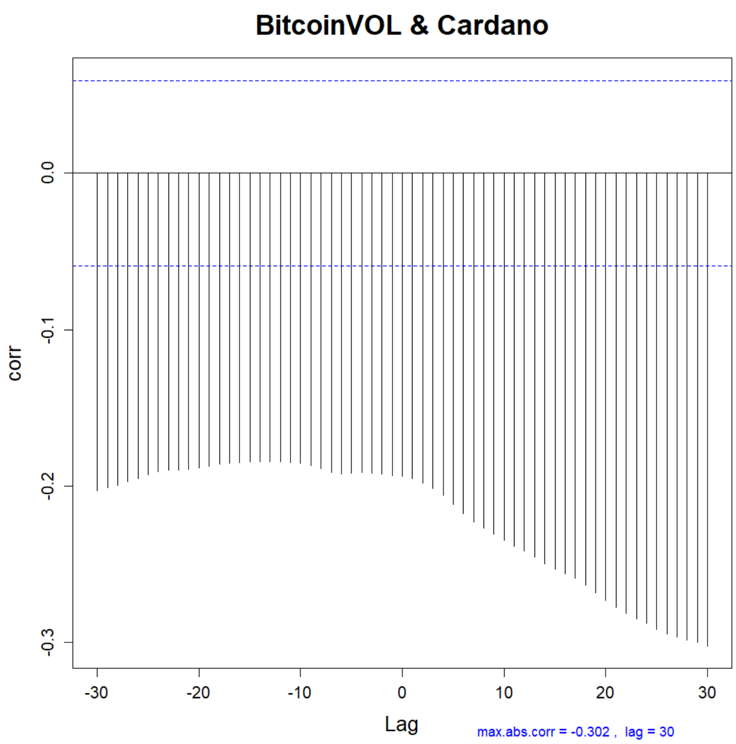

BIT & XRPVOL
corr
0.0
0.1
0.2
0.3
0.4
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.437 , lag = -2

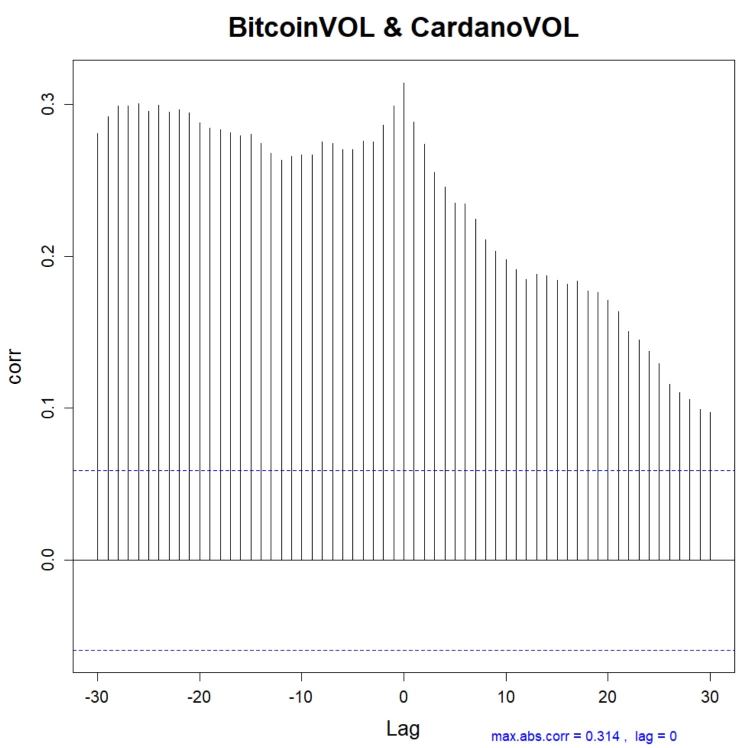

BitcoinVOL & Binance
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.519 , lag = -2

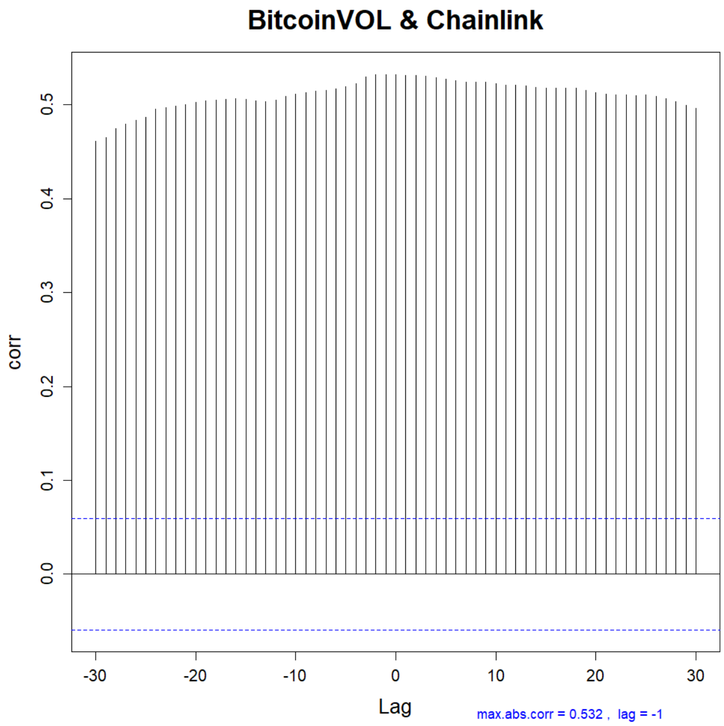

BitcoinVOL & BinanceVOL
corr
0.0
0.2
0.4
0.6
0.8
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.797 , lag = 0

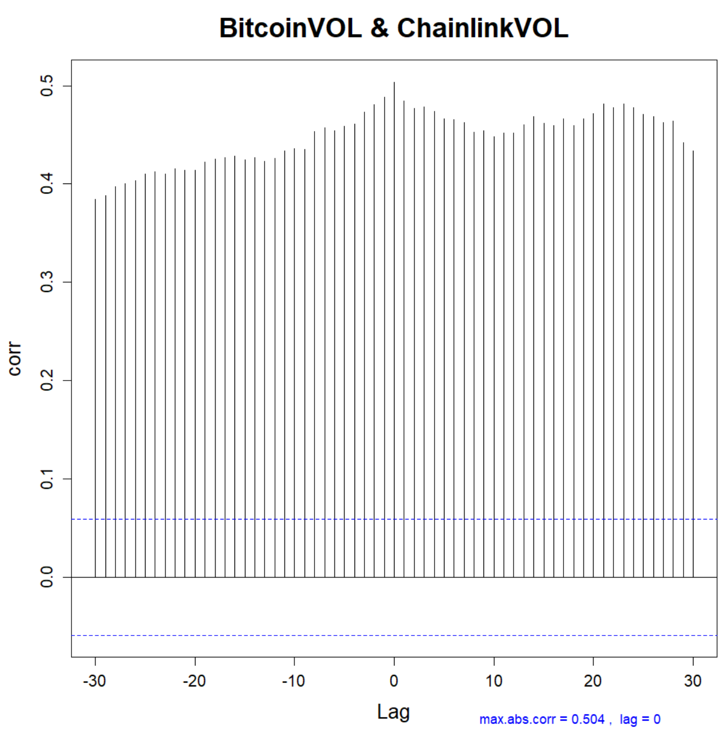

BitcoinVOL & BitcoinCash
corr
0.0
-0.1
-0.2
-0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.355 , lag = 30

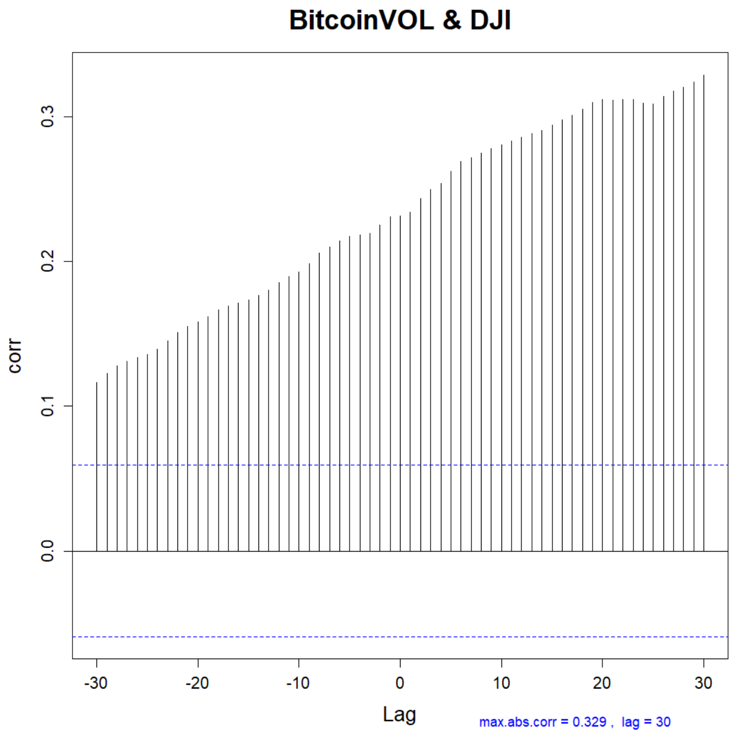

BitcoinVOL & BitcoinCashVOL
corr
0.6
0.4
0.2
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.682 , lag = 0

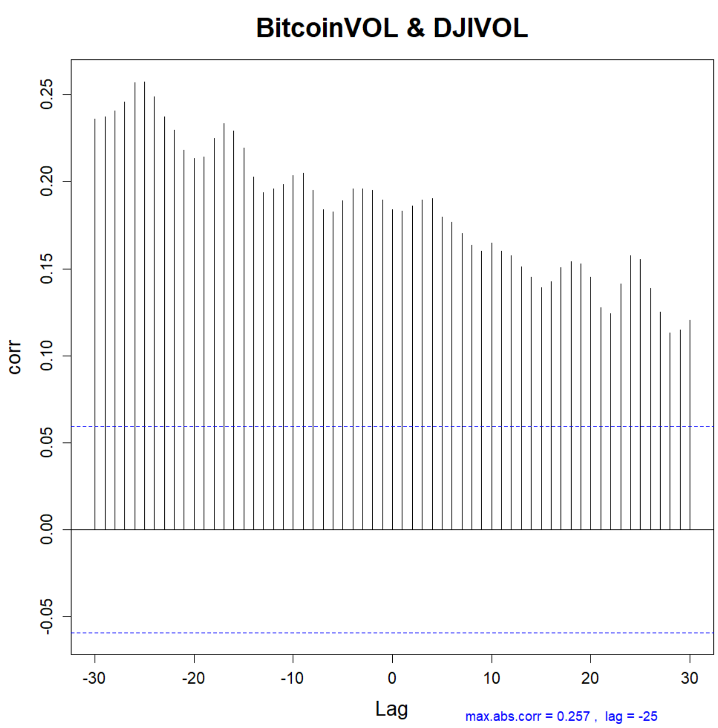

BitcoinVOL & Cardano
corr
0.0
-0.1
-0.2
-0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.302 , lag = 30

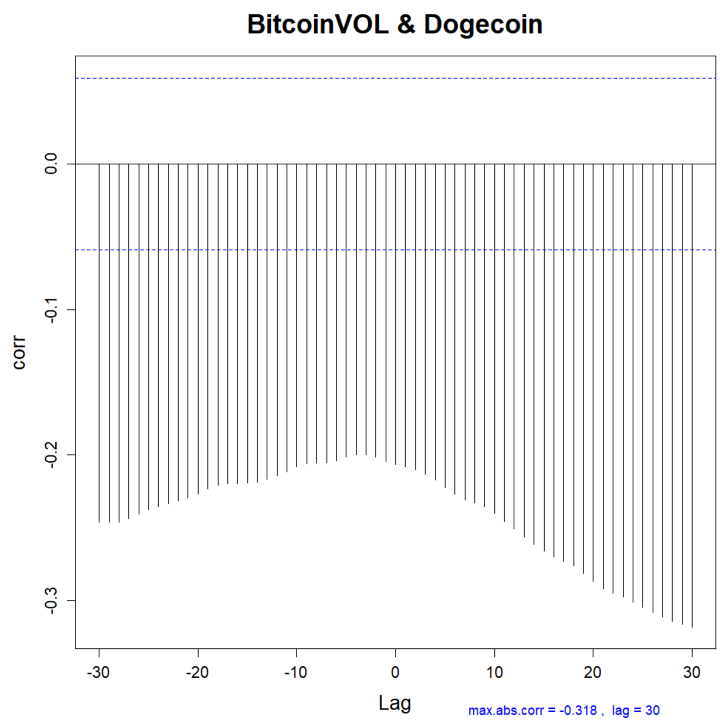

BitcoinVOL & CardanoVOL
corr
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.314 , lag = 0

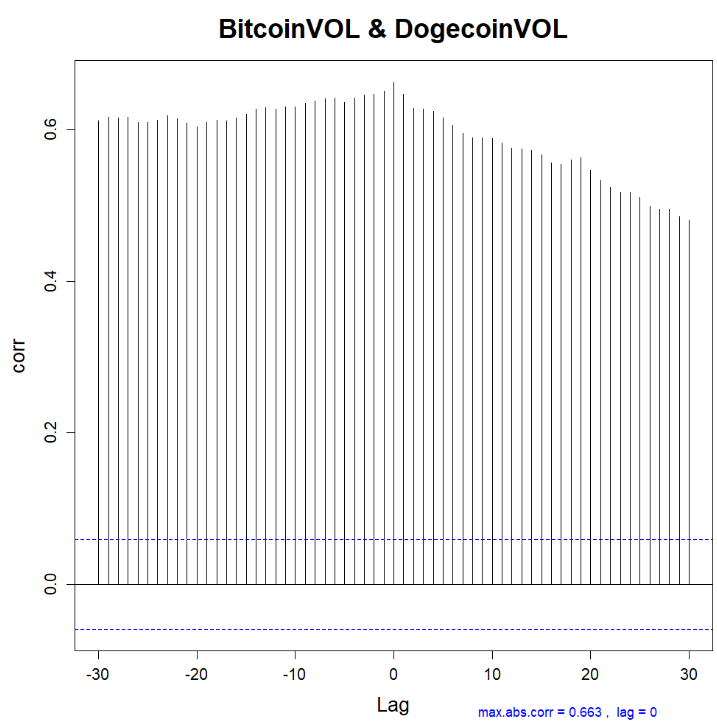

BitcoinVOL & Chainlink
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.532 , lag = -1

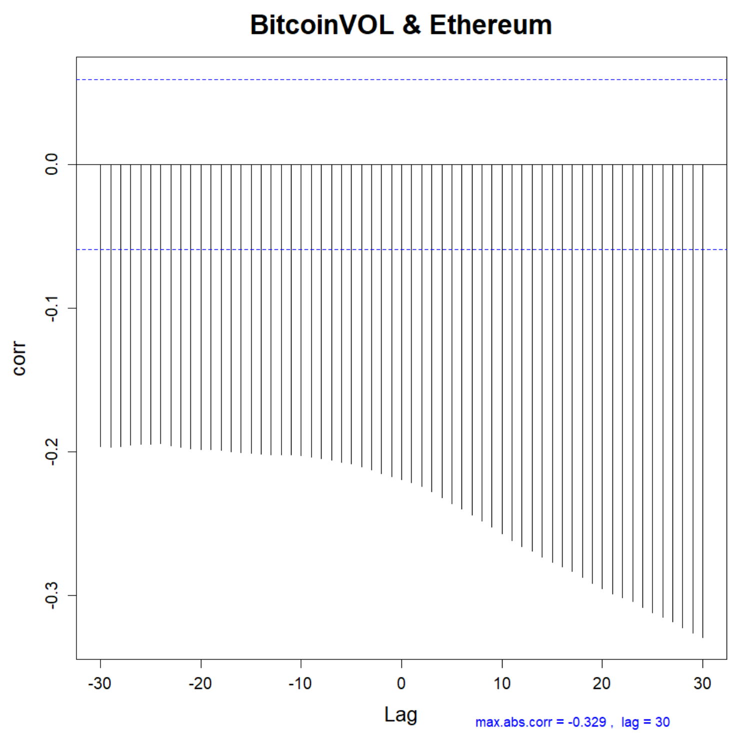

BitcoinVOL & ChainlinkVOL
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.504 , lag = 0

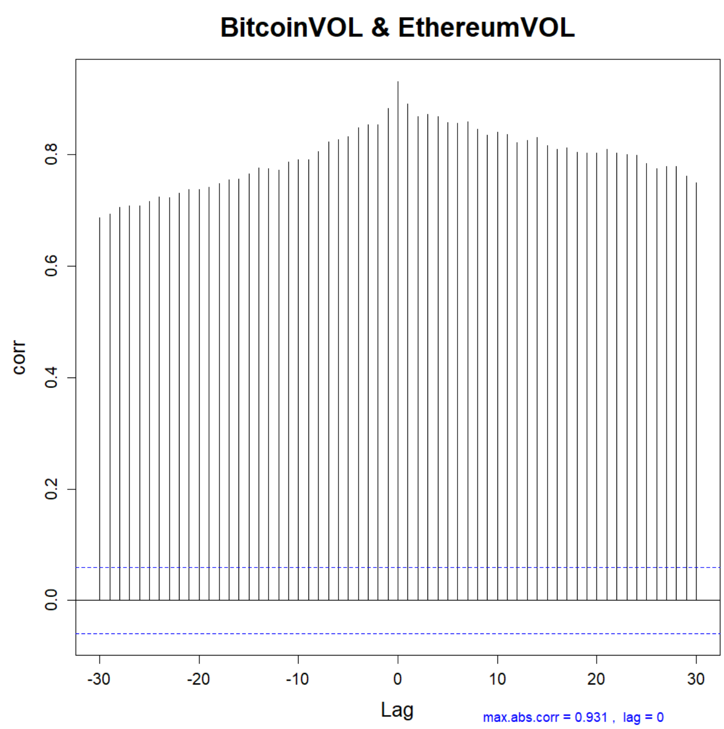

BitcoinVOL & DJI
corr
0.0
0.1
0.2
0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.329 , lag = 30

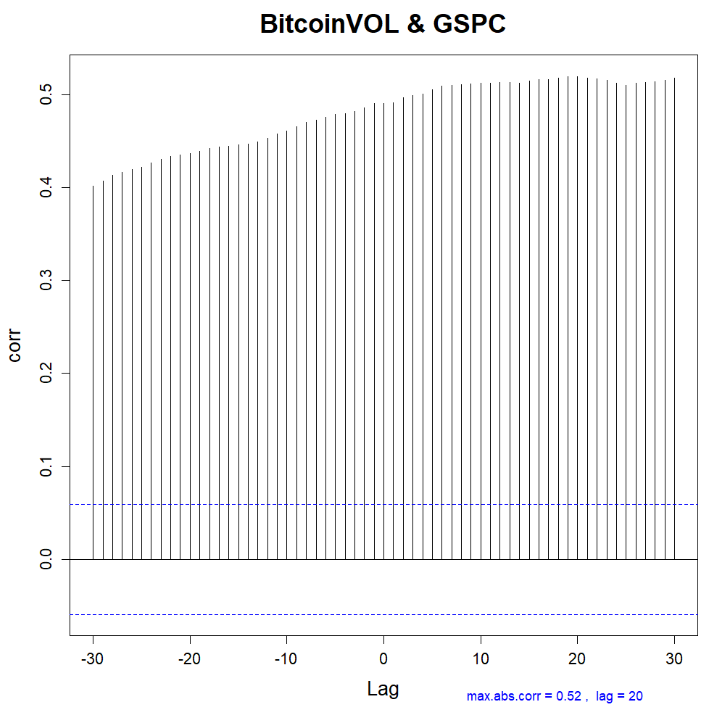

BitcoinVOL & DJIVOL
corr
-0.05
0.00
0.05
0.10
0.15
0.20
0.25
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.257 , lag = -25

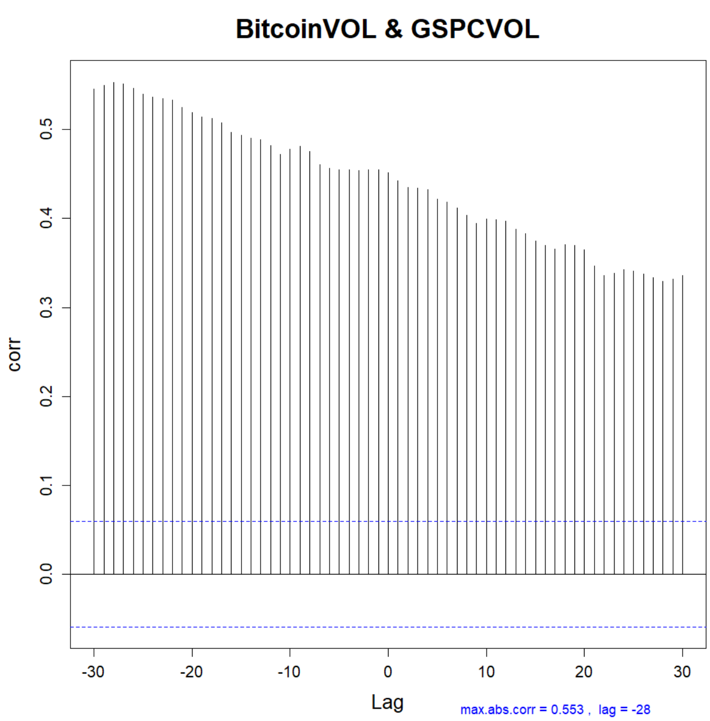

BitcoinVOL & Dogecoin
corr
0.0
-0.1
-0.2
-0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.318 , lag = 30

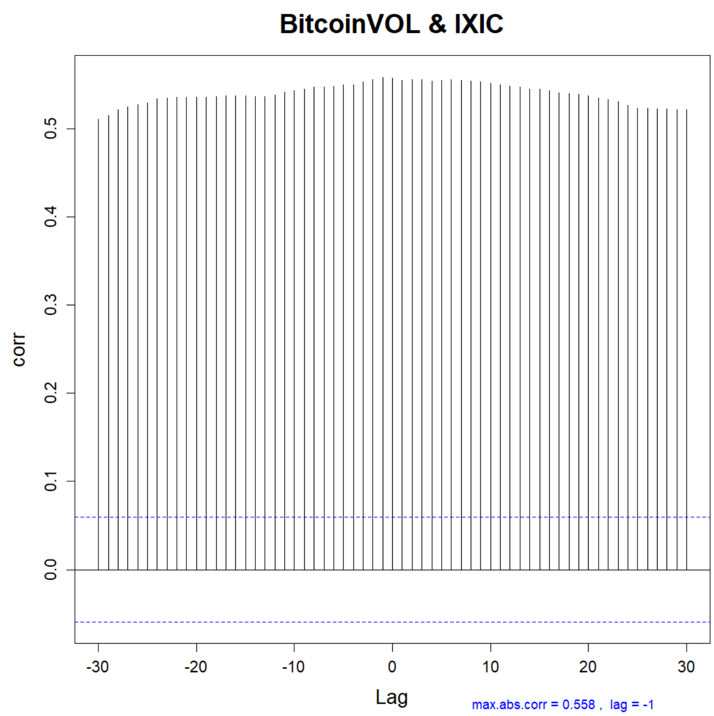

BitcoinVOL & DogecoinVOL
corr
0.6
0.4
0.2
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.663 , lag = 0

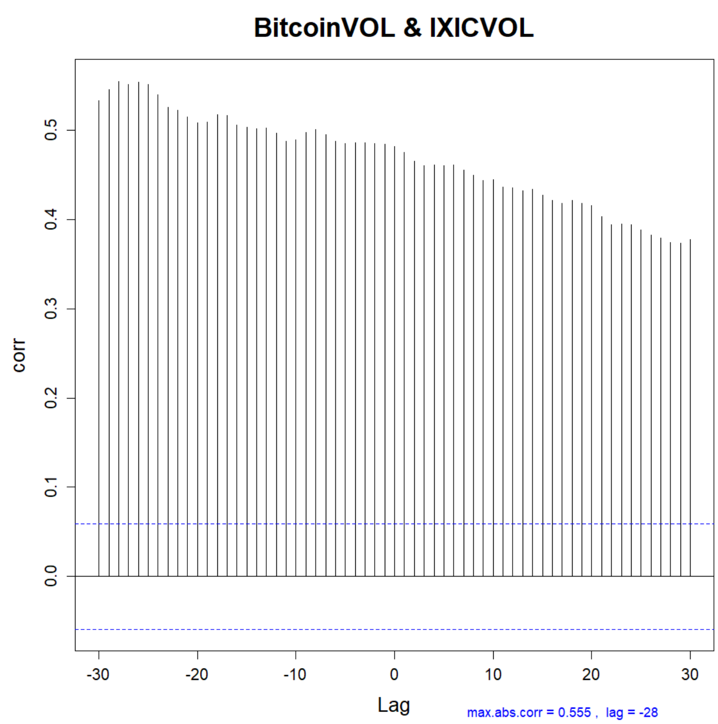

BitcoinVOL & Ethereum
corr
0.0
-0.1
-0.2
-0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.329 , lag = 30

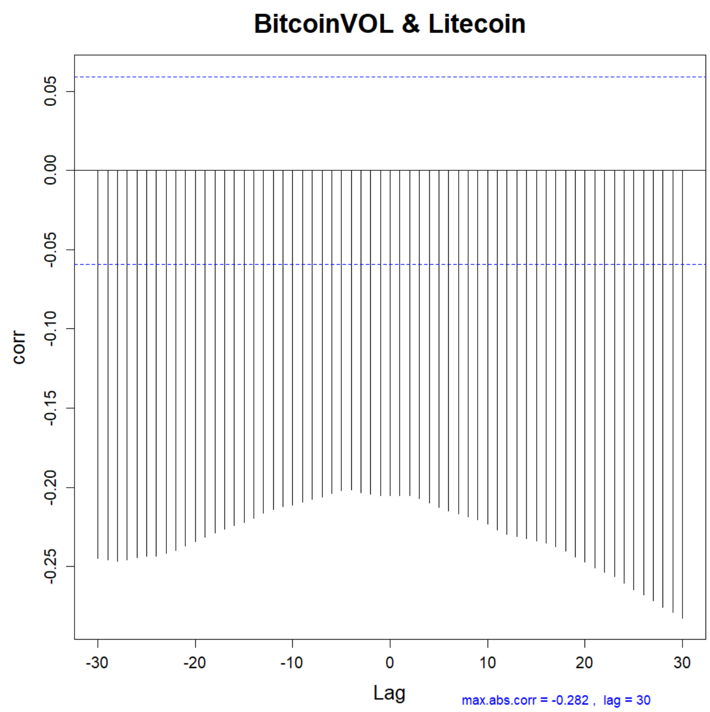

BitcoinVOL & EthereumVOL
corr
0.8
0.6
0.4
0.2
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.931 , lag = 0

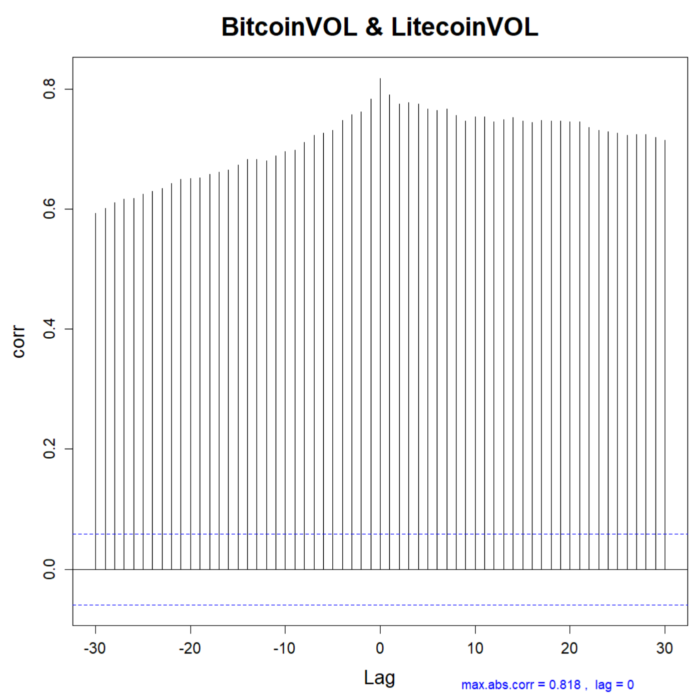

BitcoinVOL & GSPC
corr
0.5
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.52 , lag = 20

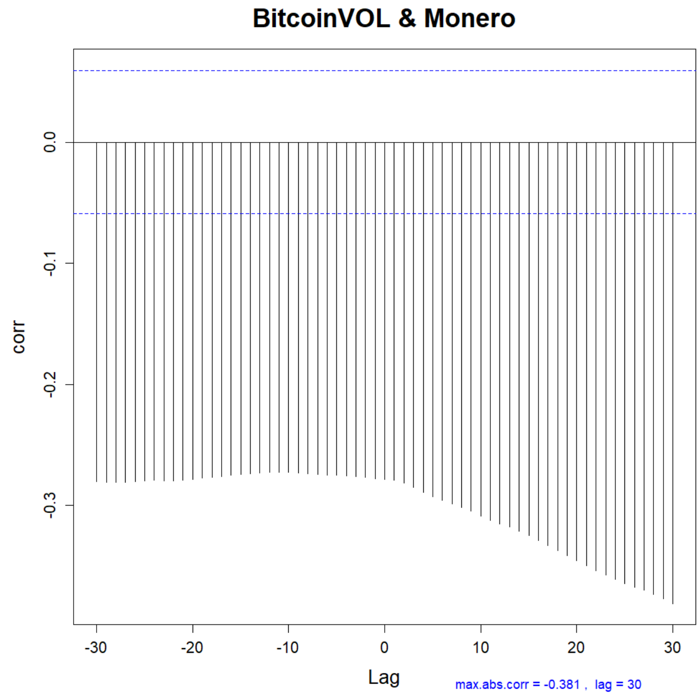

BitcoinVOL & GSPCVOL
corr
0.5
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.553 , lag = -28


BitcoinVOL & IXIC
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.558 , lag = -1

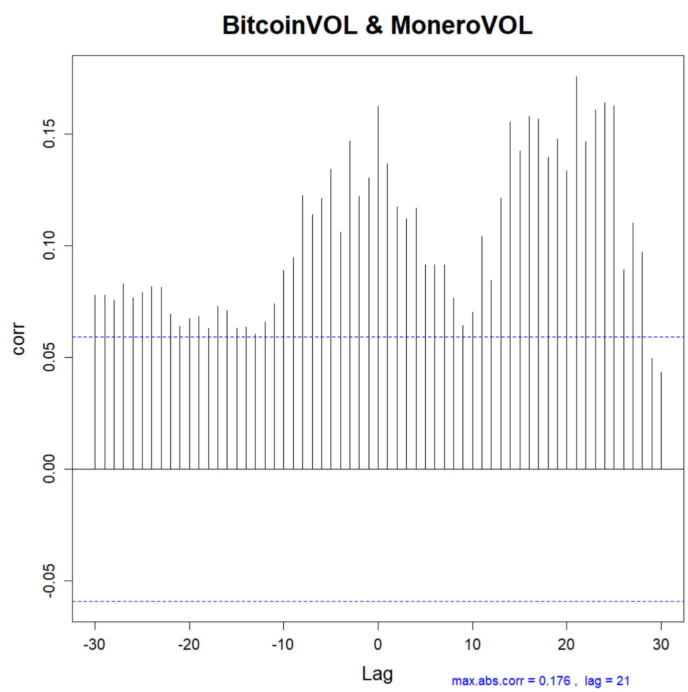

BitcoinVOL & IXICVOL
corr
0.0
0.1
0.2
0.3
0.4
0.5
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.555 , lag = -28


BitcoinVOL & Litecoin
corr
0.05
0.00
-0.05
-0.10
-0.15
-0.20
-0.25
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.282 , lag = 30

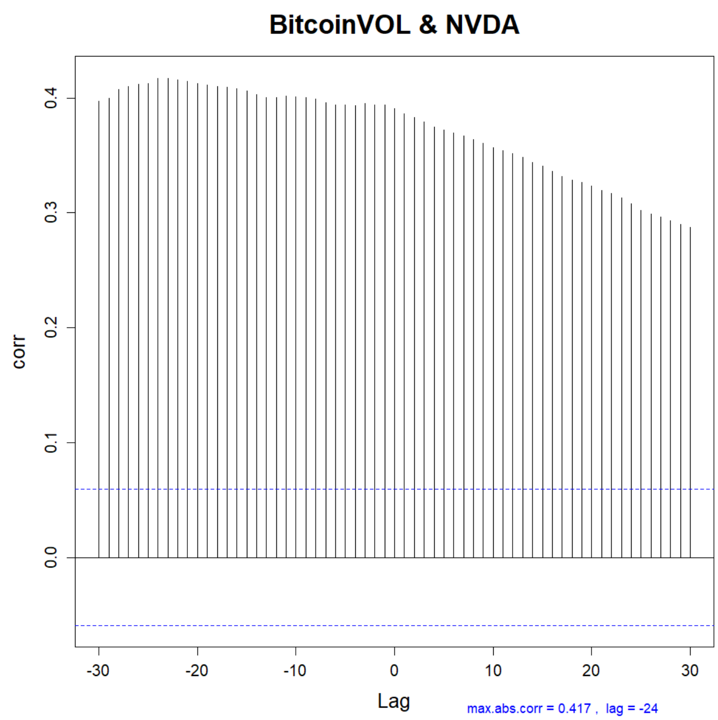

BitcoinVOL & LitecoinVOL
corr
0.8
0.6
0.4
0.2
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.818 , lag = 0

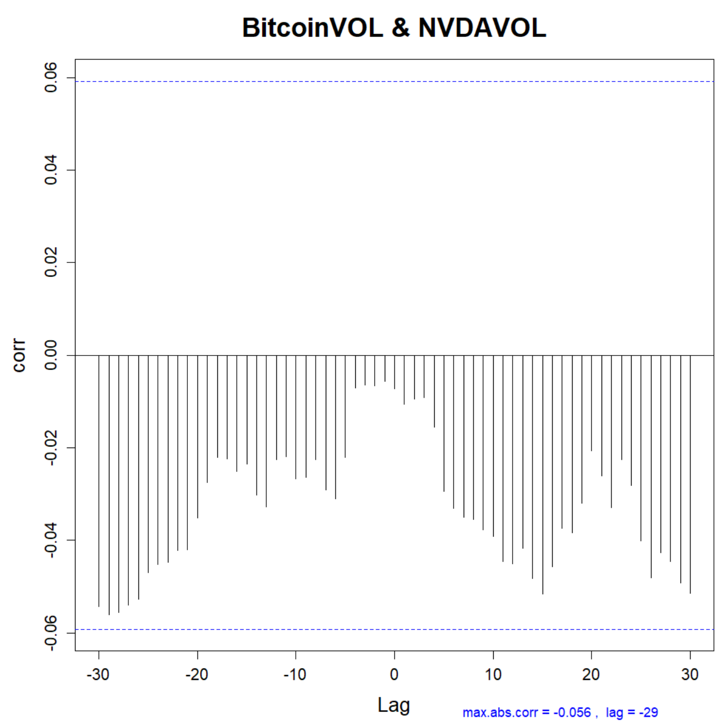


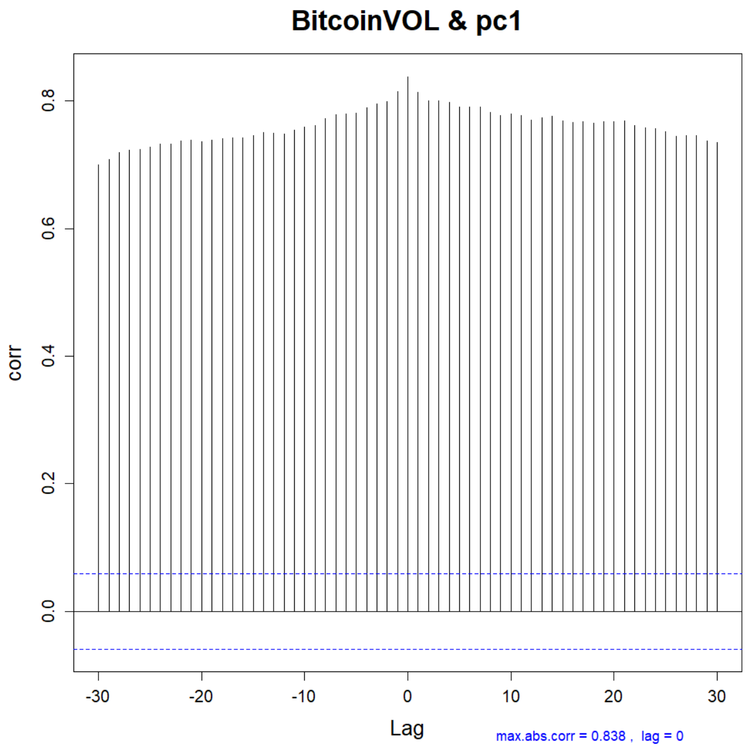
BitcoinVOL & Monero
corr
0.0
-0.1
-0.2
-0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.381 , lag = 30

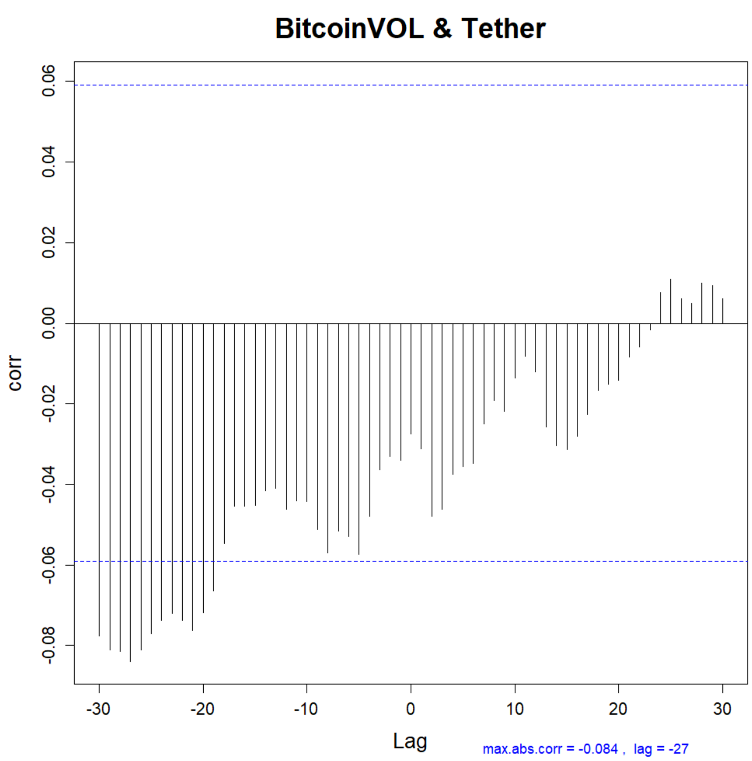


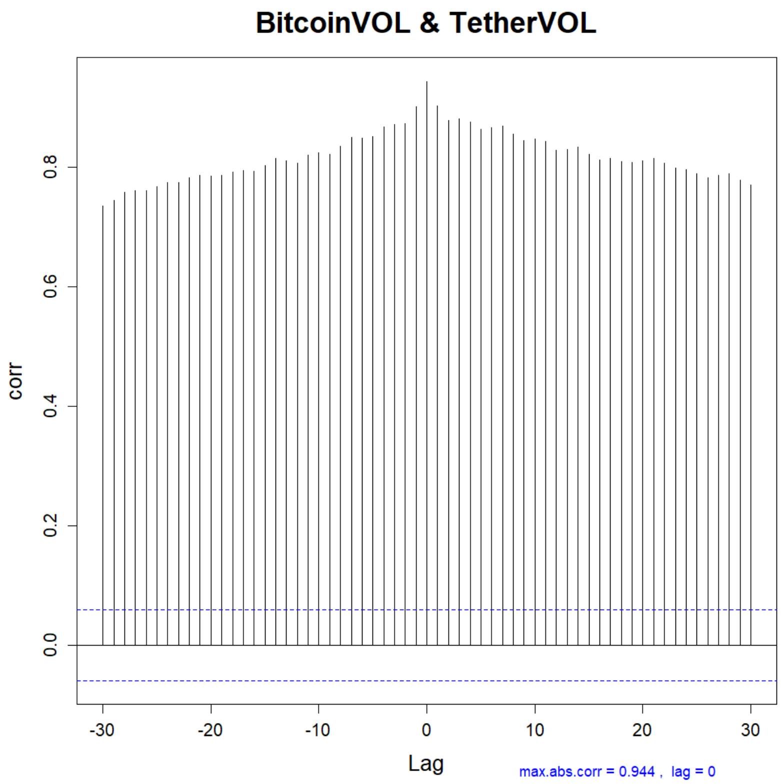
BitcoinVOL & MoneroVOL
corr
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.176 , lag = 21

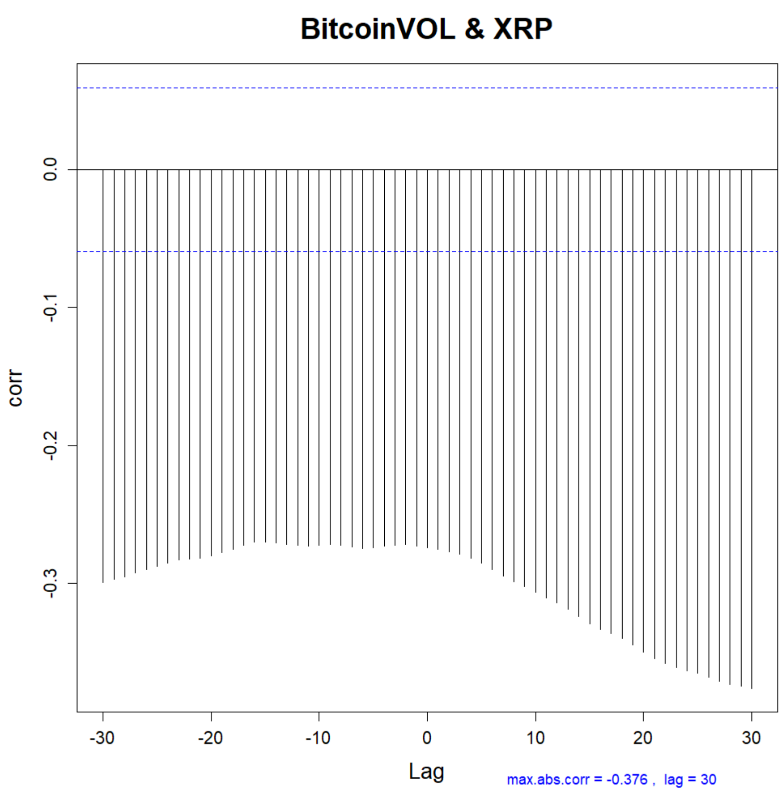

BitcoinVOL & NVDA
corr
0.4
0.3
0.2
0.1
0.0
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.417 , lag = -24

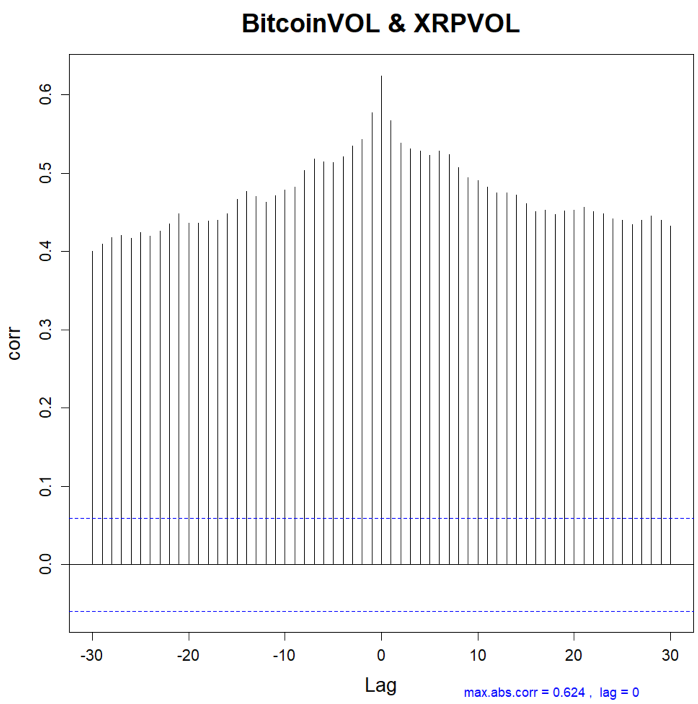

BitcoinVOL & NVDAVOL
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.056 , lag = -29


BitcoinVOL & pc1
corr
Lag
max.abs.corr = 0.838 , lag = 0

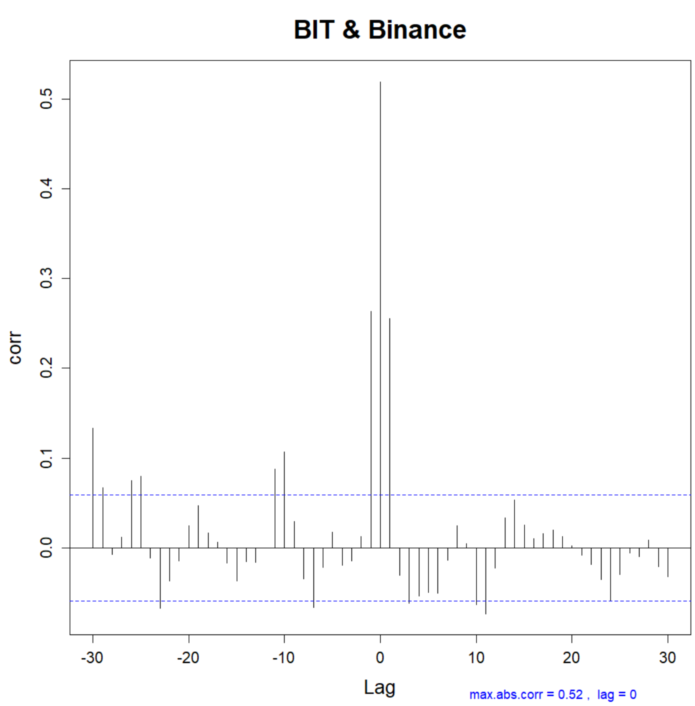

BitcoinVOL & Tether
corr
Lag
max.abs.corr = -0.084 , lag = -27

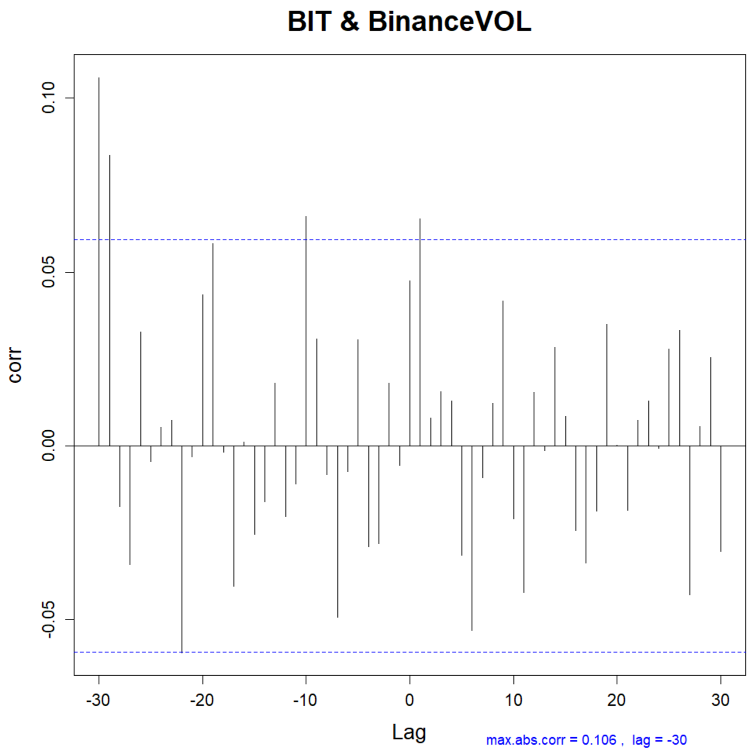

BitcoinVOL & TetherVOL
corr
0.0
0.2
0.4
0.6
0.8
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.944 , lag = 0

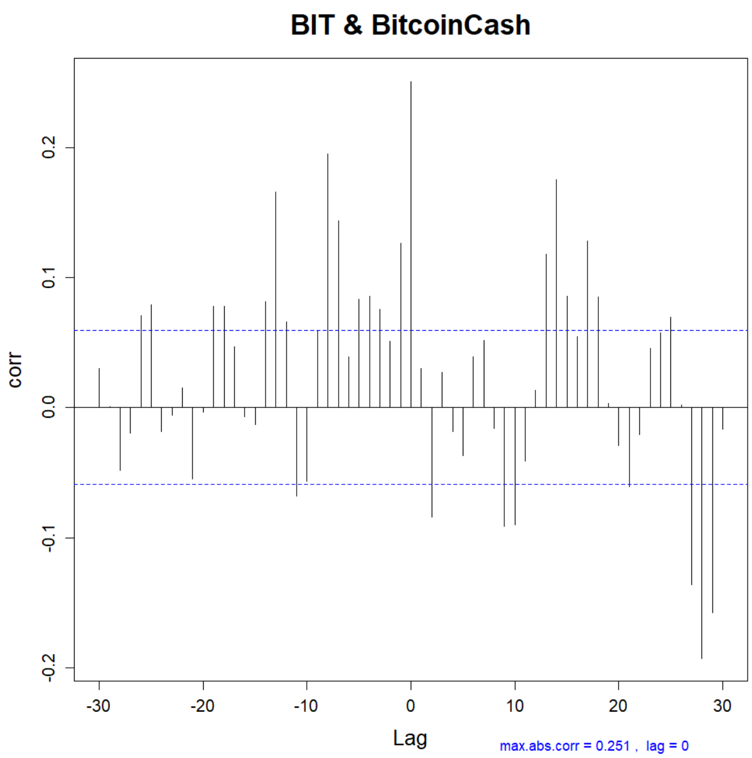

BitcoinVOL & XRP
corr
0.0
-0.1
-0.2
-0.3
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.376 , lag = 30

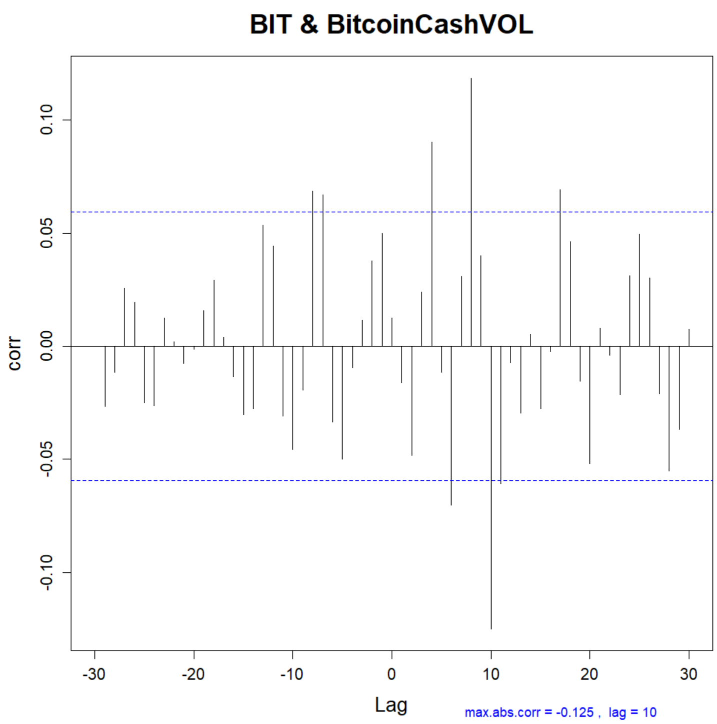


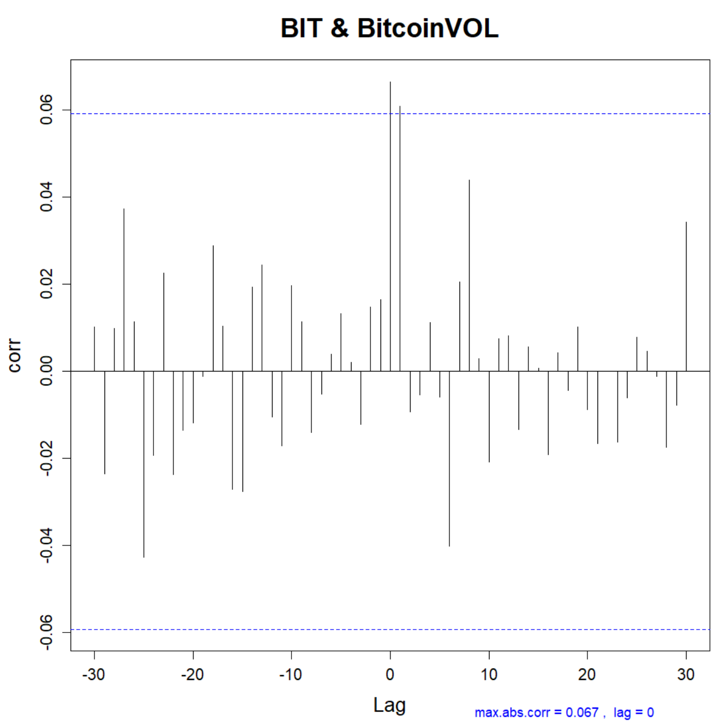
A.2.2 Differenced CCF

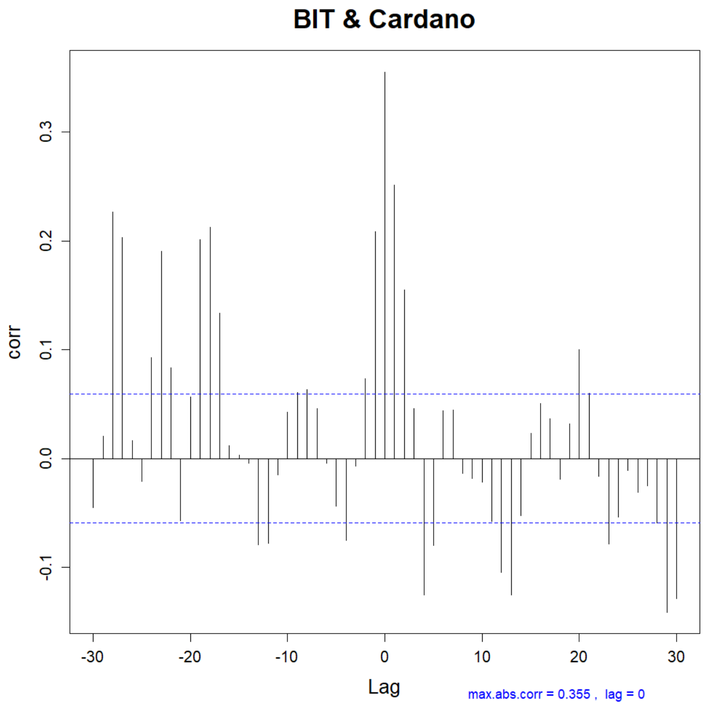


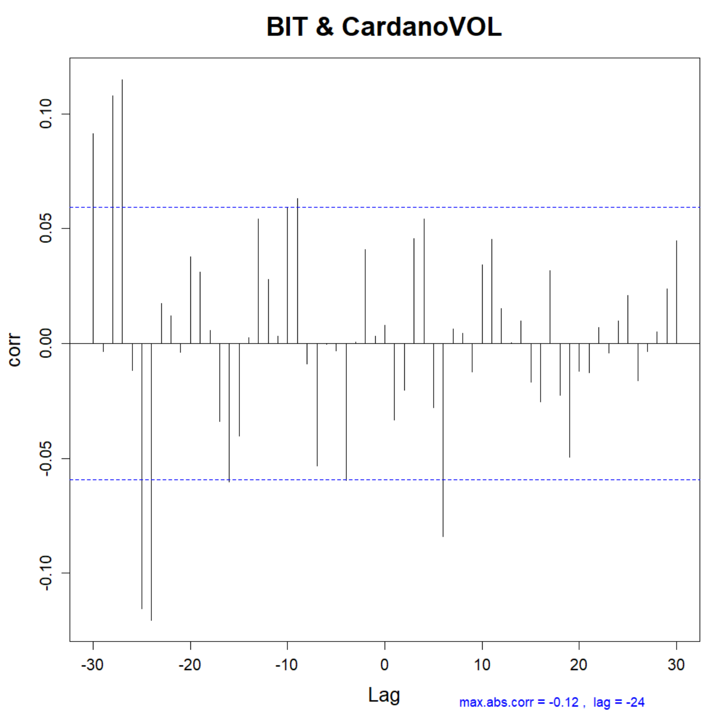

BIT & BinanceVOL
corr
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.106 , lag = -30

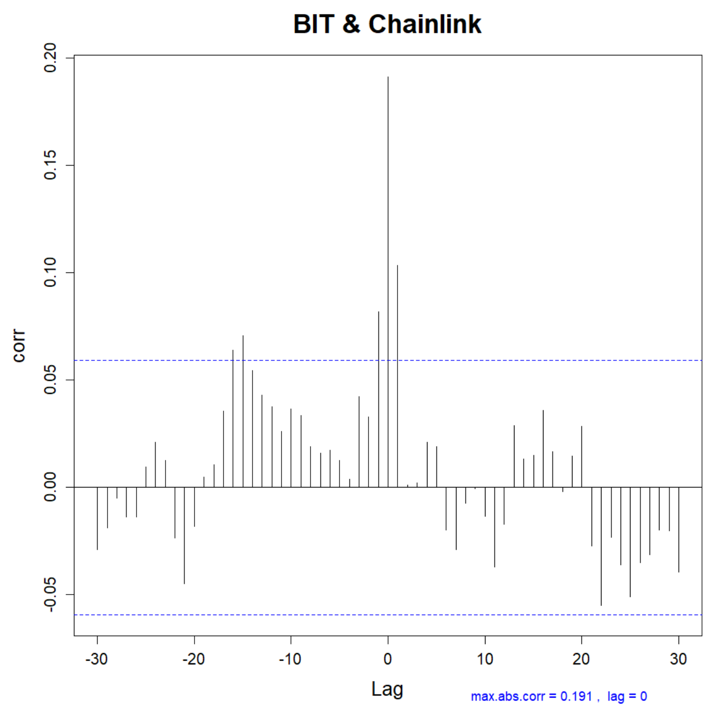

BIT & BitcoinCash
corr
0.2
0.1
0.0
-0.1
-0.2
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.251 , lag = 0

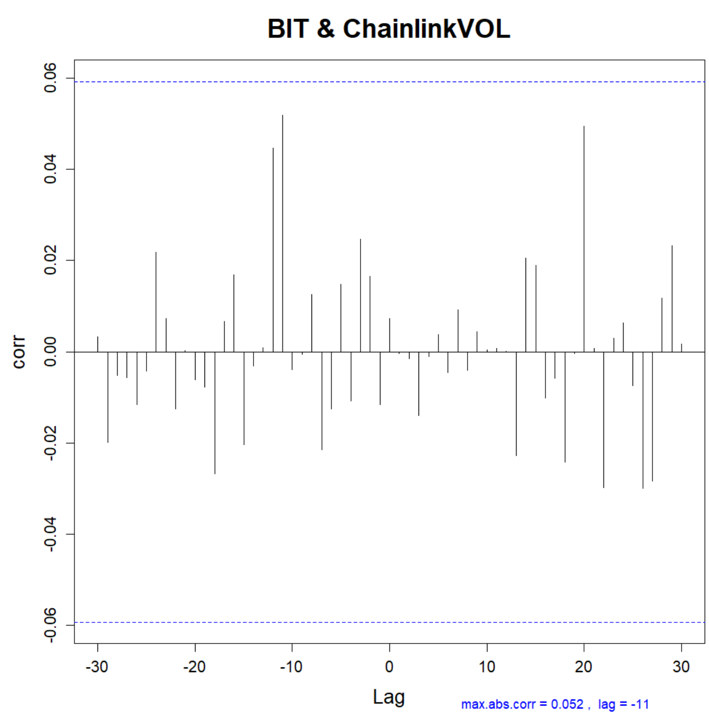

BIT & BitcoinCashVOL
corr
0.10
0.05
0.00
-0.05
-0.10
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.125 , lag = 10

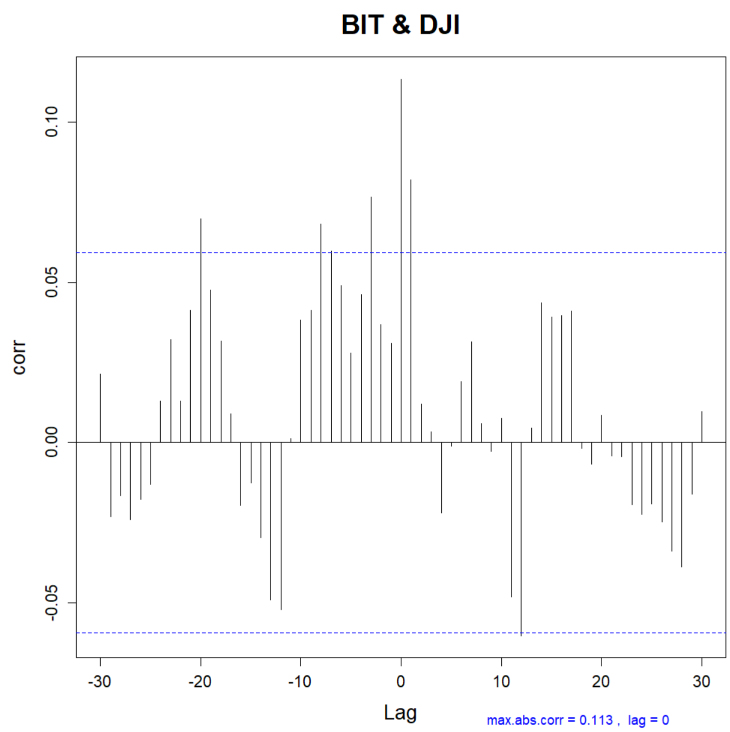

BIT & BitcoinVOL
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.067 , lag = 0

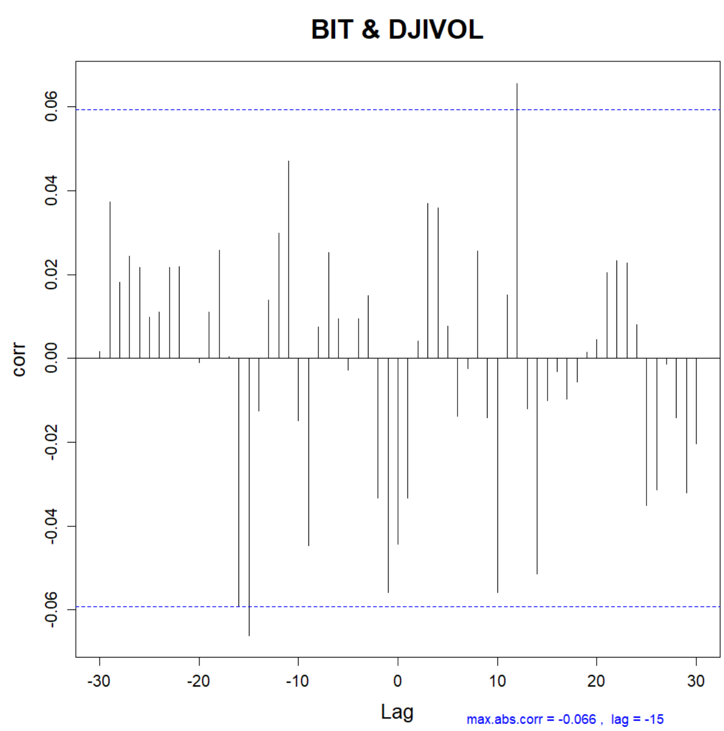

BIT & Cardano
corr
0.3
0.2
0.1
0.0
-0.1
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.355 , lag = 0

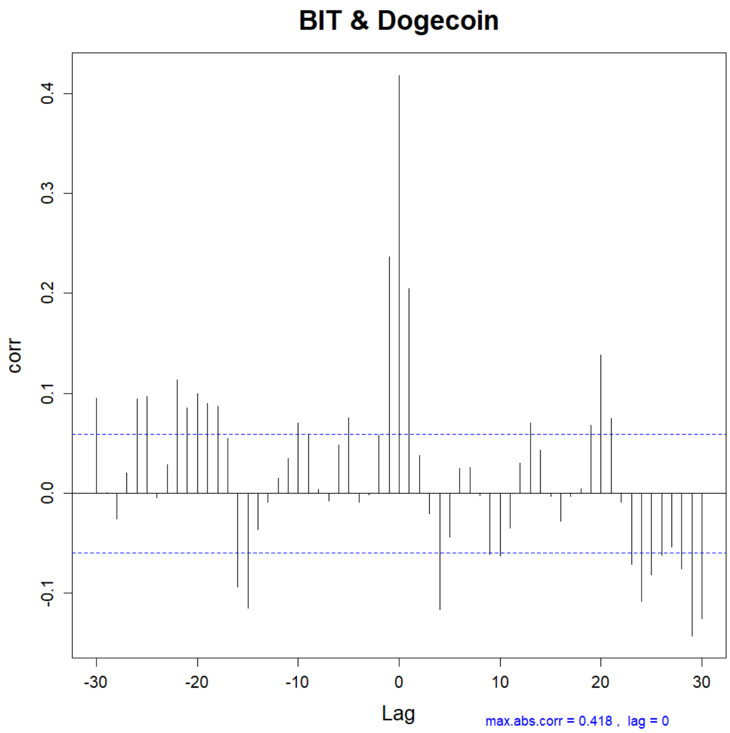

BIT & CardanoVOL
corr
0.10
0.05
0.00
-0.05
-0.10
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.12 , lag = -24

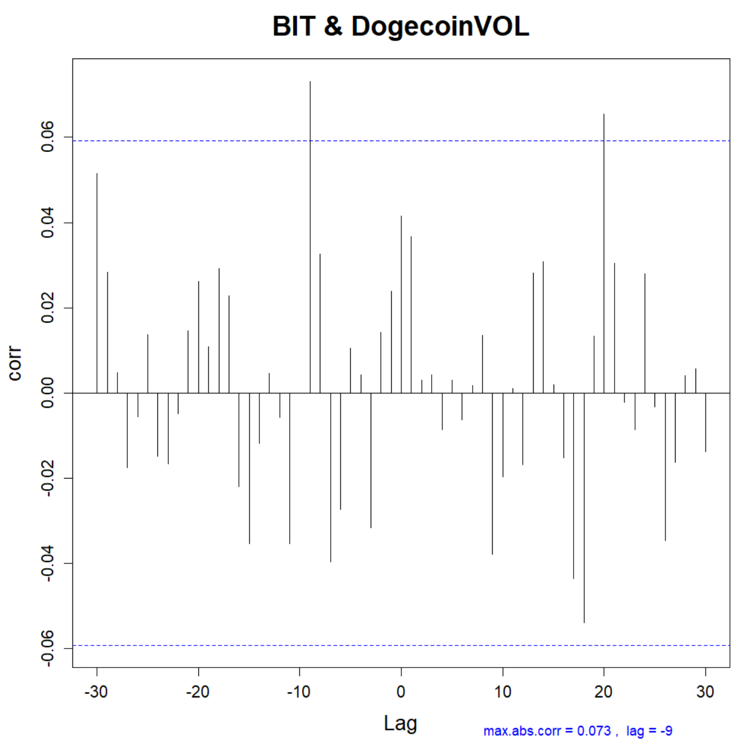

BIT & Chainlink
corr
0.20
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.191 , lag = 0

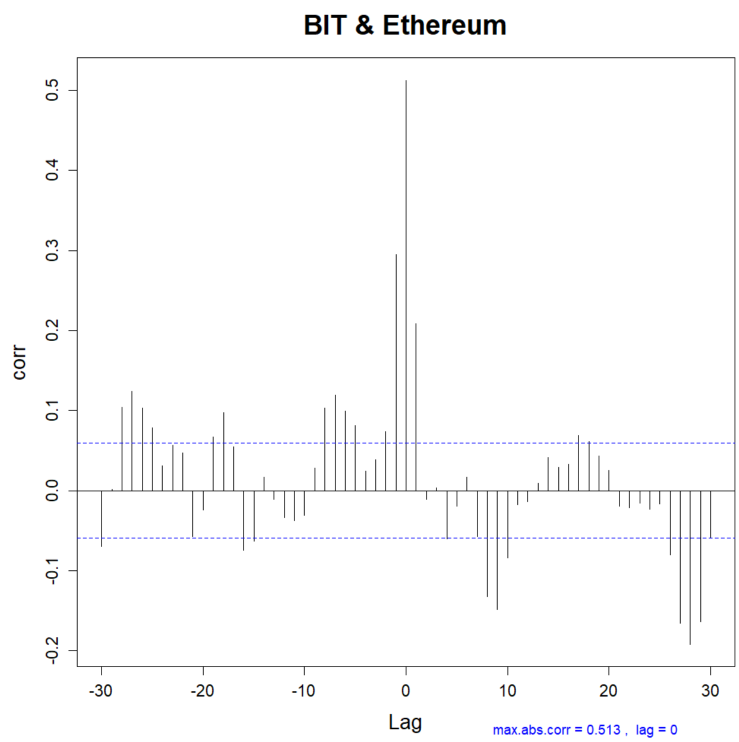

BIT & ChainlinkVOL
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.052 , lag = -11


BIT & DJI
corr
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.113 , lag = 0

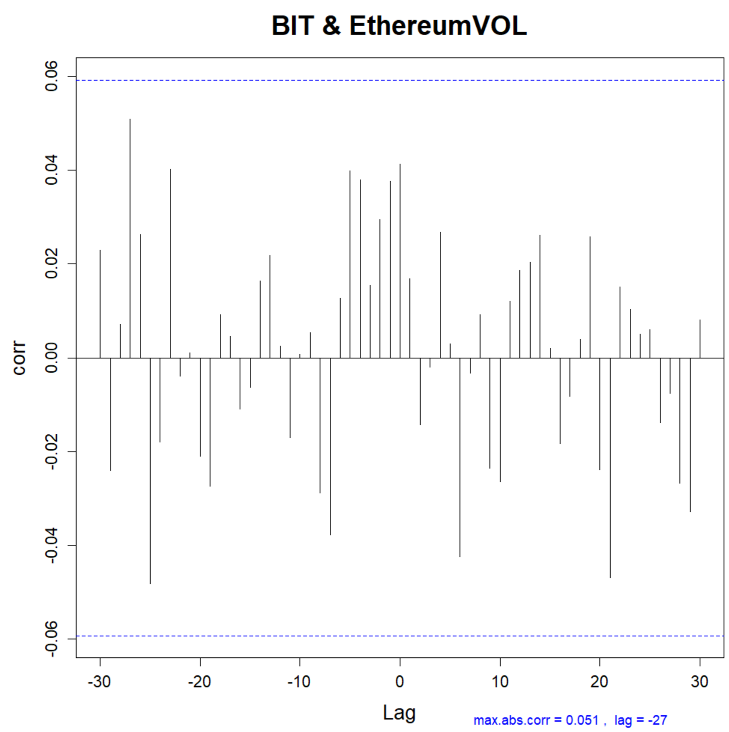

BIT & DJIVOL
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.066 , lag = -15


BIT & Dogecoin
corr
0.4
0.3
0.2
0.1
0.0
-0.1
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.418 , lag = 0

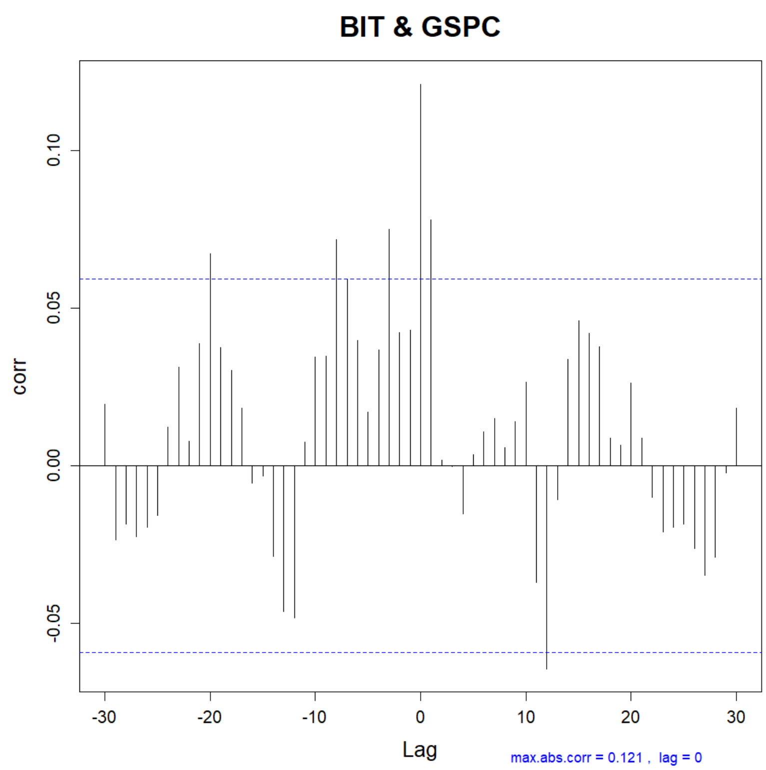

BIT & DogecoinVOL
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.073 , lag = -9

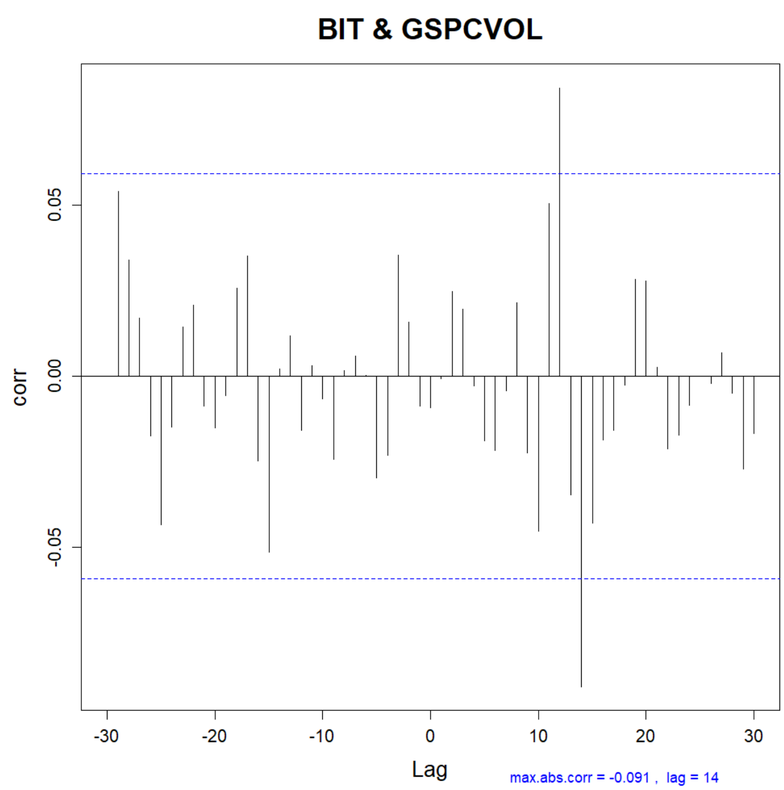


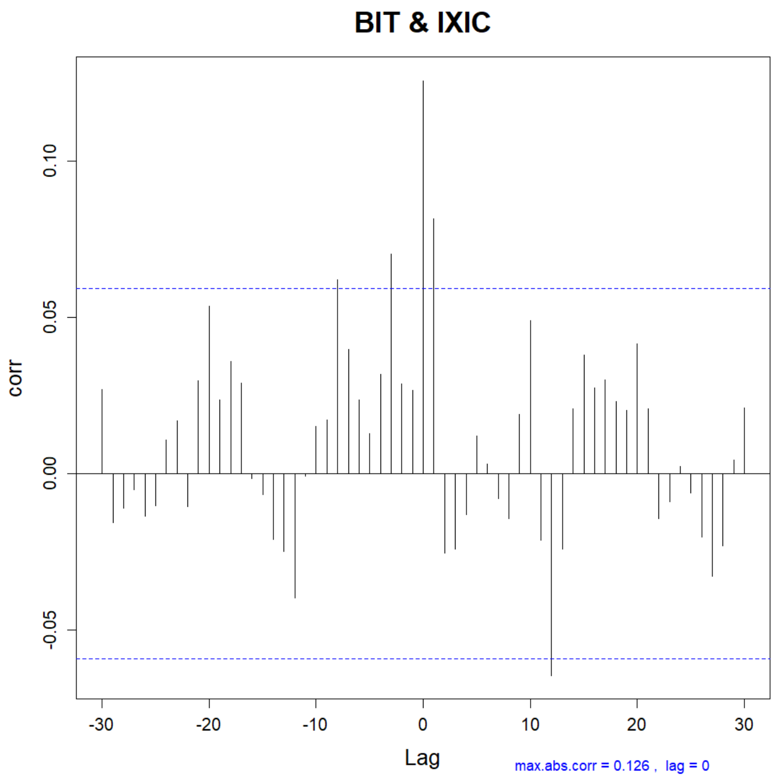
BIT & Ethereum
corr
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.513 , lag = 0

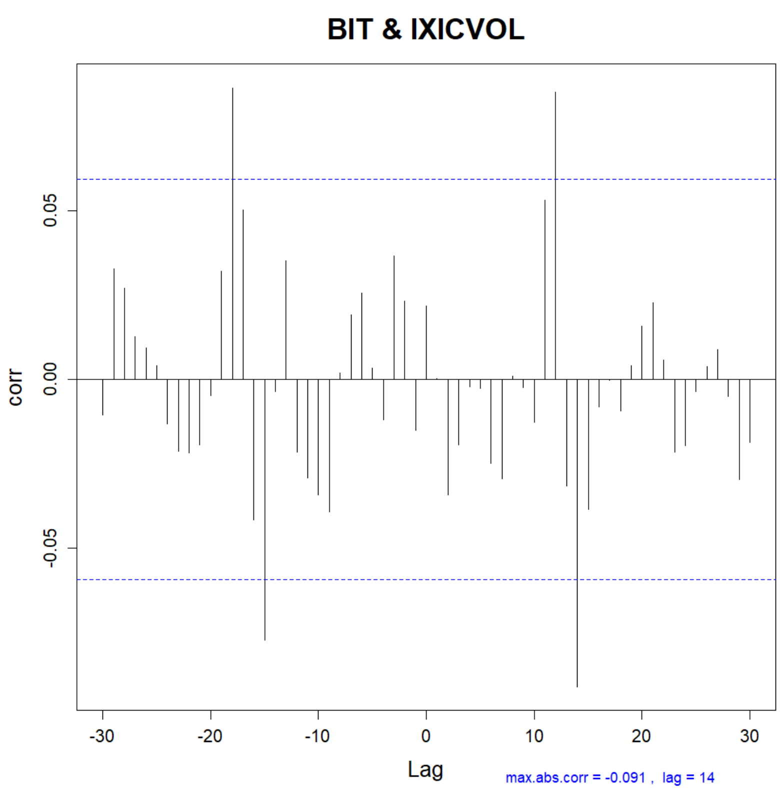


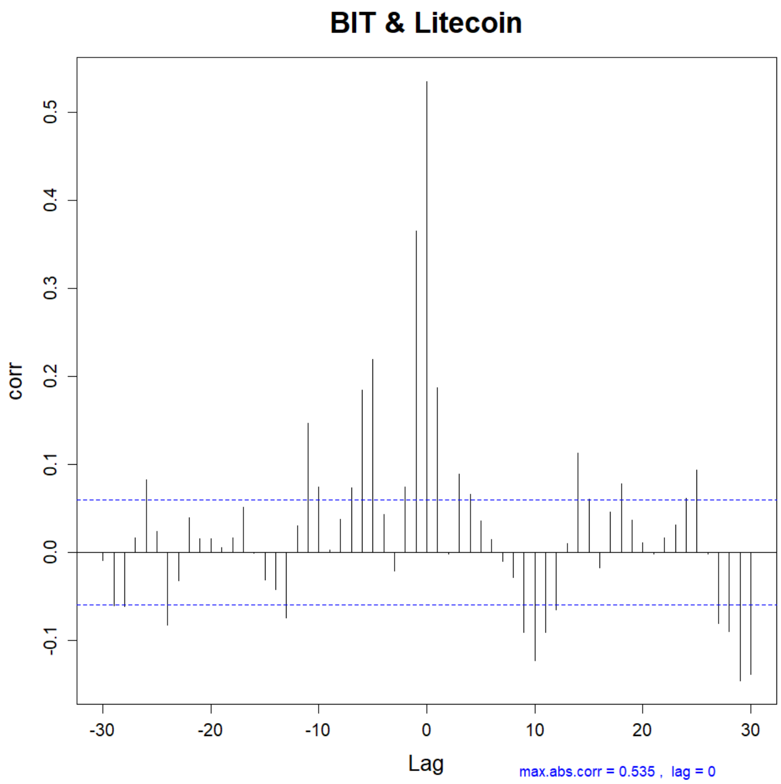
BIT & EthereumVOL
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.051 , lag = -27

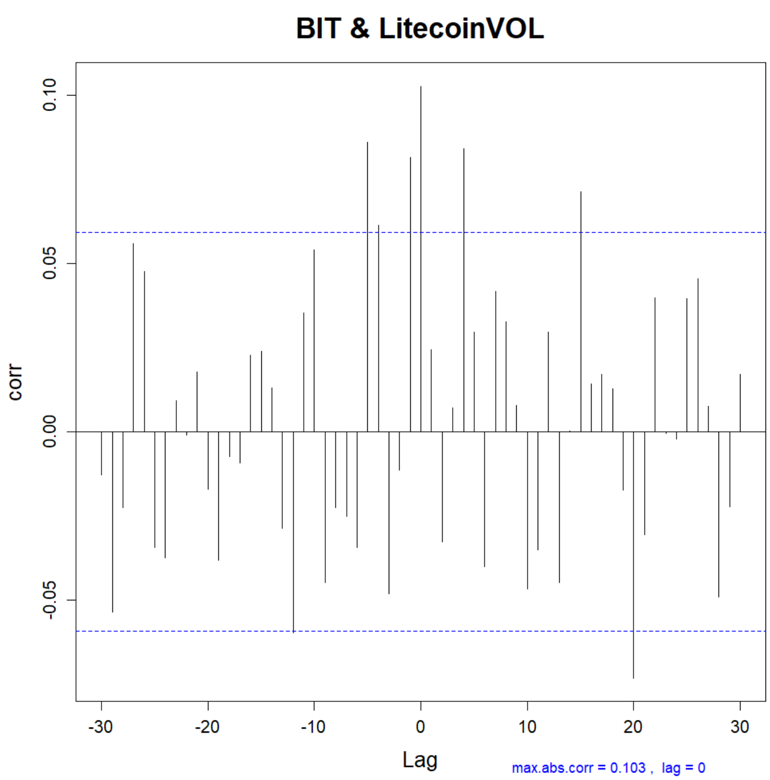

BIT & GSPC
corr
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.121 , lag = 0

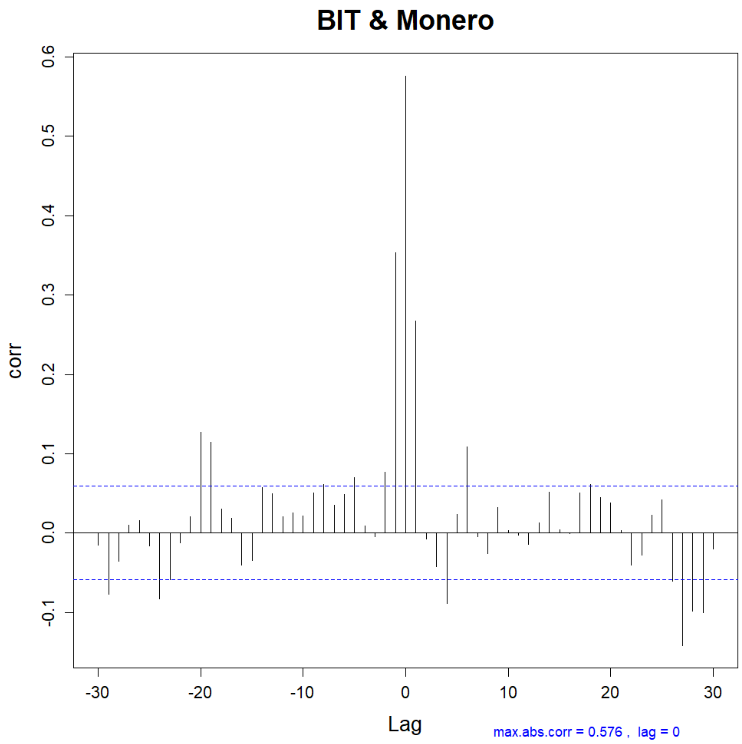

BIT & GSPCVOL
corr
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.091 , lag = 14

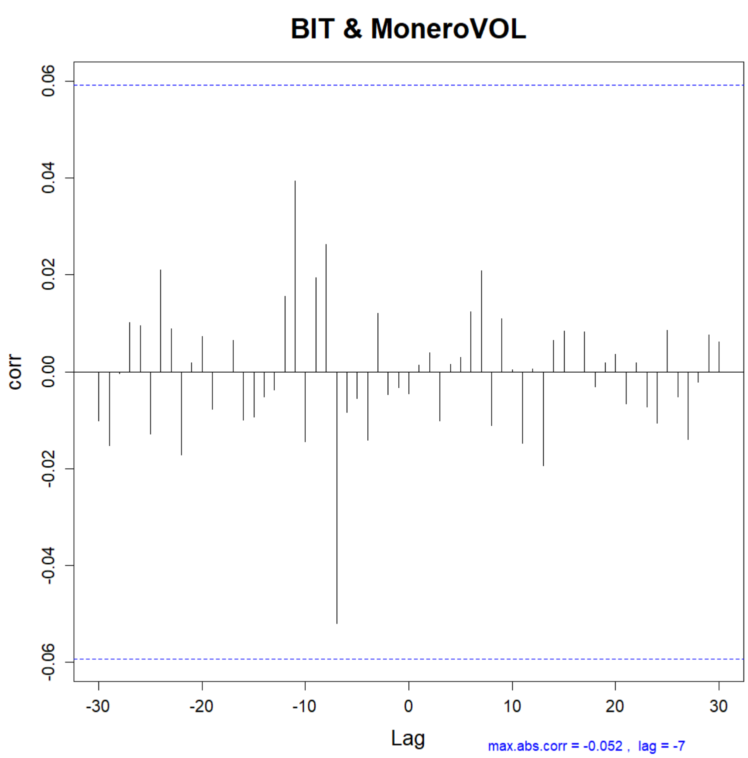

BIT & IXIC
corr
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.126 , lag = 0

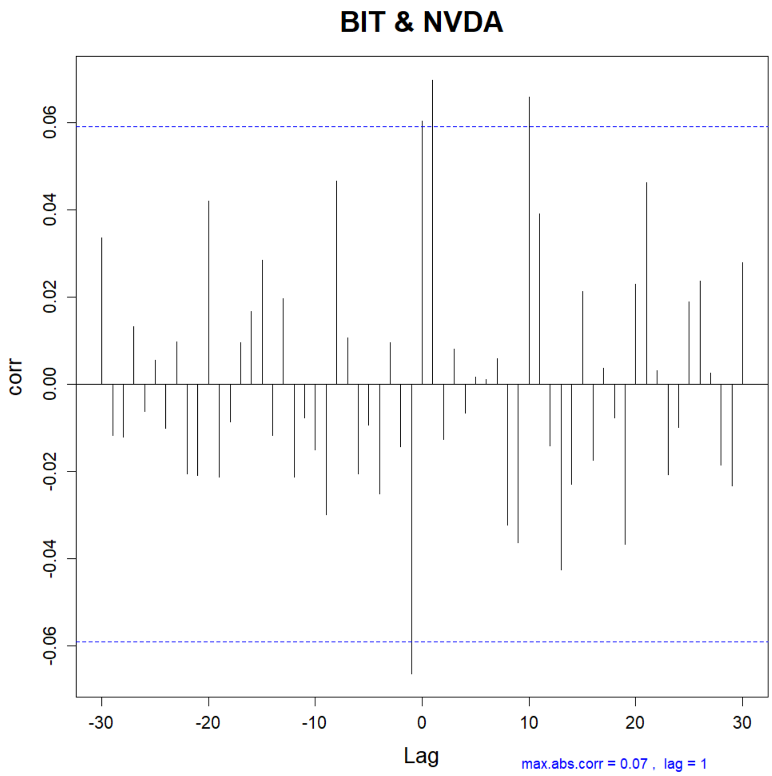

BIT & IXICVOL
corr
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.091 , lag = 14

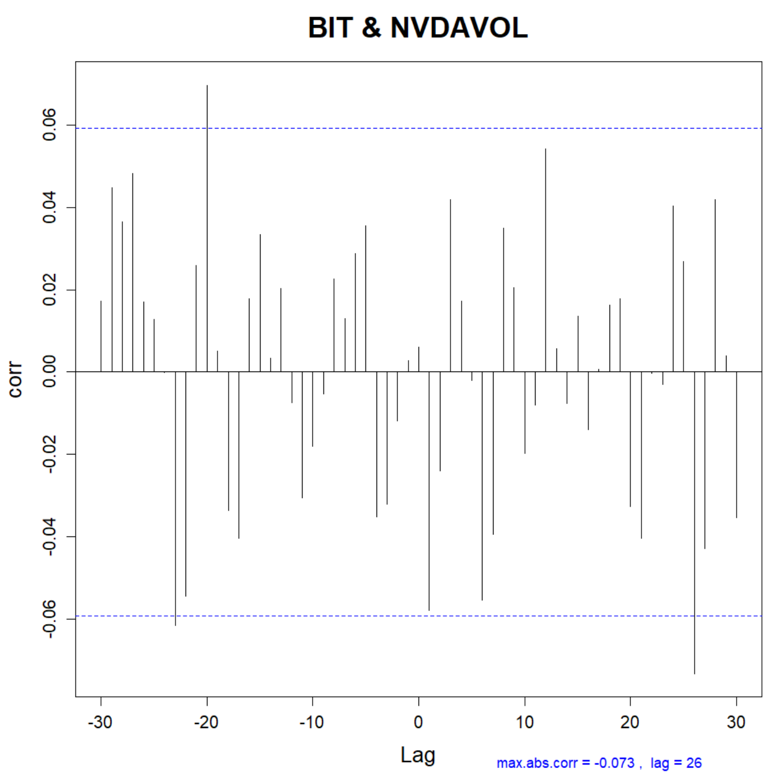

BIT & Litecoin
corr
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.535 , lag = 0

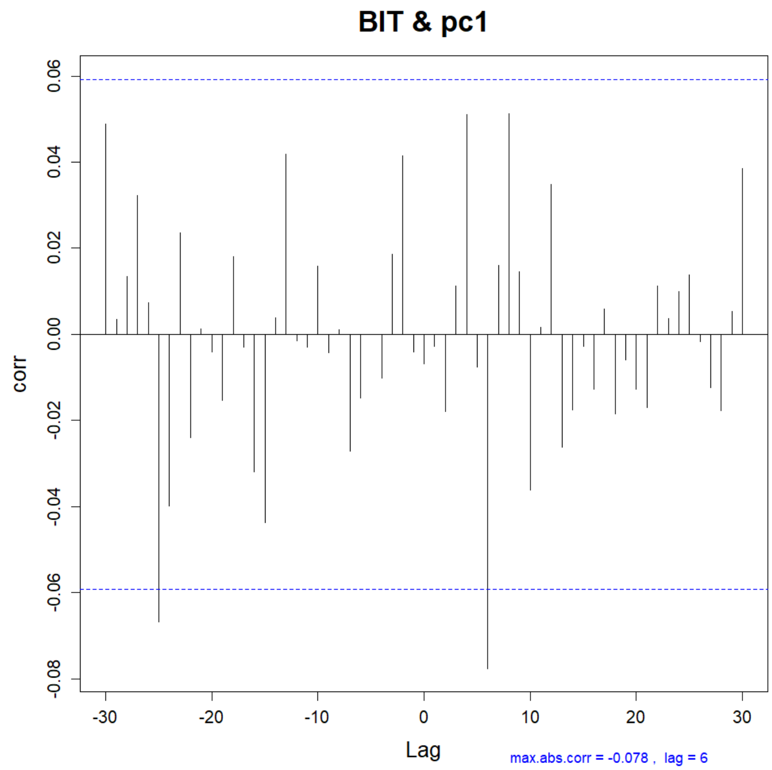

BIT & LitecoinVOL
corr
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.103 , lag = 0

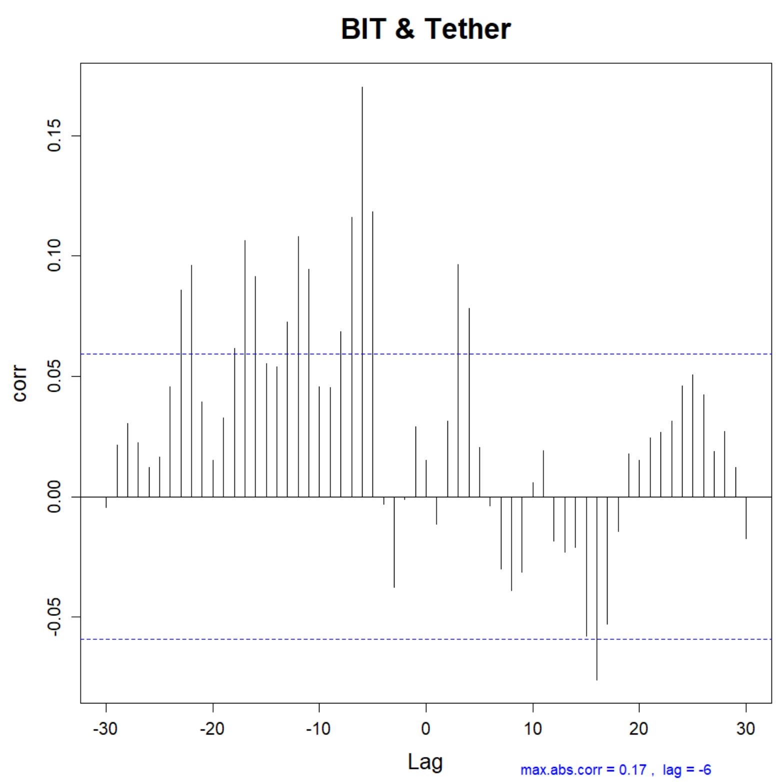

BIT & Monero
corr
Lag
max.abs.corr = 0.576 , lag = 0

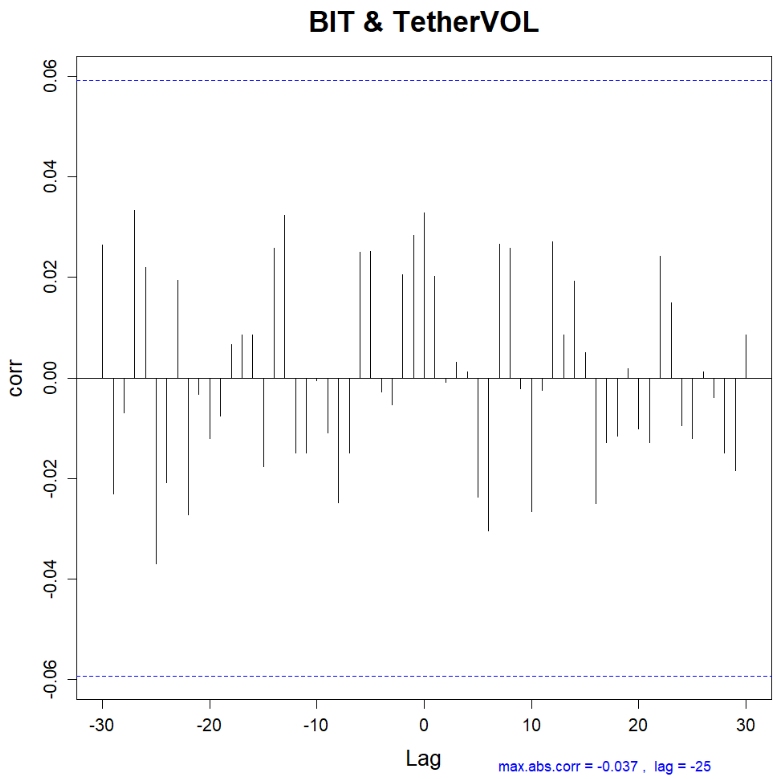

BIT & MoneroVOL
corr
Lag
max.abs.corr = -0.052 , lag = -7

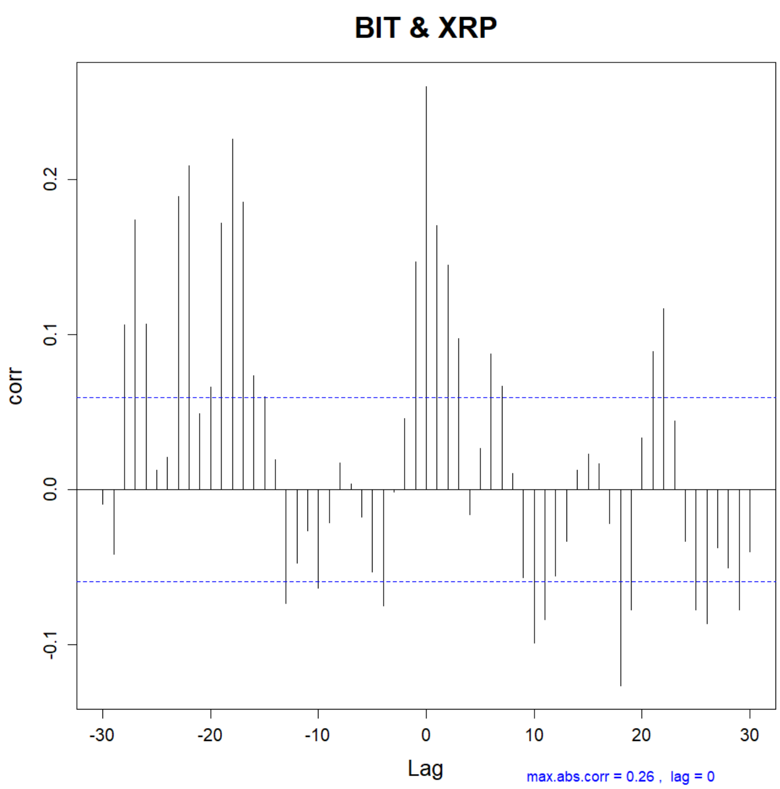

BIT & NVDA
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.07 , lag = 1

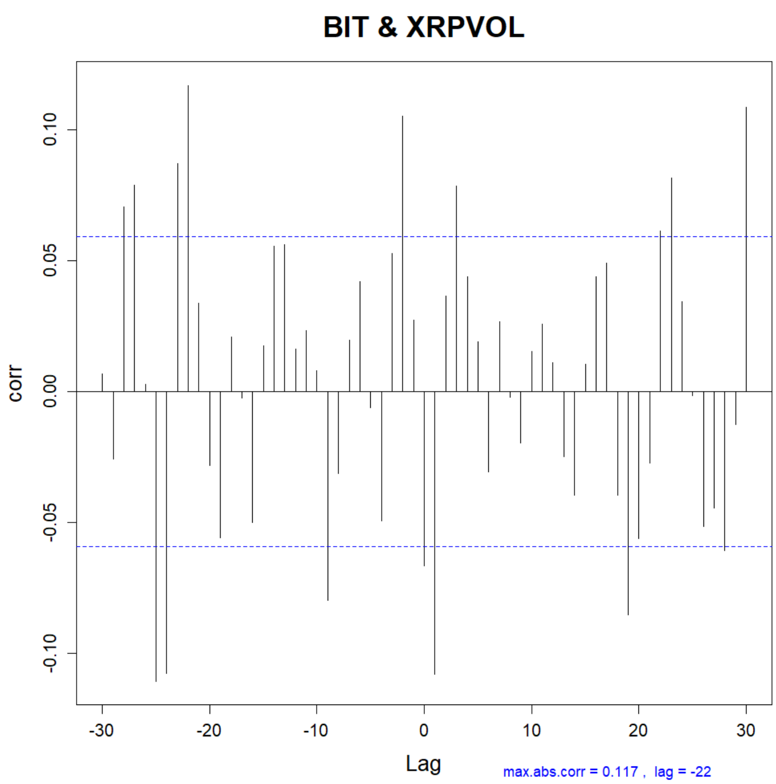

BIT & NVDAVOL
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.073 , lag = 26

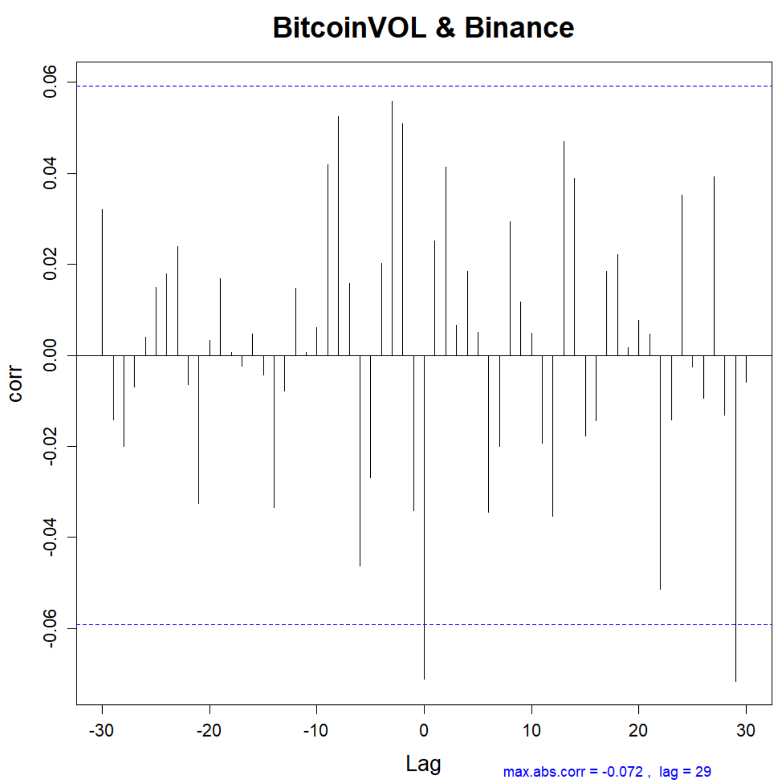

BIT & pc1
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-0.08
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.078 , lag = 6


BIT & Tether
corr
0.15
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.17 , lag = -6

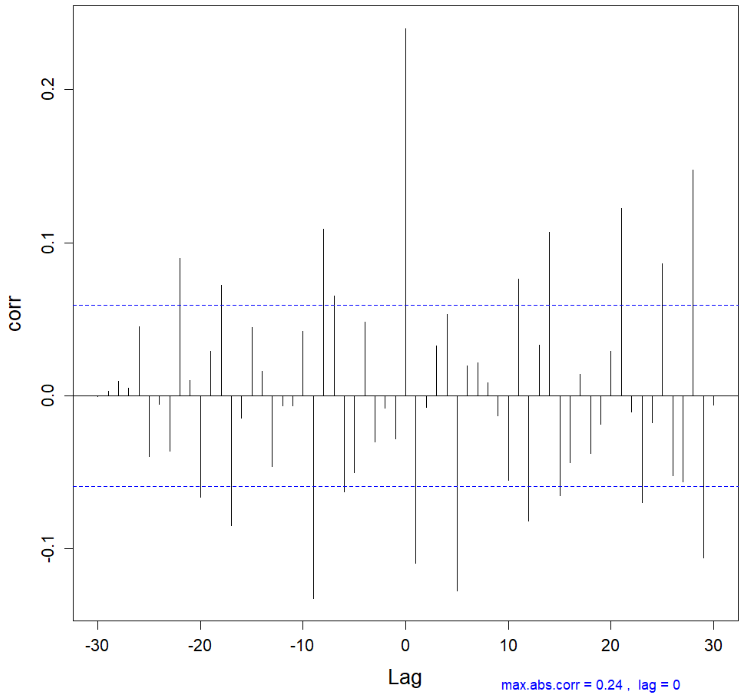

BIT & TetherVOL
corr
Lag
max.abs.corr = -0.037 , lag = -25


BIT & XRP
corr
Lag
max.abs.corr = 0.26 , lag = 0

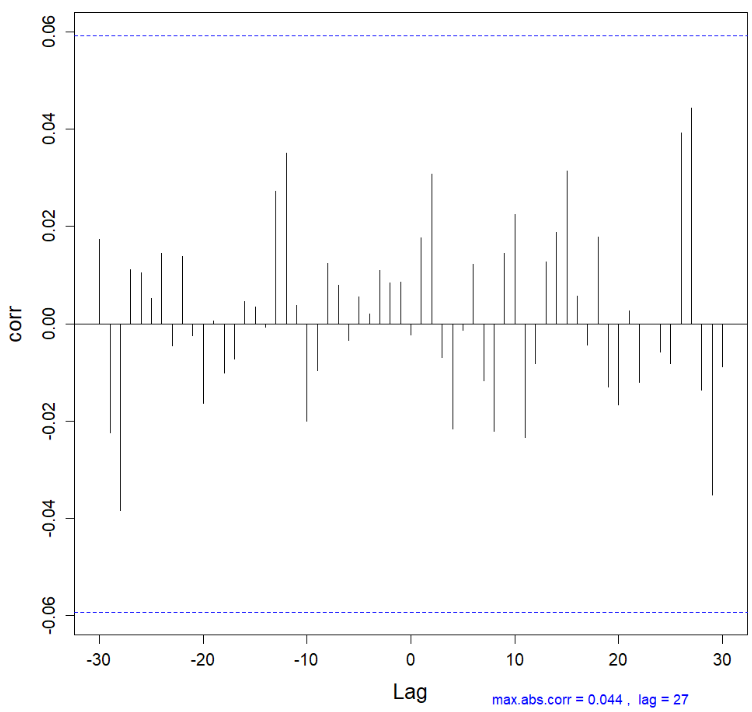

BIT & XRPVOL
corr
0.10
0.05
0.00
-0.05
-0.10
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.117 , lag = -22

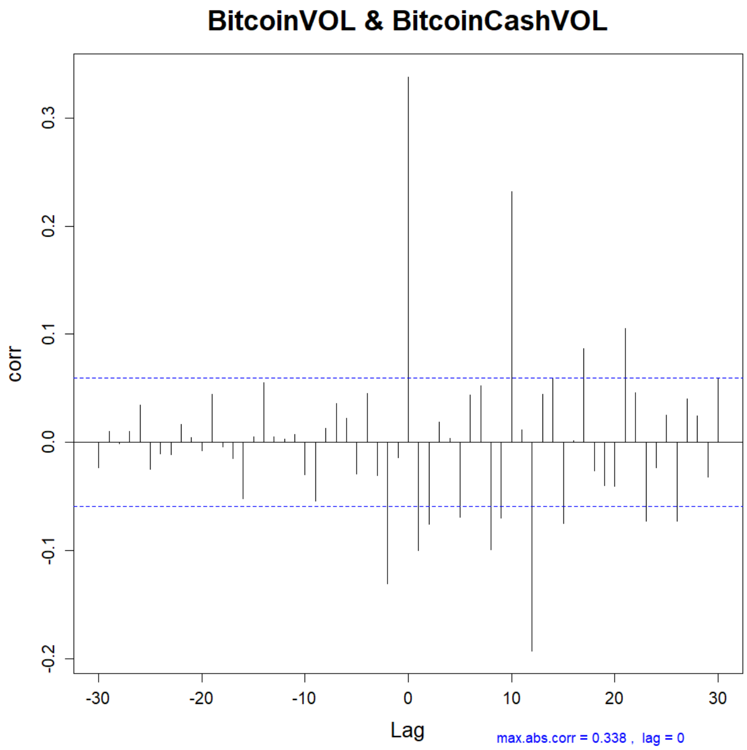

BitcoinVOL & Binance
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.072 , lag = 29

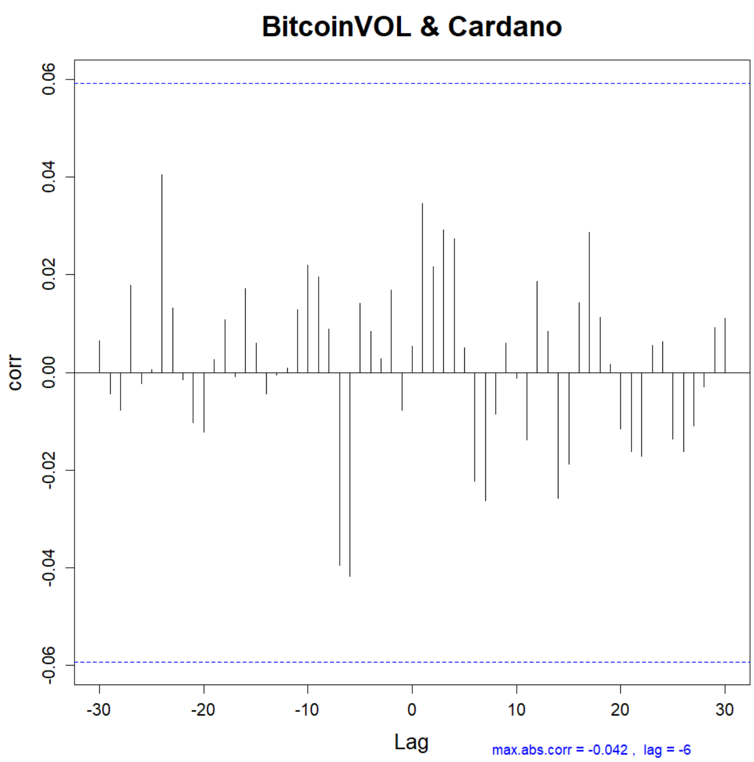
BitcoinVOL & BinanceVOL

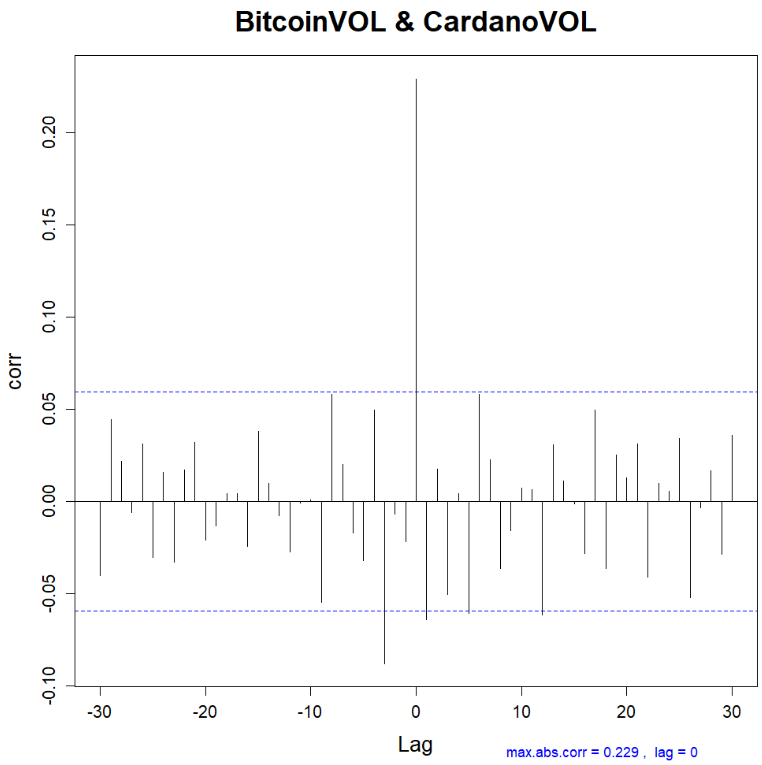


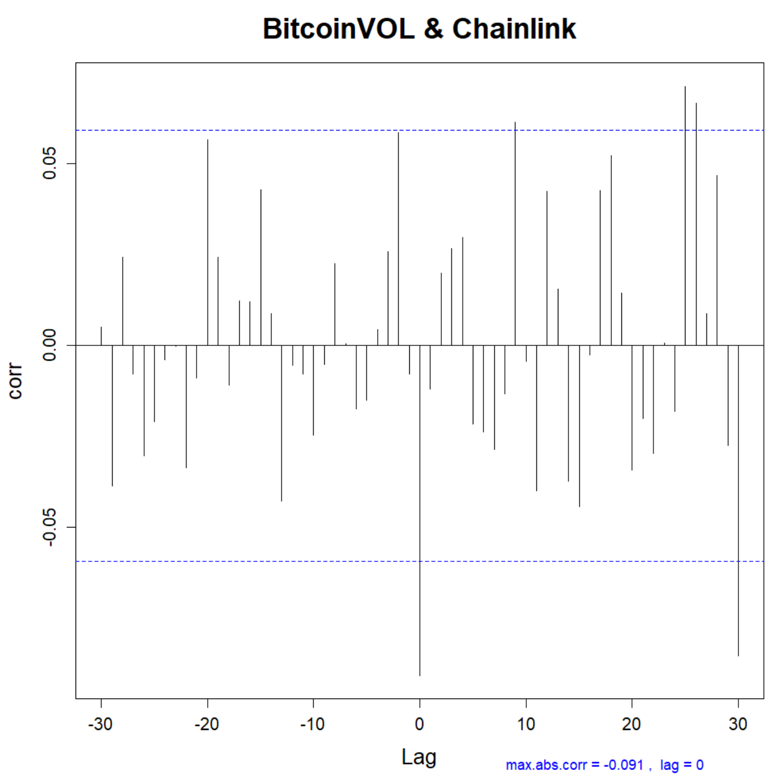
BitcoinVOL & BitcoinCash

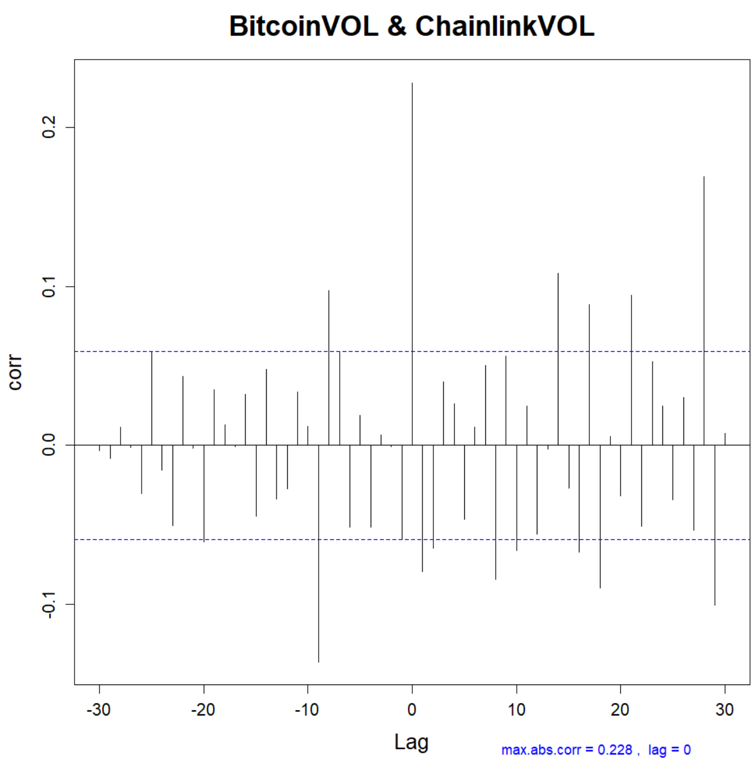


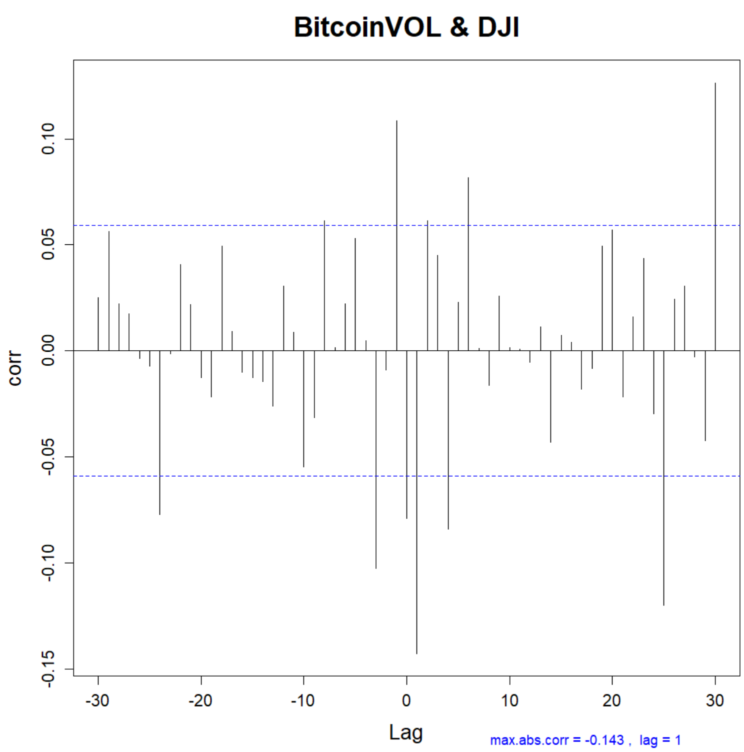

BitcoinVOL & BitcoinCashVOL
corr
0.3
0.2
0.1
0.0
-0.1
-0.2
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.338 , lag = 0

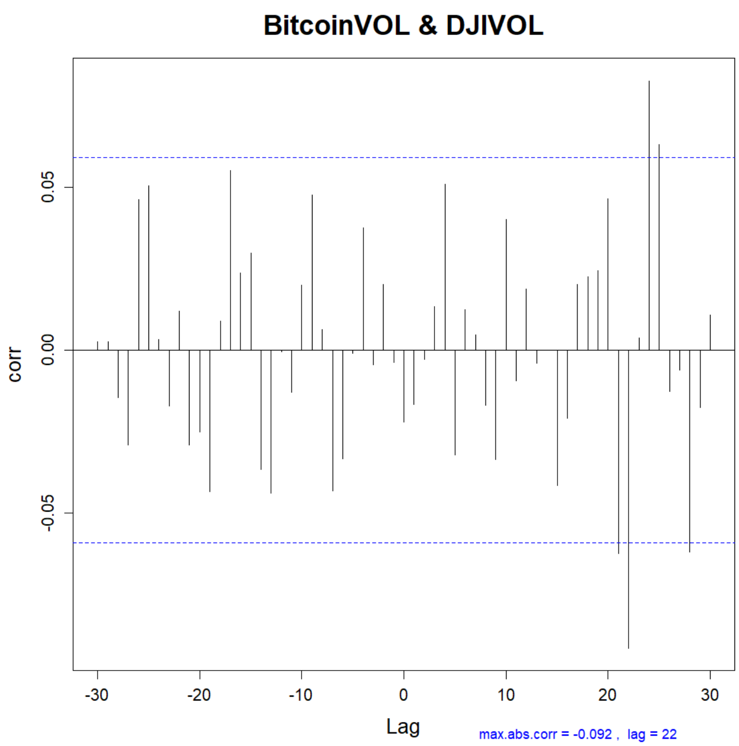

BitcoinVOL & Cardano
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.042 , lag = -6

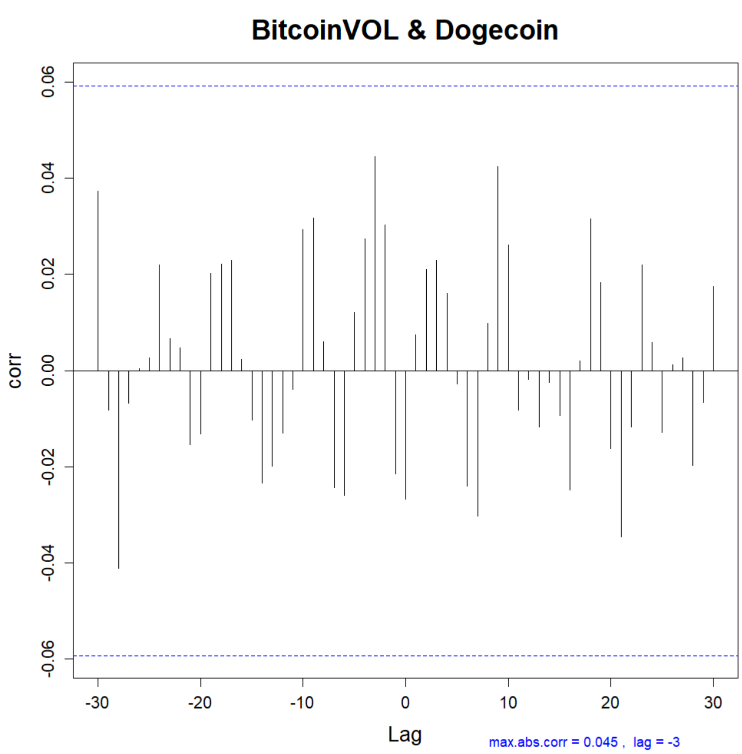

BitcoinVOL & CardanoVOL
corr
0.20
0.15
0.10
0.05
0.00
-0.05
-0.10
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.229 , lag = 0

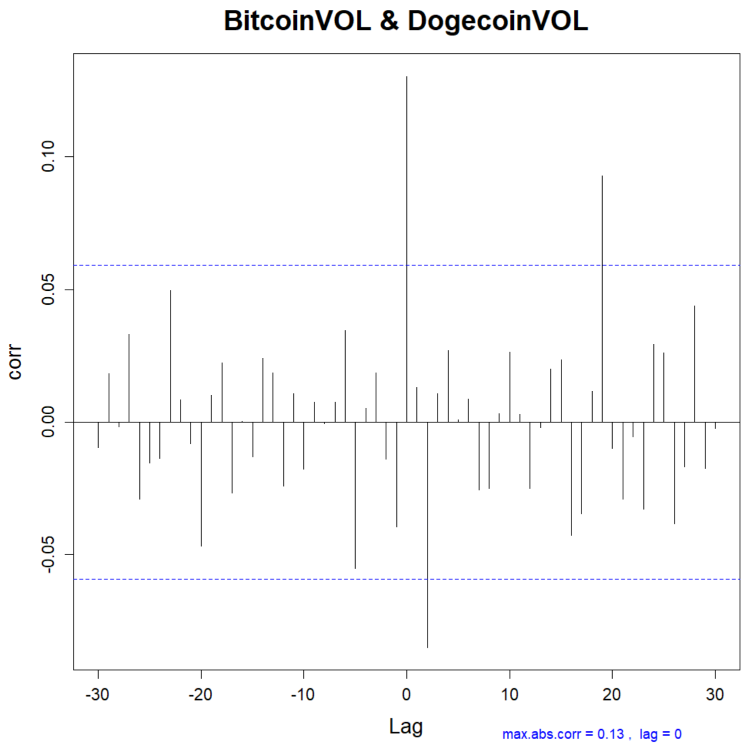

BitcoinVOL & Chainlink
corr
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.091 , lag = 0

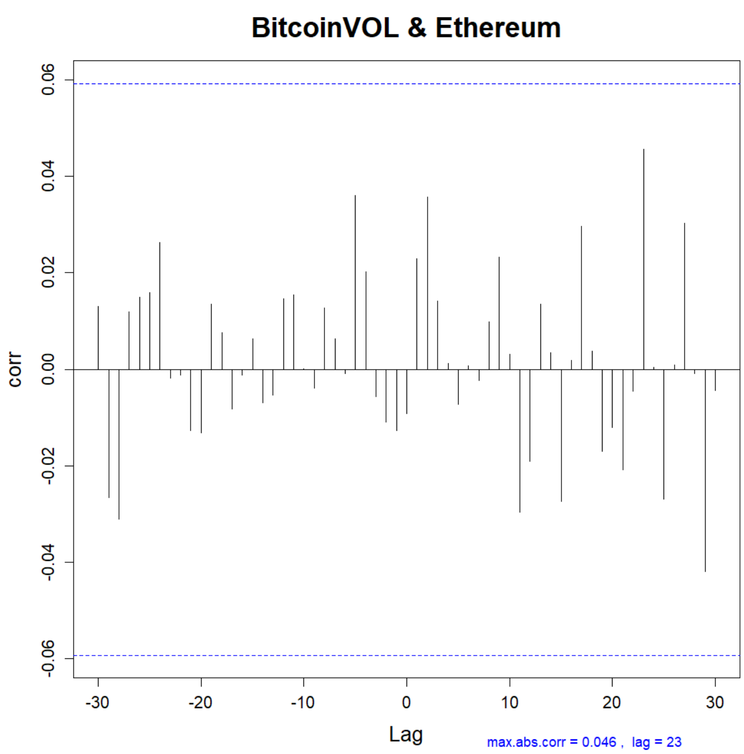

BitcoinVOL & ChainlinkVOL
corr
Lag
max.abs.corr = 0.228 , lag = 0

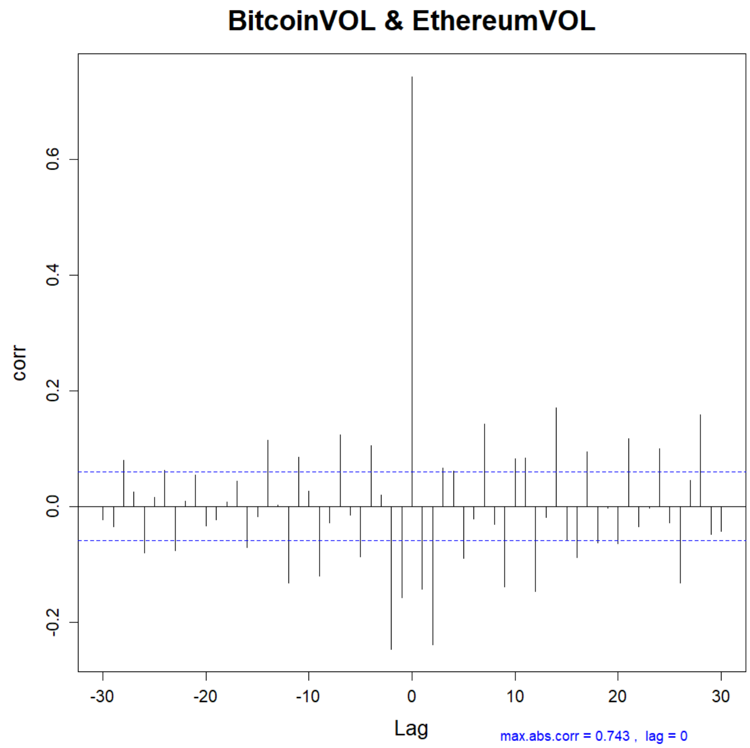

BitcoinVOL & DJI
corr
Lag
max.abs.corr = -0.143 , lag = 1

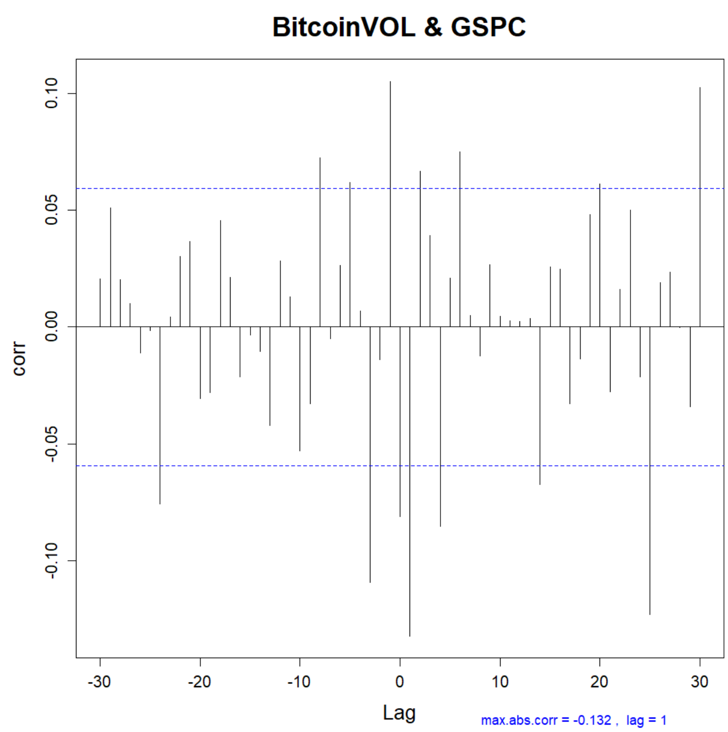

BitcoinVOL & DJIVOL
corr
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.092 , lag = 22

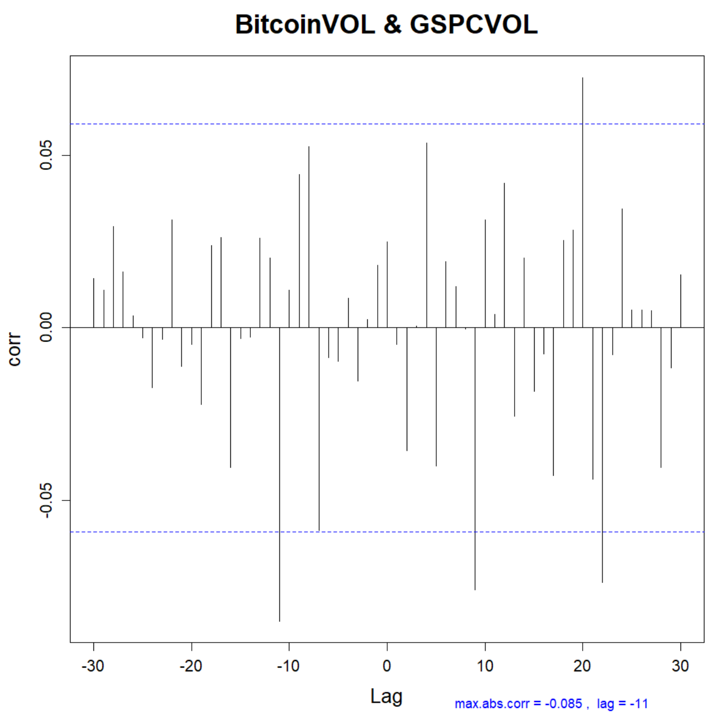

BitcoinVOL & Dogecoin
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.045 , lag = -3

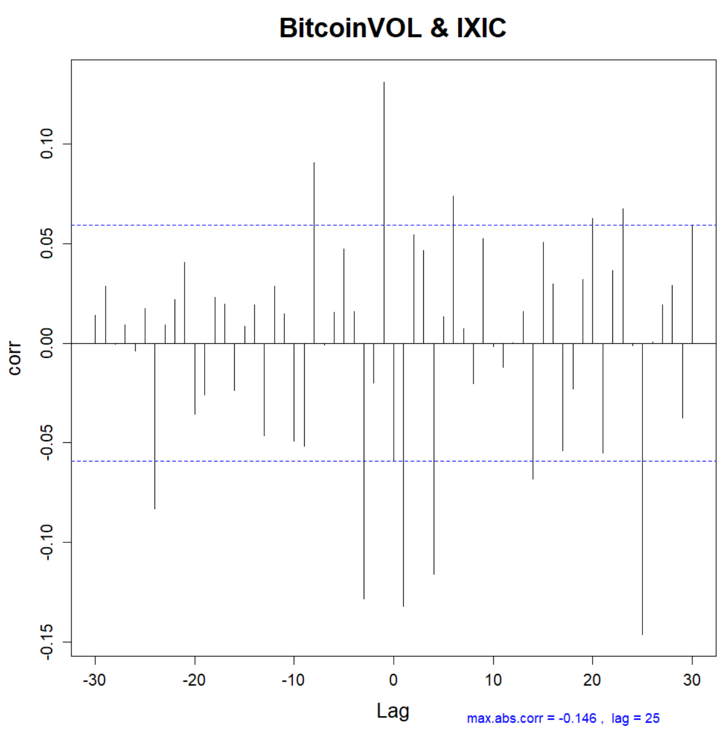

BitcoinVOL & DogecoinVOL
corr
0.10
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.13 , lag = 0

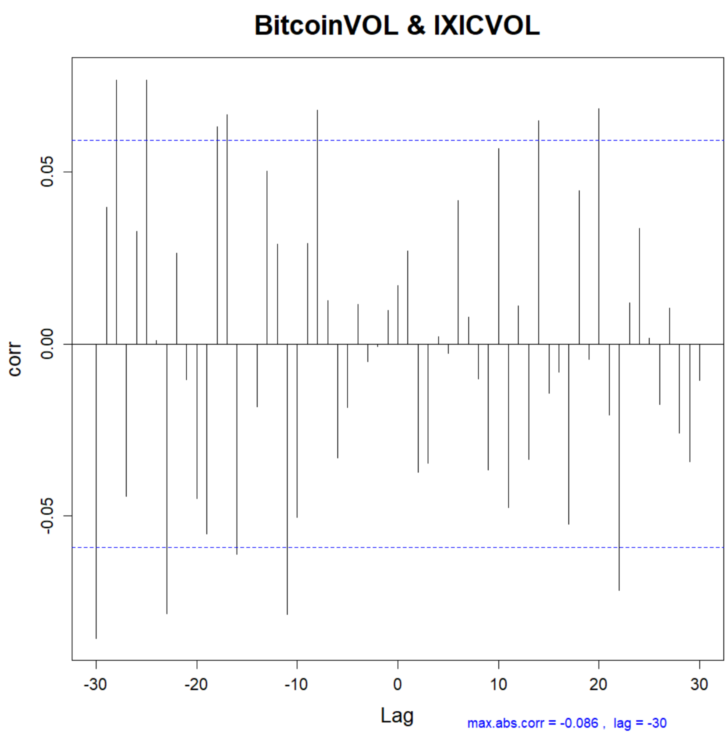

BitcoinVOL & Ethereum
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.046 , lag = 23

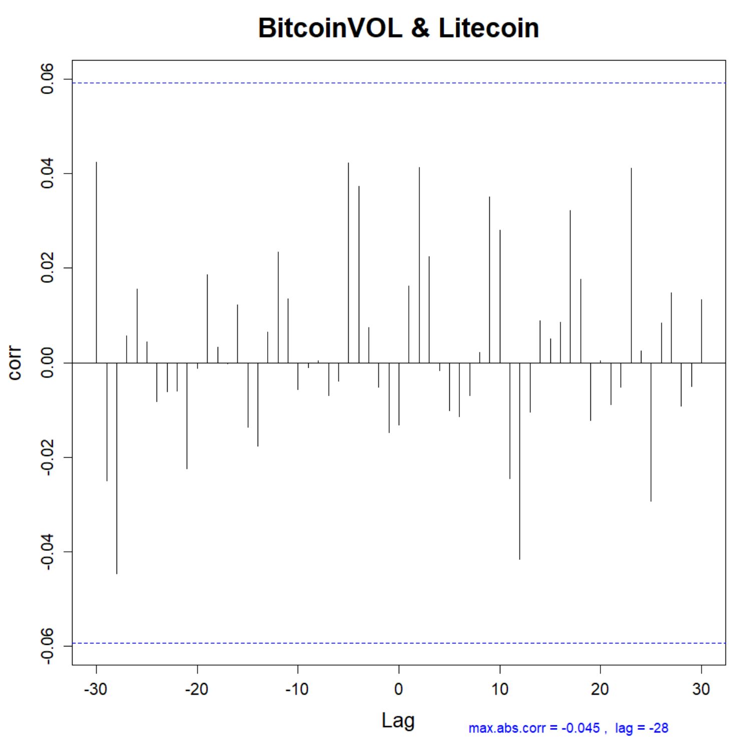

BitcoinVOL & EthereumVOL
corr
Lag
max.abs.corr = 0.743 , lag = 0

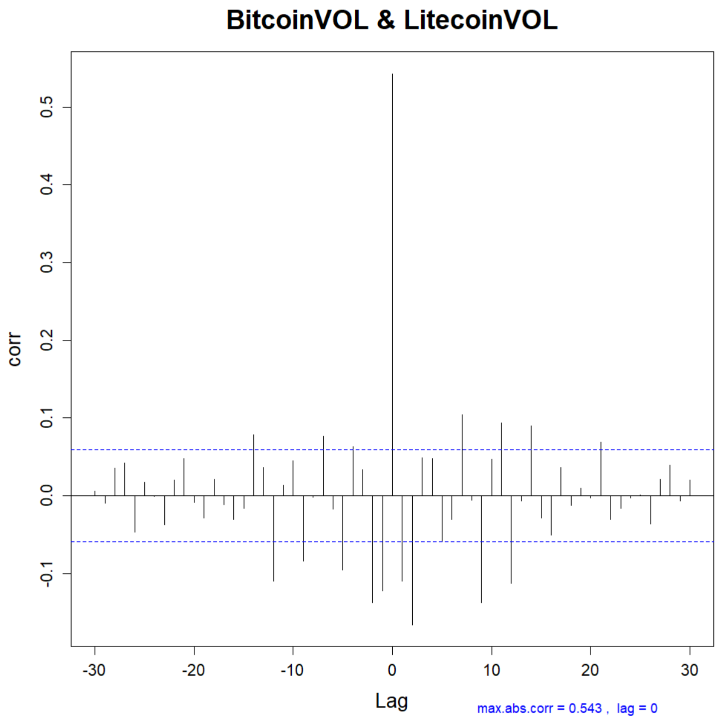

BitcoinVOL & GSPC
corr
Lag
max.abs.corr = -0.132 , lag = 1

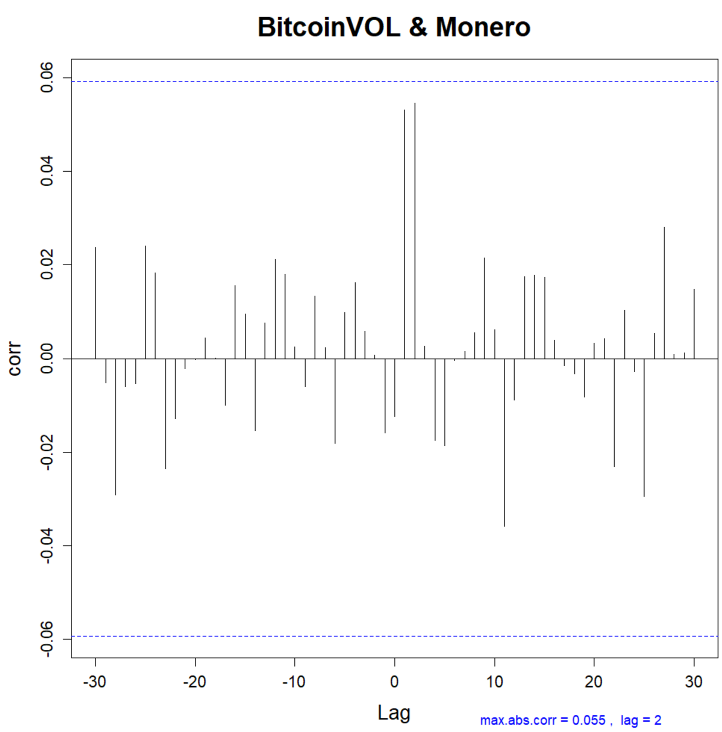

BitcoinVOL & GSPCVOL
corr
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.085 , lag = -11

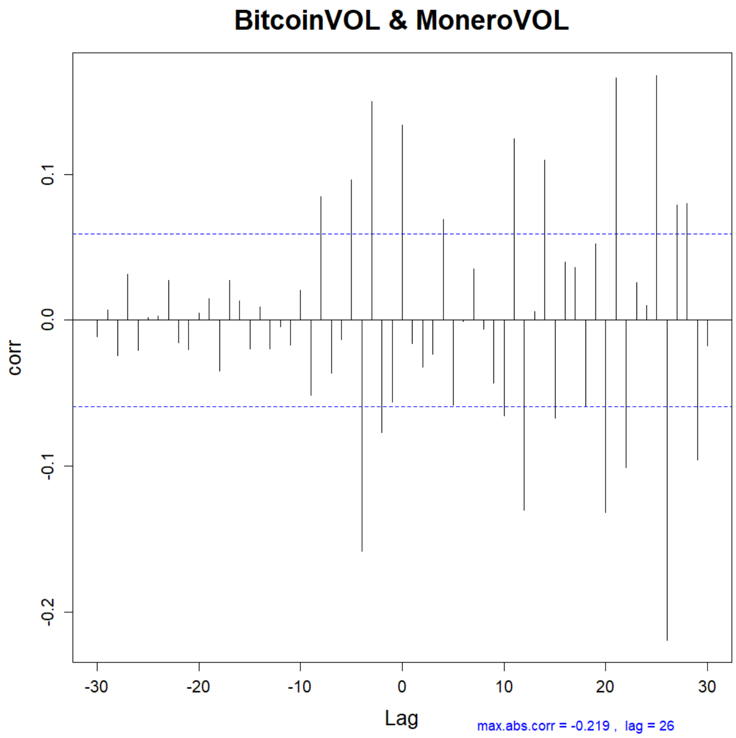

BitcoinVOL & IXIC
corr
0.10
0.05
0.00
-0.05
-0.10
-0.15
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.146 , lag = 25

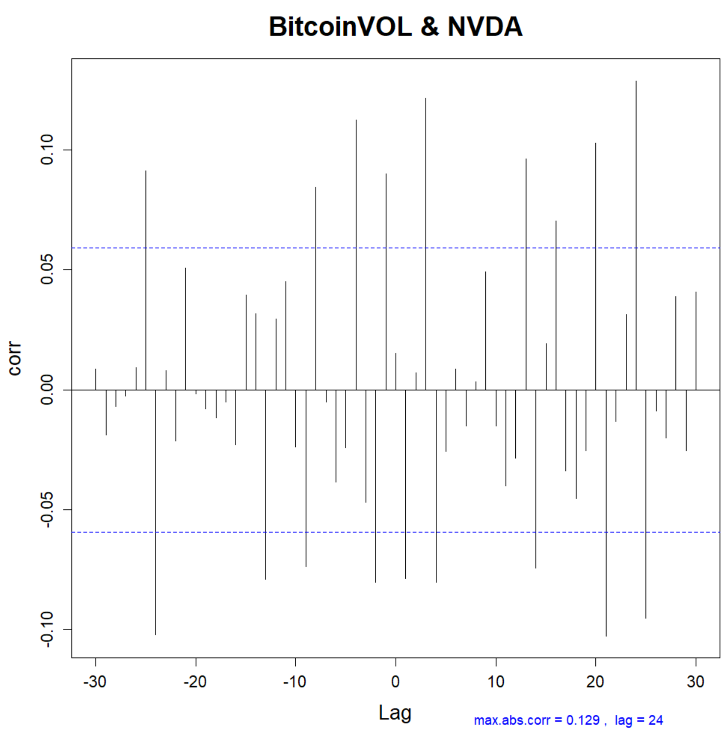

BitcoinVOL & IXICVOL
corr
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.086 , lag = -30

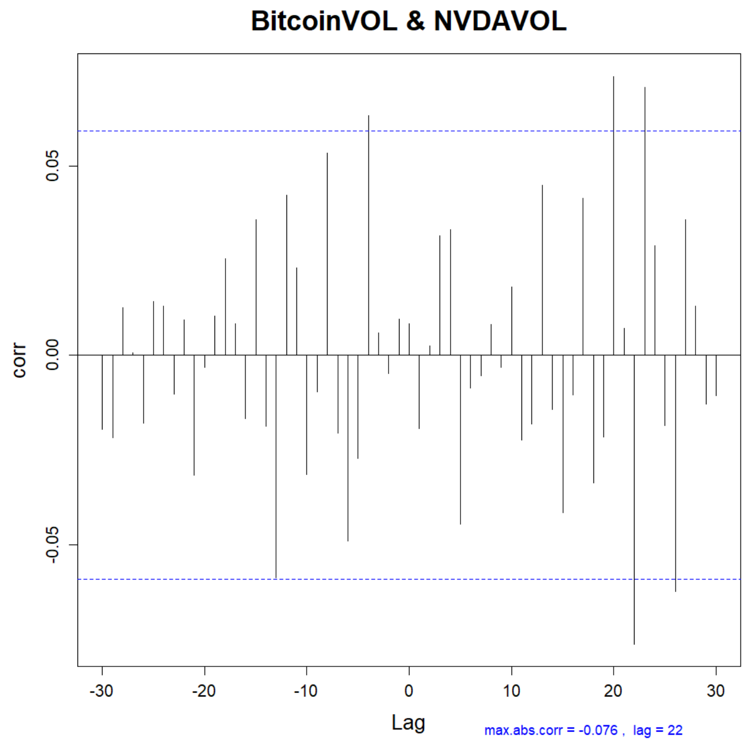

BitcoinVOL & Litecoin
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.045 , lag = -28

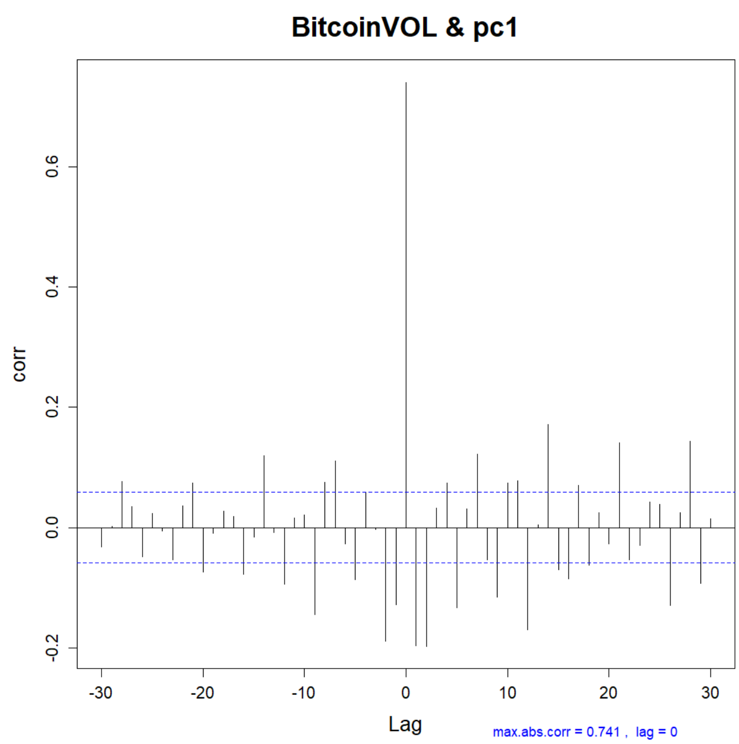

BitcoinVOL & LitecoinVOL
corr
0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.543 , lag = 0

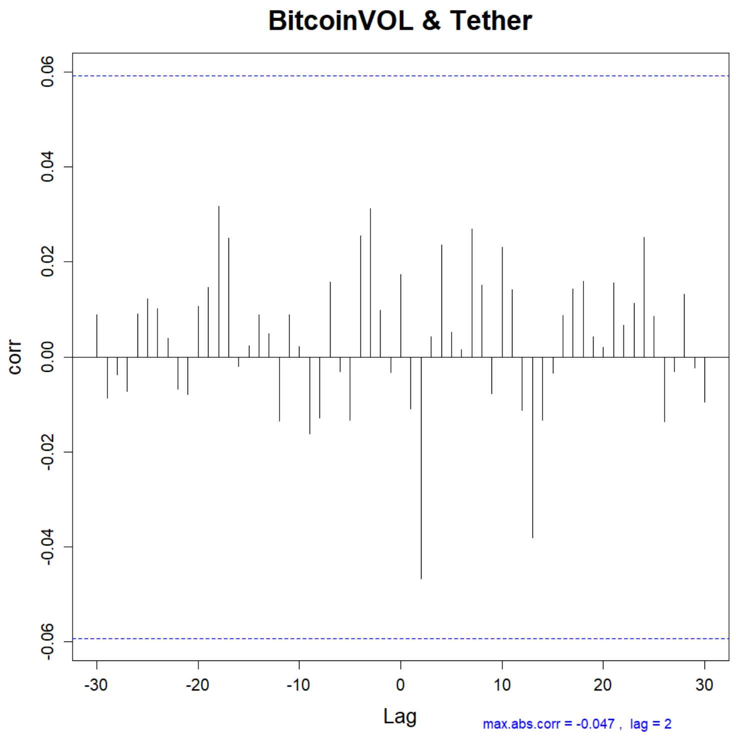

BitcoinVOL & Monero
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.055 , lag = 2

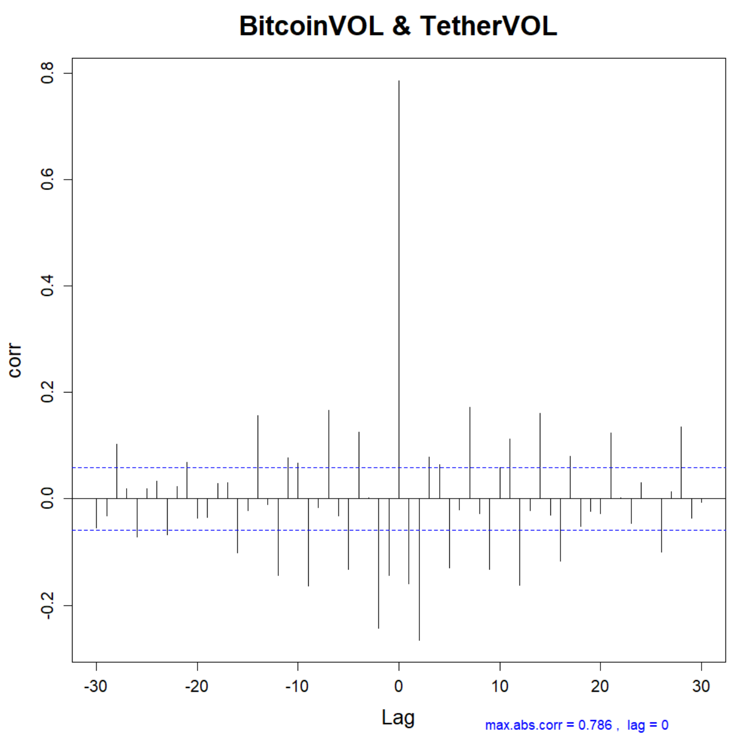

BitcoinVOL & MoneroVOL
corr
0.1
0.0
-0.1
-0.2
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.219 , lag = 26

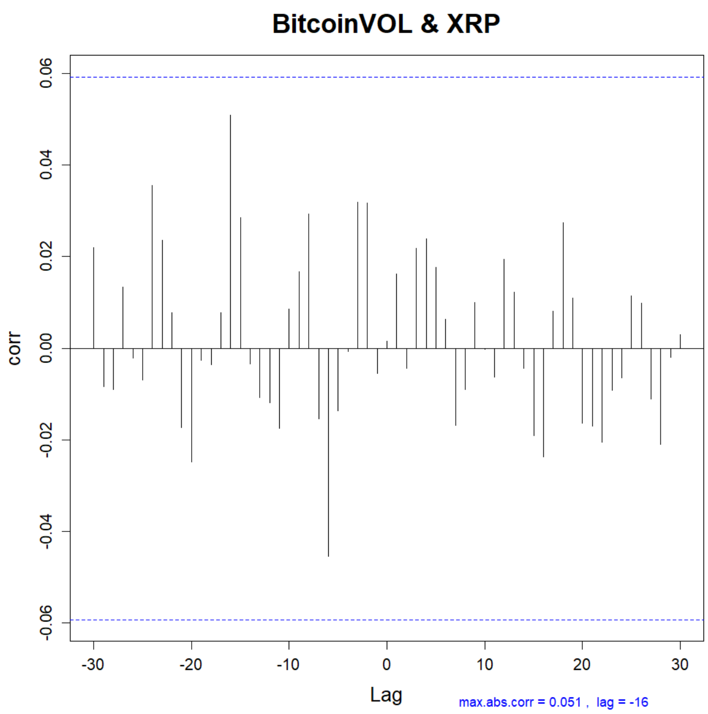

BitcoinVOL & NVDA
corr
0.10
0.05
0.00
-0.05
-0.10
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.129 , lag = 24

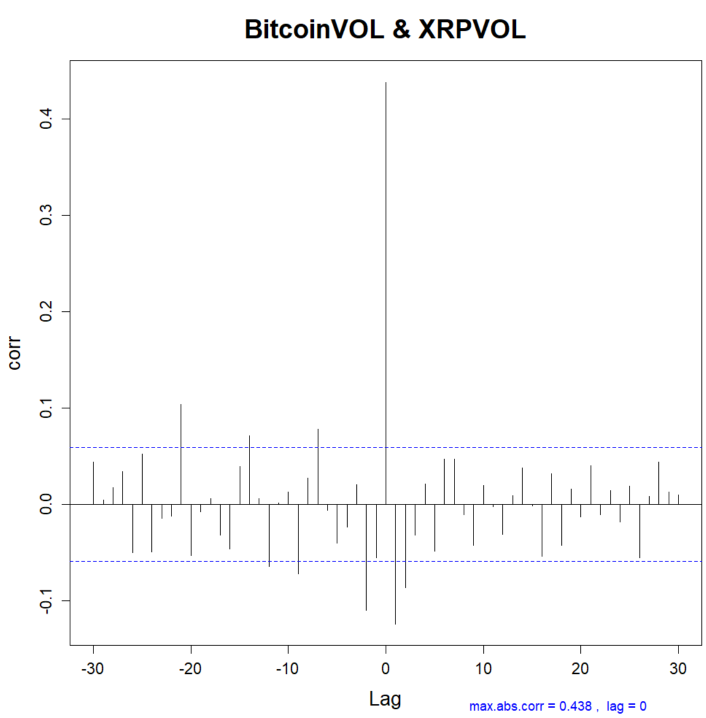

BitcoinVOL & NVDAVOL
corr
0.05
0.00
-0.05
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.076 , lag = 22


BitcoinVOL & pc1
corr
0.6
0.4
0.2
0.0
-0.2
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.741 , lag = 0

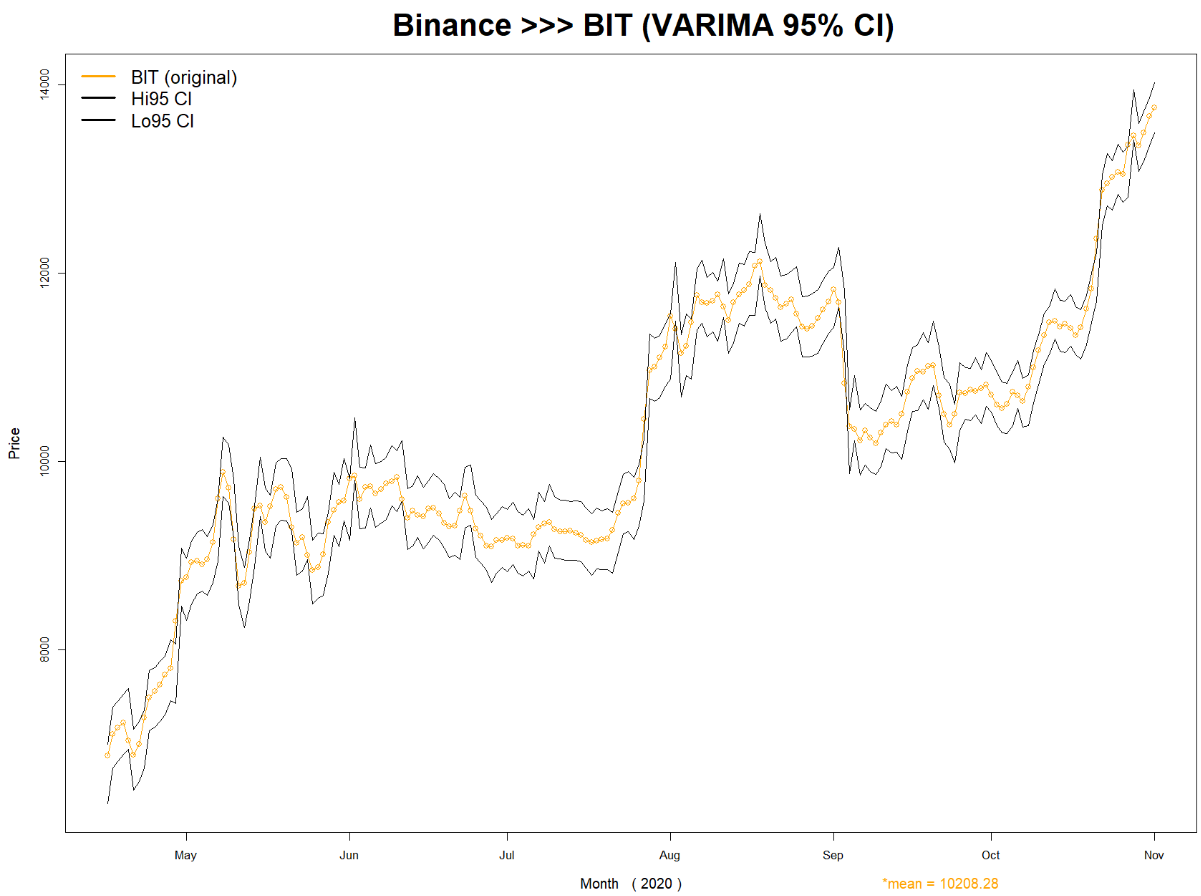

BitcoinVOL & Tether
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = -0.047 , lag = 2


BitcoinVOL & TetherVOL
corr
0.8
0.6
0.4
0.2
0.0
-0.2
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.786 , lag = 0

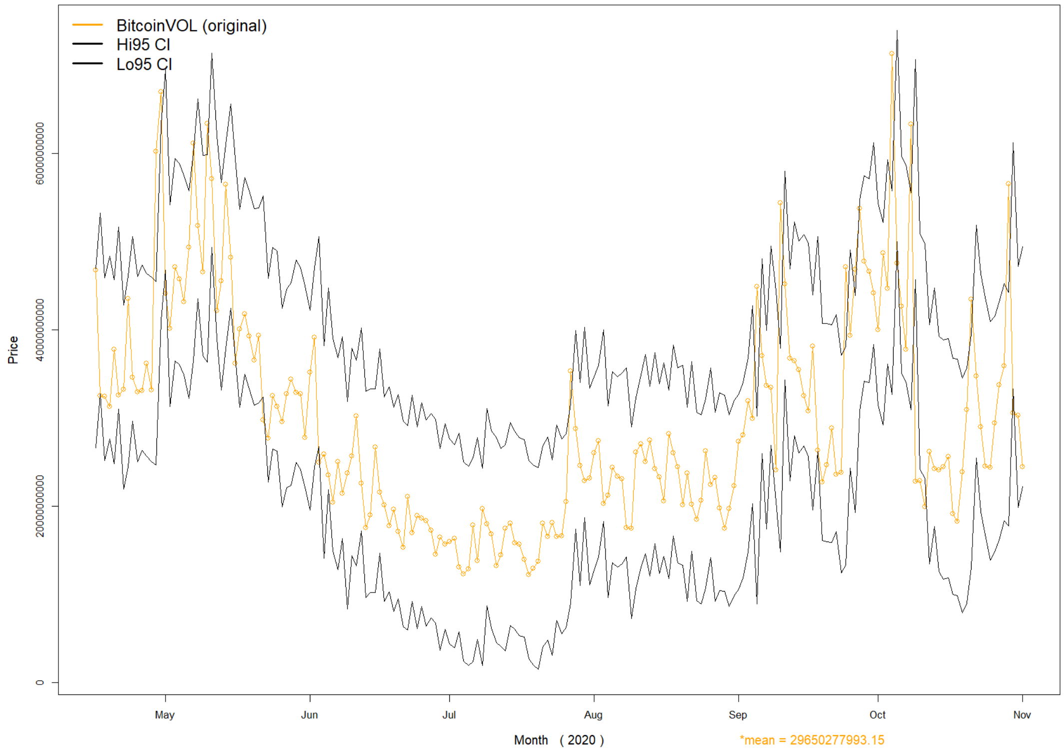

BitcoinVOL & XRP
corr
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.051 , lag = -16


BitcoinVOL & XRPVOL
corr
0.4
0.3
0.2
0.1
0.0
-0.1
-30
-20
-10
0
10
20
30
Lag
max.abs.corr = 0.438 , lag = 0

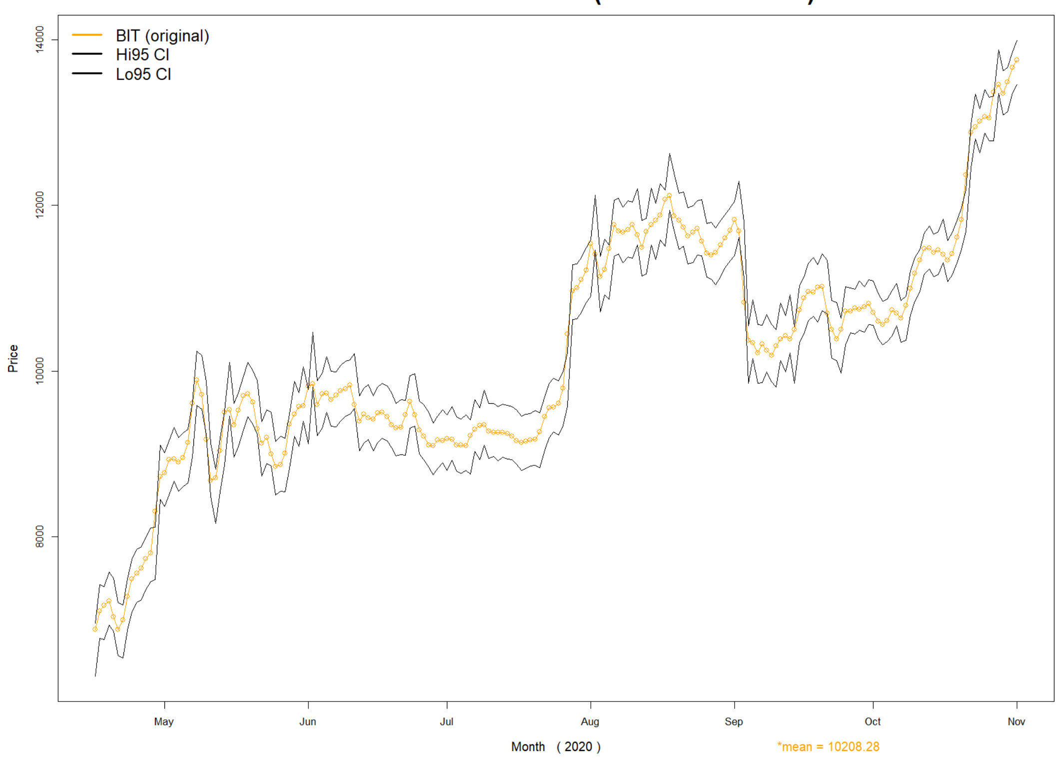
A.3 Forecasted VARIMA Confidence Intervals

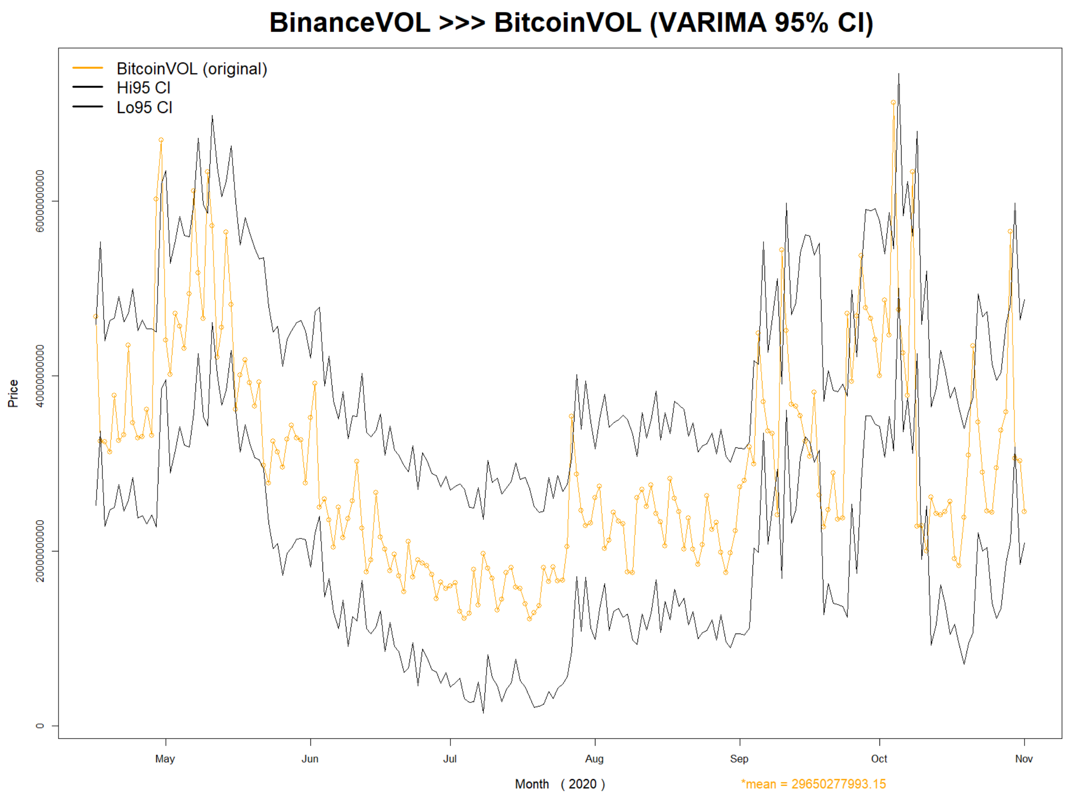


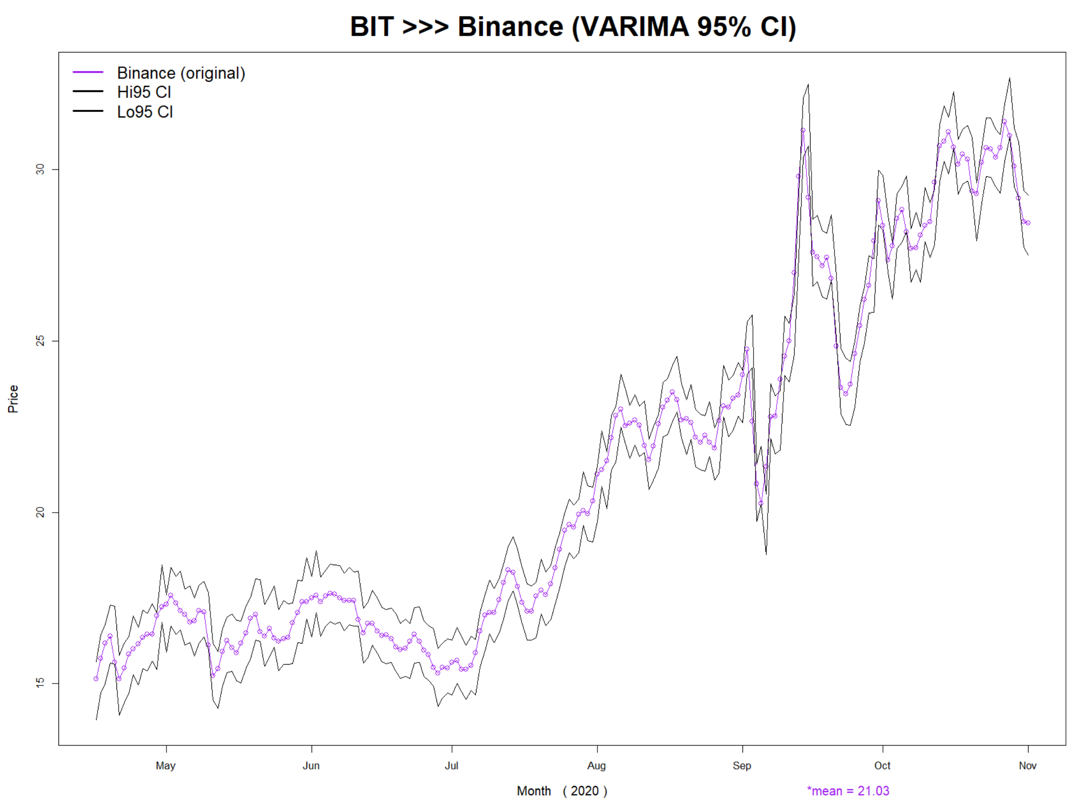
Binance >>> BitcoinVOL (VARIMA 95% CI)

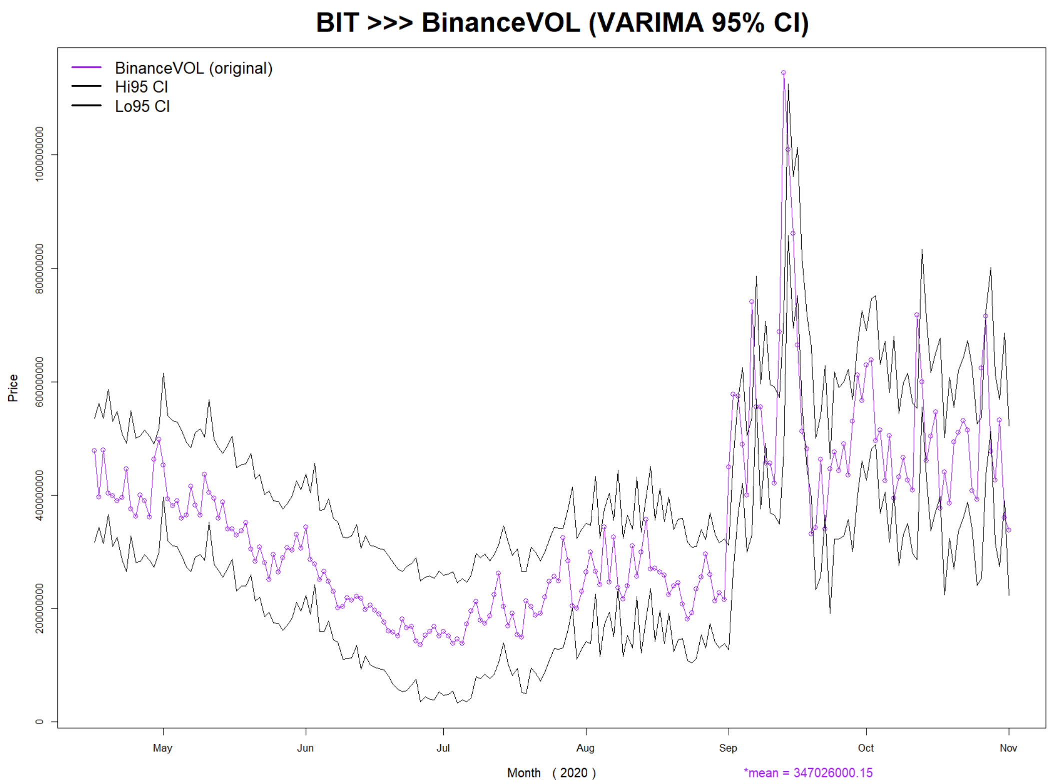


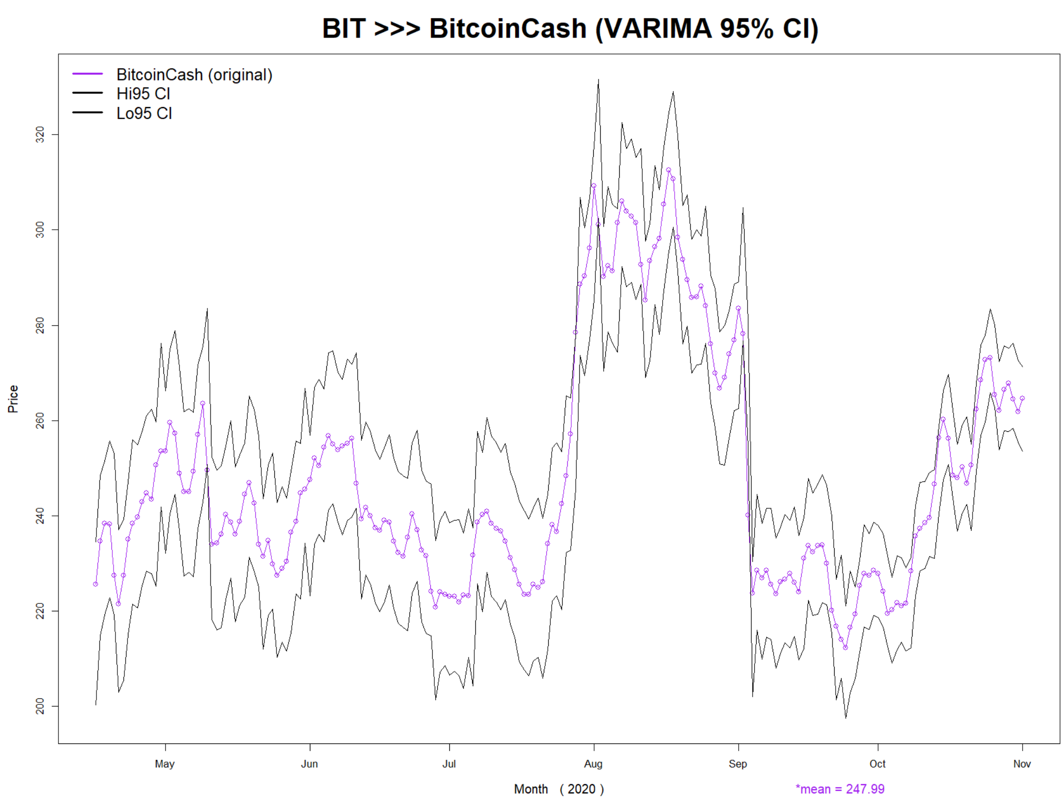
BinanceVOL >>> BIT (VARIMA 95% CI)

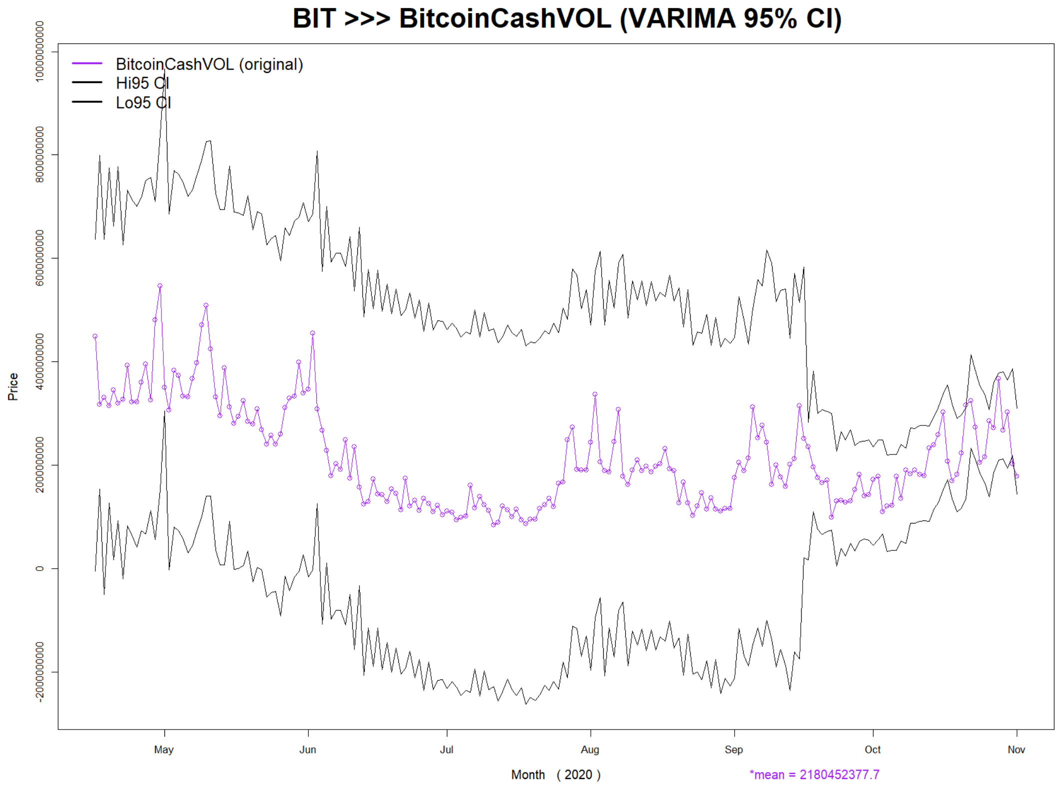



BinanceVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

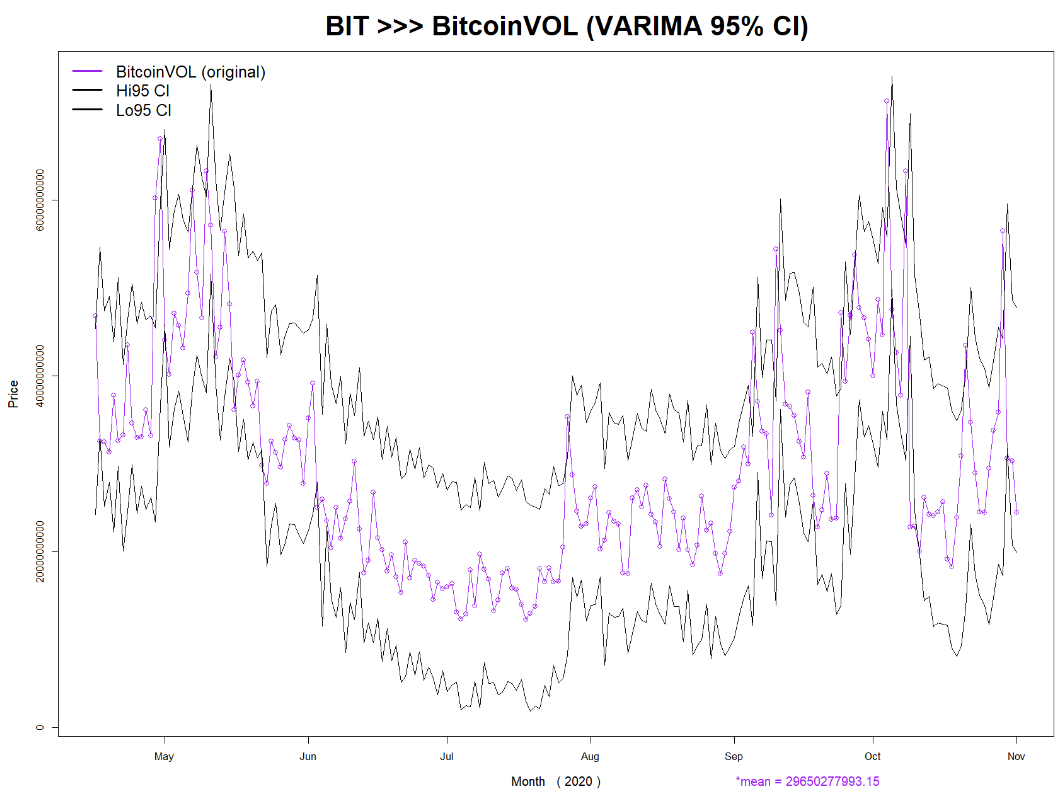

BIT >>> Binance (VARIMA 95% CI)
Binance (original)
Hi95 CI
Lo95 CI
Price
15
20
25
30
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 21.03


BIT >>> BinanceVOL (VARIMA 95% CI)
BinanceVOL (original)
Hi95 CI
Lo95 CI
Price
0
200000000
400000000
600000000
800000000
1000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 347026000.15

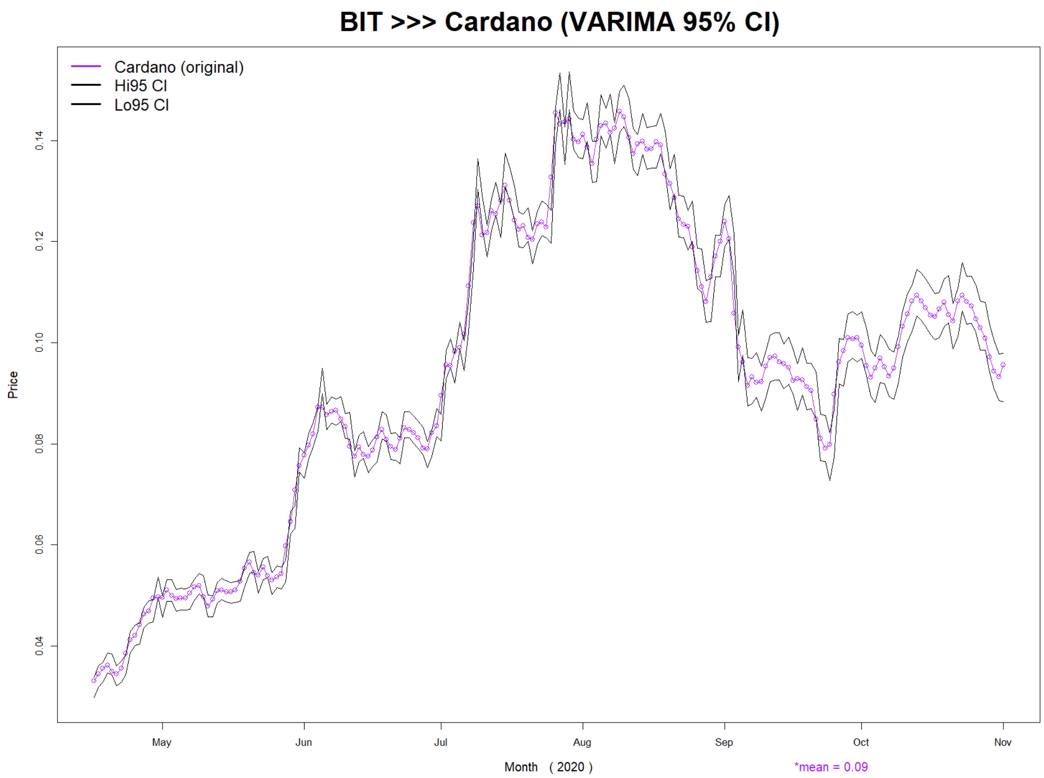

BIT >>> BitcoinCash (VARIMA 95% CI)
BitcoinCash (original)
Hi95 CI
Lo95 CI
Price
200
220
240
260
280
300
320
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 247.99

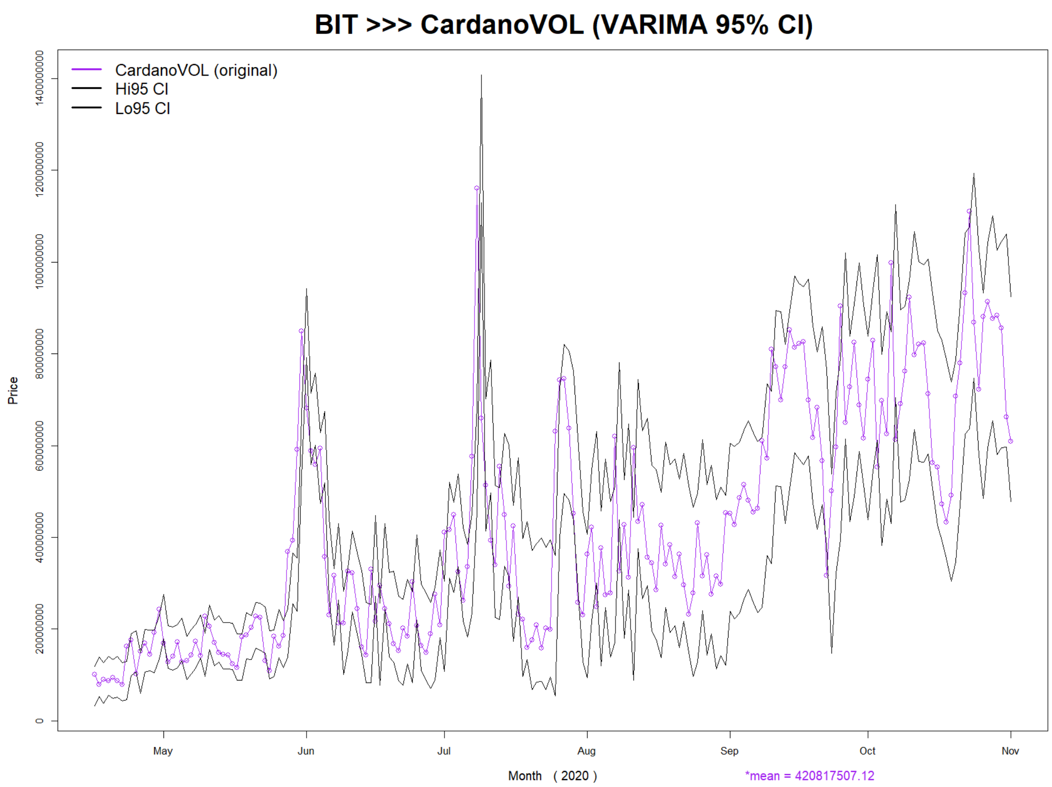


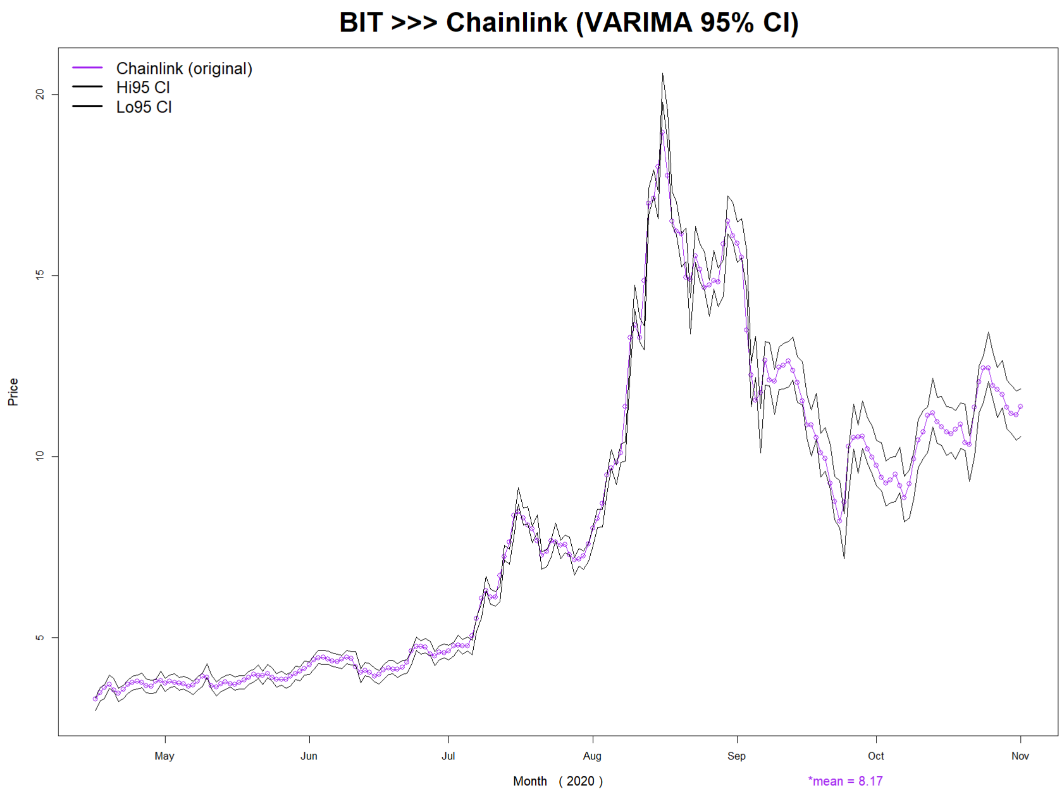
BIT >>> BitcoinCashVOL (VARIMA 95% CI)
BitcoinCashVOL (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 2180452377.7

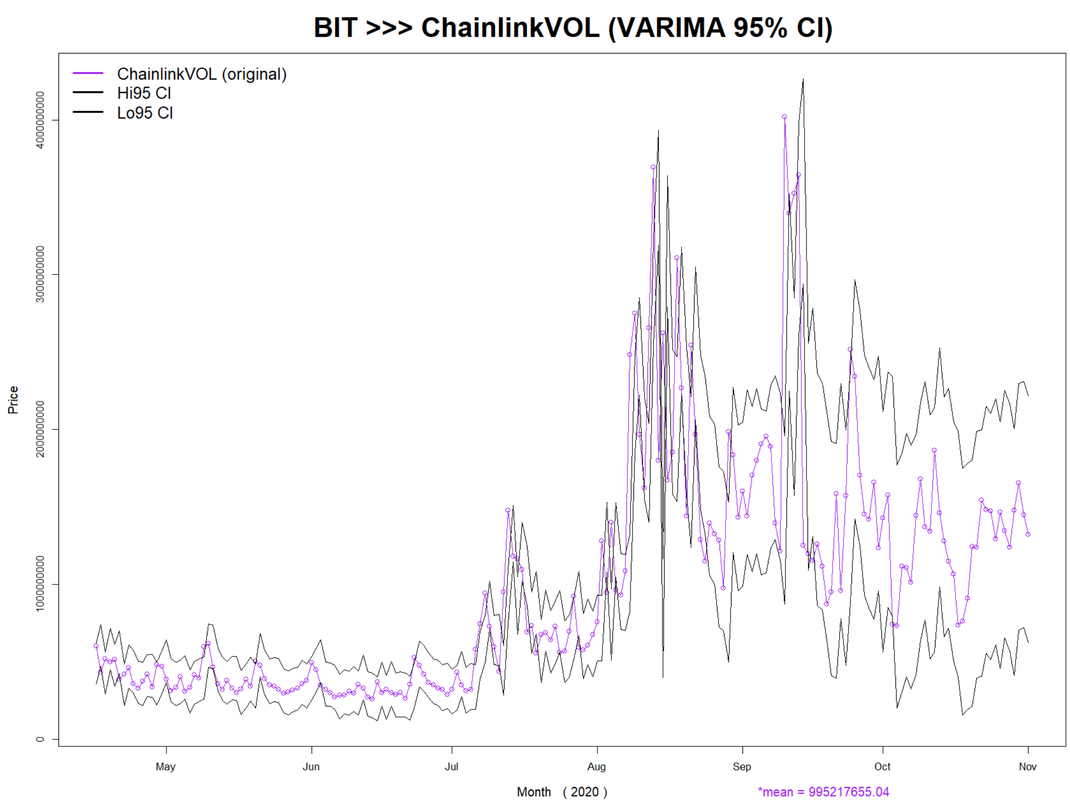


BIT >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 29650277993.15


BIT >>> Cardano (VARIMA 95% CI)
Cardano (original)
Hi95 CI
Lo95 CI
Price
0.04
0.06
0.08
0.10
0.12
0.14
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 0.09

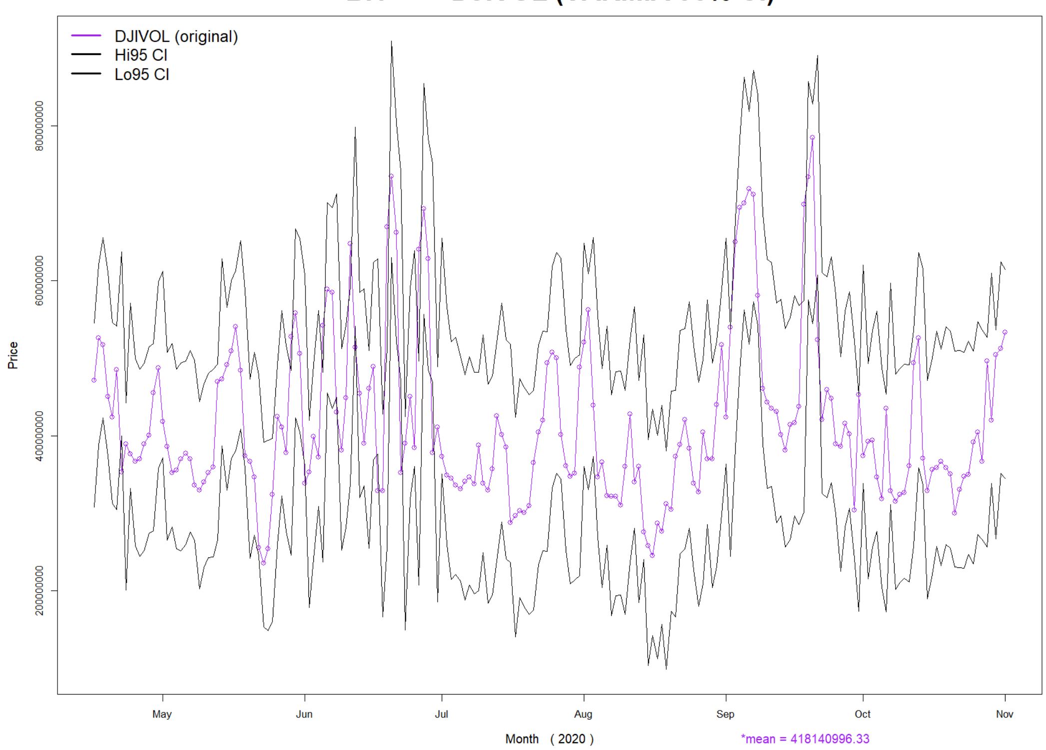

BIT >>> CardanoVOL (VARIMA 95% CI)
CardanoVOL (original)
Hi95 CI
Lo95 CI
Price
0
200000000
400000000
600000000
800000000
1000000000
1200000000
1400000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 420817507.12

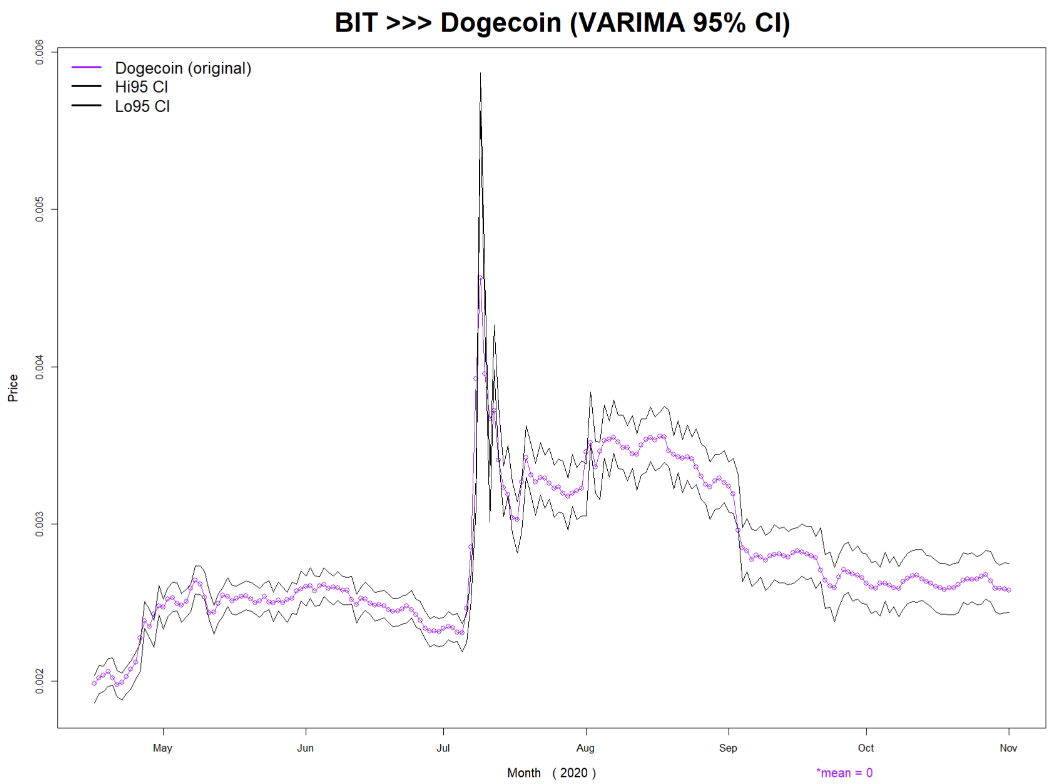

BIT >>> Chainlink (VARIMA 95% CI)
Chainlink (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 8.17

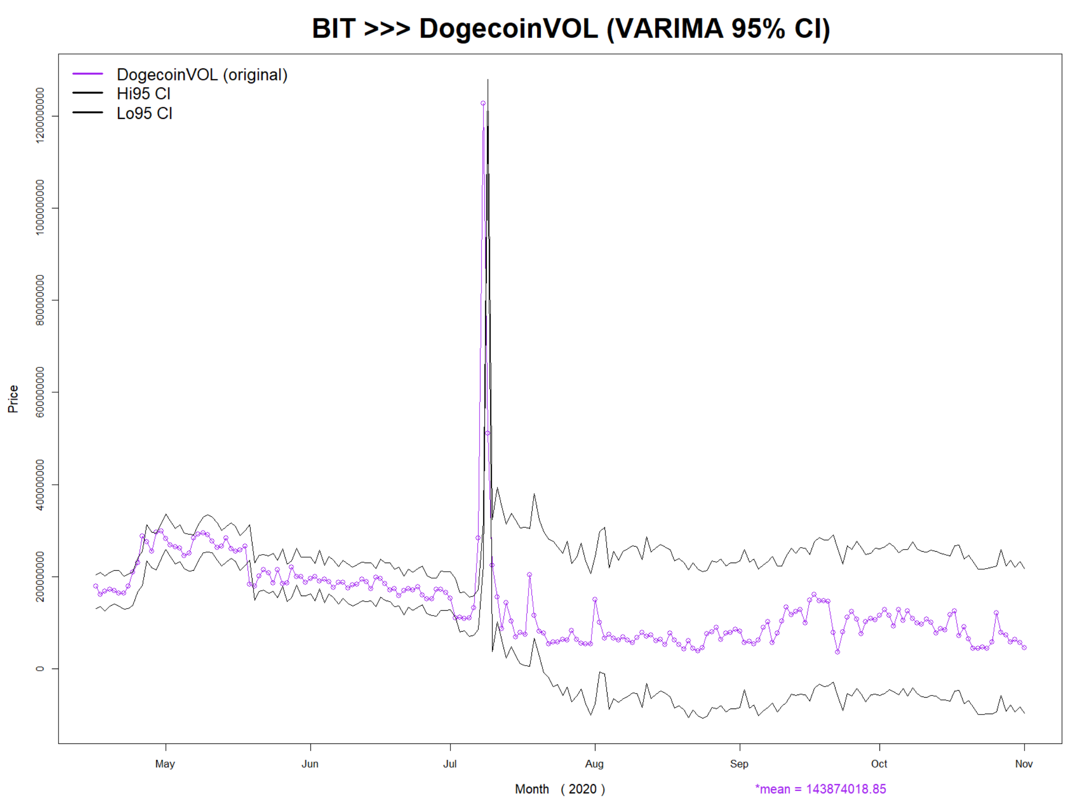

BIT >>> ChainlinkVOL (VARIMA 95% CI)
ChainlinkVOL (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 995217655.04

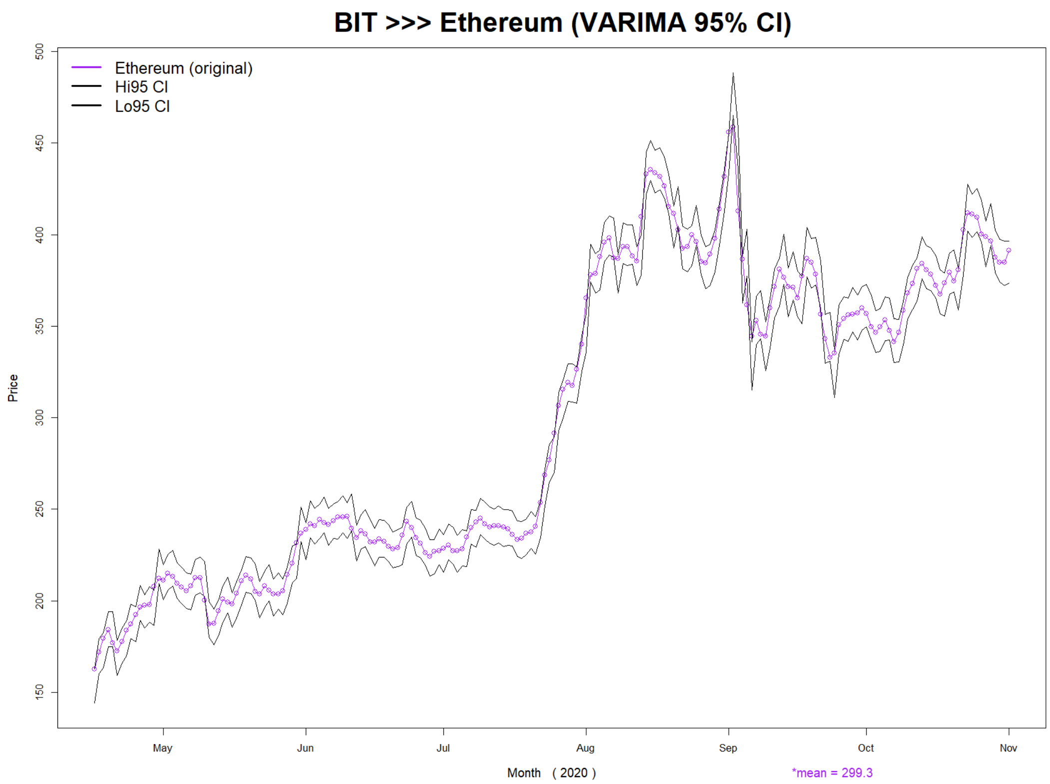
BIT >>> DJI (VARIMA 95% CI)

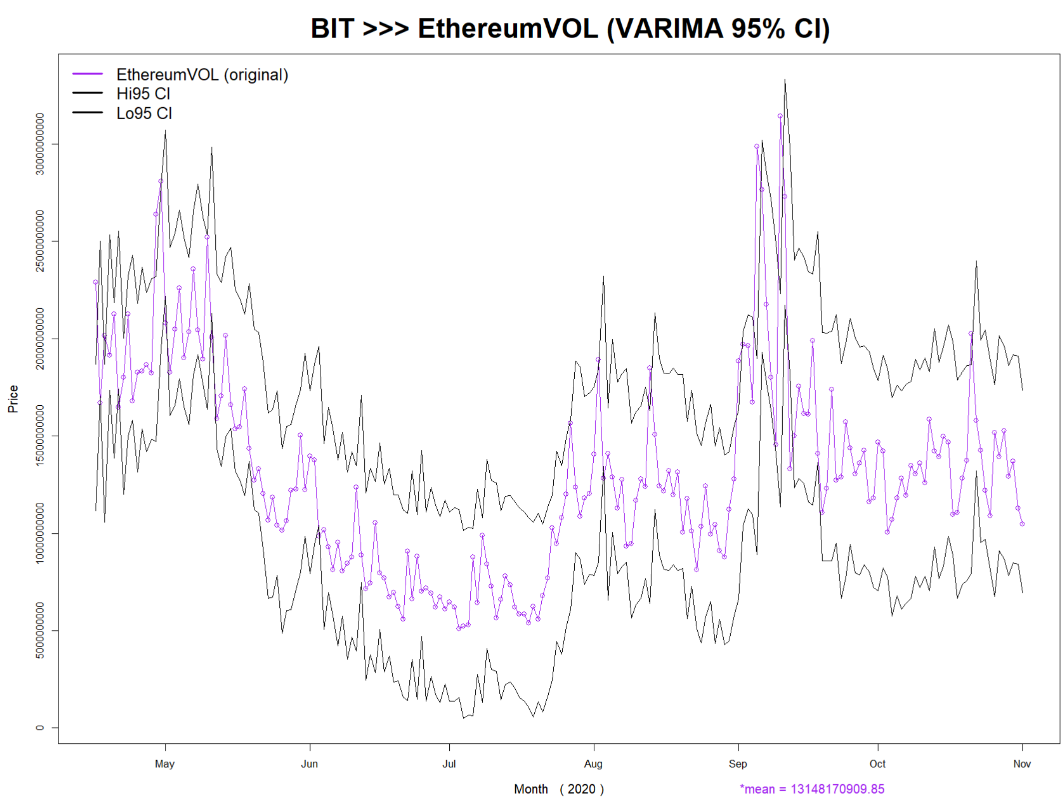
BIT >>> DJIVOL (VARIMA 95% CI)

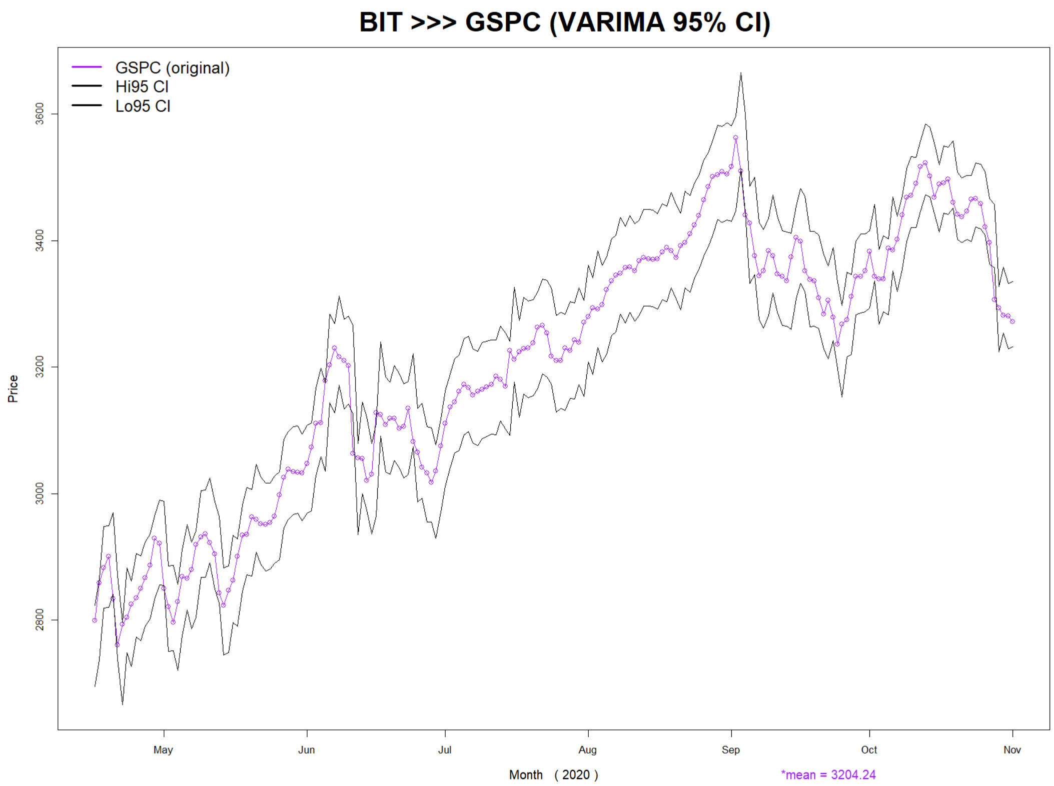


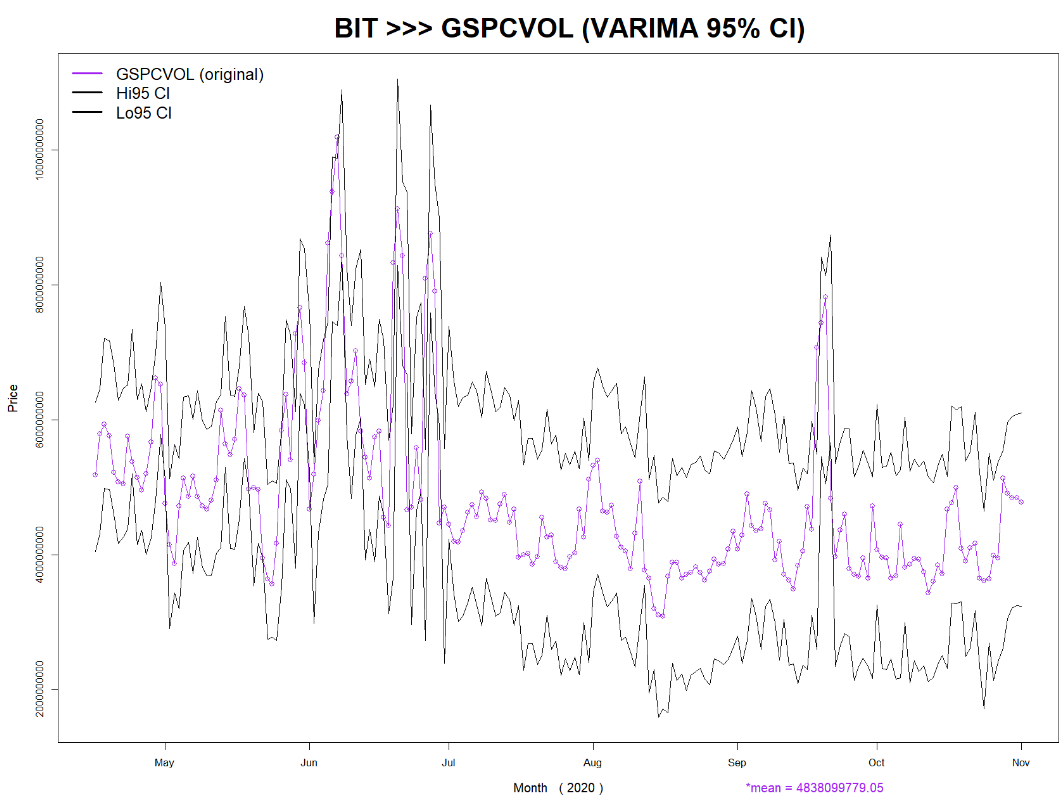

BIT >>> Dogecoin (VARIMA 95% CI)
Dogecoin (original)
Hi95 CI
Lo95 CI
Price
0.006
0.005
0.004
0.003
0.002
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 0


BIT >>> DogecoinVOL (VARIMA 95% CI)
DogecoinVOL (original)
Hi95 CI
Lo95 CI
Price
1200000000
1000000000
800000000
600000000
400000000
200000000
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 143874018.85

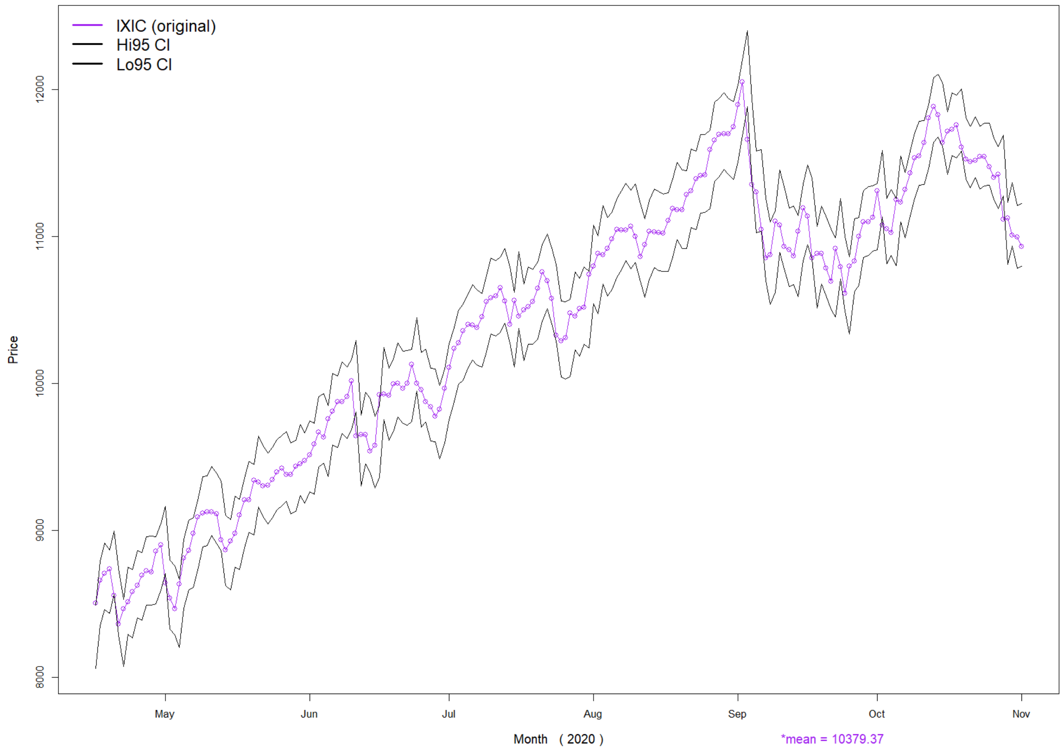

BIT >>> Ethereum (VARIMA 95% CI)
Ethereum (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 299.3


BIT >>> EthereumVOL (VARIMA 95% CI)
EthereumVOL (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 13148170909.85

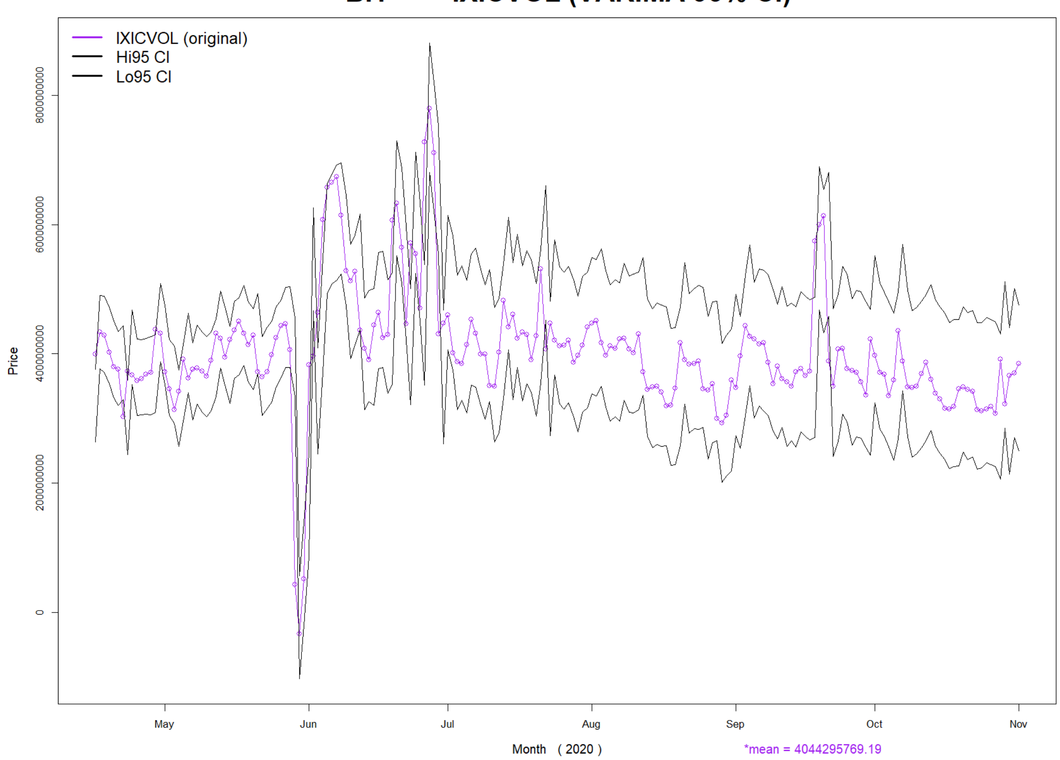

BIT >>> GSPC (VARIMA 95% CI)
GSPC (original)
Hi95 CI
Lo95 CI
Price
3600
3400
3200
3000
2800
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 3204.24

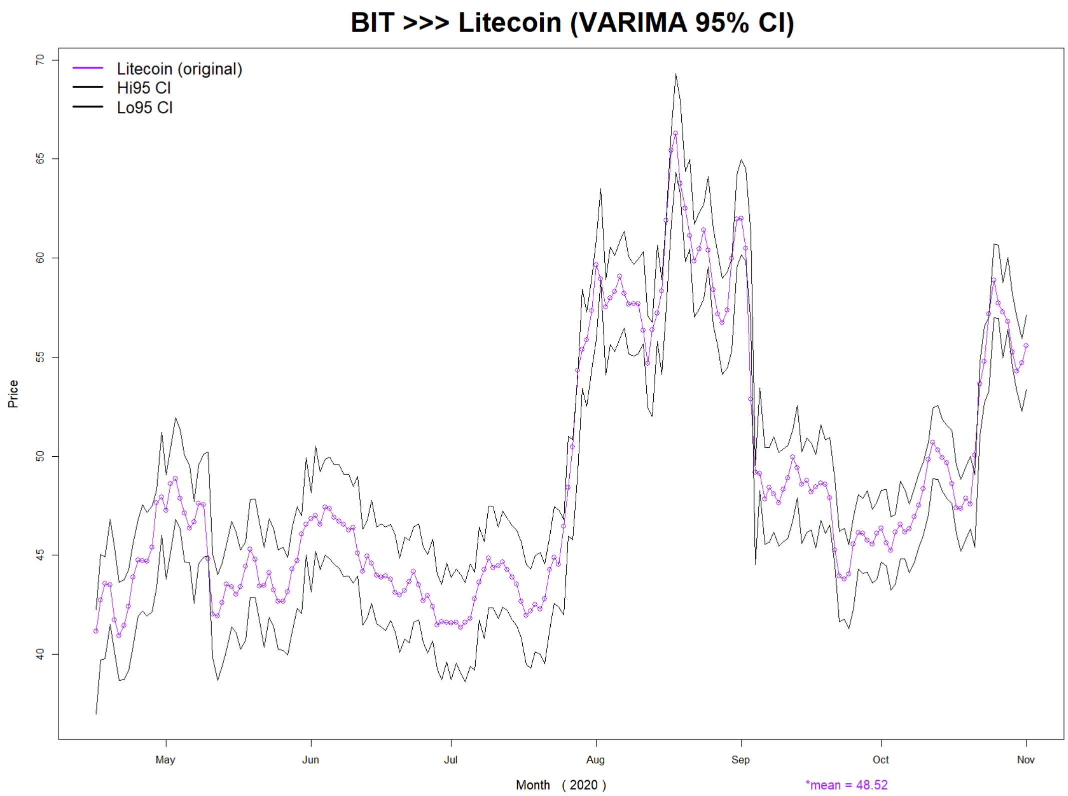

BIT >>> GSPCVOL (VARIMA 95% CI)
GSPCVOL (original)
Hi95 CI
Lo95 CI
Price
10000000000
8000000000
6000000000
4000000000
2000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 4838099779.05

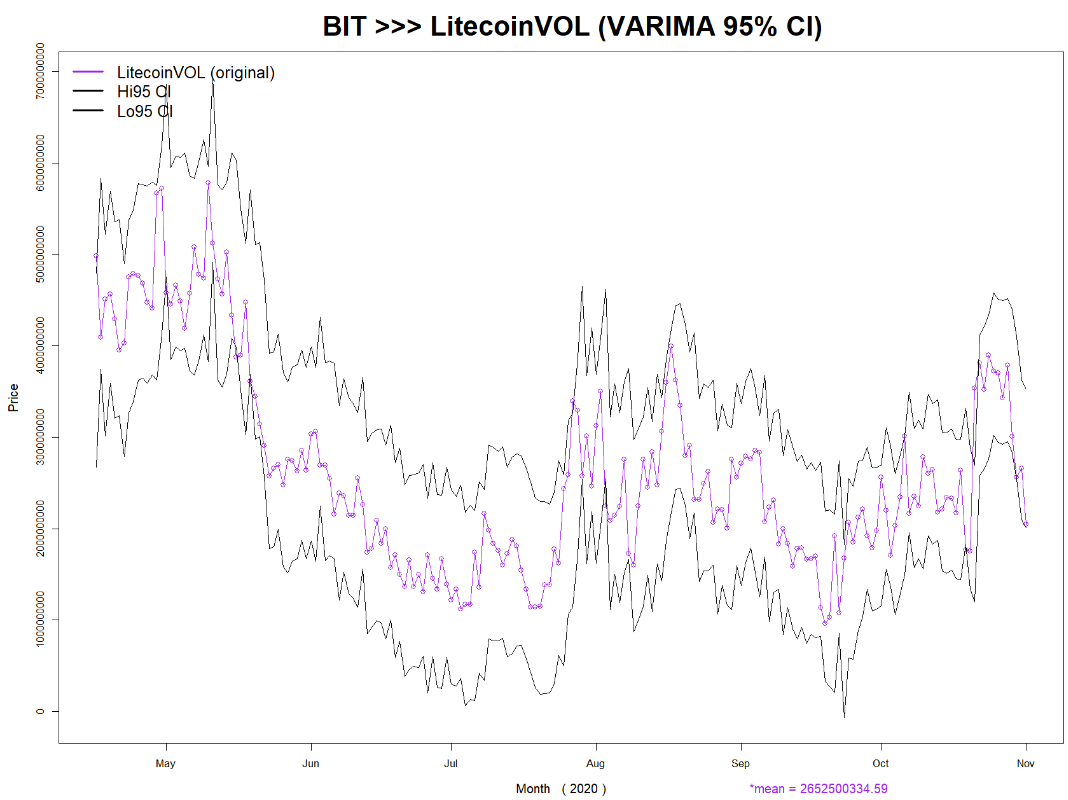
BIT >>> IXIC (VARIMA 95% CI)

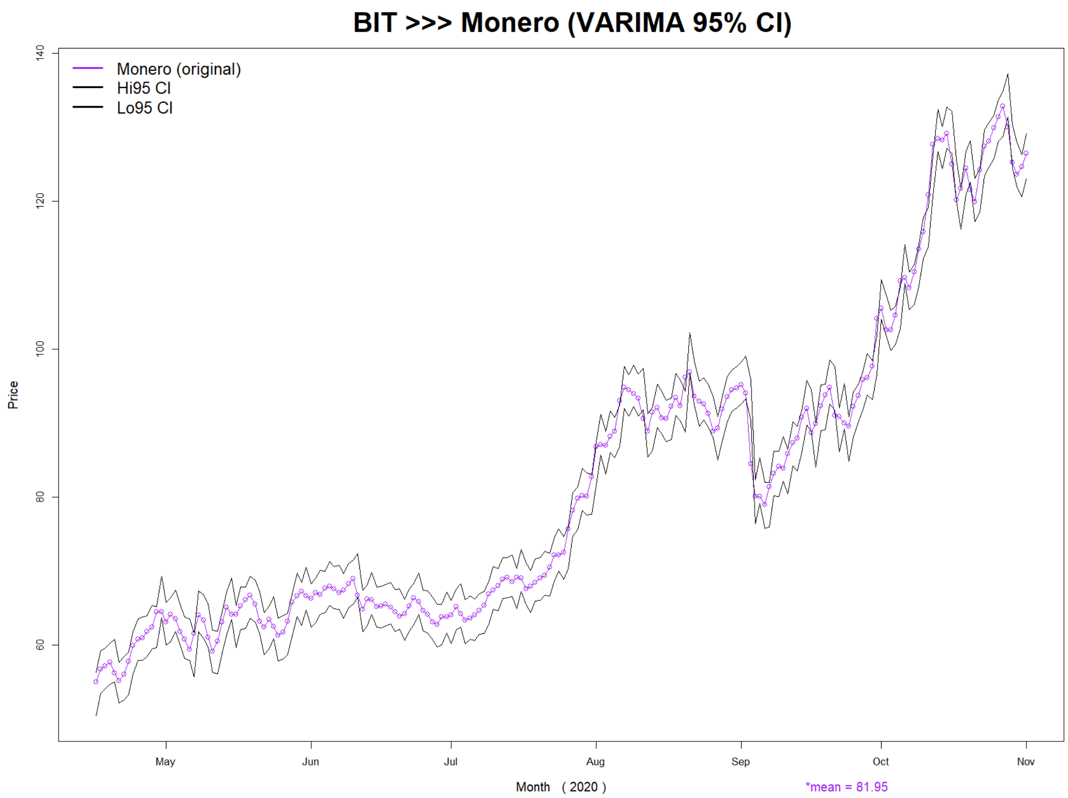


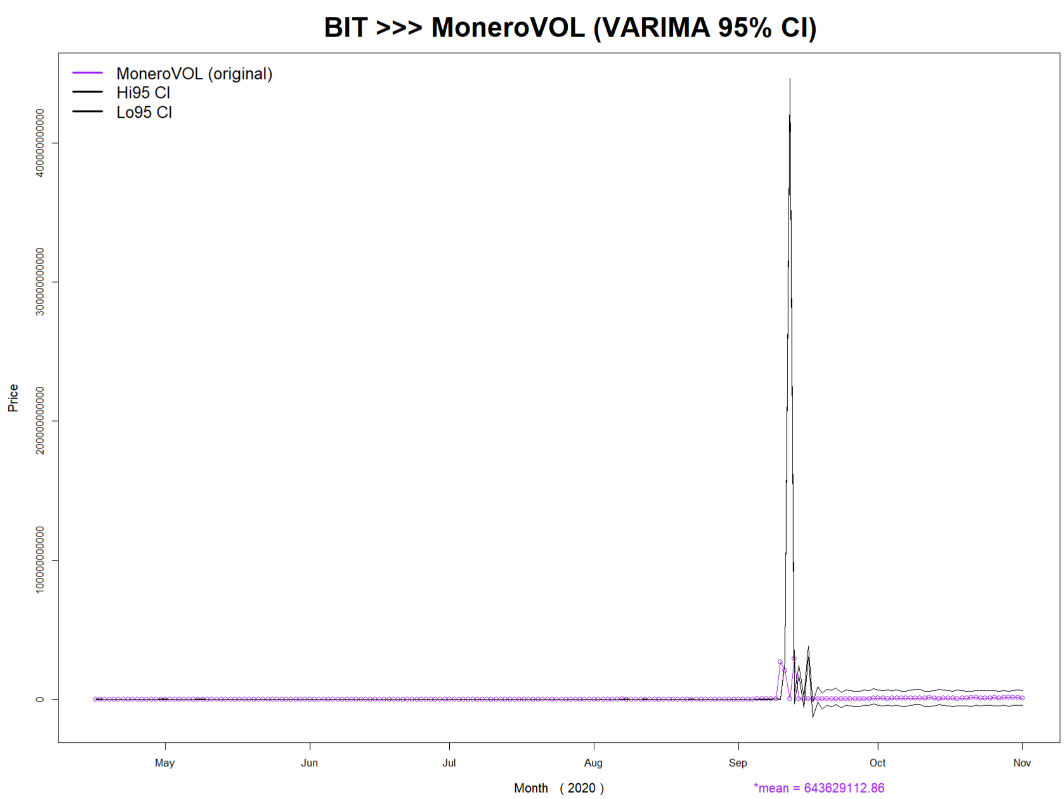
BIT >>> IXICVOL (VARIMA 95% CI)

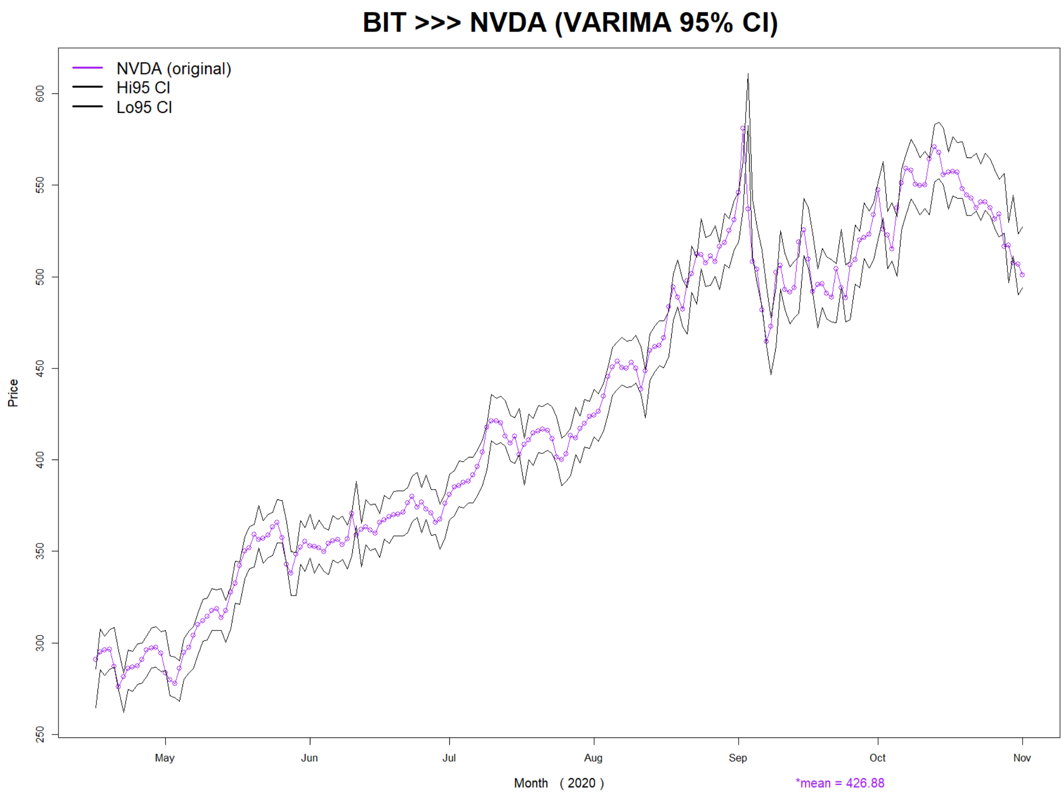


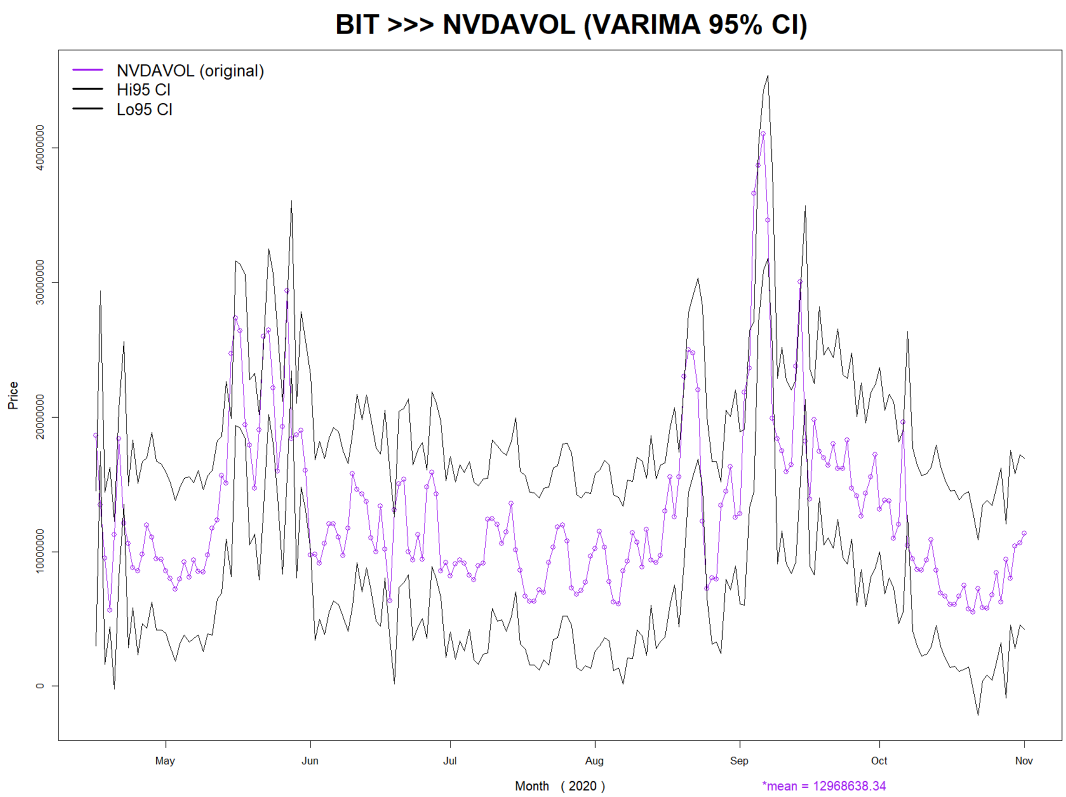

BIT >>> Litecoin (VARIMA 95% CI)
Litecoin (original)
Hi95 CI
Lo95 CI
Price
70
65
60
55
50
45
40
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 48.52

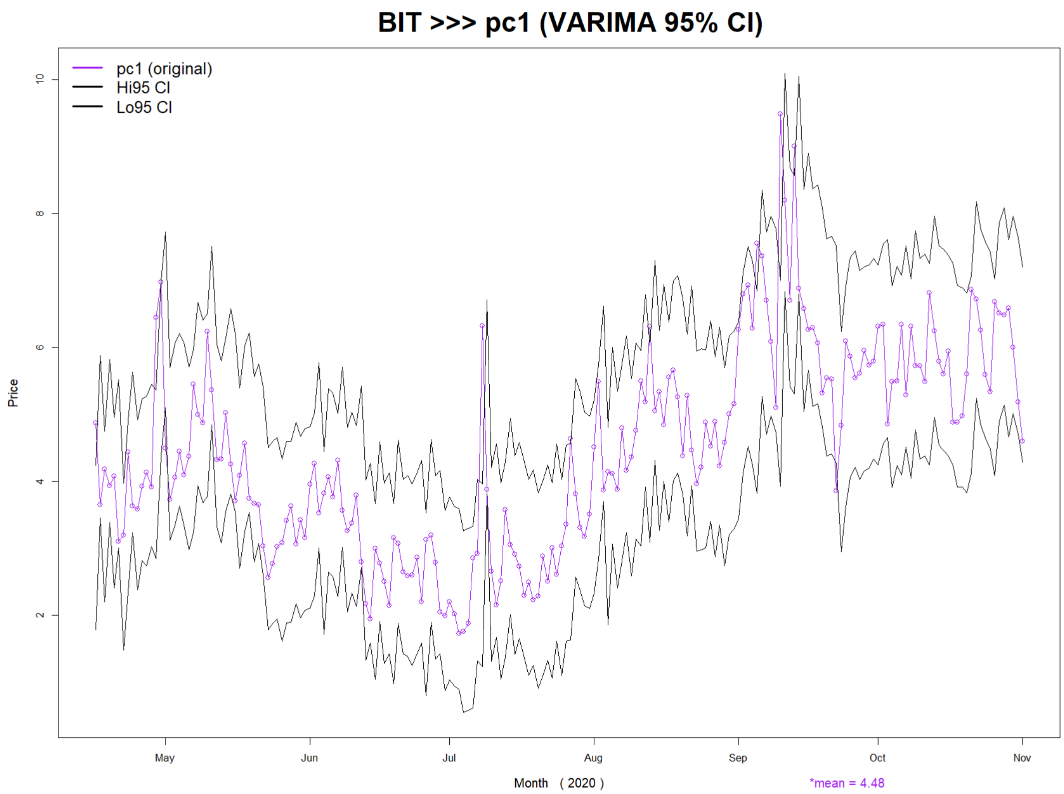

BIT >>> LitecoinVOL (VARIMA 95% CI)
LitecoinVOL (original)
Hi95 CI
Lo95 CI
Price
7000000000
6000000000
5000000000
4000000000
3000000000
2000000000
1000000000
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 2652500334.59

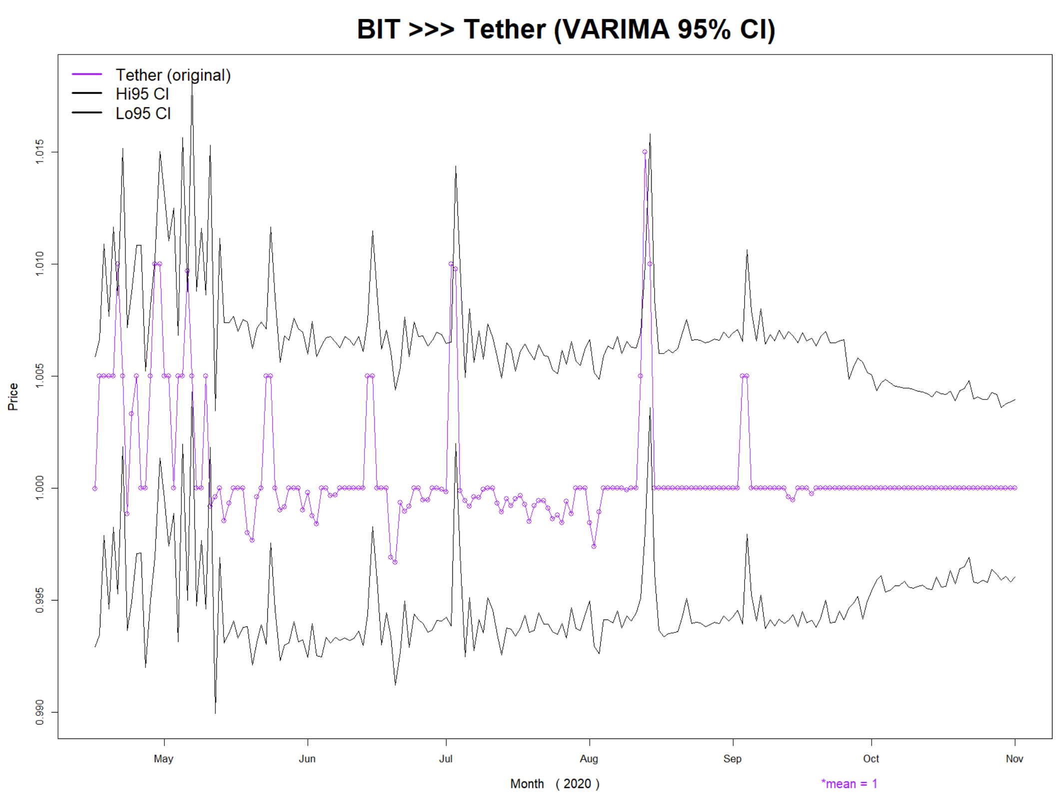

BIT >>> Monero (VARIMA 95% CI)
Monero (original)
Hi95 CI
Lo95 CI
Price
140
120
100
80
60
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 81.95

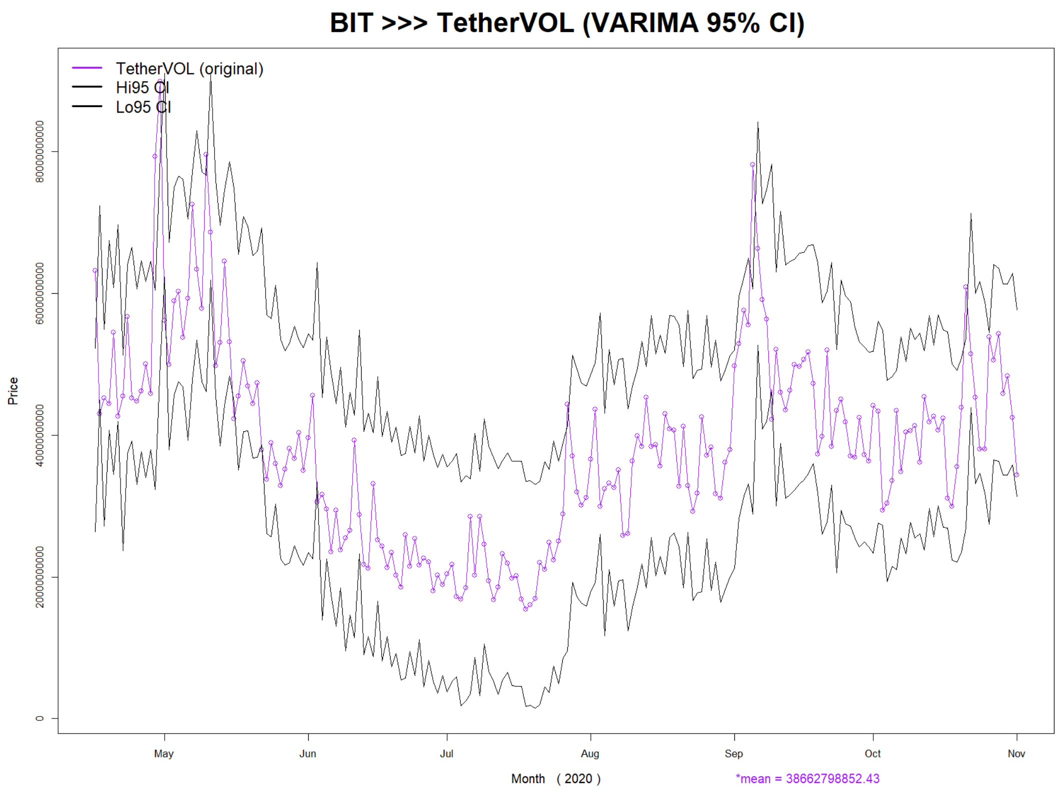

BIT >>> MoneroVOL (VARIMA 95% CI)
MoneroVOL (original)
Hi95 CI
Lo95 CI
Price
400000000000
300000000000
200000000000
100000000000
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 643629112.86

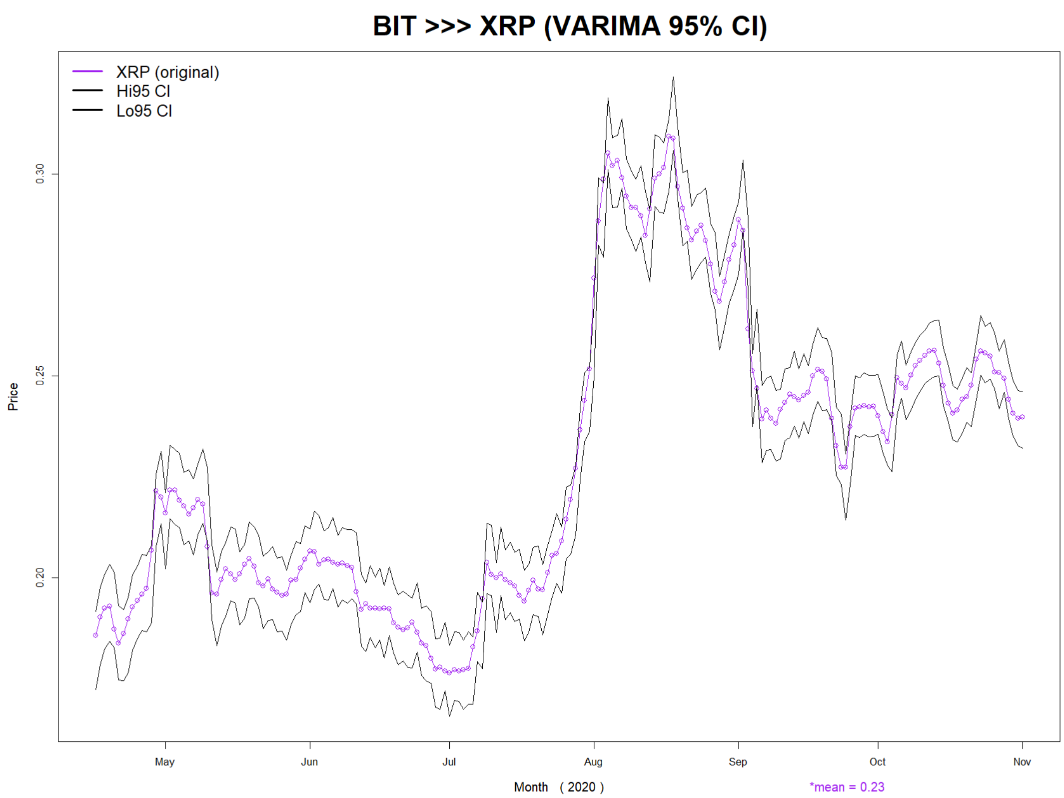

BIT >>> NVDA (VARIMA 95% CI)
NVDA (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 426.88

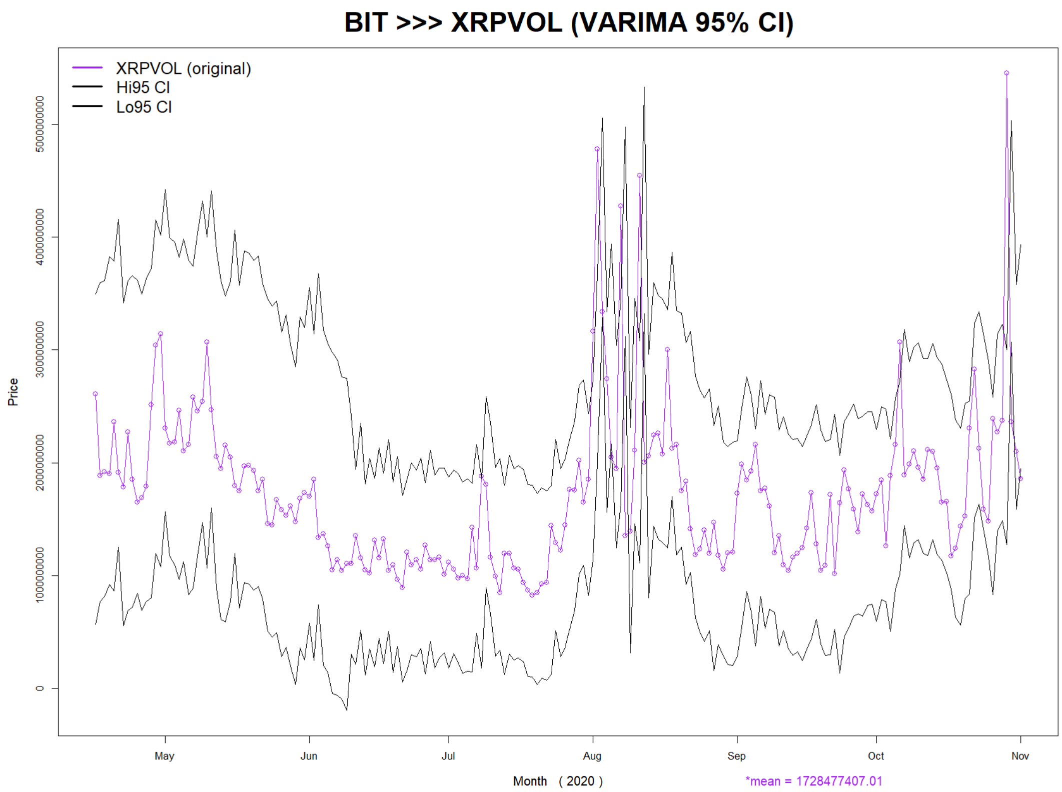

BIT >>> NVDAVOL (VARIMA 95% CI)
NVDAVOL (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 12968638.34

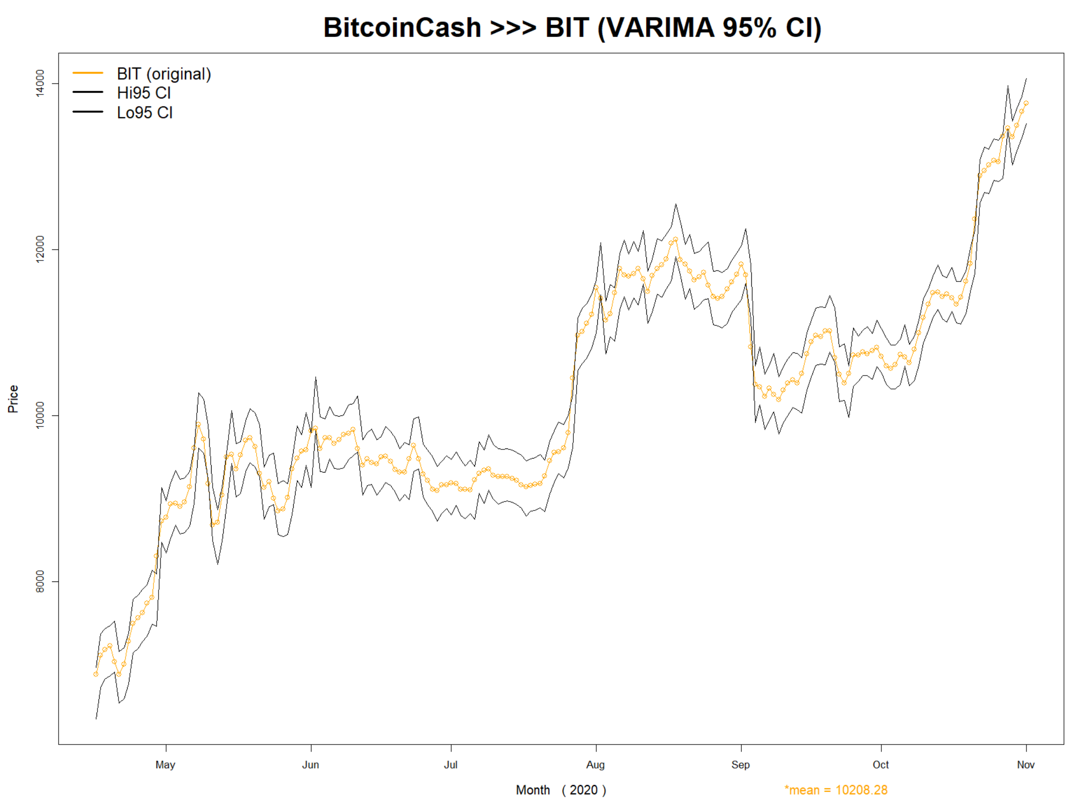

BIT >>> pc1 (VARIMA 95% CI)
pc1 (original)
Hi95 CI
Lo95 CI
Price
10
8
6
4
2
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 4.48

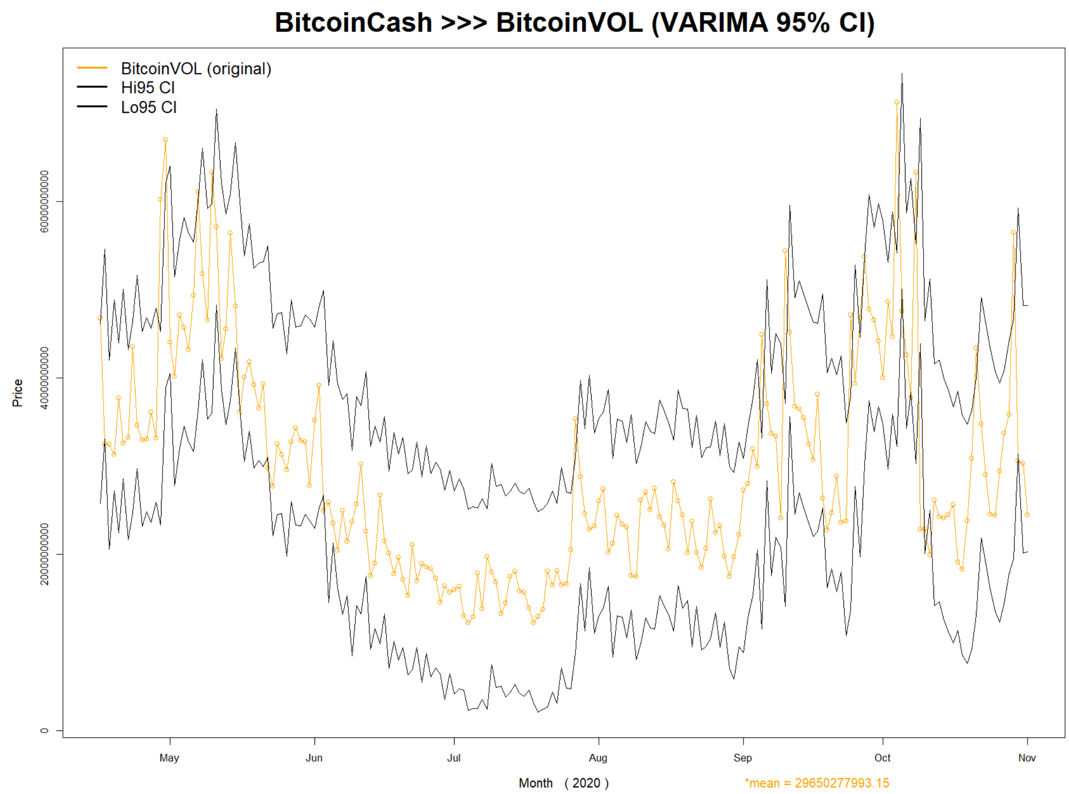

BIT >>> Tether (VARIMA 95% CI)
Tether (original)
Hi95 CI
Lo95 CI
Price
1.015
1.010
1.005
1.000
0.995
0.990
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 1

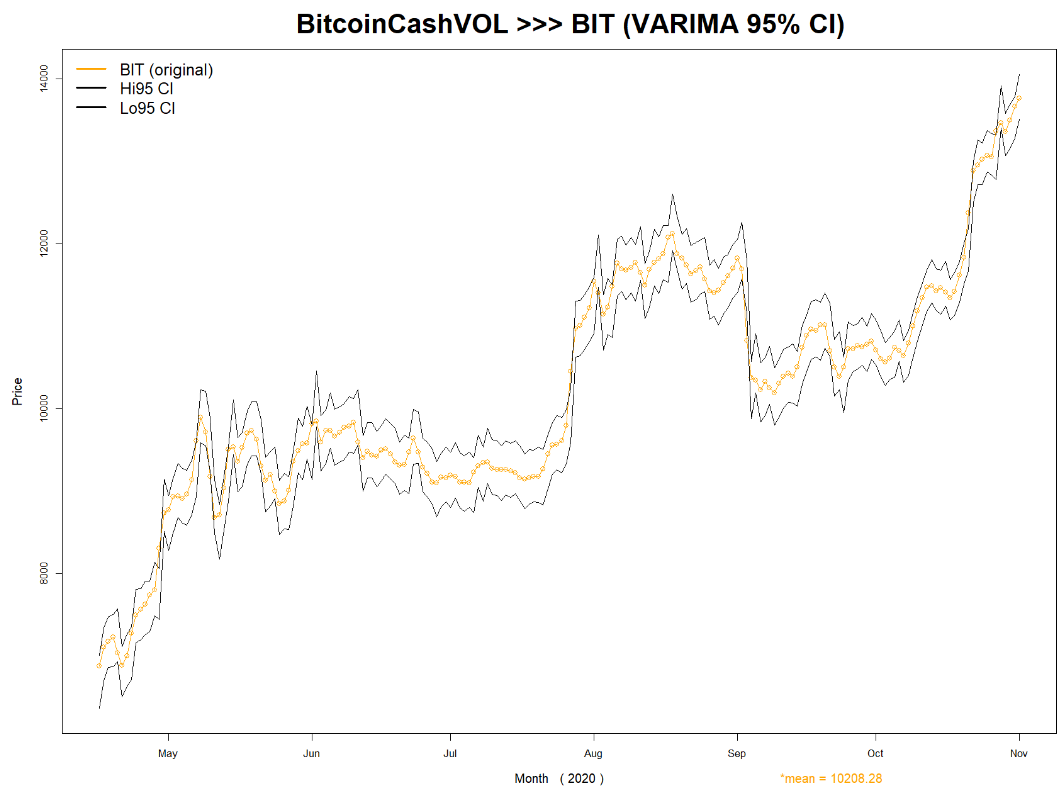

BIT >>> TetherVOL (VARIMA 95% CI)
TetherVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
80000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 38662798852.43


BIT >>> XRP (VARIMA 95% CI)
XRP (original)
Hi95 CI
Lo95 CI
Price
0.20
0.25
0.30
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 0.23

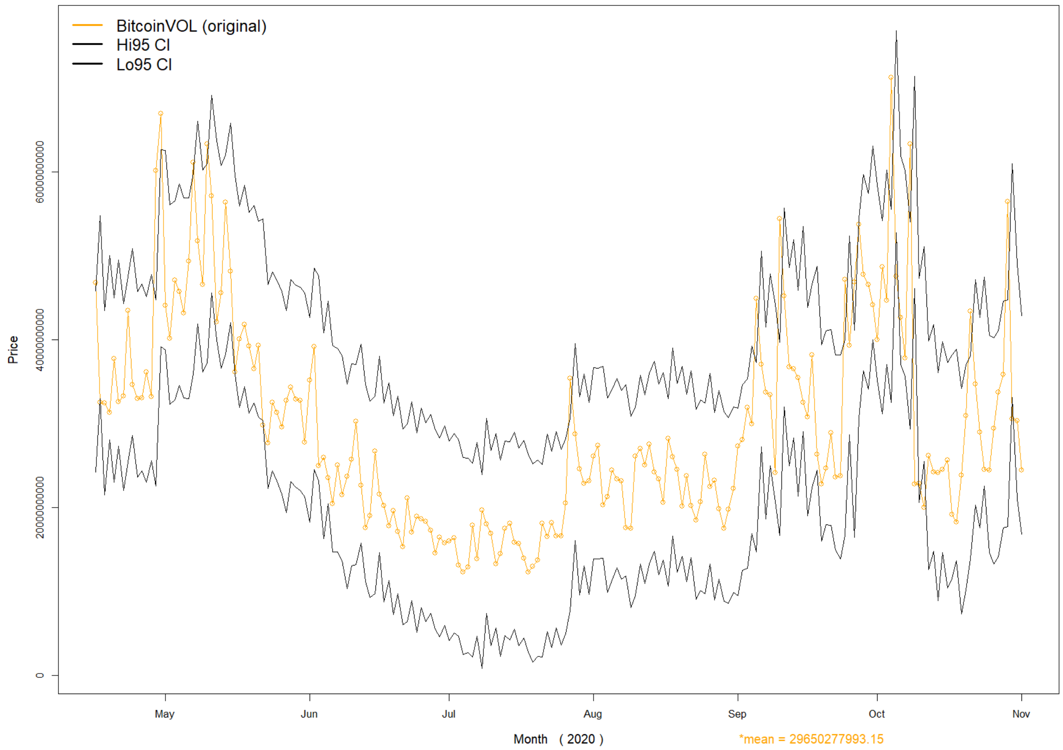

BIT >>> XRPVOL (VARIMA 95% CI)
XRPVOL (original)
Hi95 CI
Lo95 CI
Price
0
1000000000
2000000000
3000000000
4000000000
5000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 1728477407.01


BitcoinCash >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

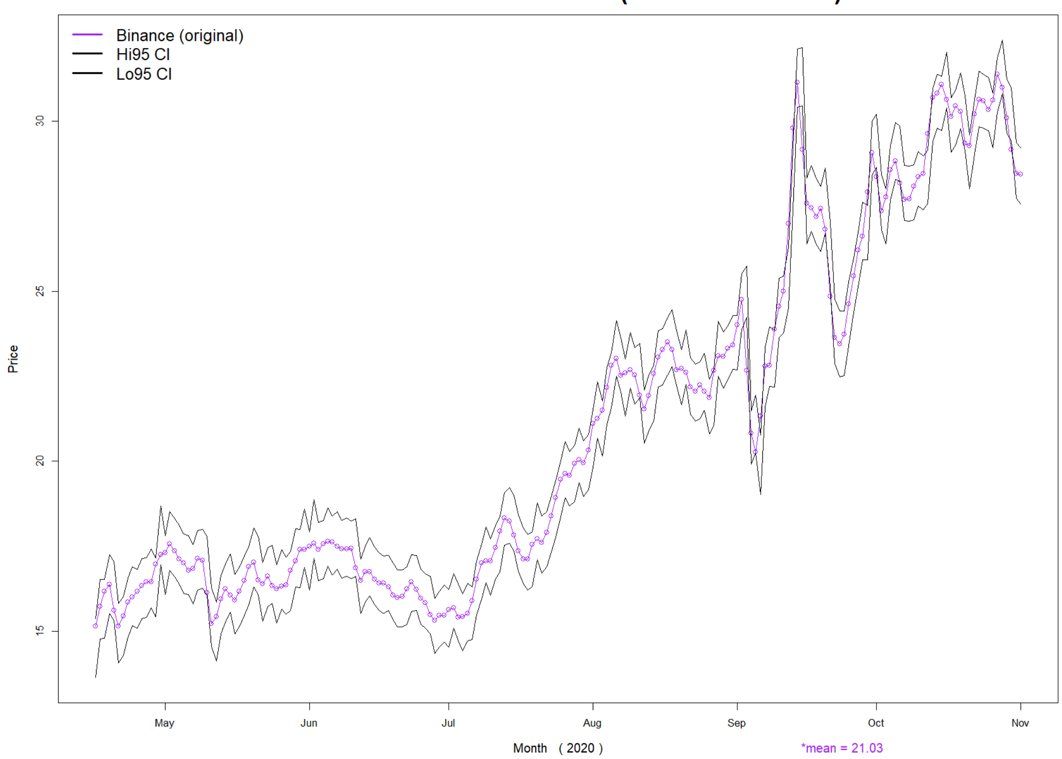

BitcoinCash >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

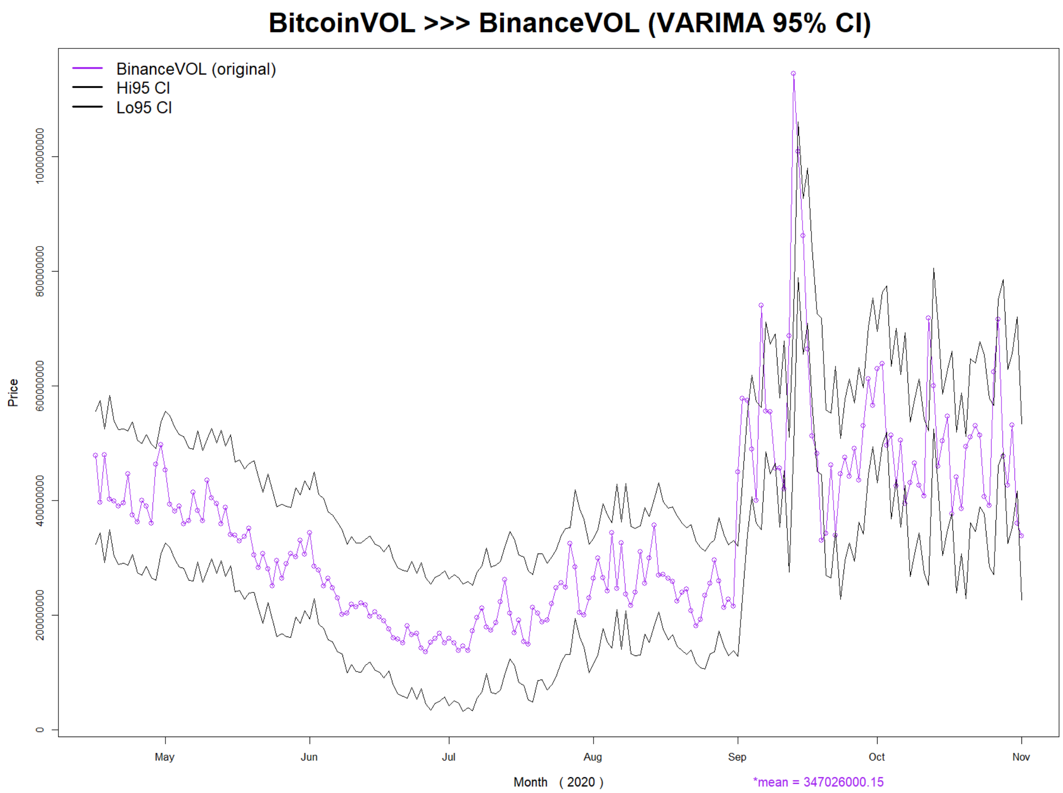

BitcoinCashVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

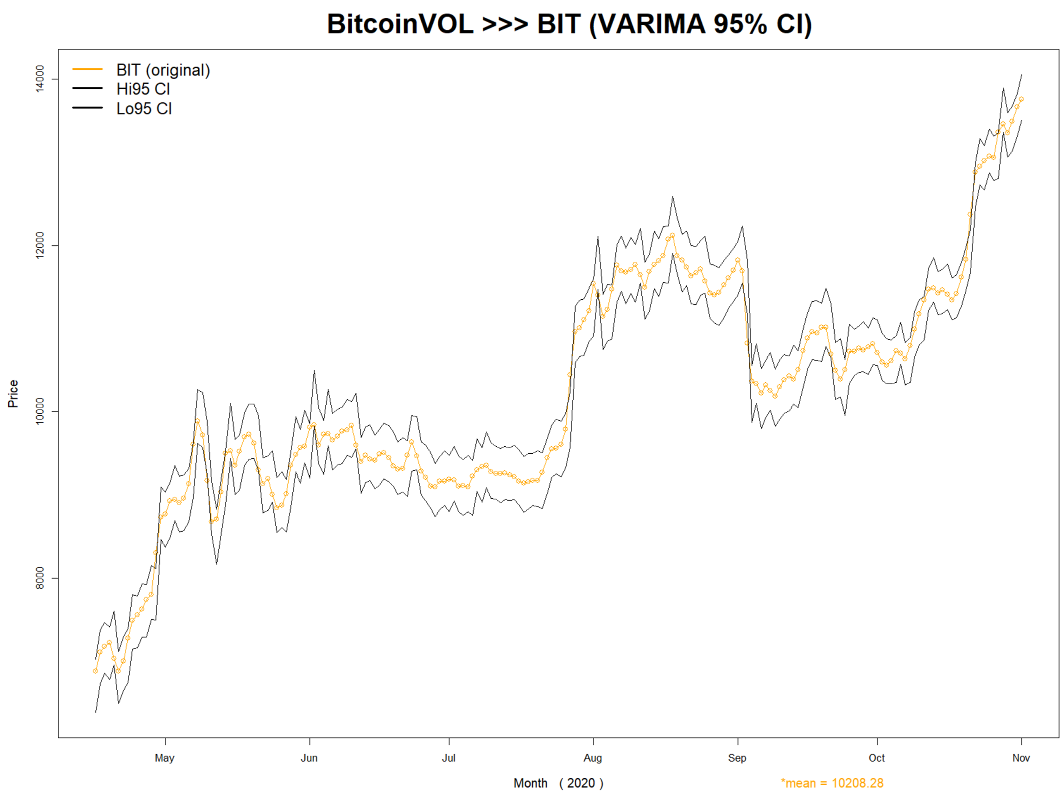
BitcoinCashVOL >>> BitcoinVOL (VARIMA 95% CI)

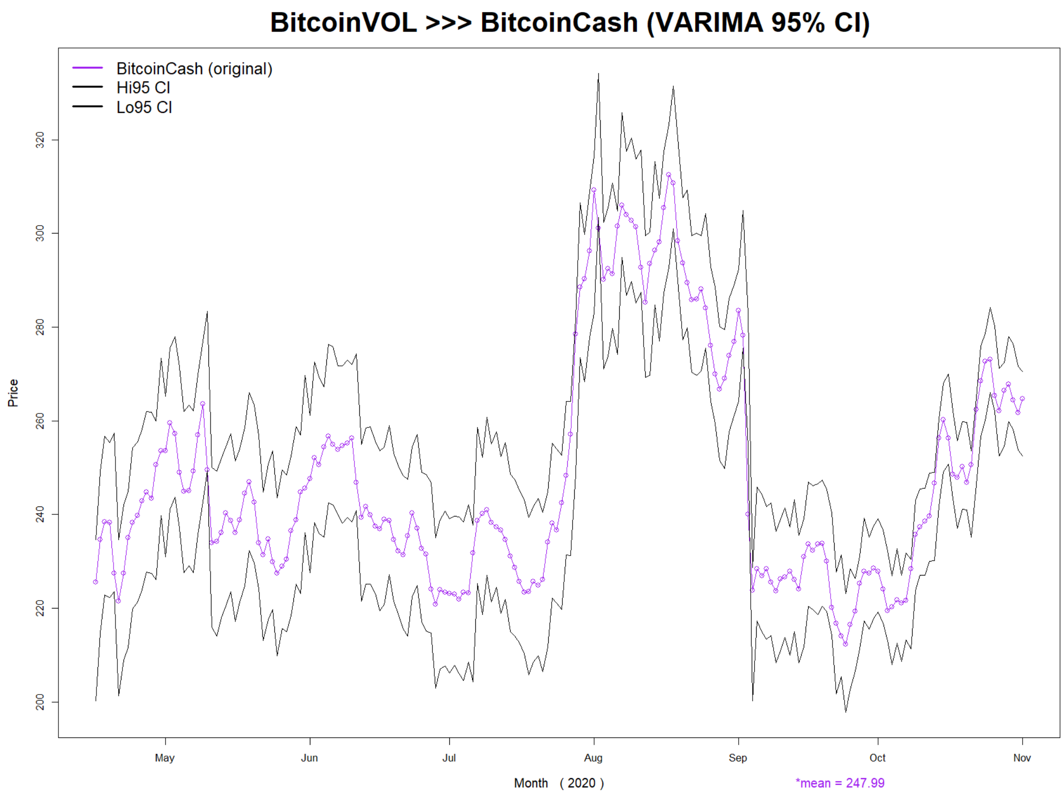


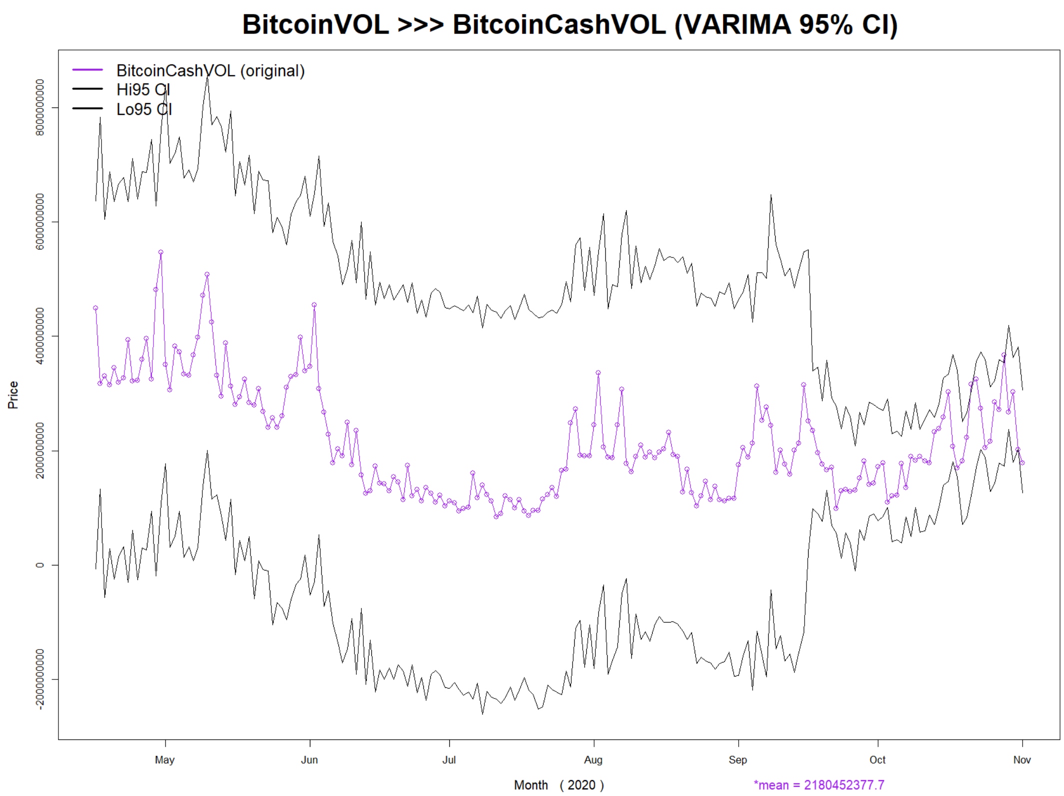
BitcoinVOL >>> Binance (VARIMA 95% CI)

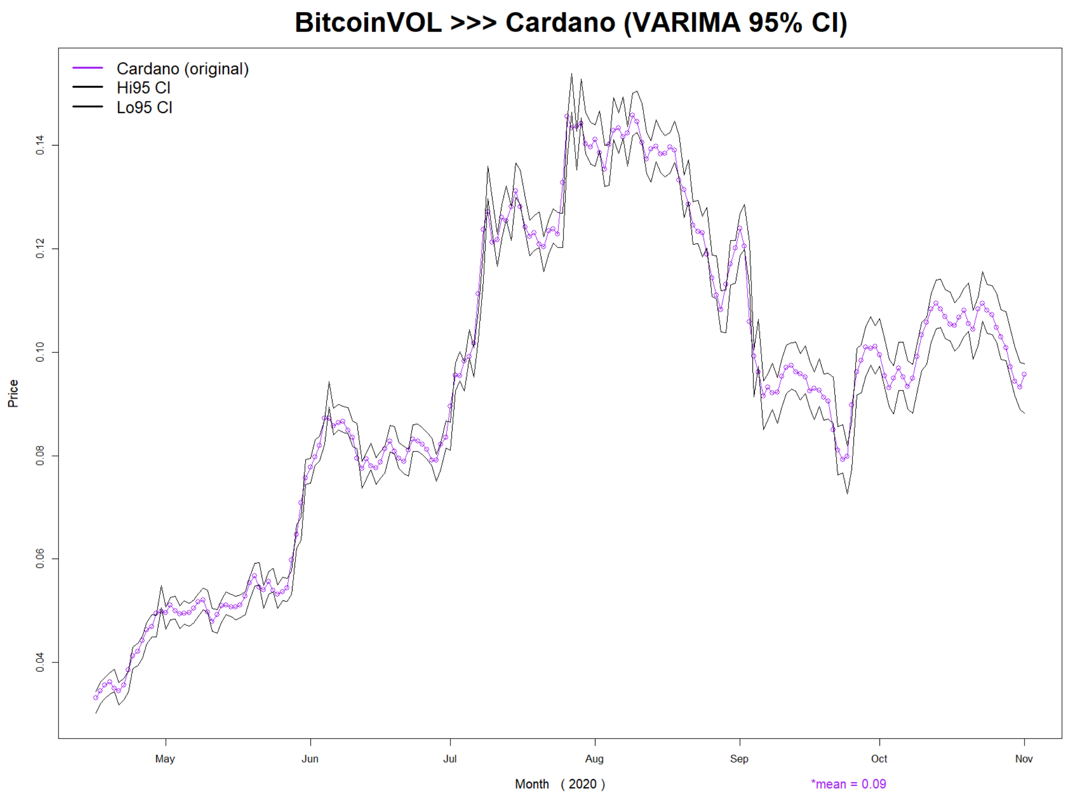


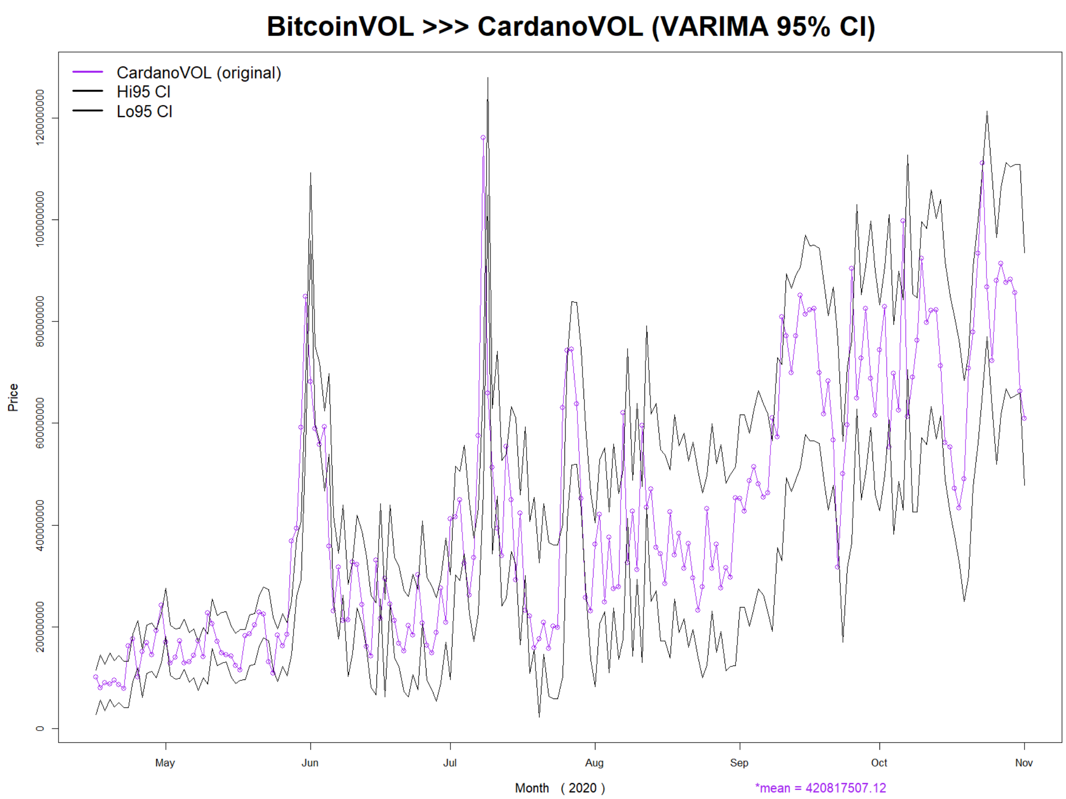

BitcoinVOL >>> BinanceVOL (VARIMA 95% CI)
BinanceVOL (original)
Hi95 CI
Lo95 CI
Price
0
200000000
400000000
600000000
800000000
1000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 347026000.15

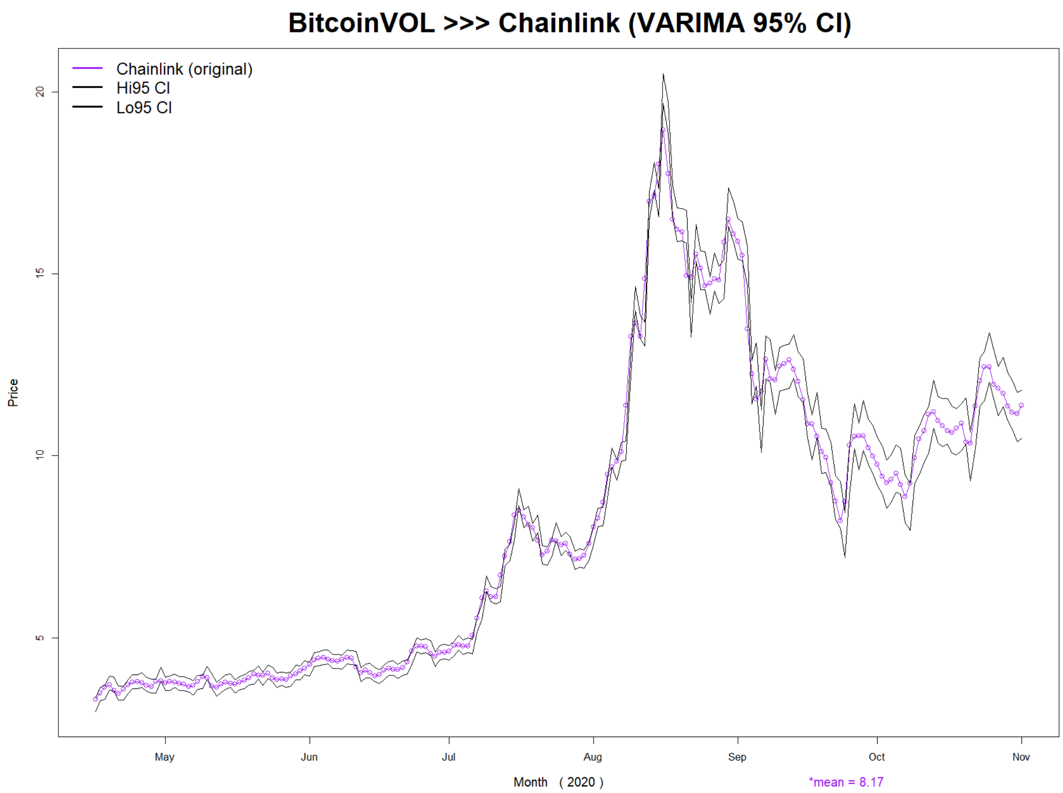

BitcoinVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28


BitcoinVOL >>> BitcoinCash (VARIMA 95% CI)
BitcoinCash (original)
Hi95 CI
Lo95 CI
Price
200
220
240
260
280
300
320
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 247.99

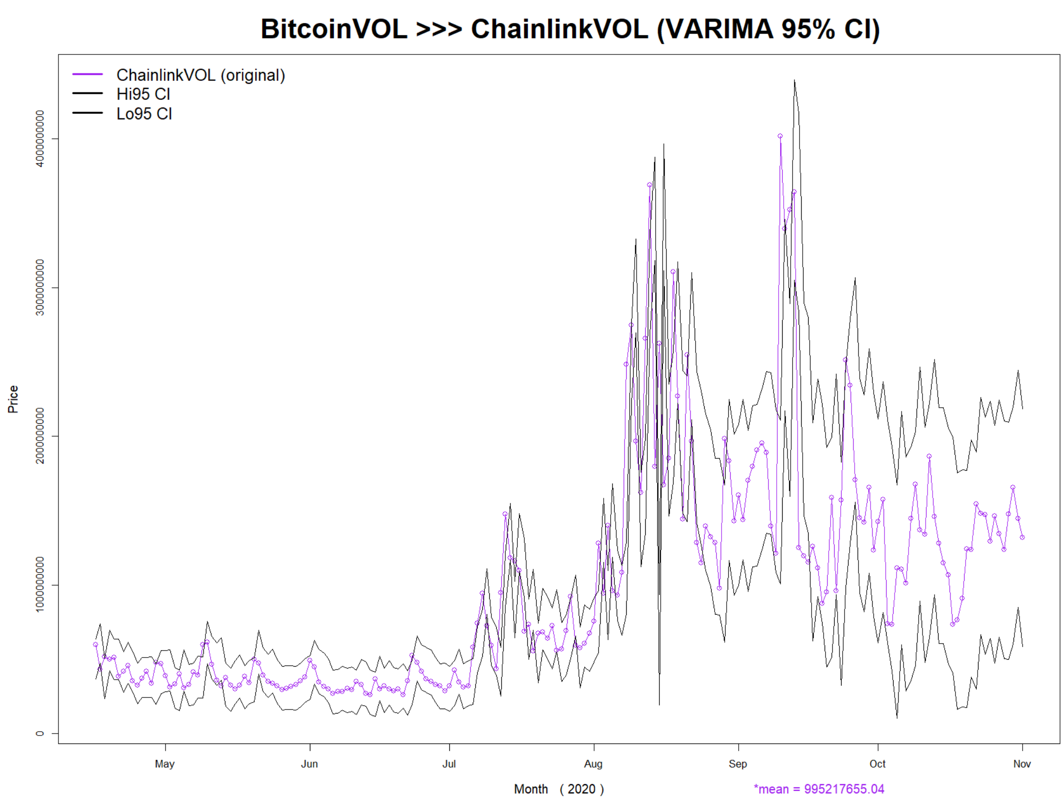

BitcoinVOL >>> BitcoinCashVOL (VARIMA 95% CI)
BitcoinCashVOL (original)
Hi95 CI
Lo95 CI
Price
-2000000000
0
2000000000
4000000000
6000000000
8000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 2180452377.7


BitcoinVOL >>> Cardano (VARIMA 95% CI)
Cardano (original)
Hi95 CI
Lo95 CI
Price
0.04
0.06
0.08
0.10
0.12
0.14
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 0.09


BitcoinVOL >>> CardanoVOL (VARIMA 95% CI)
CardanoVOL (original)
Hi95 CI
Lo95 CI
Price
0
200000000
400000000
600000000
800000000
1000000000
1200000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 420817507.12



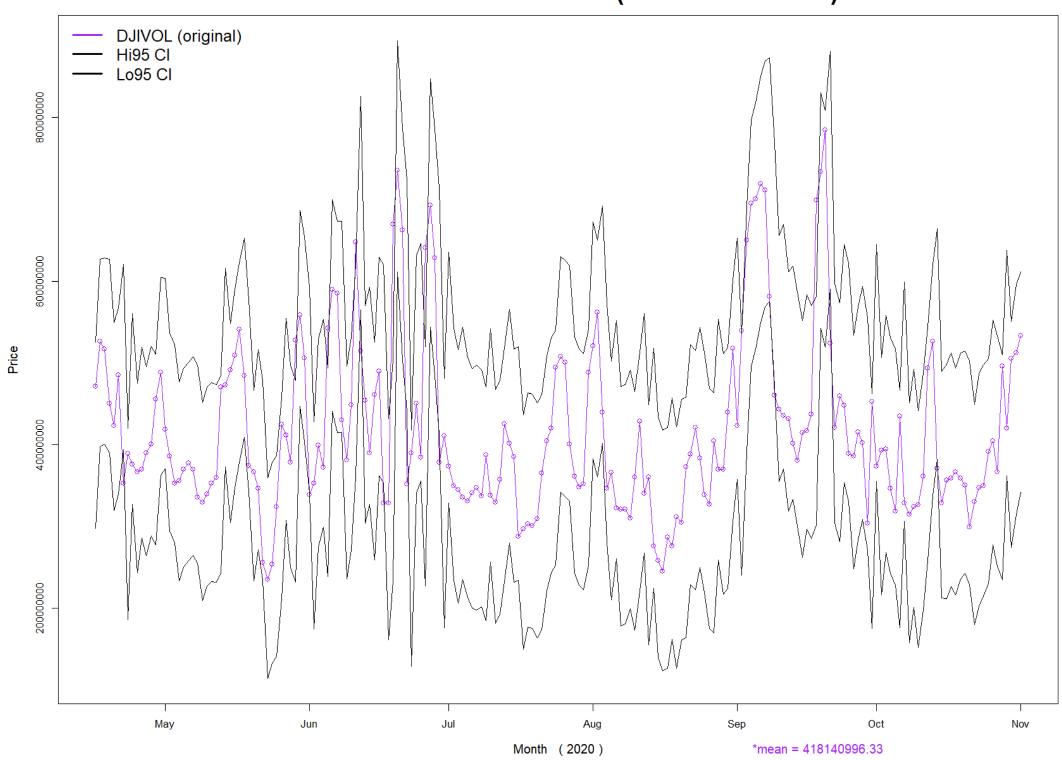
BitcoinVOL >>> Chainlink (VARIMA 95% CI)
Chainlink (original)
Hi95 CI
Lo95 CI
Price
20
15
10
5
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 8.17

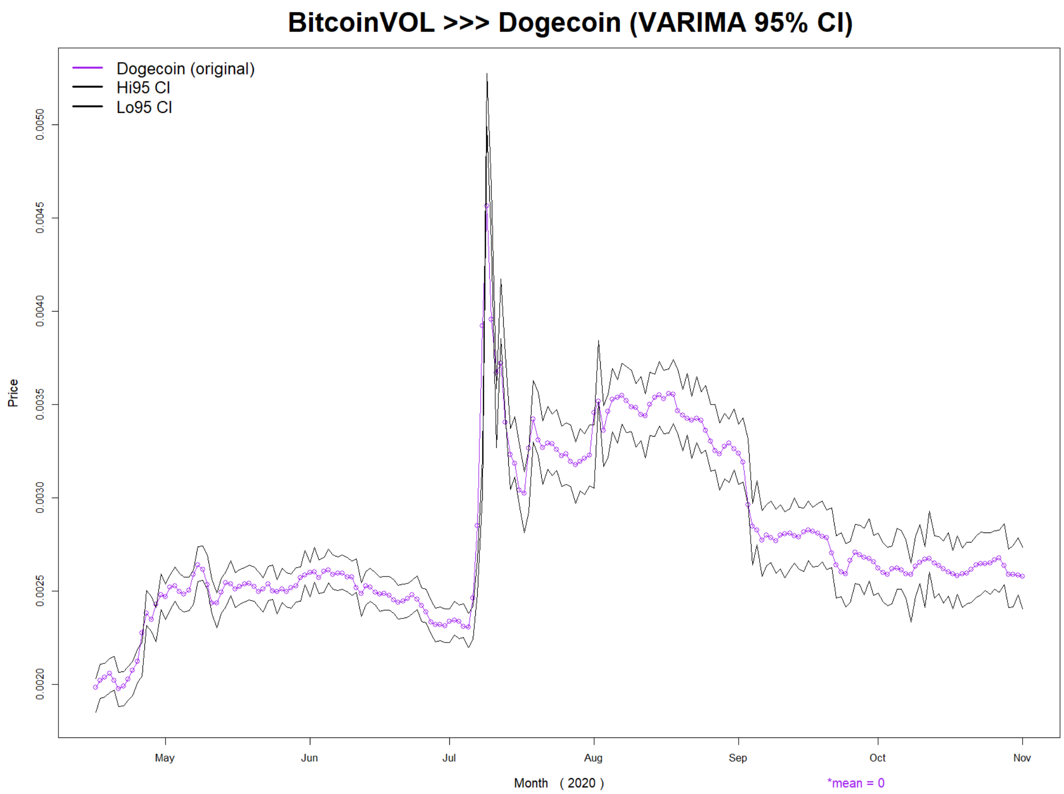


BitcoinVOL >>> ChainlinkVOL (VARIMA 95% CI)
ChainlinkVOL (original)
Hi95 CI
Lo95 CI
Price
4000000000
3000000000
2000000000
1000000000
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 995217655.04

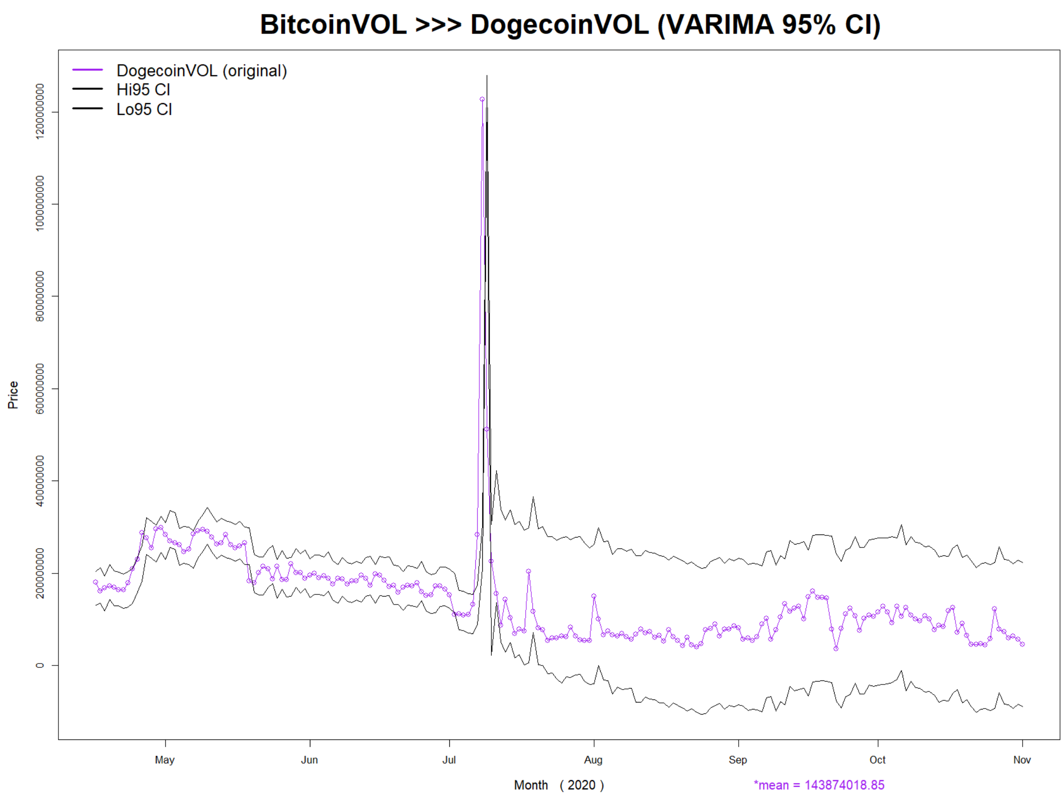
BitcoinVOL >>> DJI (VARIMA 95% CI)

BitcoinVOL >>> DJIVOL (VARIMA 95% CI)

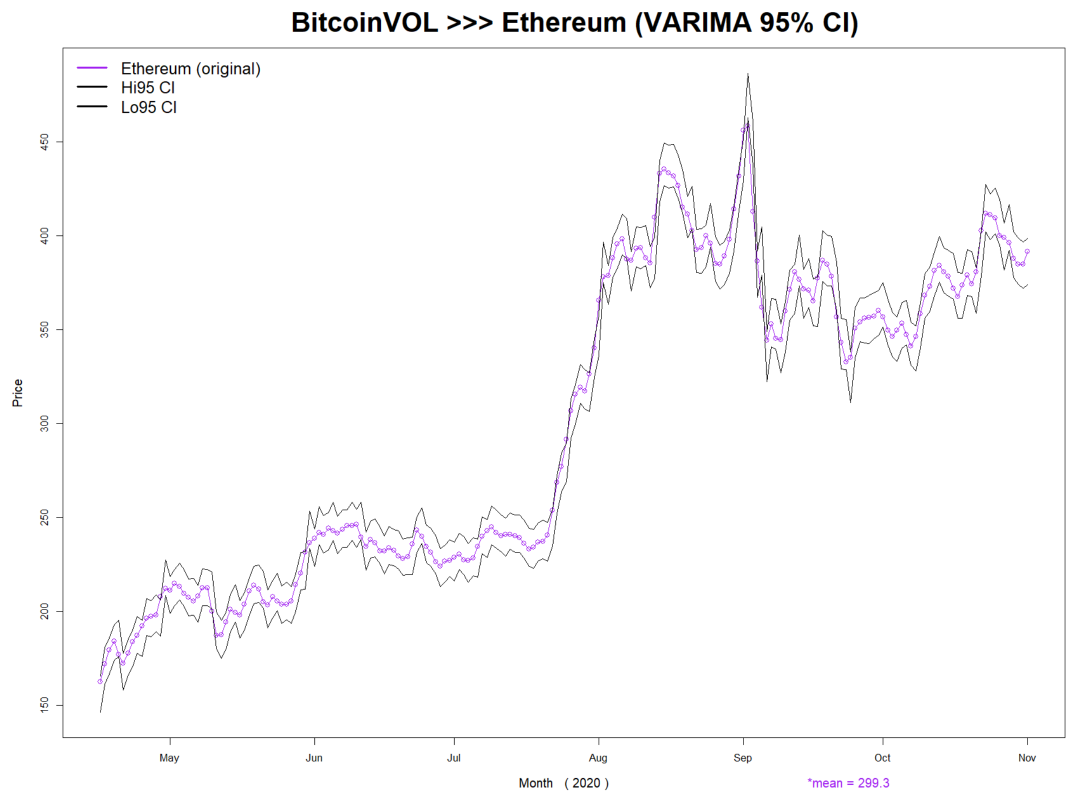


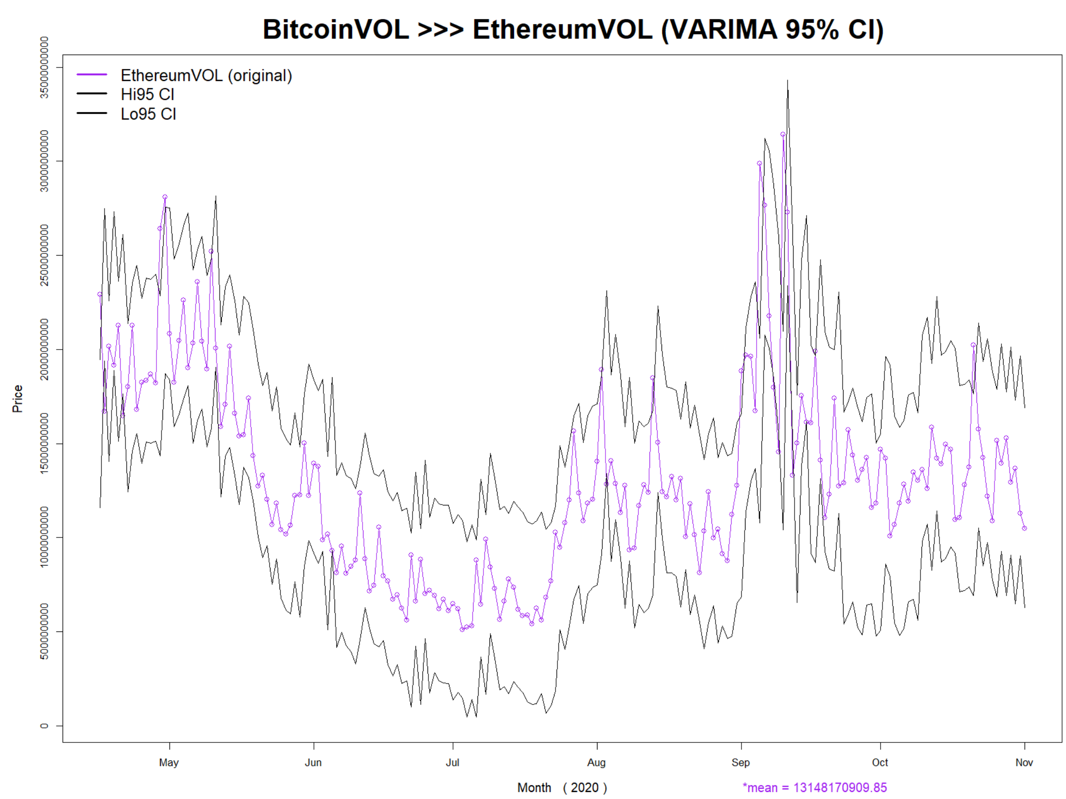


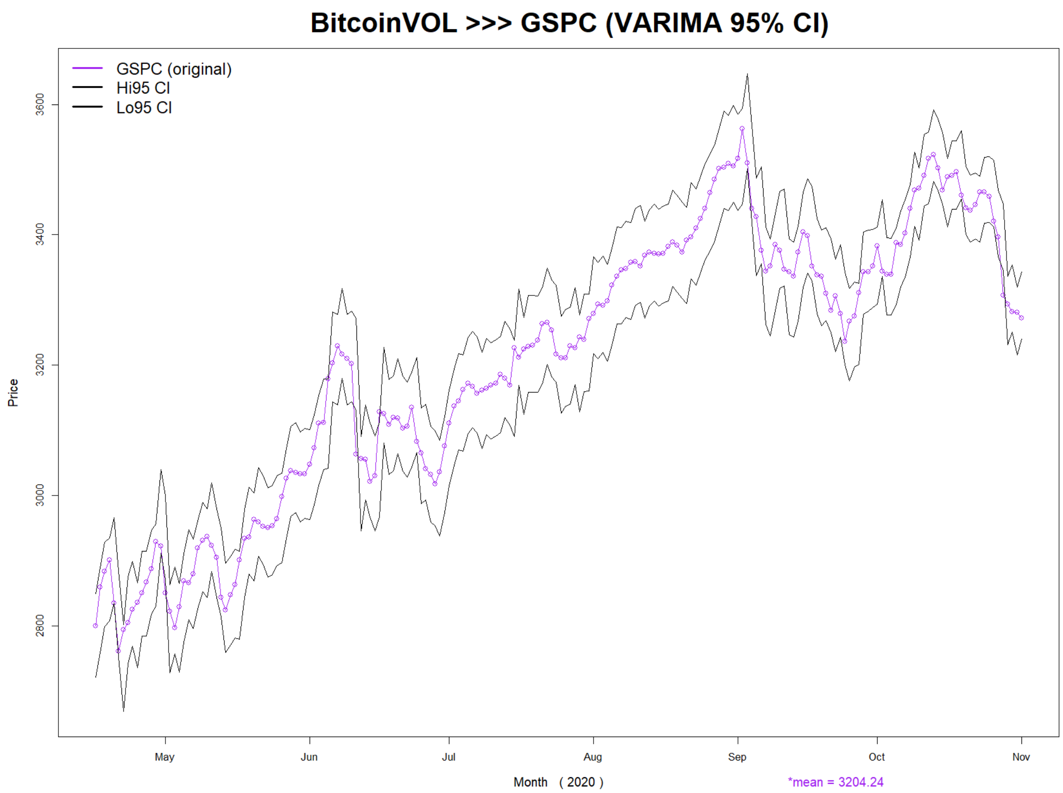
BitcoinVOL >>> Dogecoin (VARIMA 95% CI)
Dogecoin (original)
Hi95 CI
Lo95 CI
Price
0.0020
0.0025
0.0030
0.0035
0.0040
0.0045
0.0050
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 0

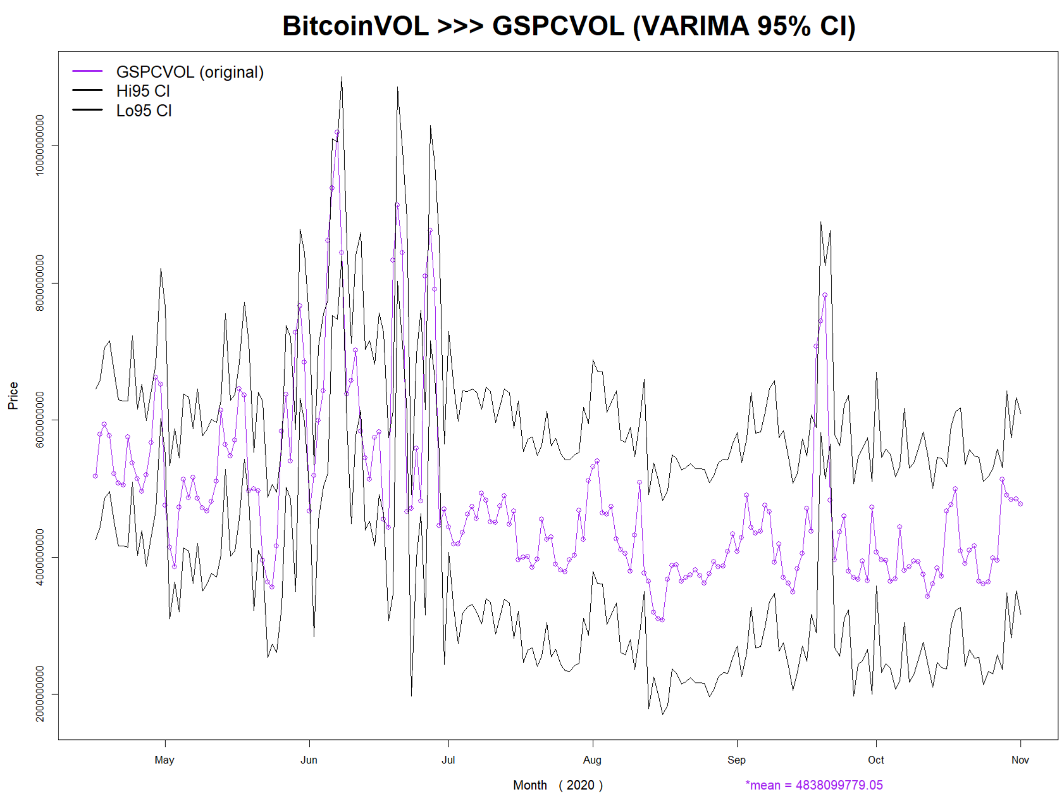


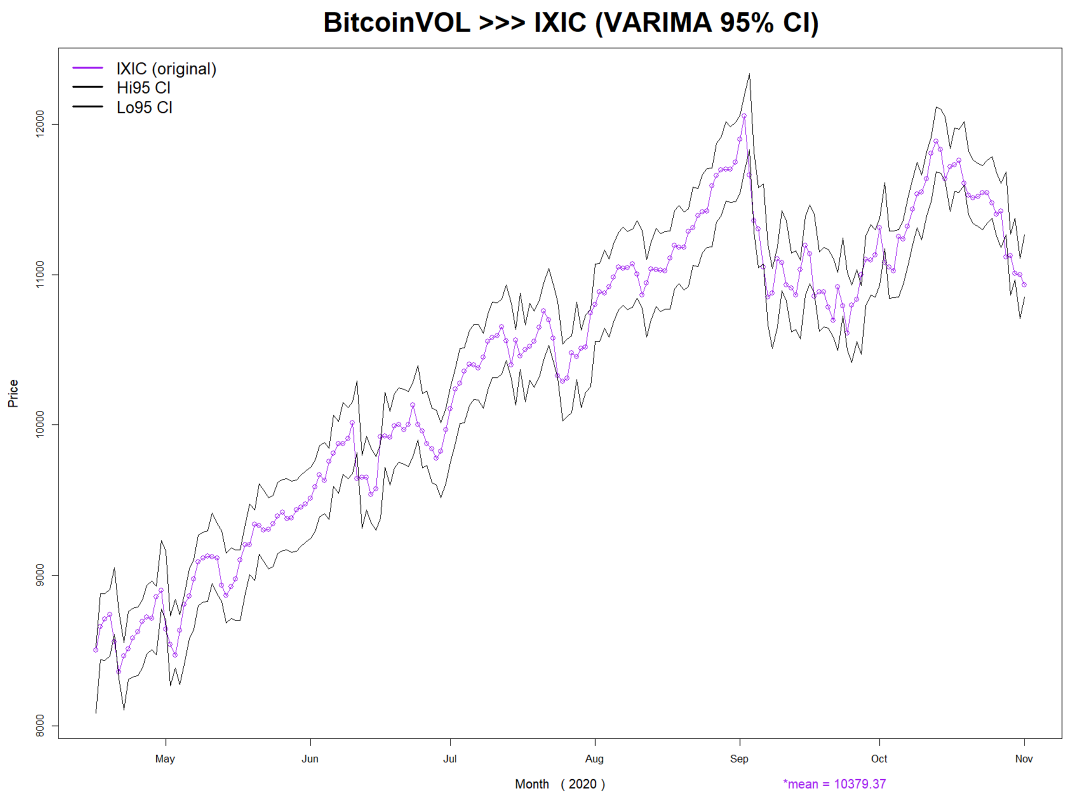
BitcoinVOL >>> DogecoinVOL (VARIMA 95% CI)
DogecoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
200000000
400000000
600000000
800000000
1000000000
1200000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 143874018.85

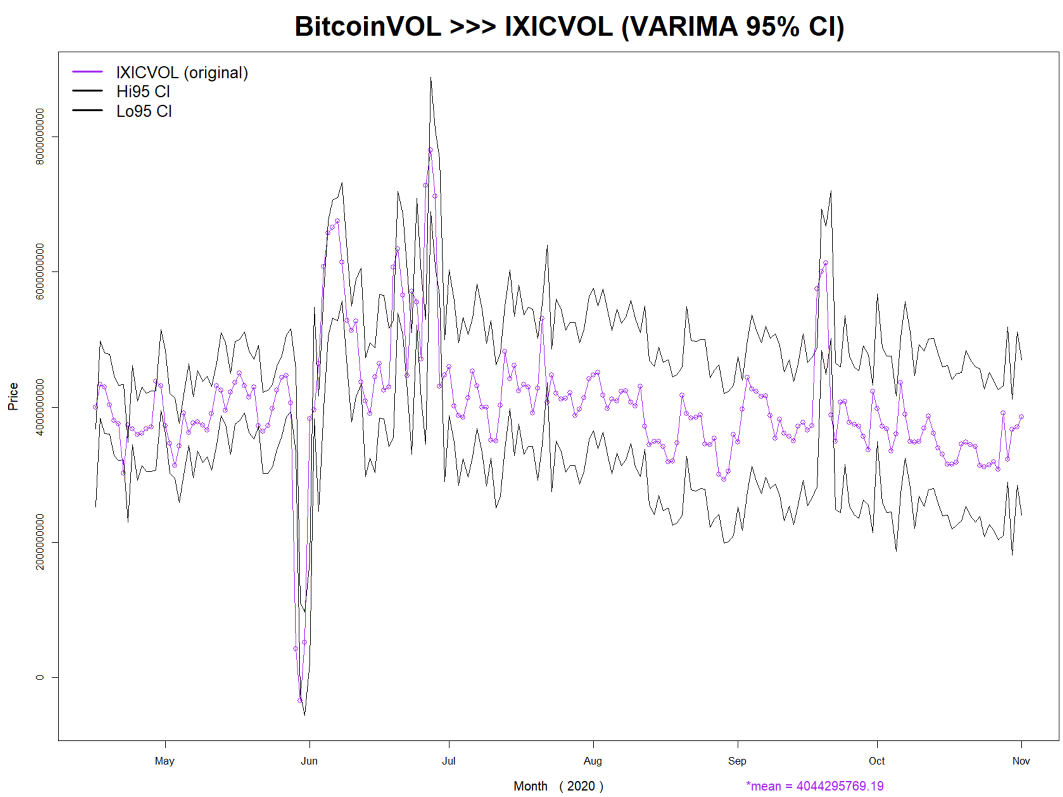

BitcoinVOL >>> Ethereum (VARIMA 95% CI)
Ethereum (original)
Hi95 CI
Lo95 CI
Price
150
200
250
300
350
400
450
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 299.3

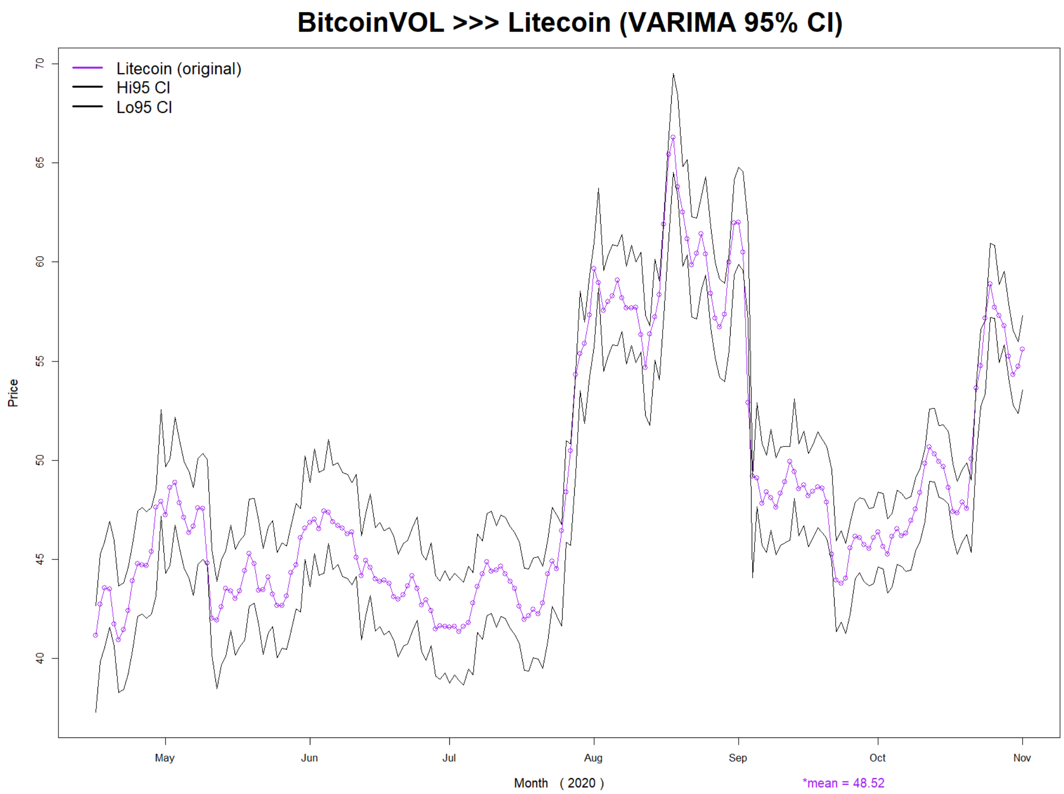

BitcoinVOL >>> EthereumVOL (VARIMA 95% CI)
EthereumVOL (original)
Hi95 CI
Lo95 CI
Price
0
5000000000
10000000000
15000000000
20000000000
25000000000
30000000000
35000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 13148170909.85

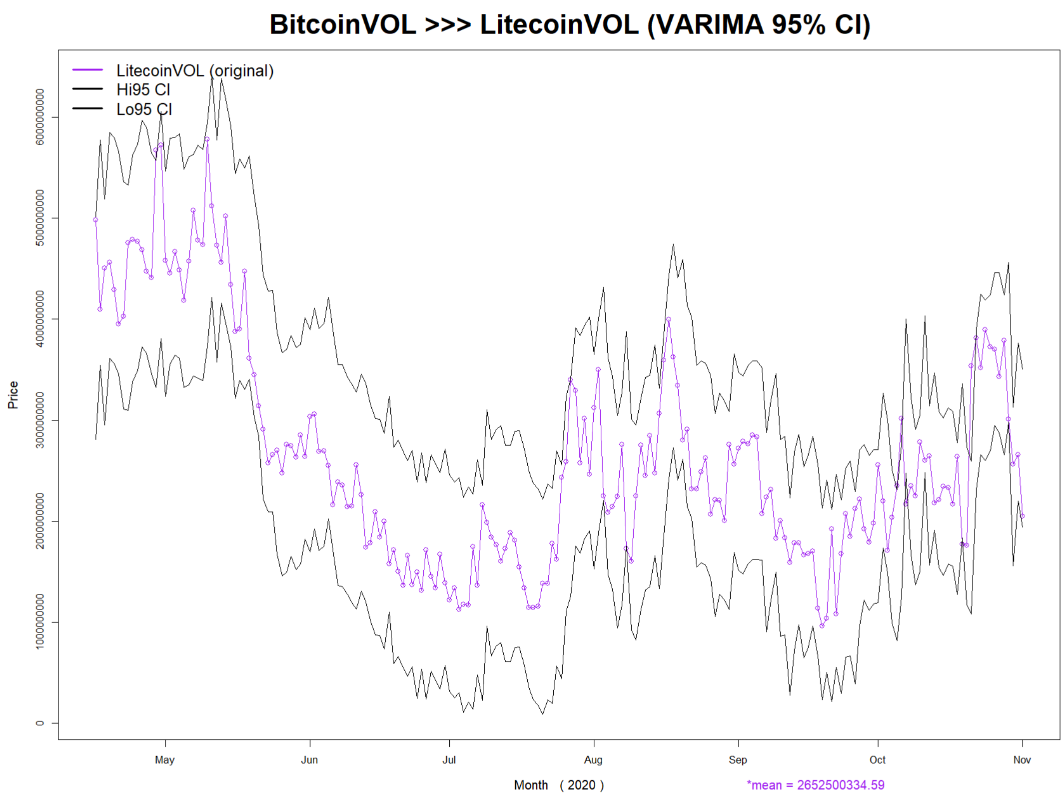

BitcoinVOL >>> GSPC (VARIMA 95% CI)
GSPC (original)
Hi95 CI
Lo95 CI
Price
3600
3400
3200
3000
2800
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 3204.24


BitcoinVOL >>> GSPCVOL (VARIMA 95% CI)
GSPCVOL (original)
Hi95 CI
Lo95 CI
Price
10000000000
8000000000
6000000000
4000000000
2000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 4838099779.05

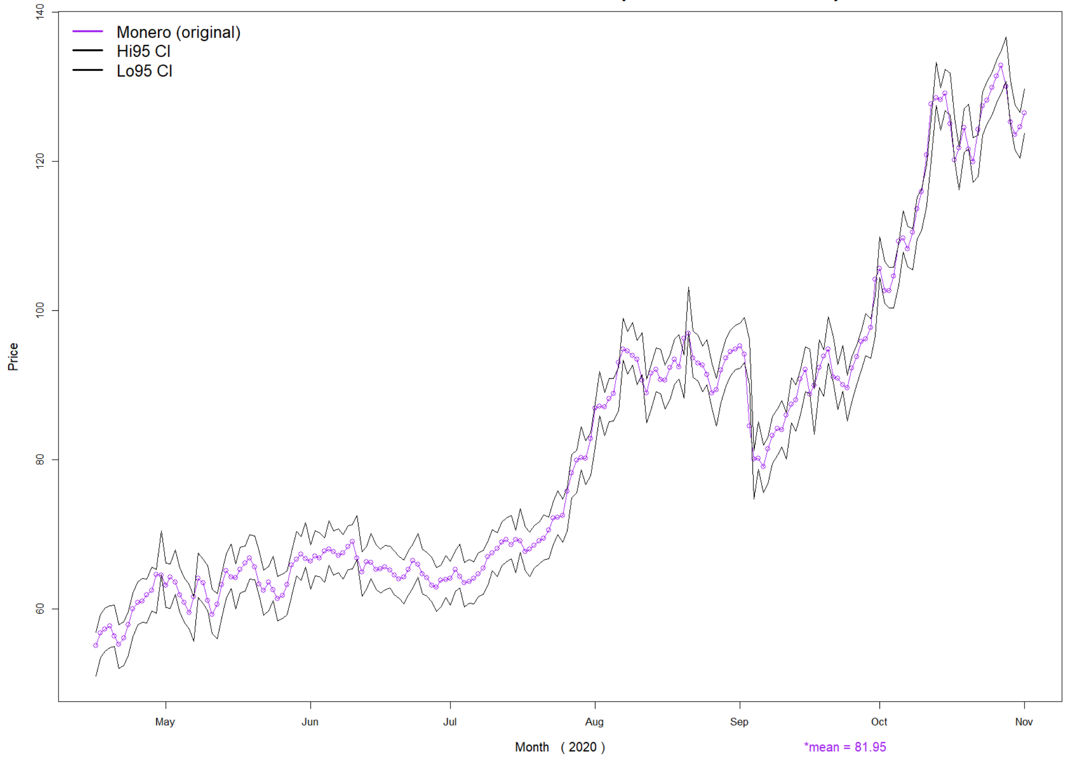

BitcoinVOL >>> IXIC (VARIMA 95% CI)
IXIC (original)
Hi95 CI
Lo95 CI
Price
8000
9000
10000
11000
12000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10379.37


BitcoinVOL >>> IXICVOL (VARIMA 95% CI)
IXICVOL (original)
Hi95 CI
Lo95 CI
Price
0
2000000000
4000000000
6000000000
8000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 4044295769.19

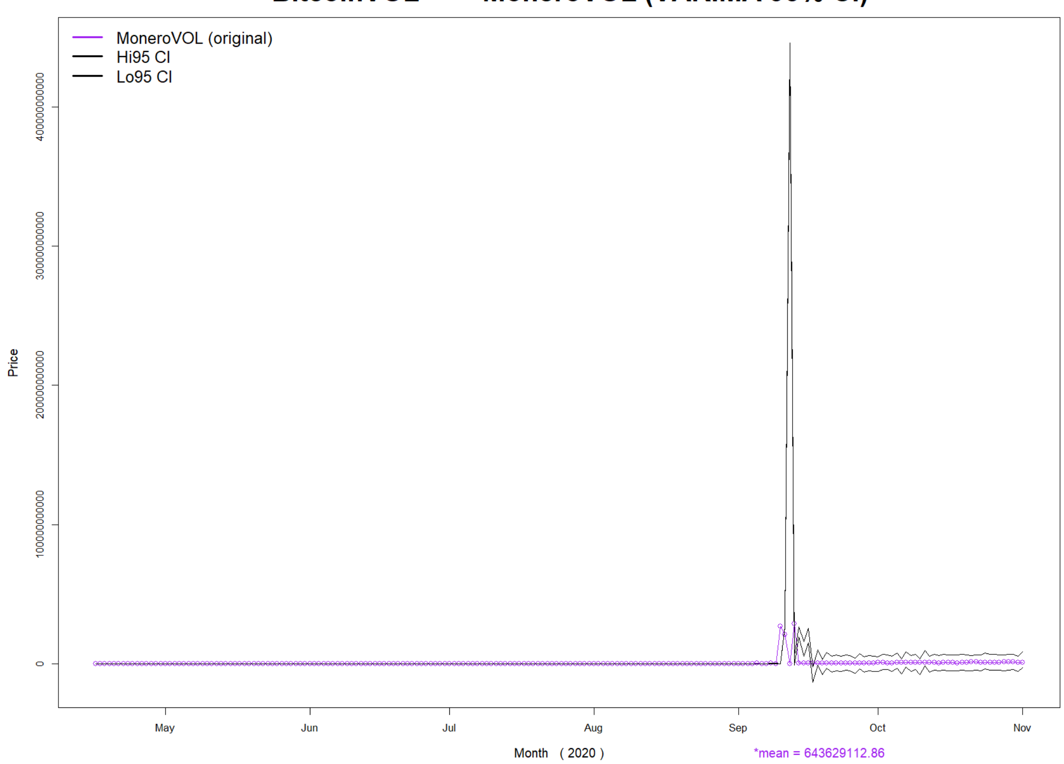

BitcoinVOL >>> Litecoin (VARIMA 95% CI)
Litecoin (original)
Hi95 CI
Lo95 CI
Price
70
65
60
55
50
45
40
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 48.52

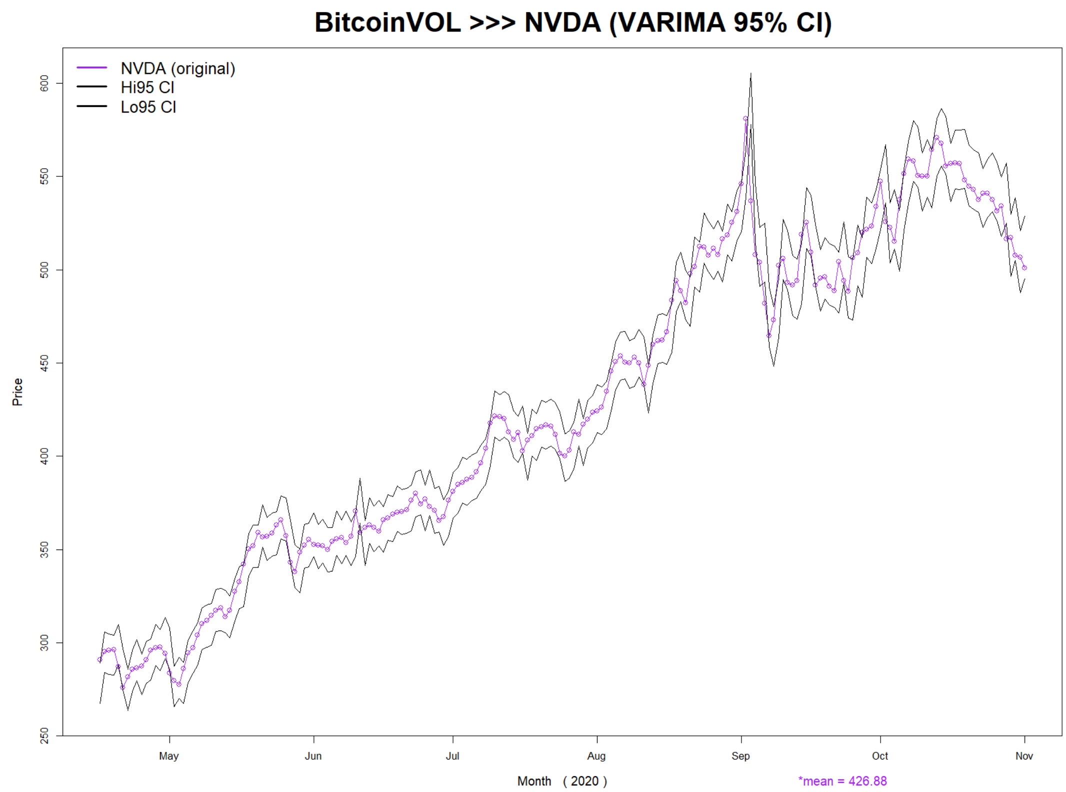

BitcoinVOL >>> LitecoinVOL (VARIMA 95% CI)
LitecoinVOL (original)
Hi95 CI
Lo95 CI
Price
6000000000
5000000000
4000000000
3000000000
2000000000
1000000000
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 2652500334.59

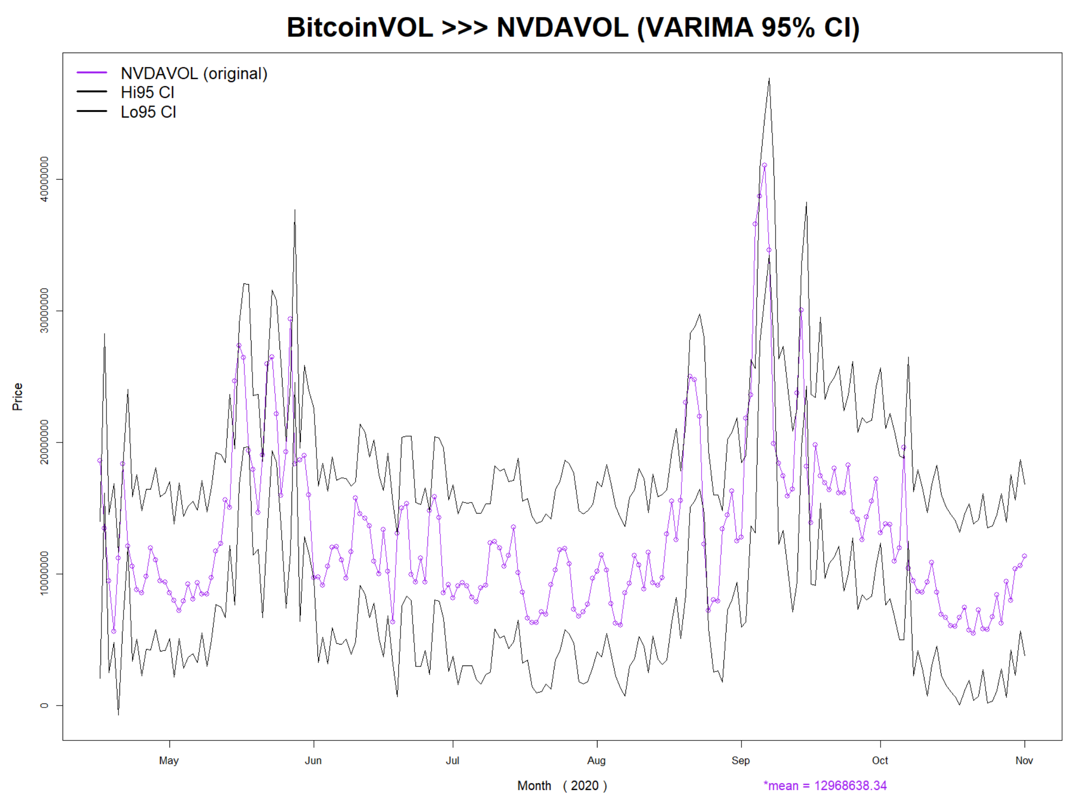
BitcoinVOL >>> Monero (VARIMA 95% CI)

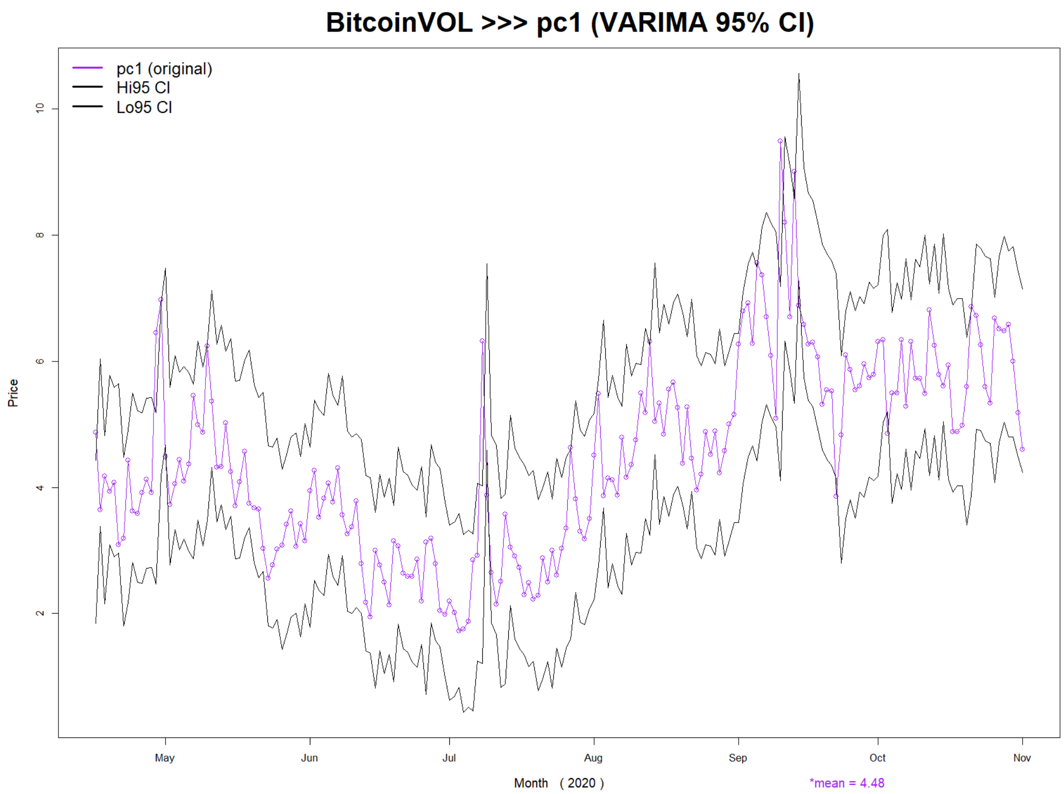


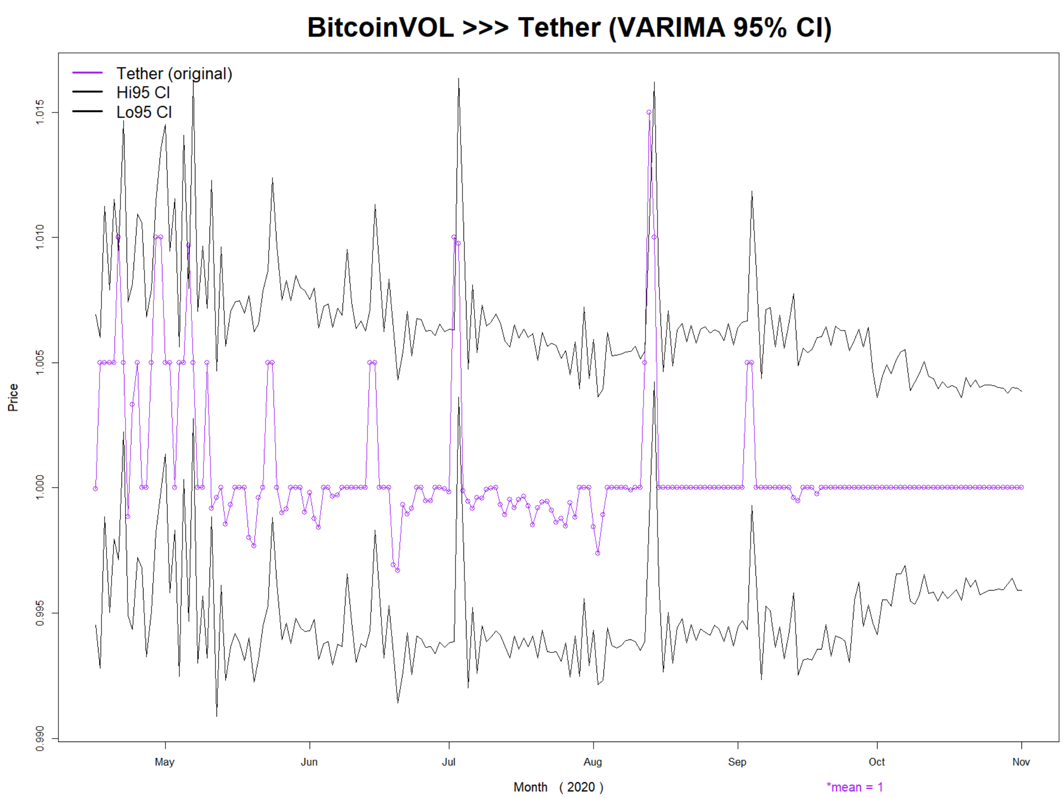
BitcoinVOL >>> MoneroVOL (VARIMA 95% CI)

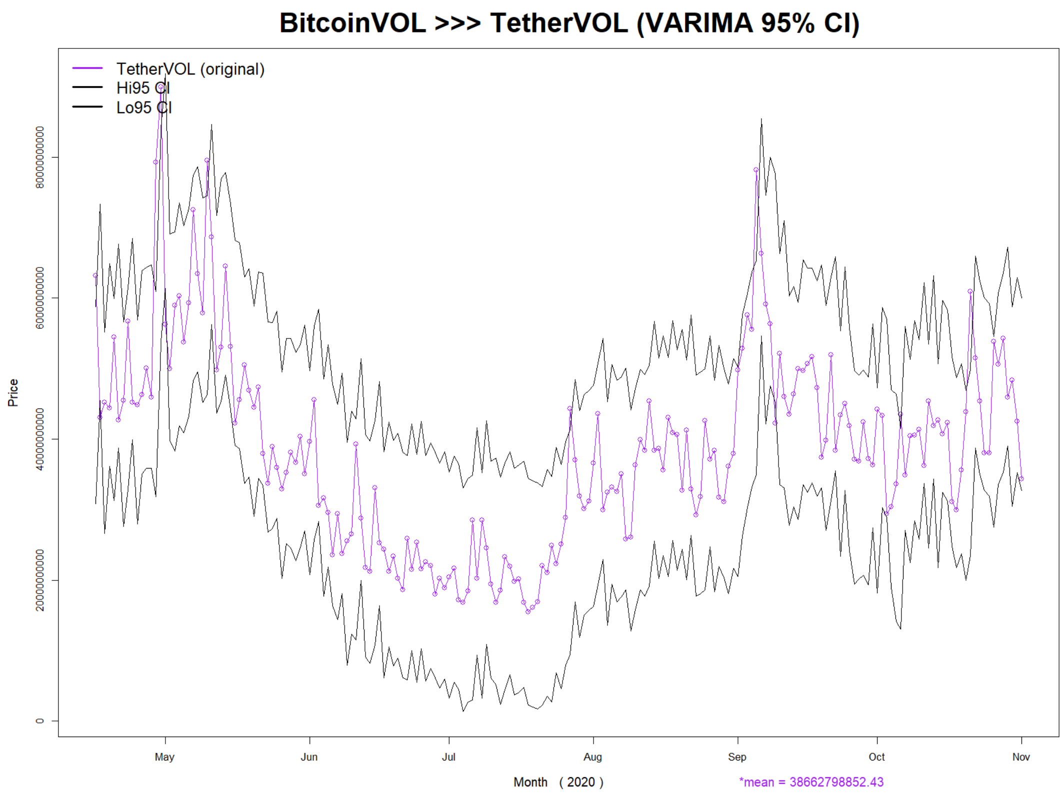


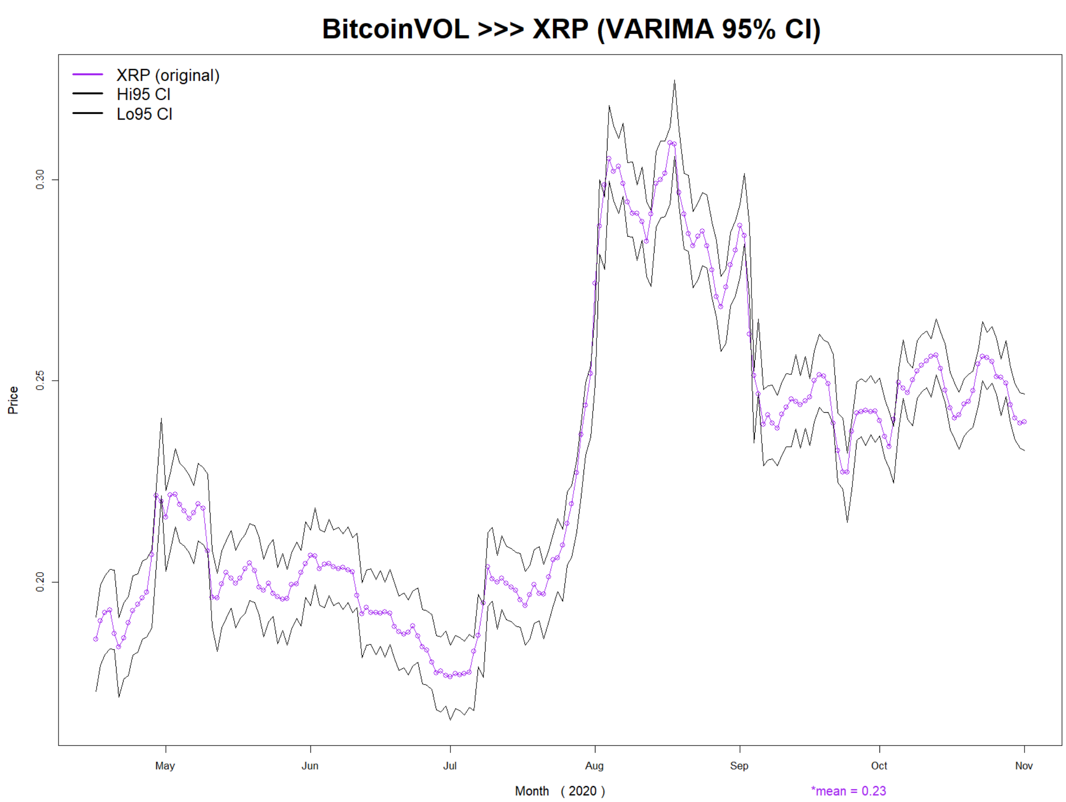

BitcoinVOL >>> NVDA (VARIMA 95% CI)
NVDA (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 426.88

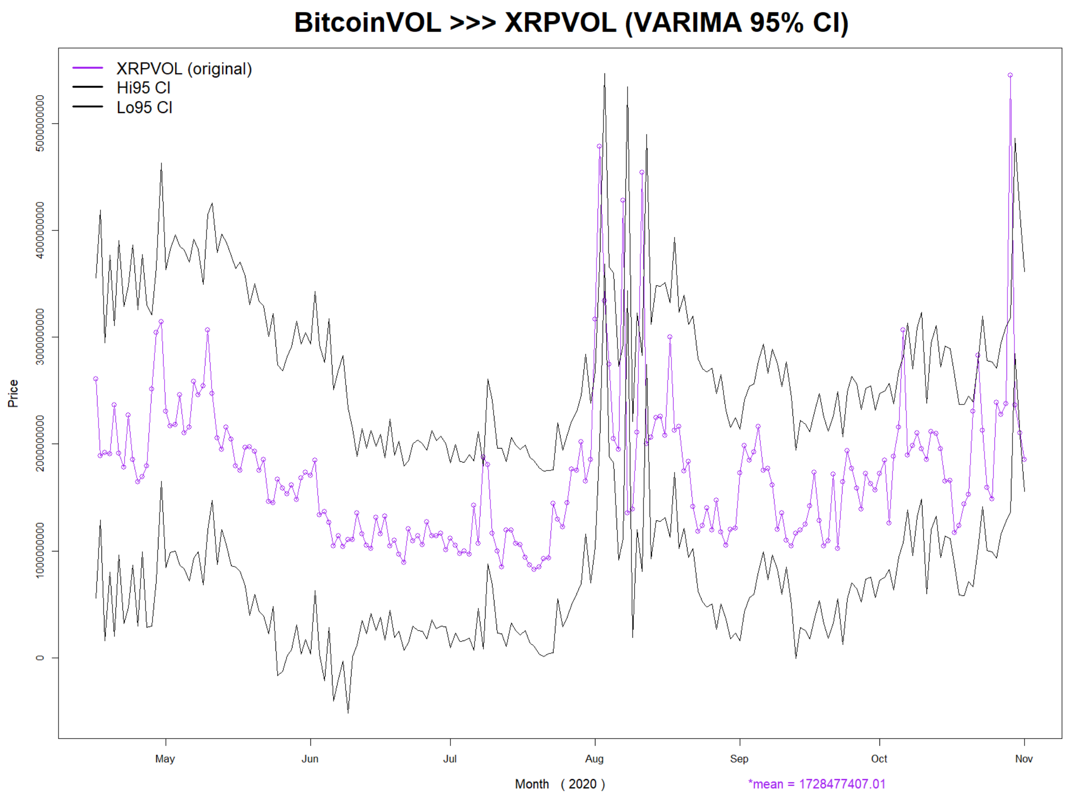

BitcoinVOL >>> NVDAVOL (VARIMA 95% CI)
NVDAVOL (original)
Hi95 CI
Lo95 CI
Price
Month (2020)
*mean = 12968638.34


BitcoinVOL >>> pc1 (VARIMA 95% CI)
pc1 (original)
Hi95 CI
Lo95 CI
Price
10
8
6
4
2
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 4.48

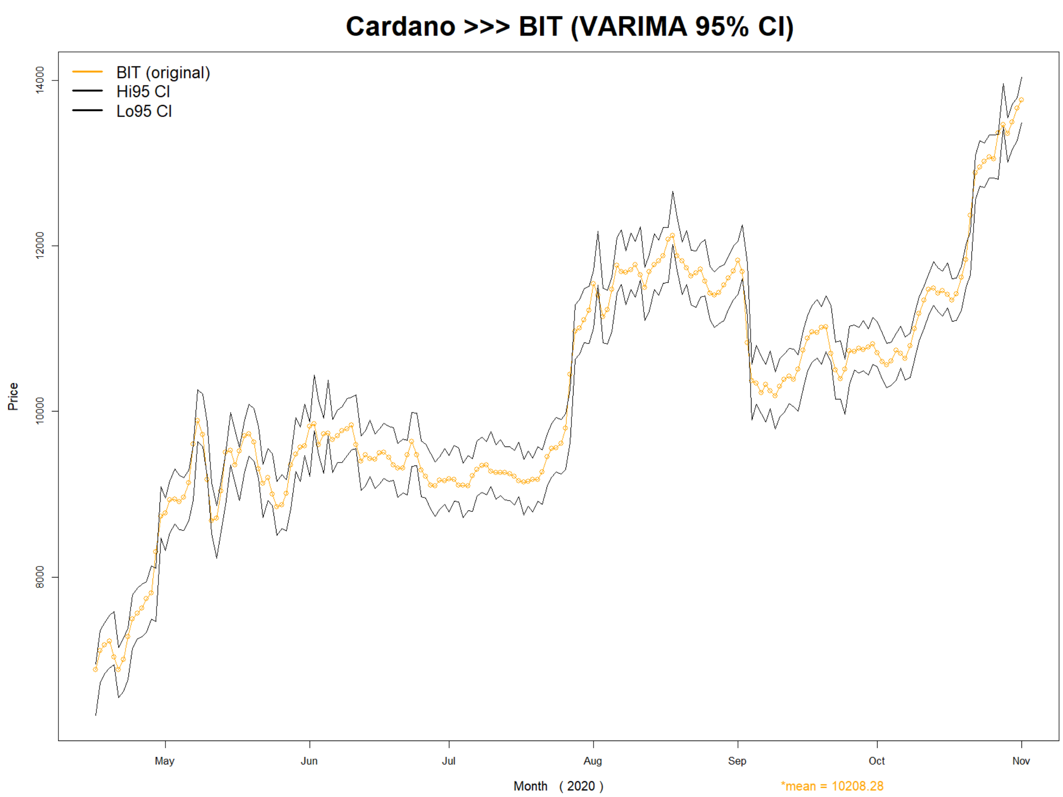

BitcoinVOL >>> Tether (VARIMA 95% CI)
Tether (original)
Hi95 CI
Lo95 CI
Price
1.015
1.010
1.005
1.000
0.995
0.990
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 1


BitcoinVOL >>> TetherVOL (VARIMA 95% CI)
TetherVOL (original)
Hi95 CI
Lo95 CI
Price
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 38662798852.43

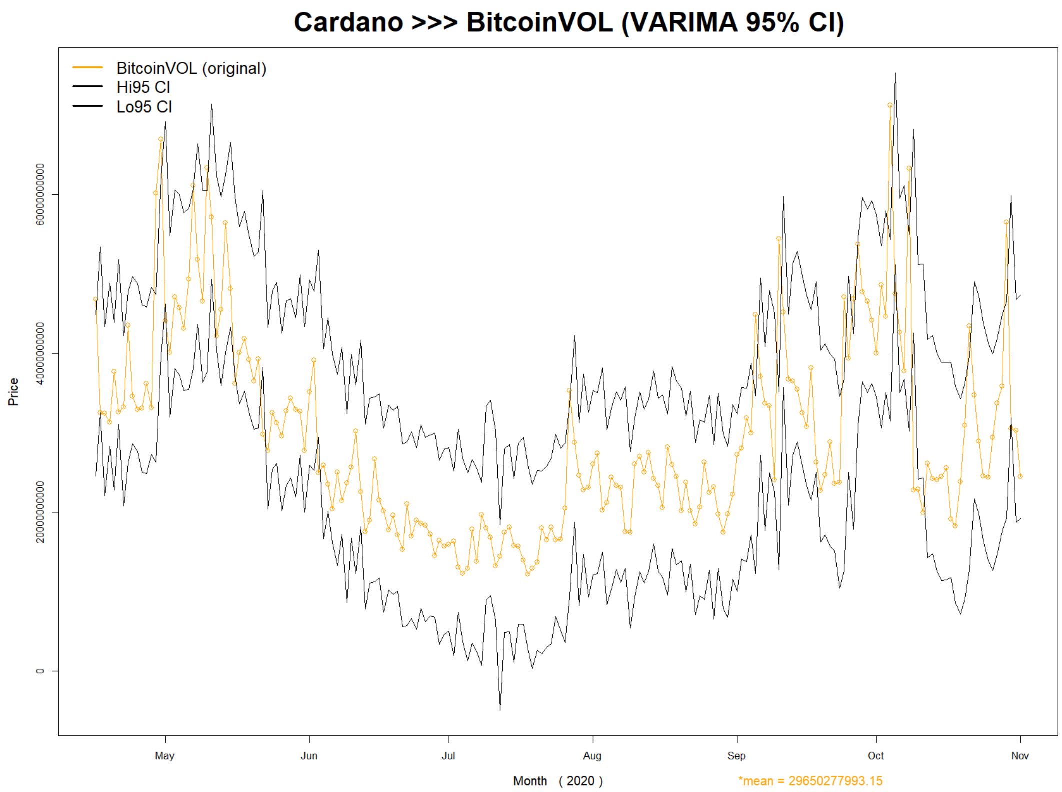

BitcoinVOL >>> XRP (VARIMA 95% CI)
XRP (original)
Hi95 CI
Lo95 CI
Price
0.20
0.25
0.30
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 0.23

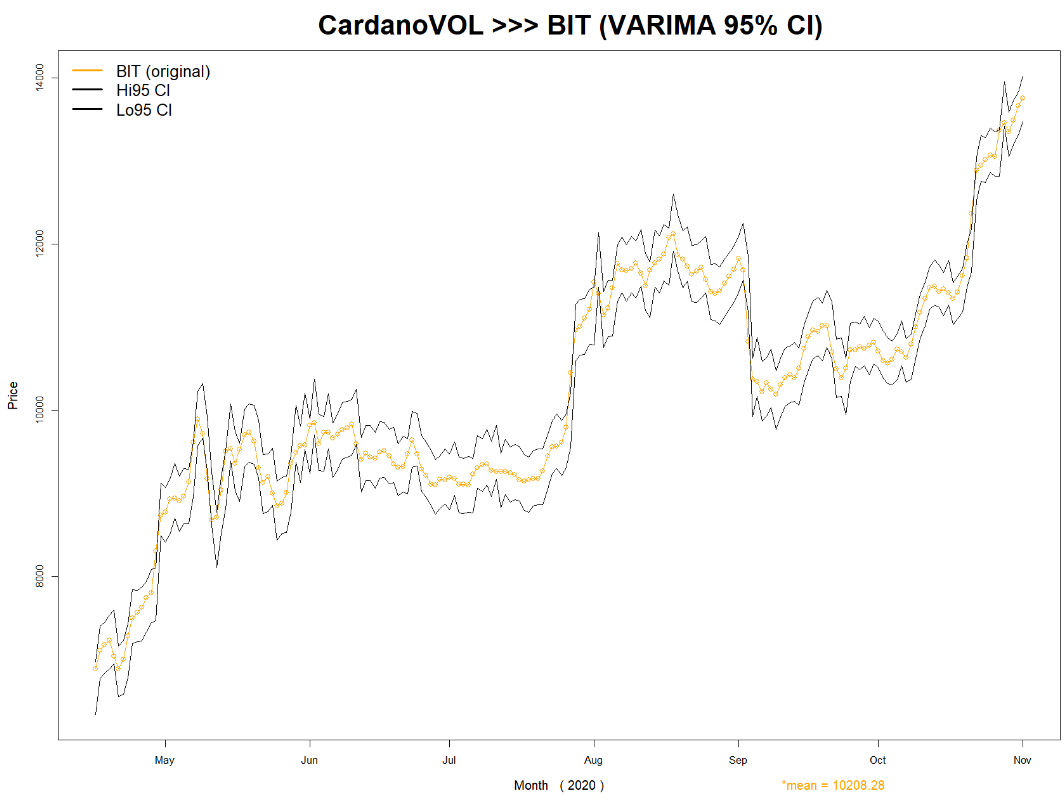


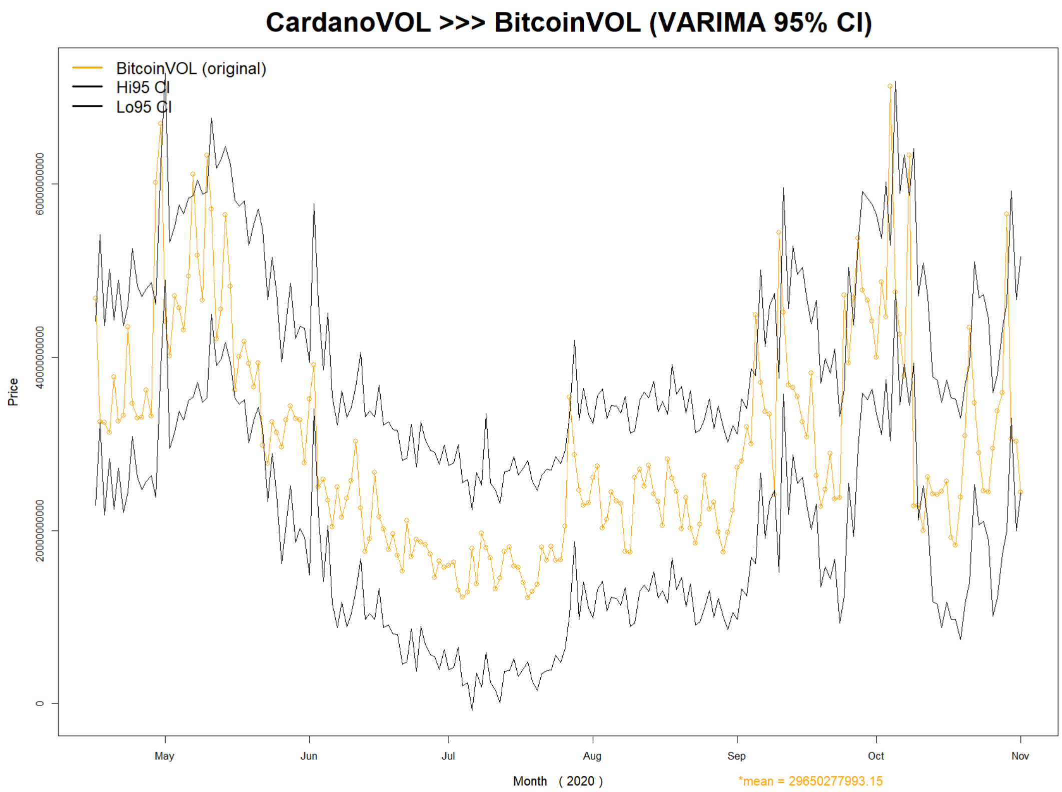
BitcoinVOL >>> XRPVOL (VARIMA 95% CI)
XRPVOL (original)
Hi95 CI
Lo95 CI
Price
0
1000000000
2000000000
3000000000
4000000000
5000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 1728477407.01



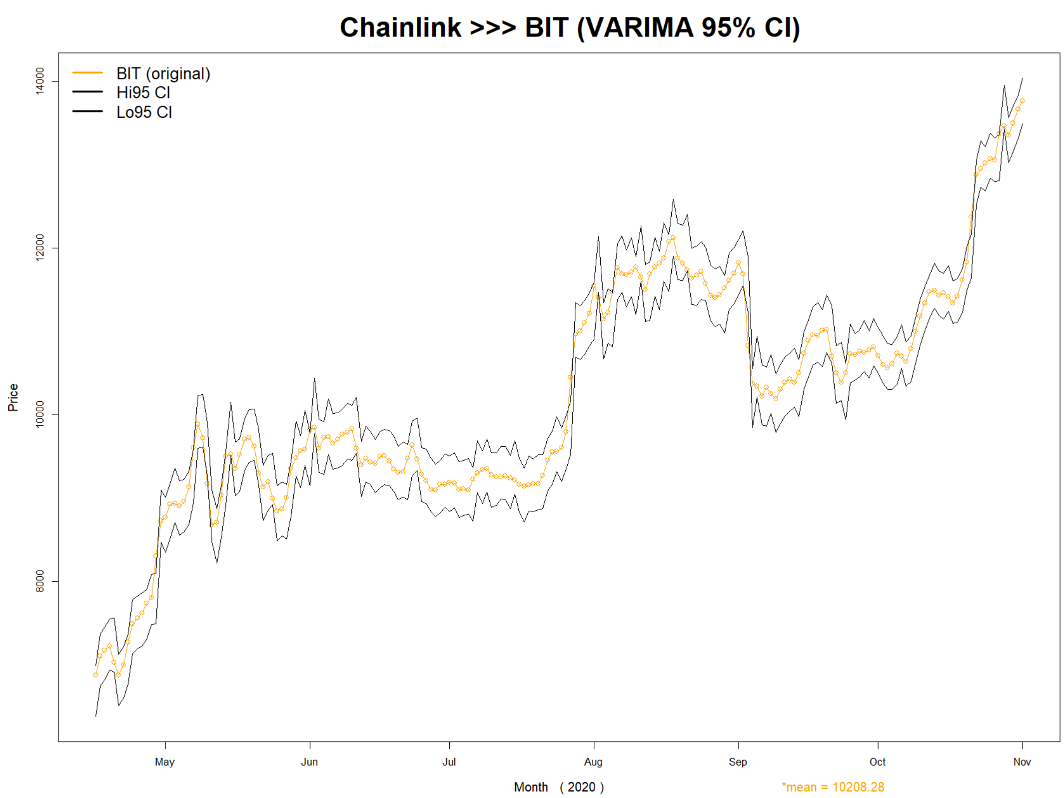
Cardano >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28


Cardano >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

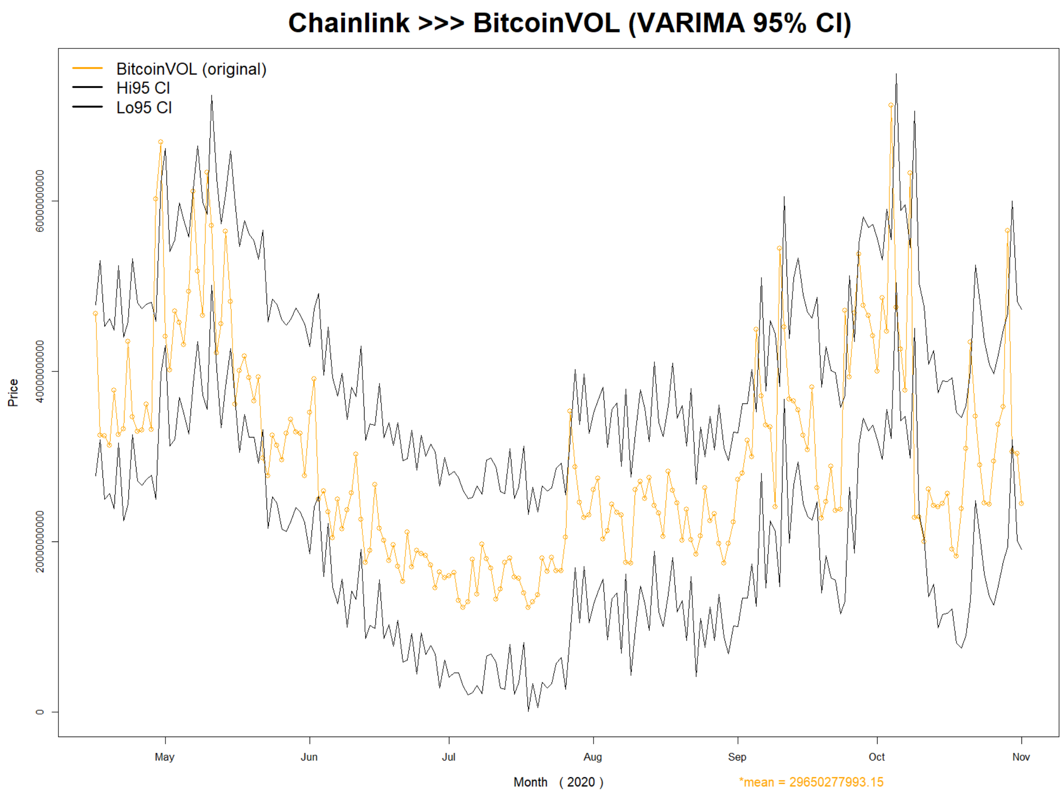

CardanoVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28



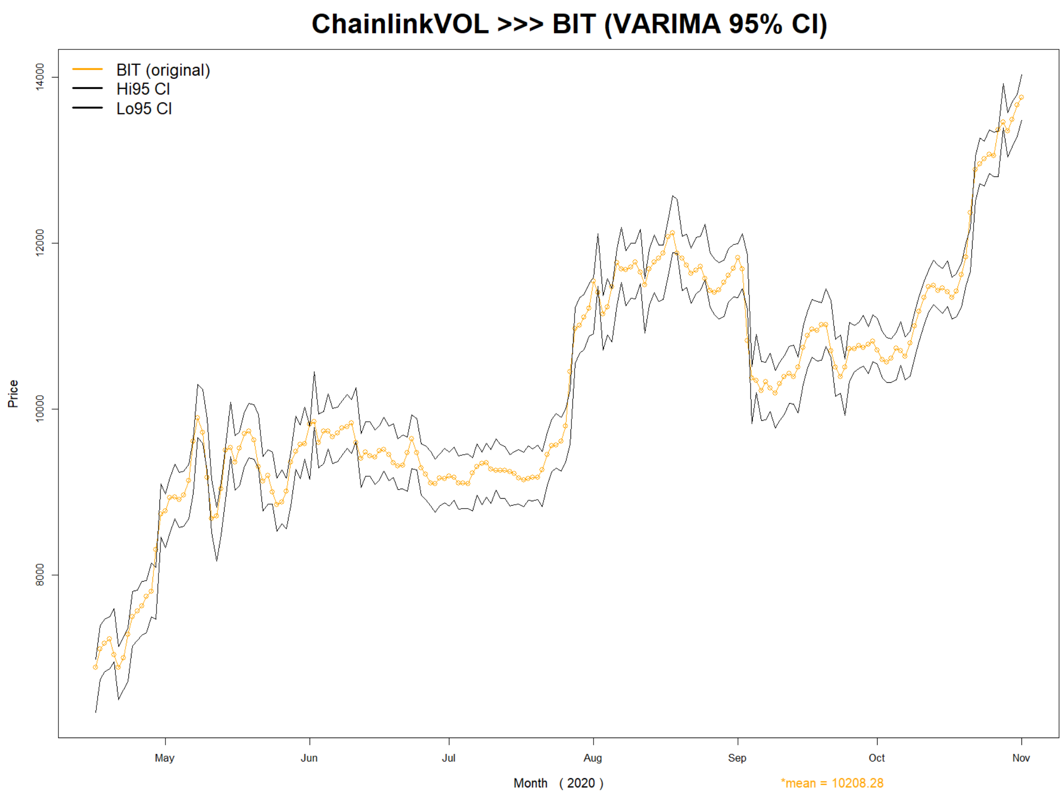
CardanoVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15



Chainlink >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

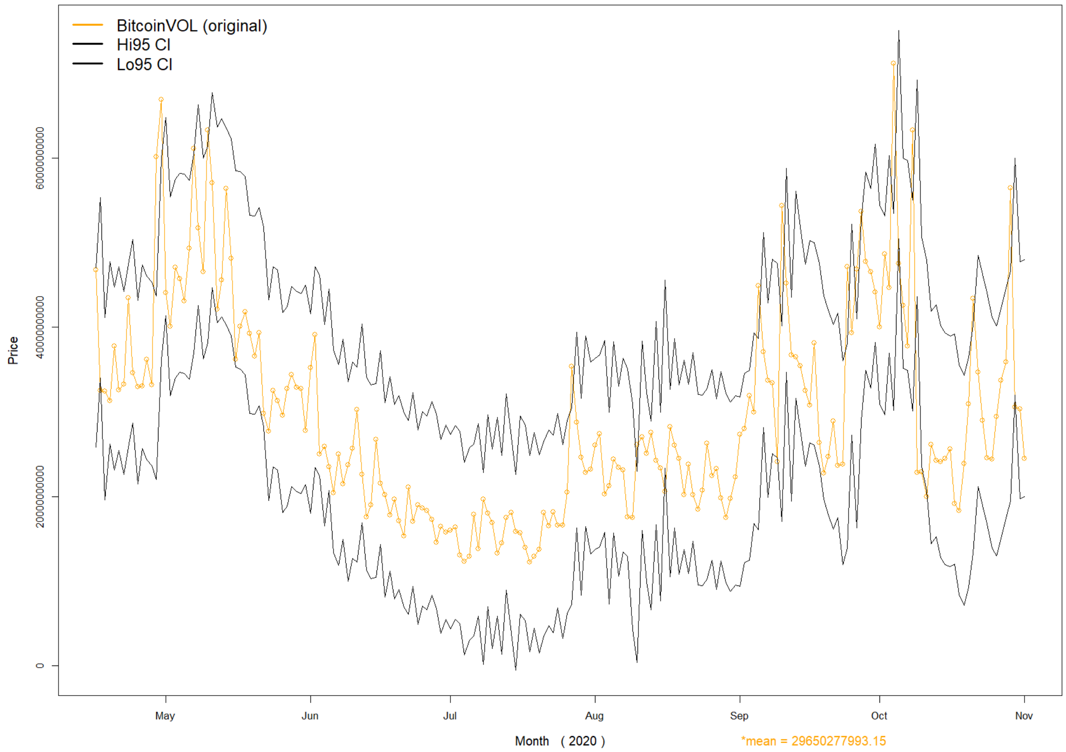


Chainlink >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
60000000000
40000000000
20000000000
0
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15



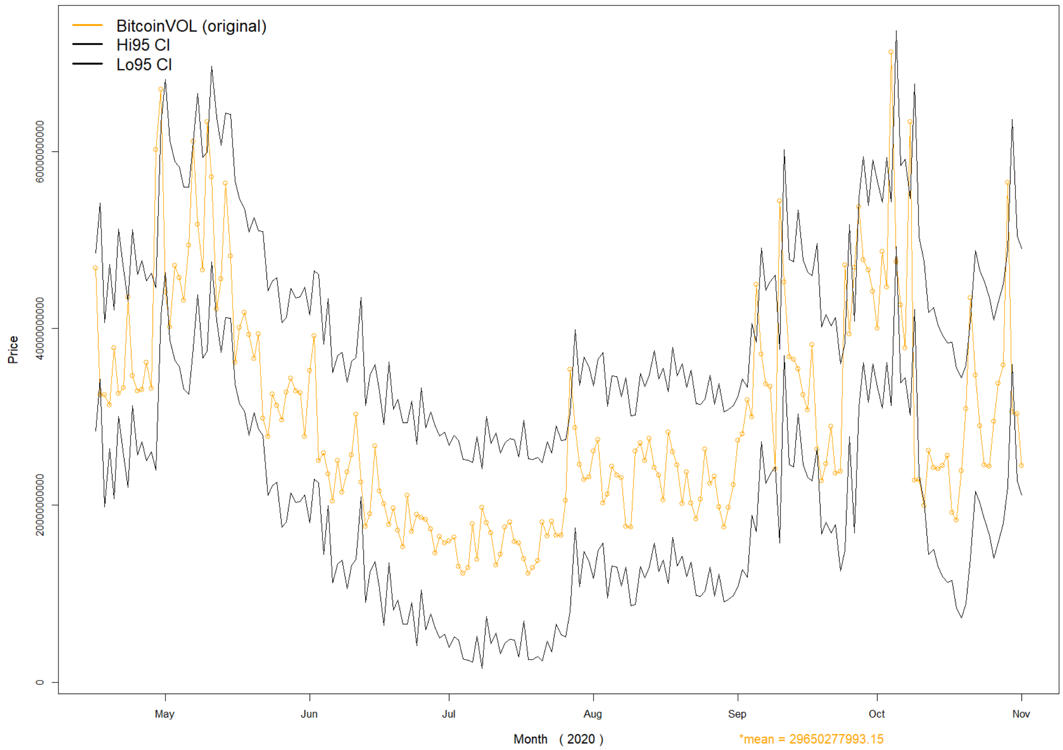
ChainlinkVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
14000
12000
10000
8000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

ChainlinkVOL >>> BitcoinVOL (VARIMA 95% CI)

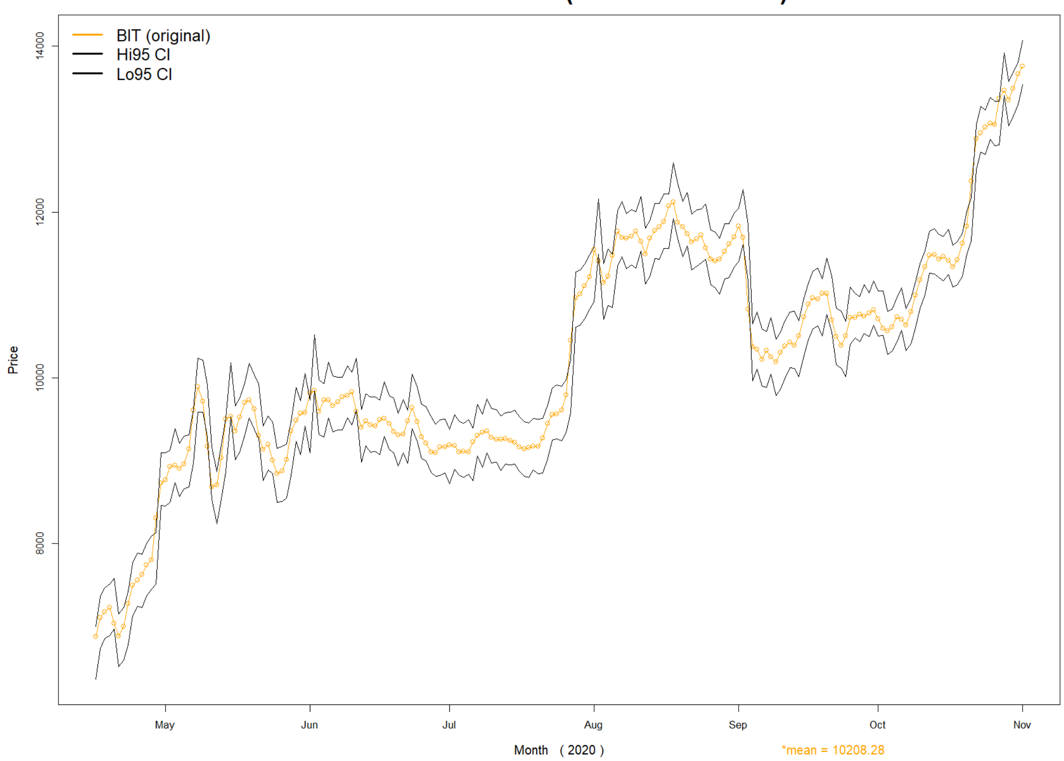


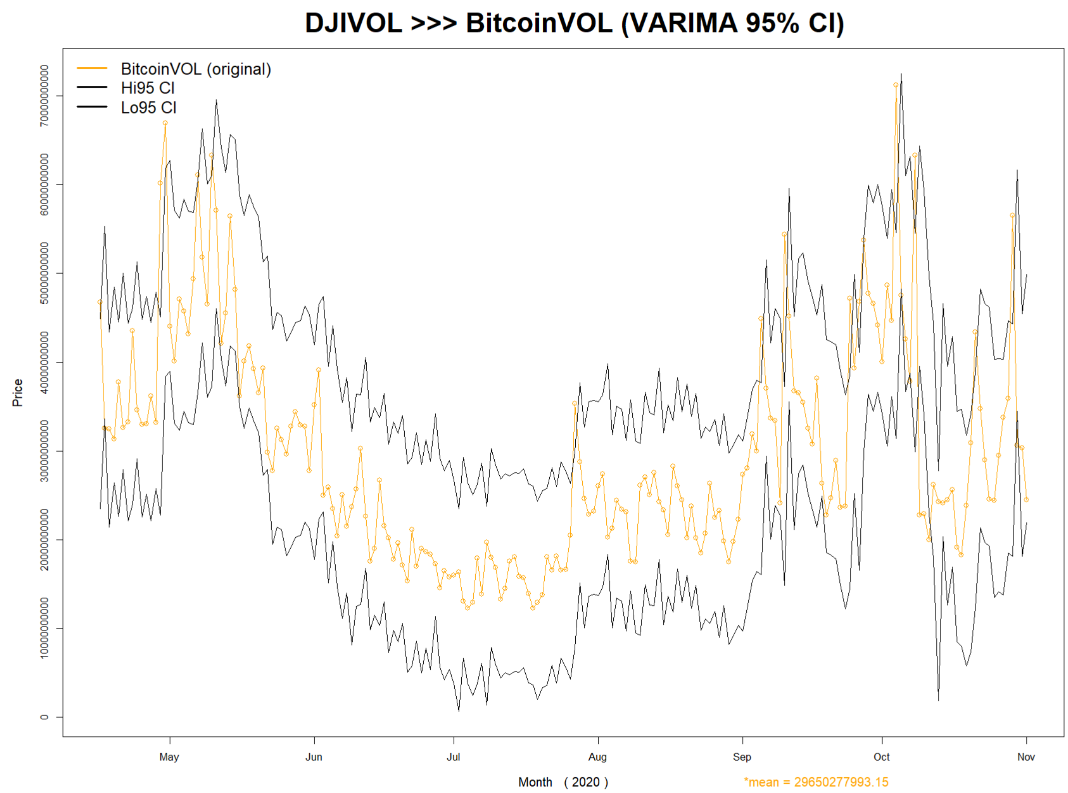
DJI >>> BIT (VARIMA 95% CI)

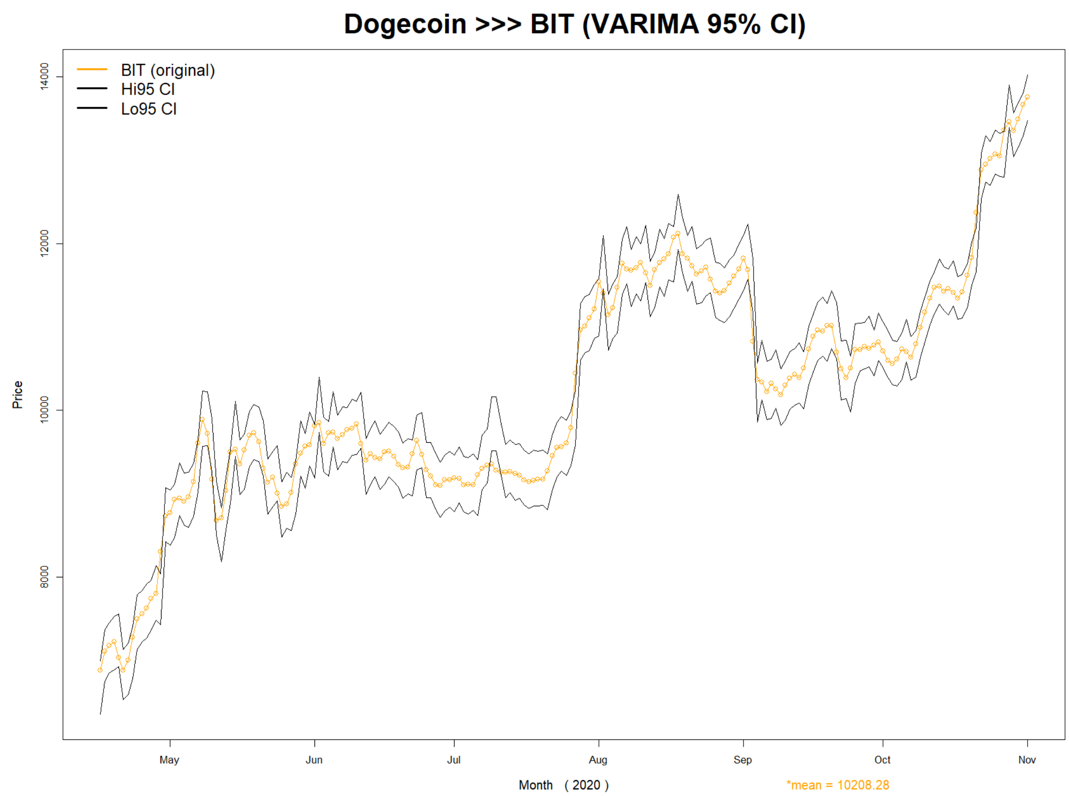
DJI >>> BitcoinVOL (VARIMA 95% CI)



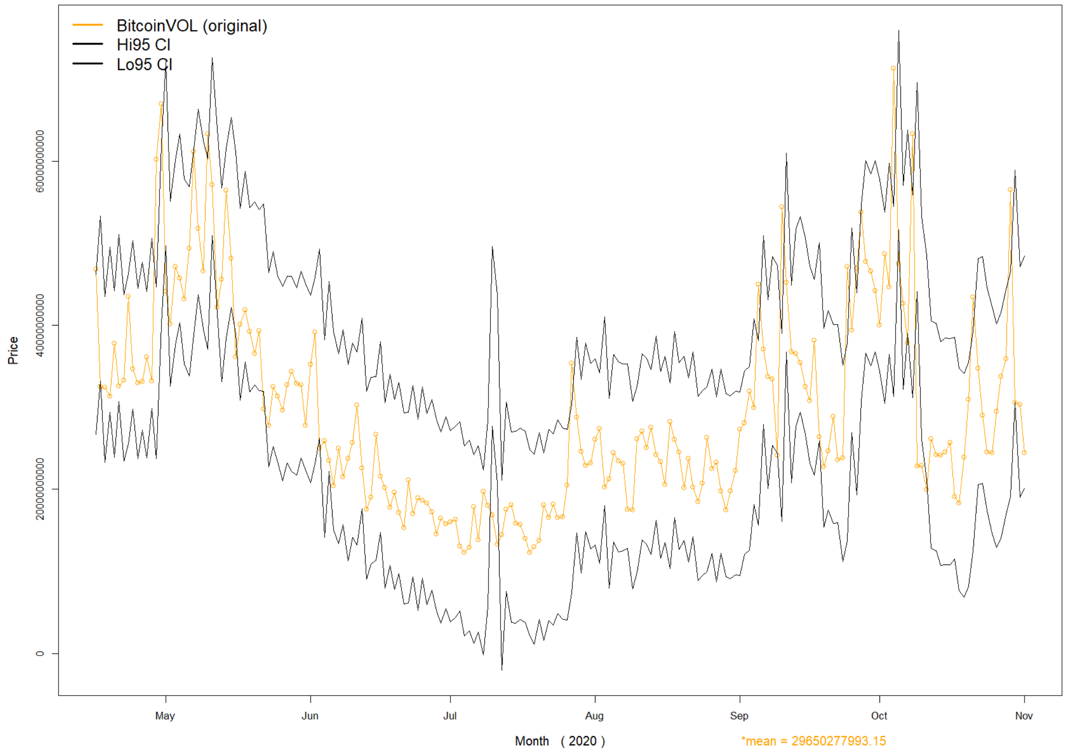
DJIVOL >>> BIT (VARIMA 95% CI)




DJIVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
10000000000
20000000000
30000000000
40000000000
50000000000
60000000000
70000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

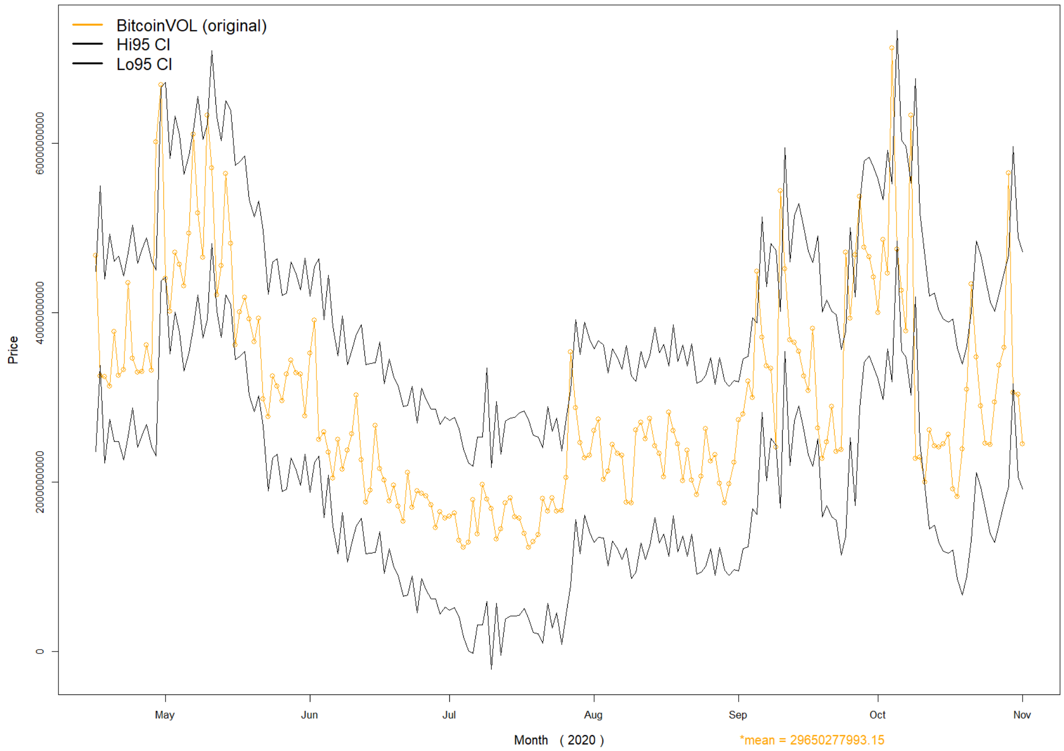

Dogecoin >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

Dogecoin >>> BitcoinVOL (VARIMA 95% CI)

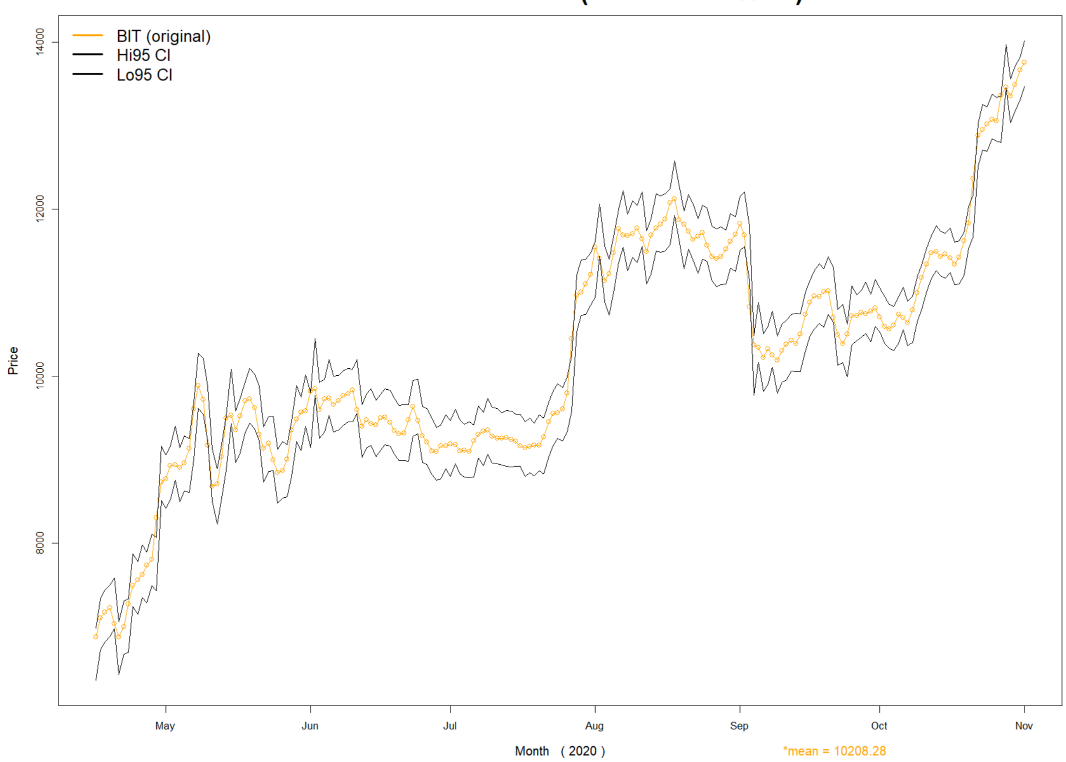


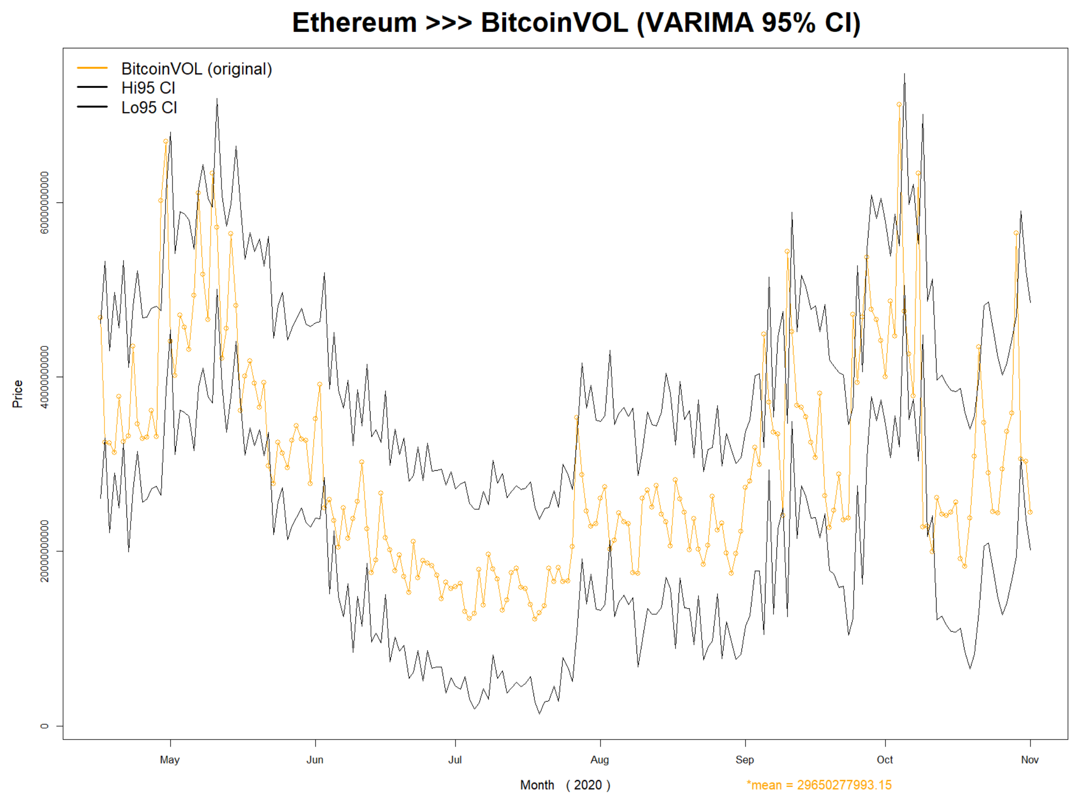
DogecoinVOL >>> BIT (VARIMA 95% CI)

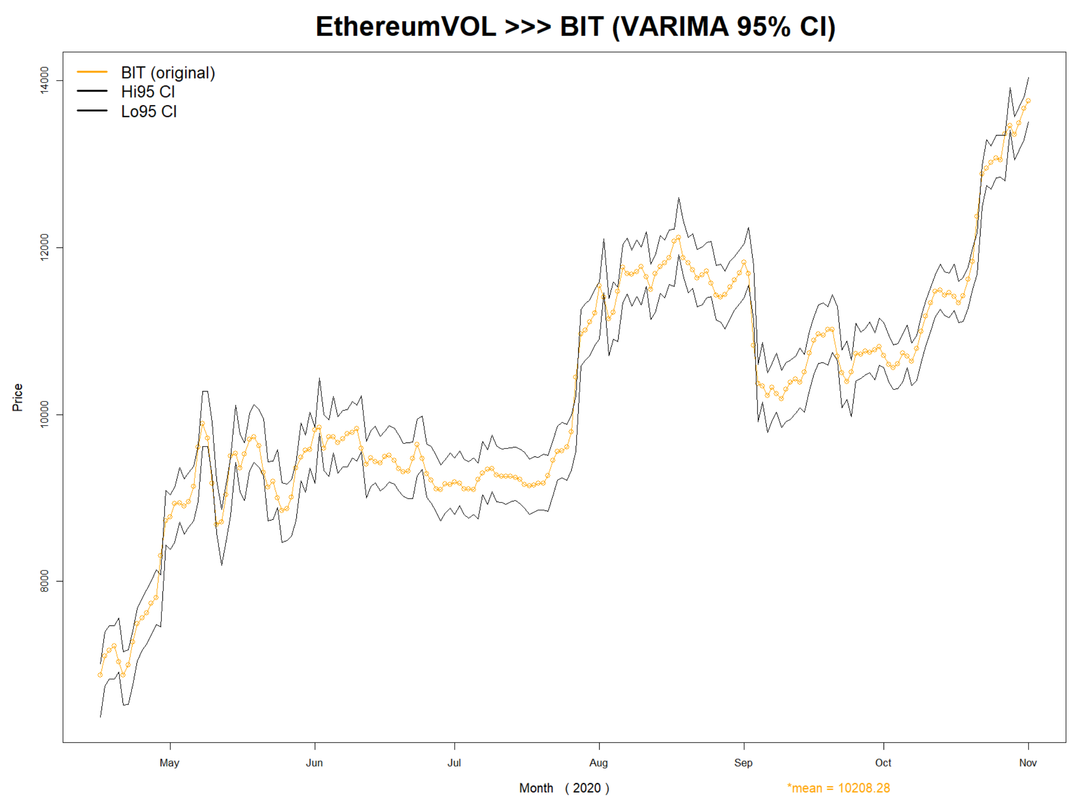
DogecoinVOL >>> BitcoinVOL (VARIMA 95% CI)

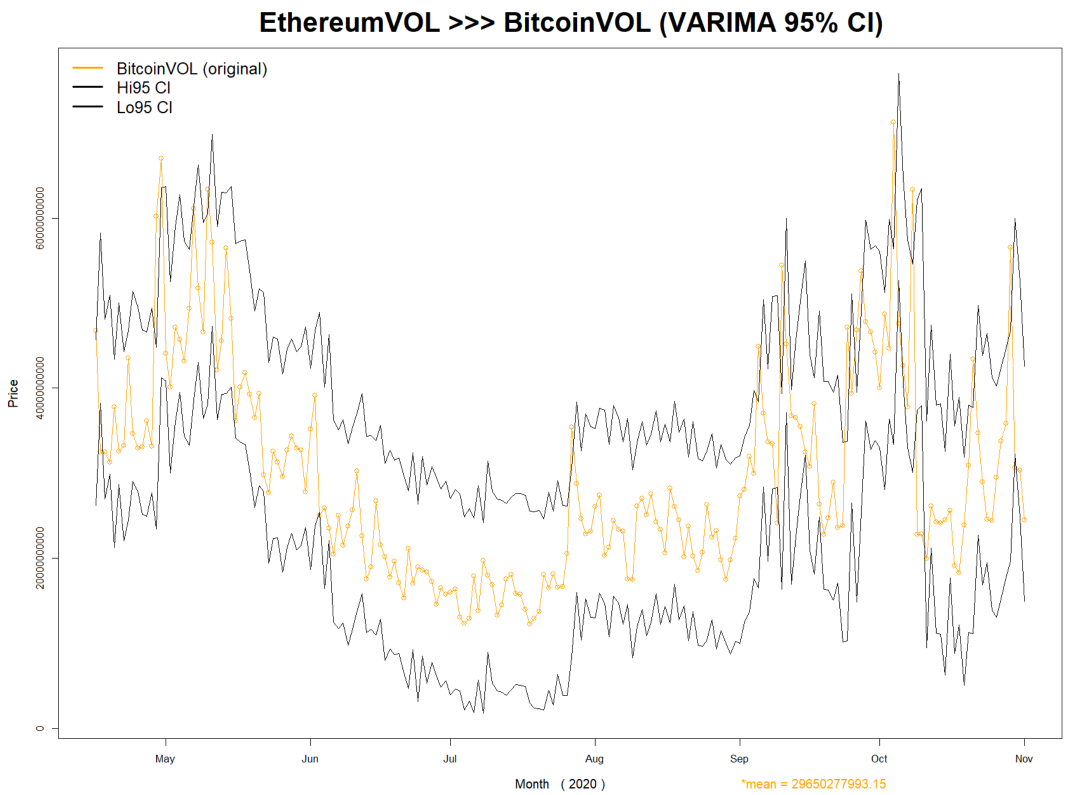


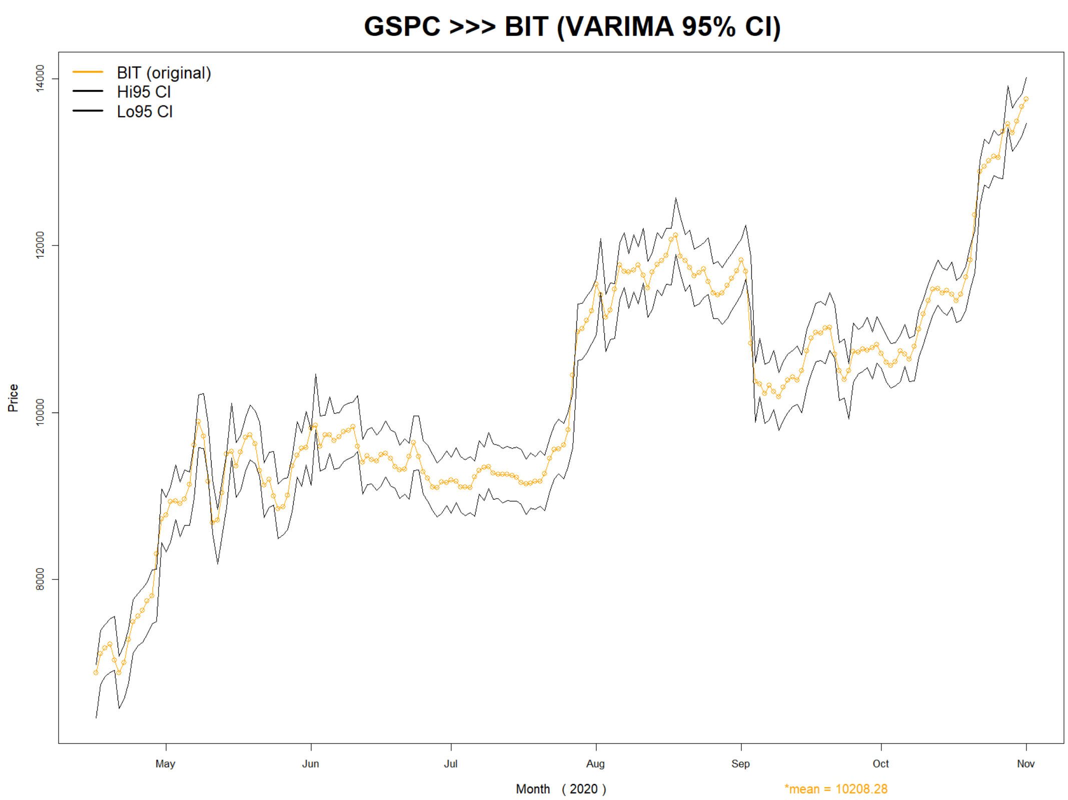
Ethereum >>> BIT (VARIMA 95% CI)



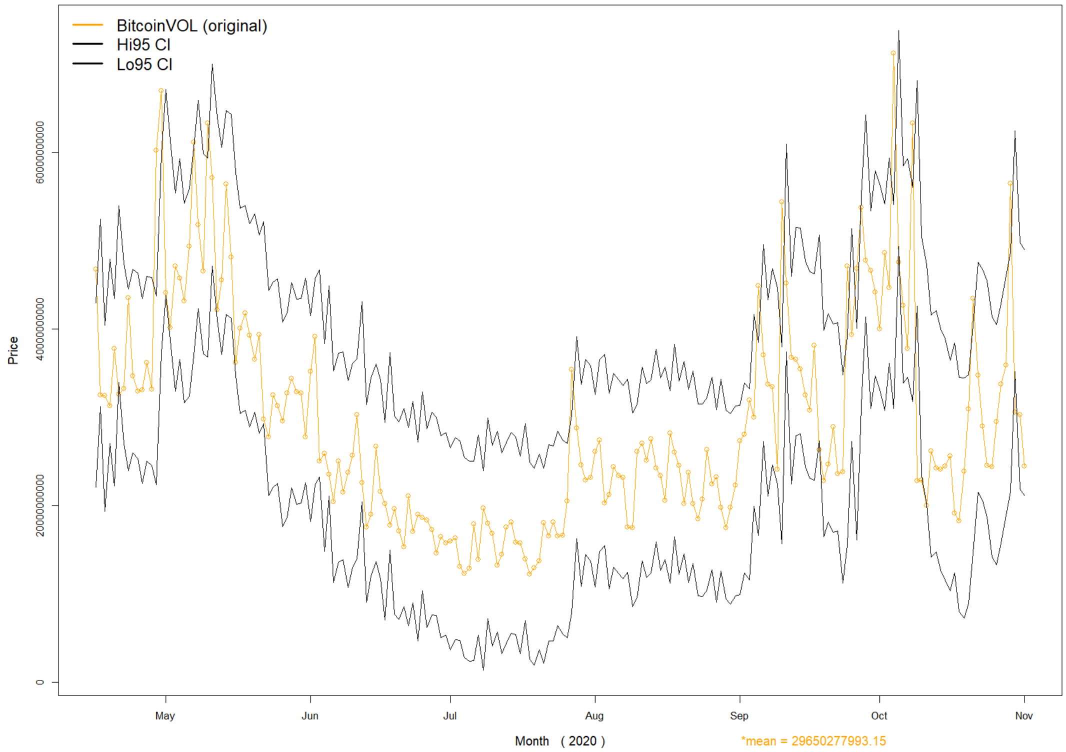

Ethereum >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15


EthereumVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

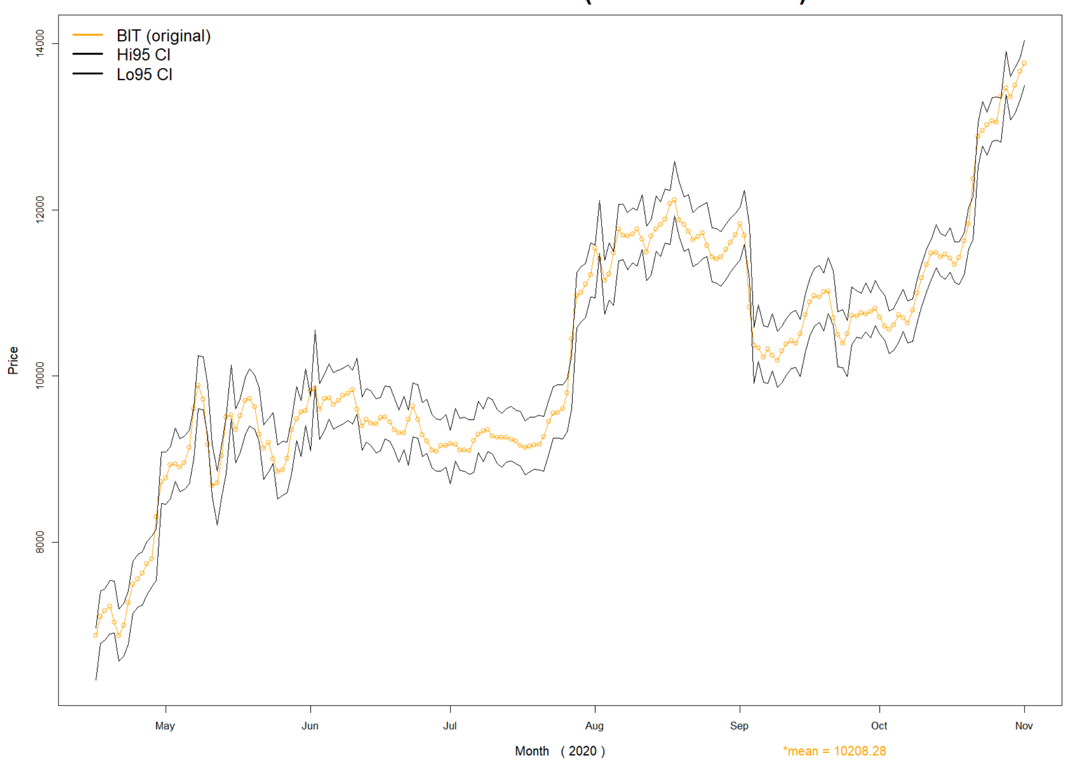

EthereumVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

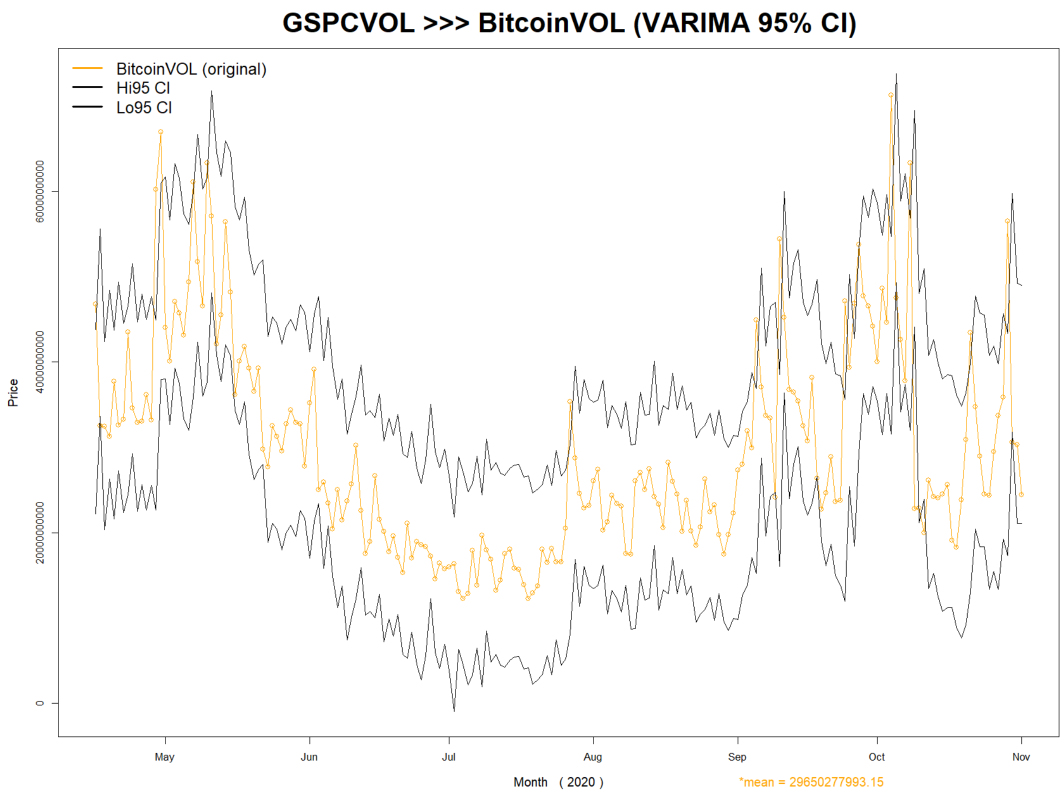

GSPC >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

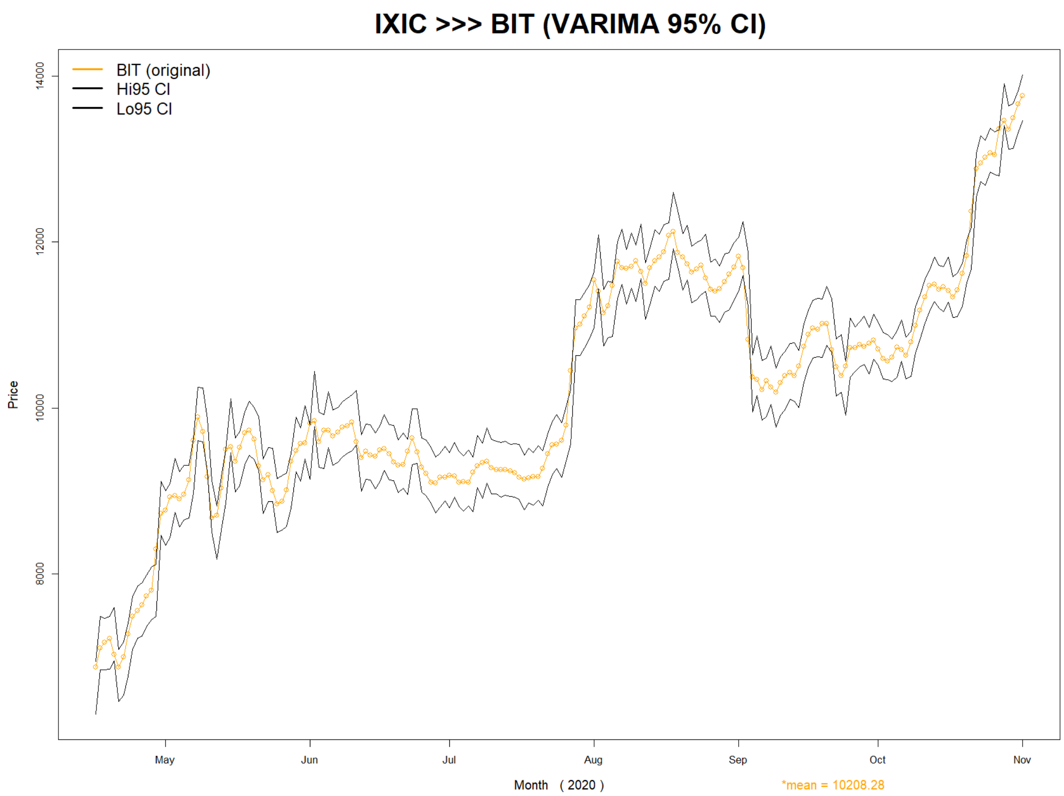
GSPC >>> BitcoinVOL (VARIMA 95% CI)



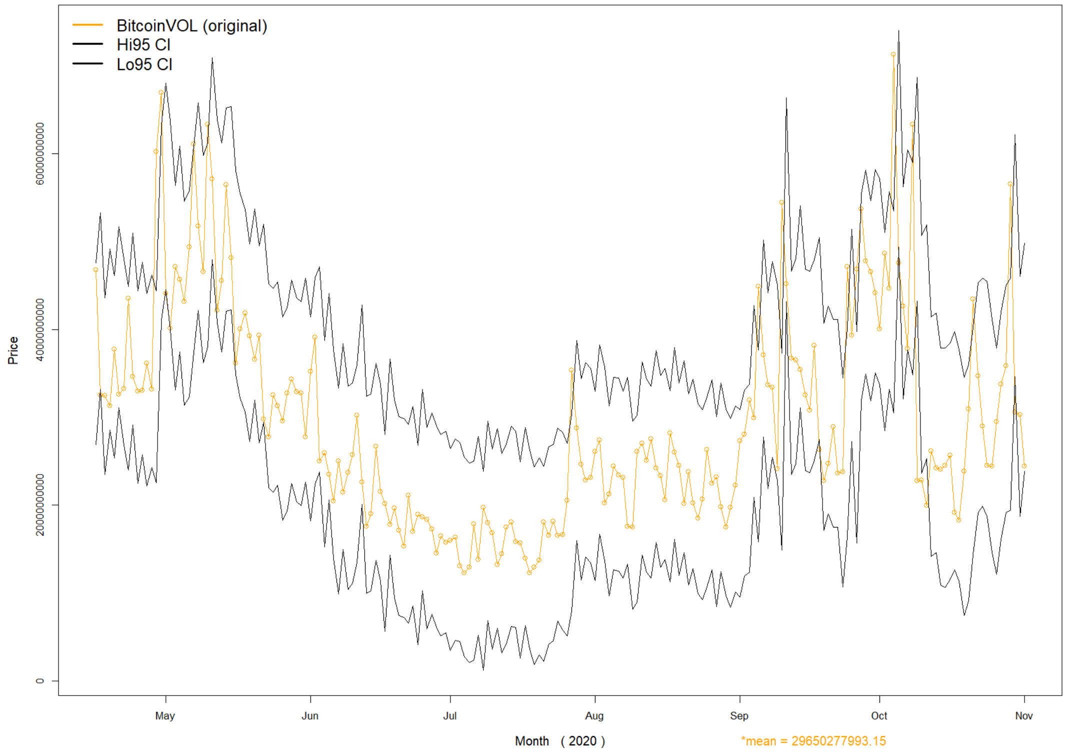
GSPCVOL >>> BIT (VARIMA 95% CI)



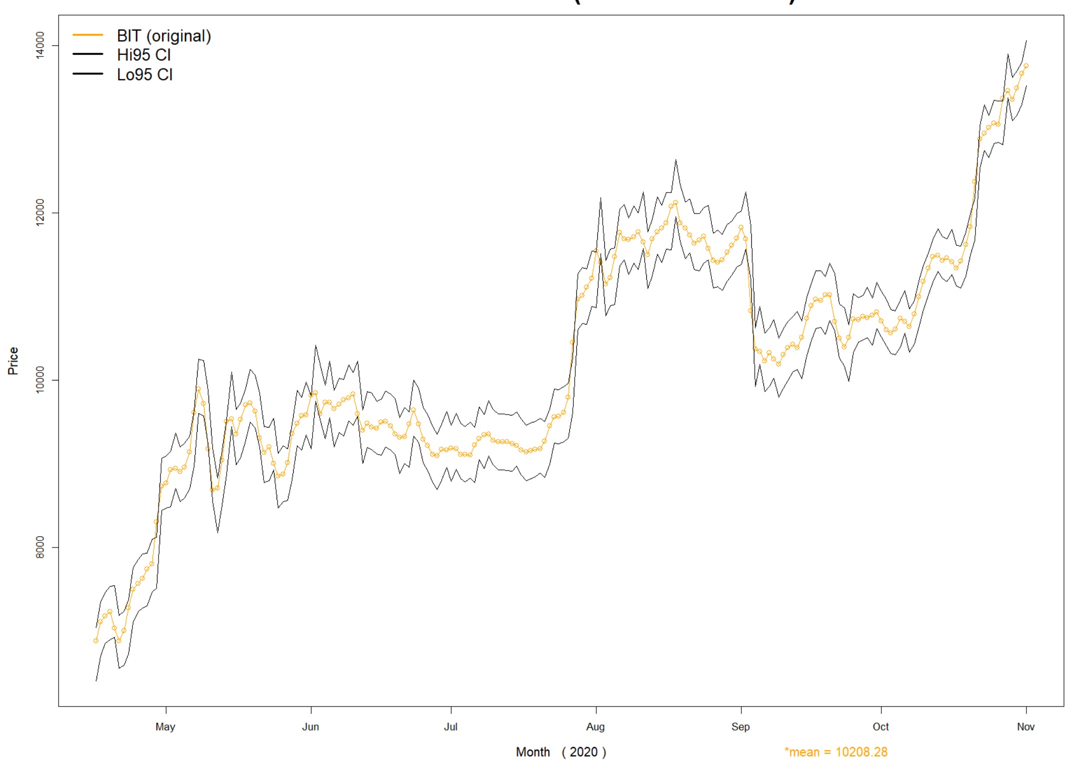

GSPCVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15


IXIC >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

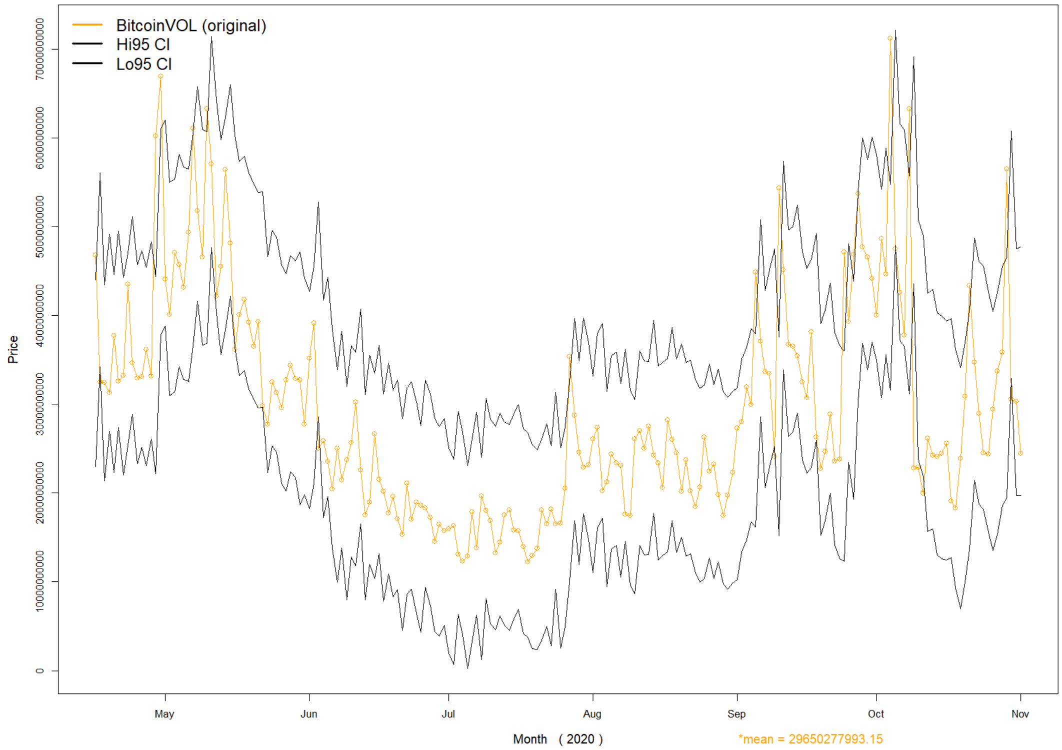
IXIC >>> BitcoinVOL (VARIMA 95% CI)



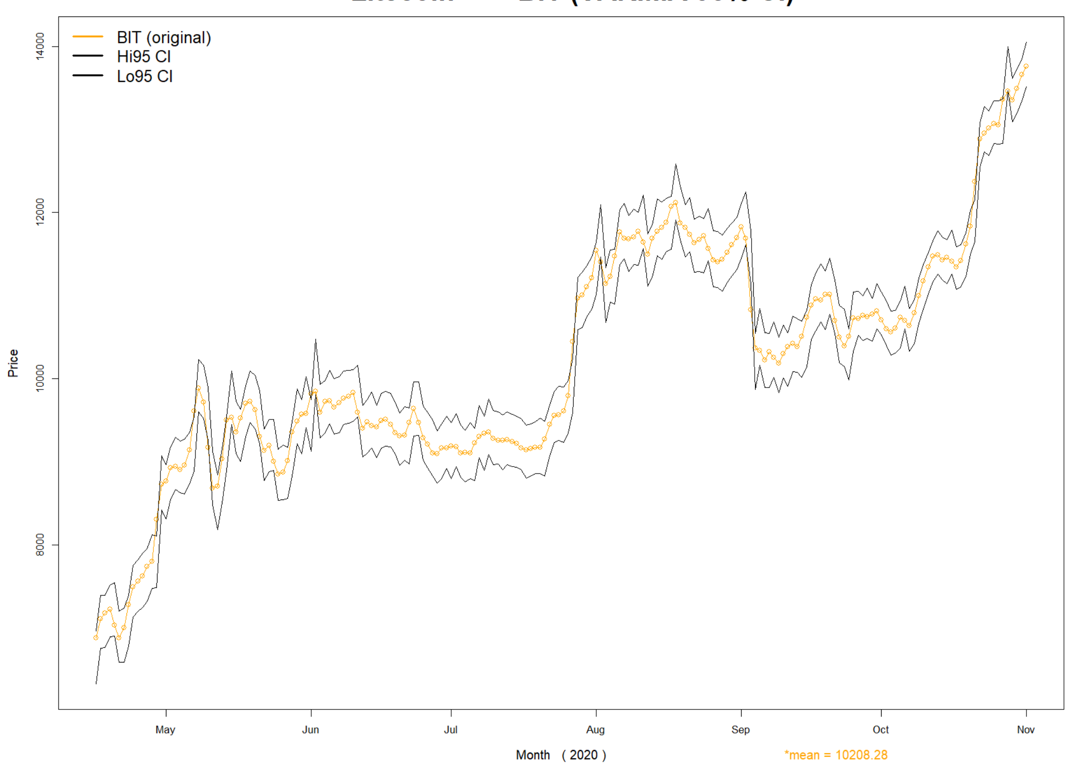
IXICVOL >>> BIT (VARIMA 95% CI)

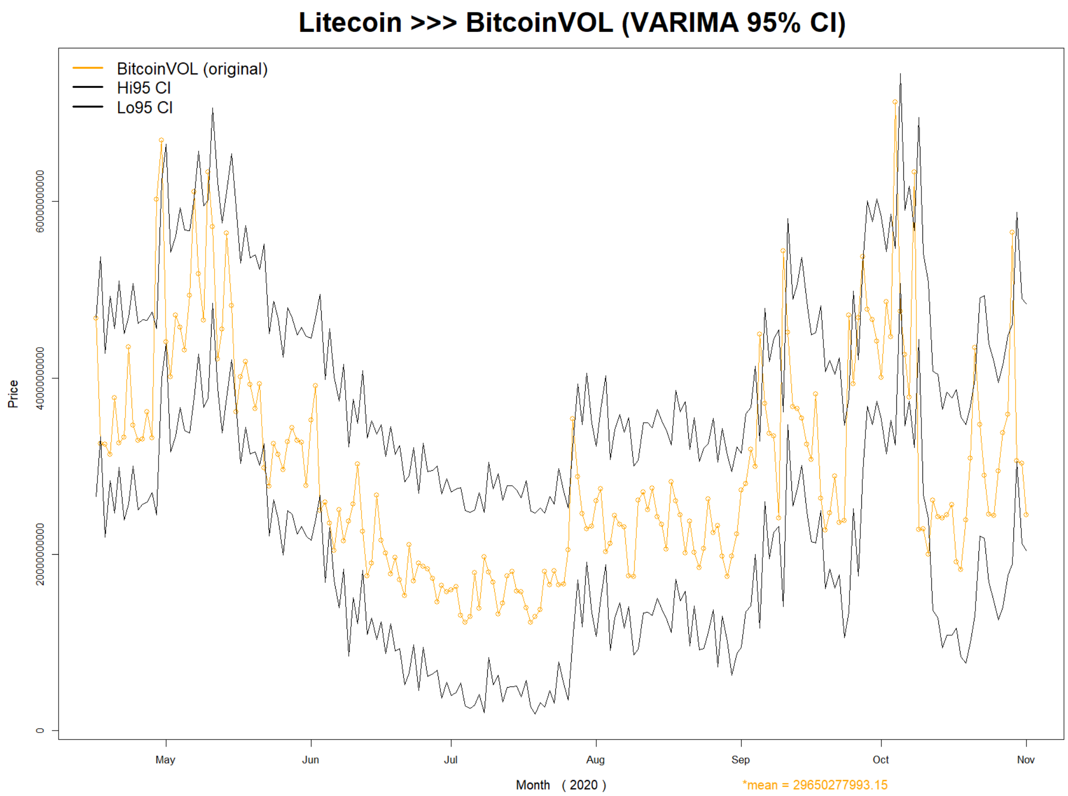


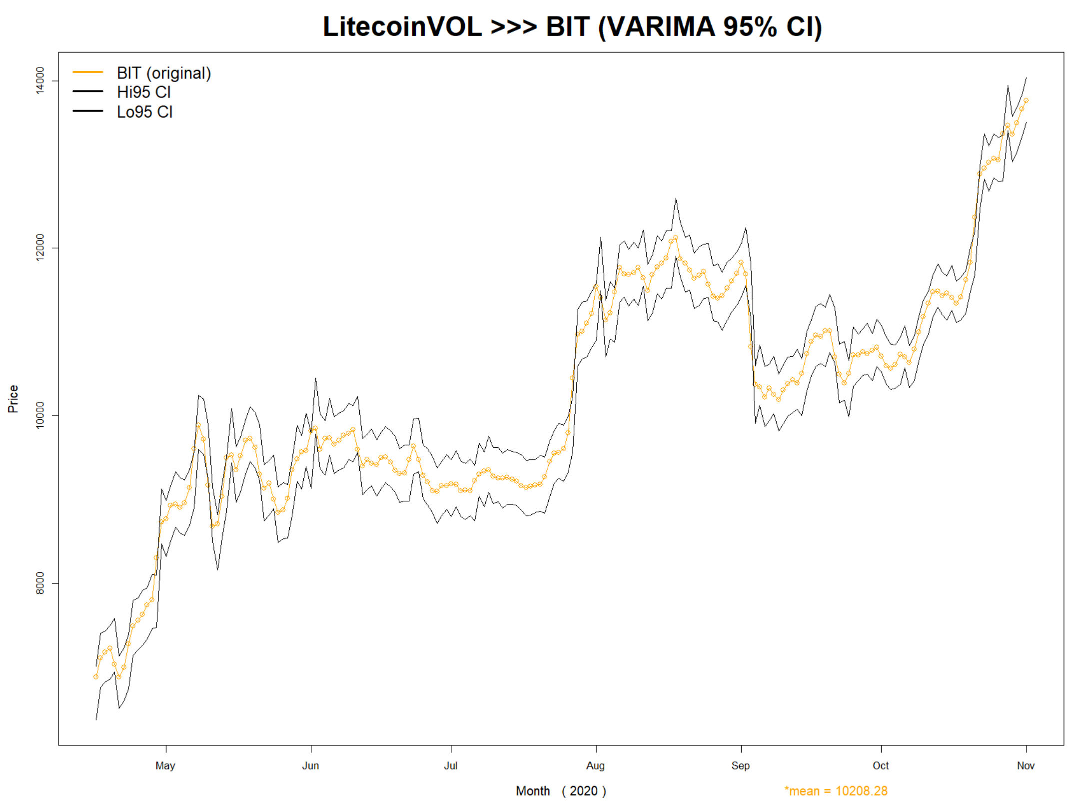
IXICVOL >>> BitcoinVOL (VARIMA 95% CI)

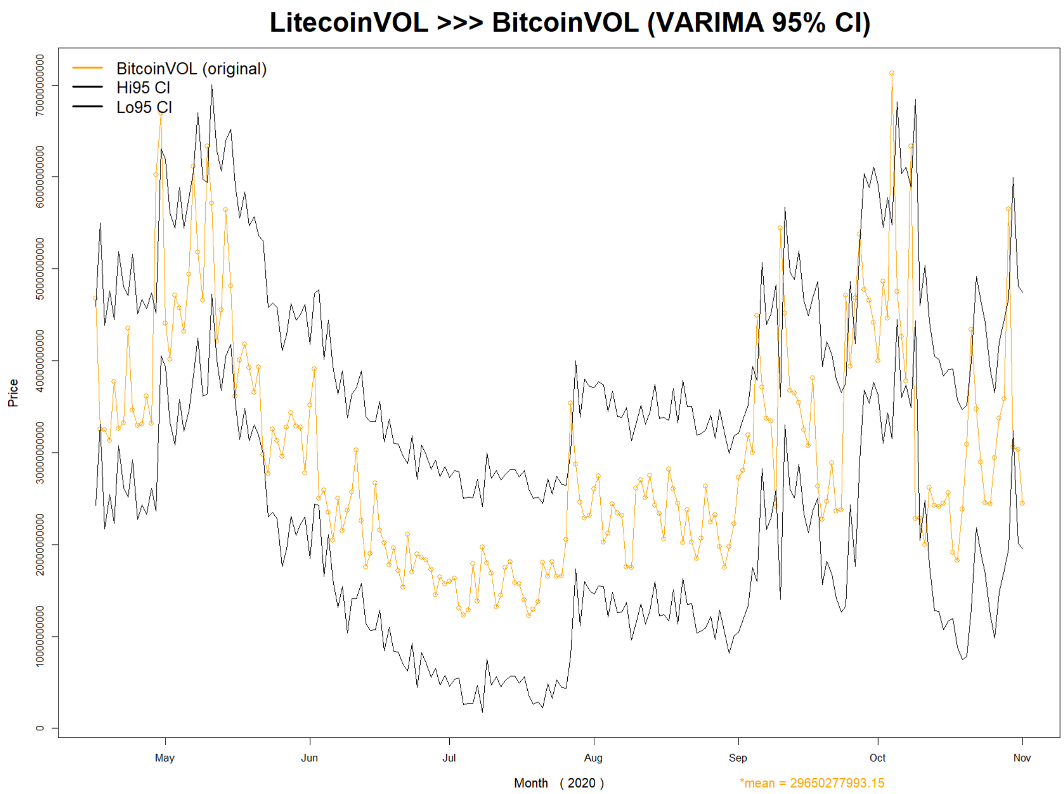


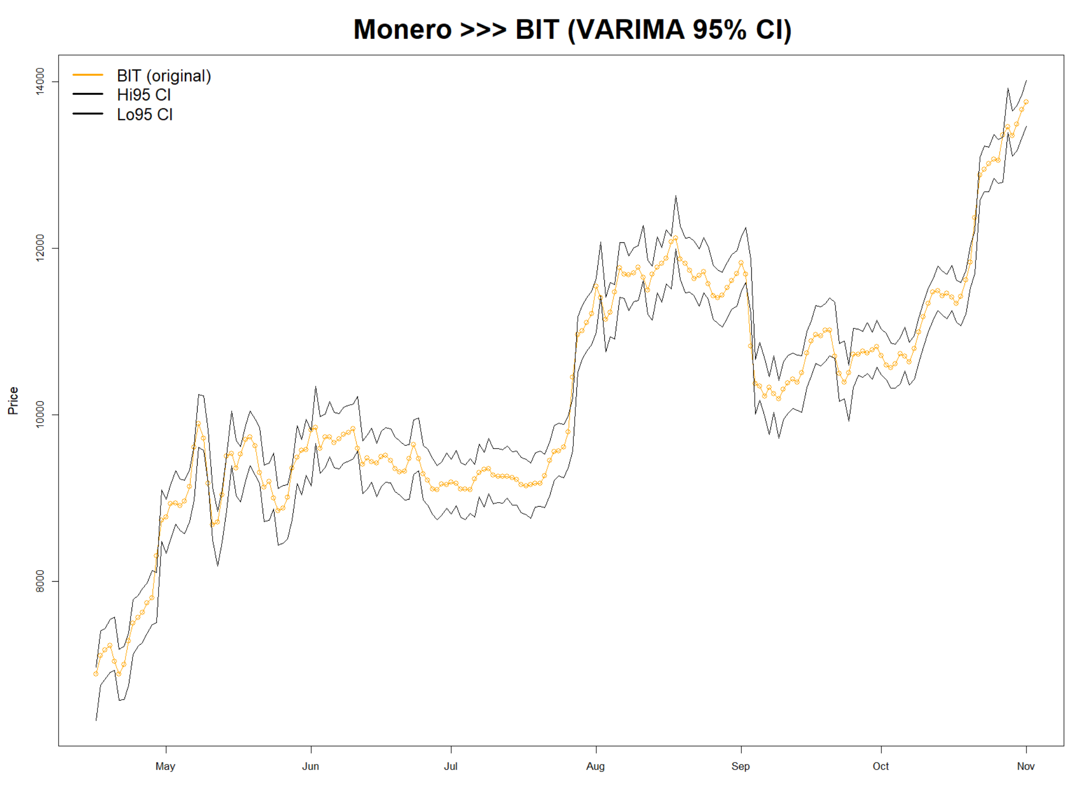
Litecoin >>> BIT (VARIMA 95% CI)



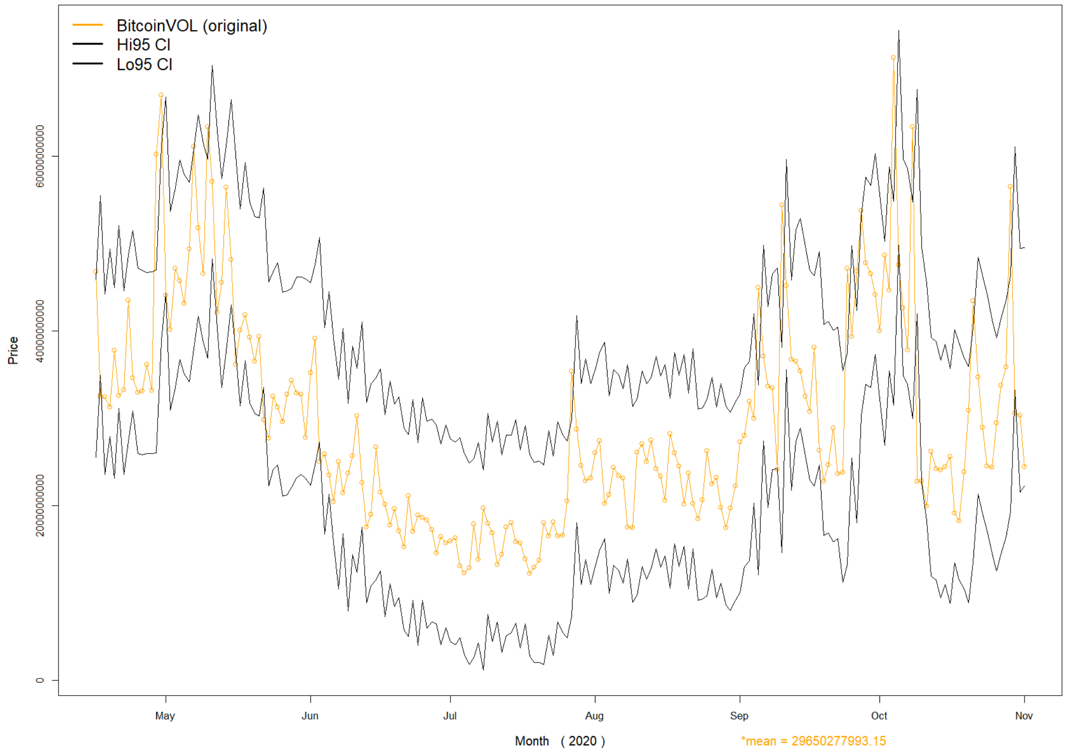

Litecoin >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15


LitecoinVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

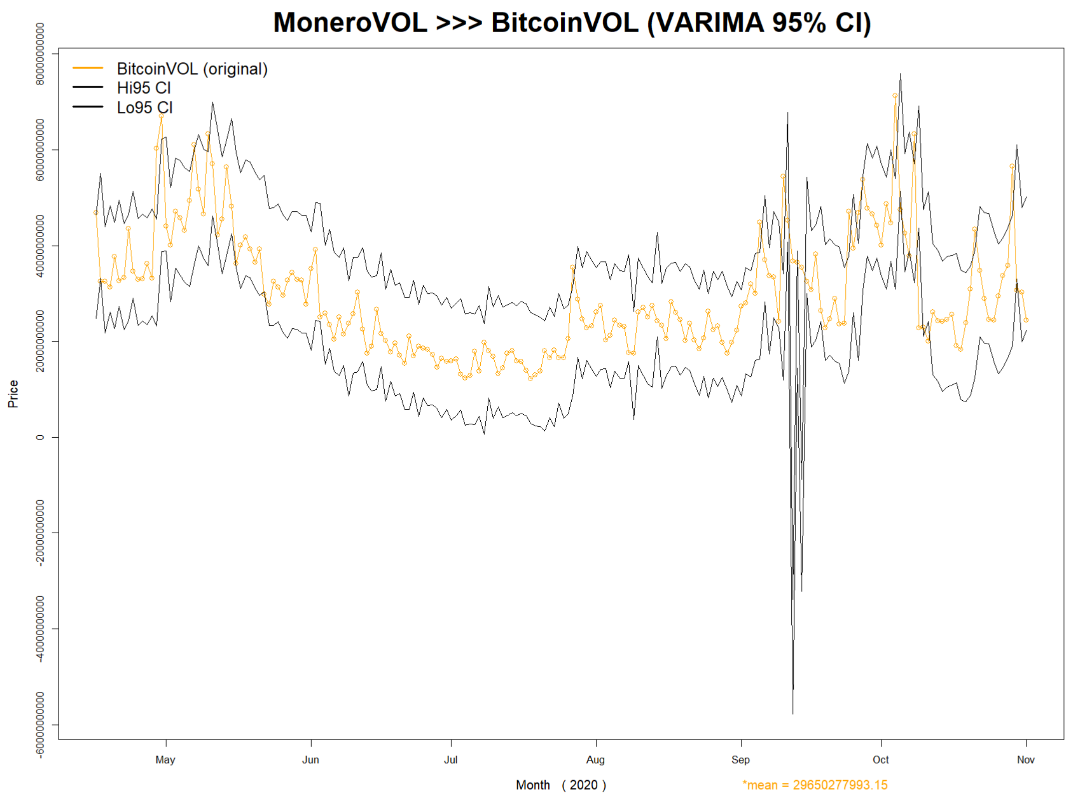

LitecoinVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
10000000000
20000000000
30000000000
40000000000
50000000000
60000000000
70000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15


Monero >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

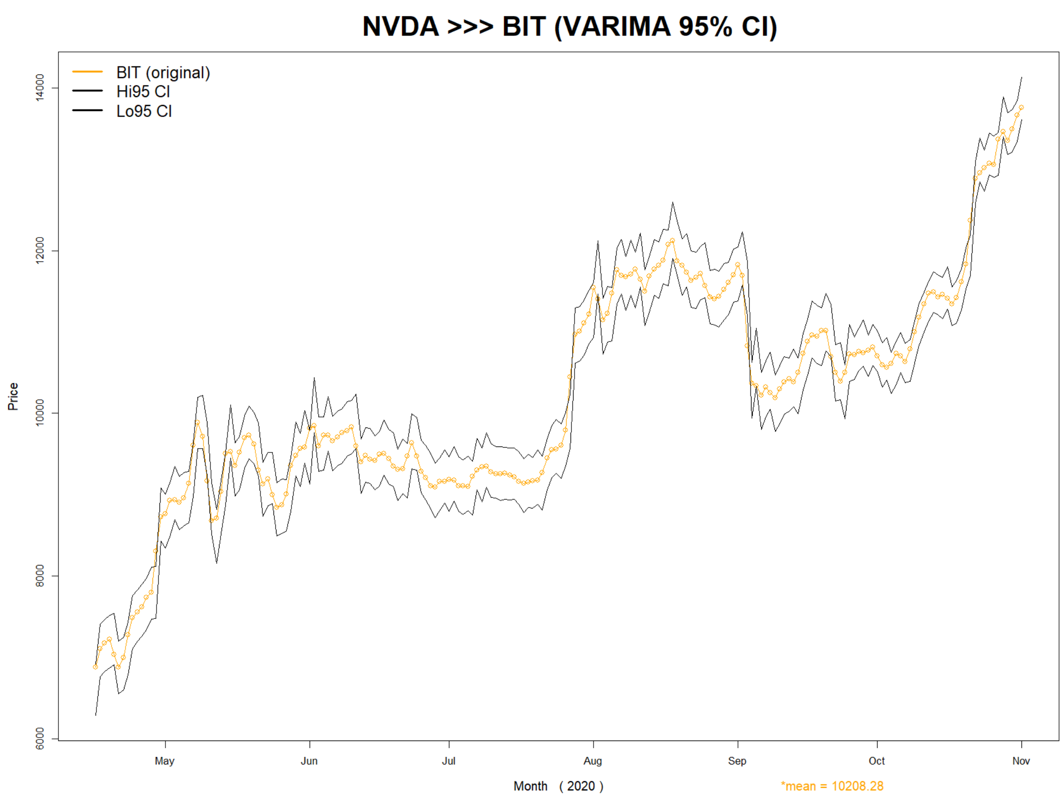
Monero >>> BitcoinVOL (VARIMA 95% CI)



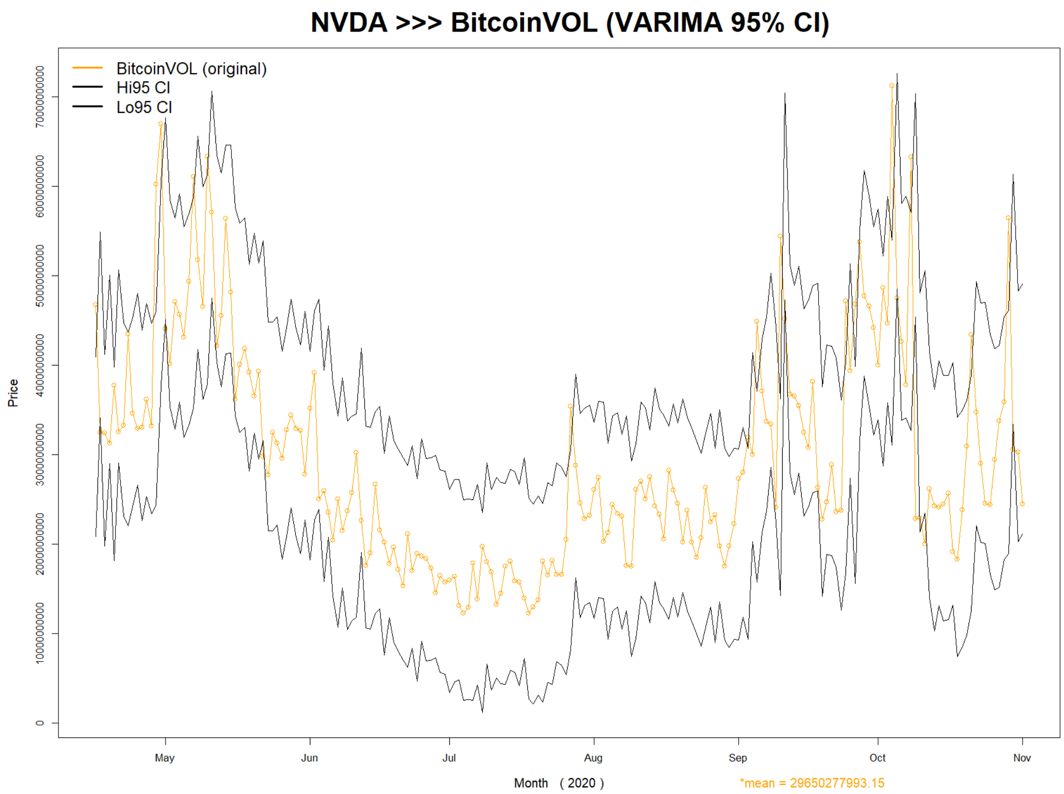
MoneroVOL >>> BIT (VARIMA 95% CI)

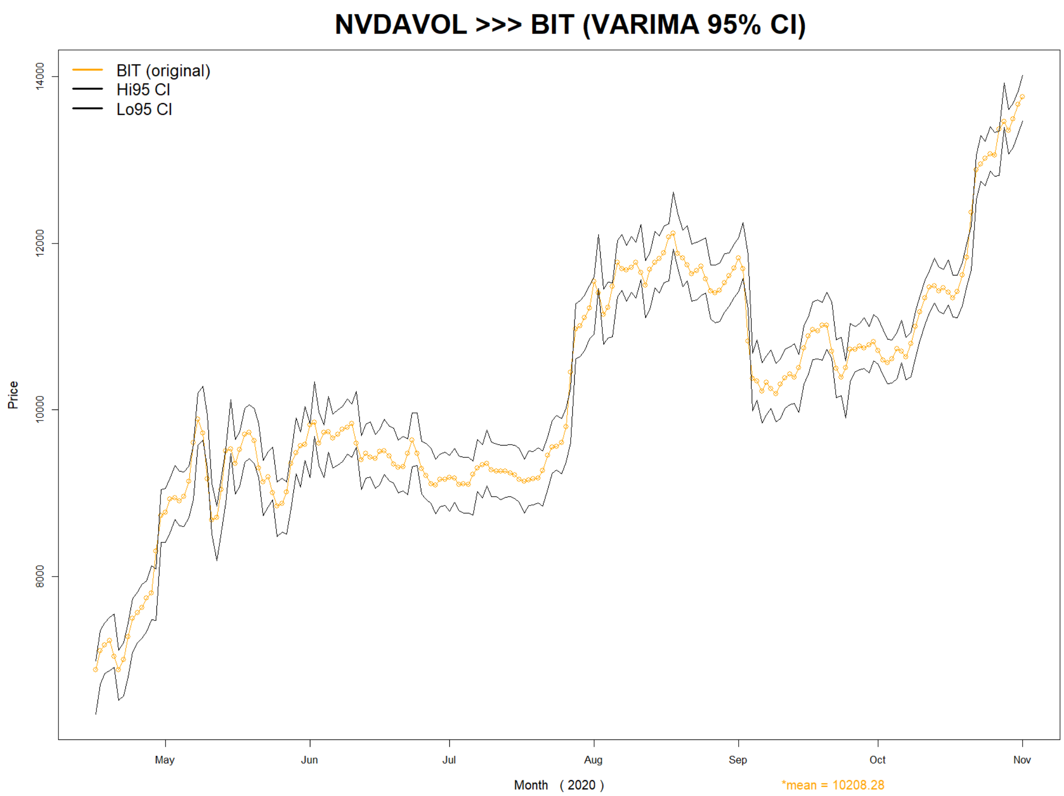


MoneroVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
80000000000
60000000000
40000000000
20000000000
0
-20000000000
-40000000000
-60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

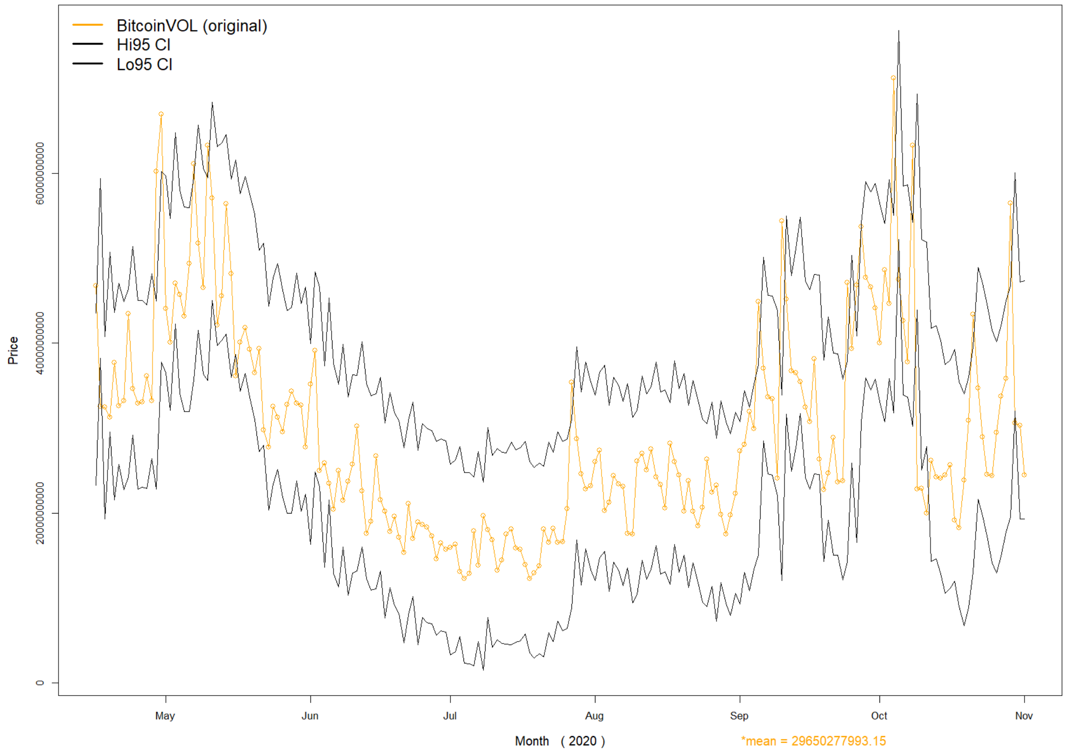


NVDA >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
14000
12000
10000
8000
6000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28


NVDA >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
10000000000
20000000000
30000000000
40000000000
50000000000
60000000000
70000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

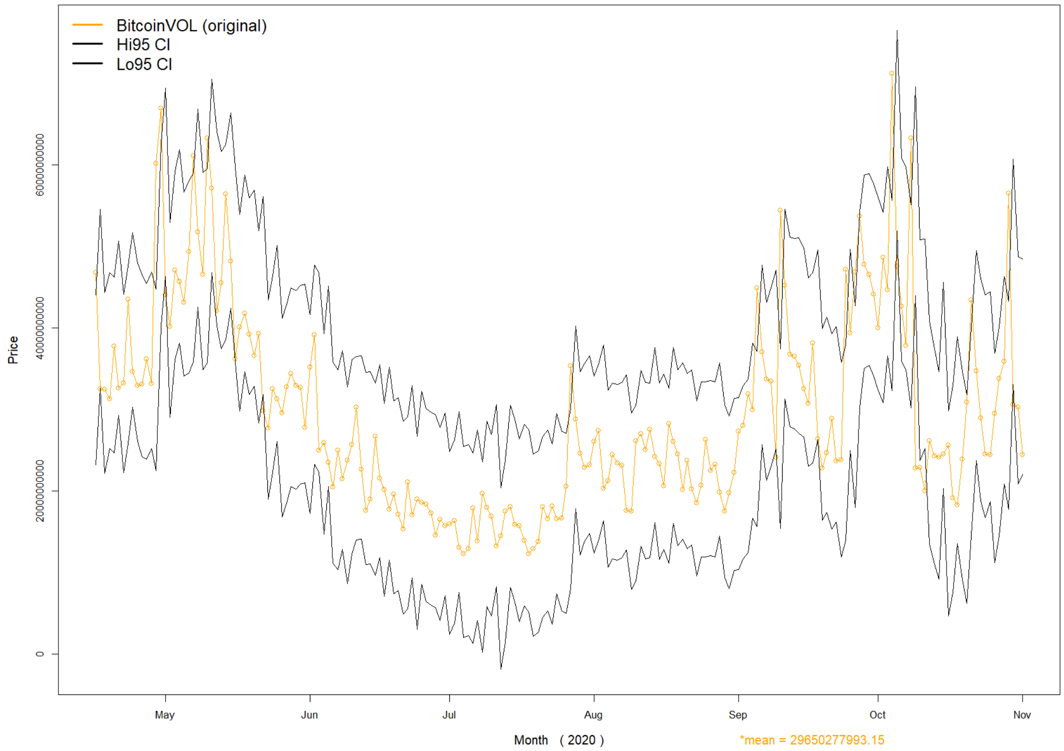

NVDAVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

NVDAVOL >>> BitcoinVOL (VARIMA 95% CI)

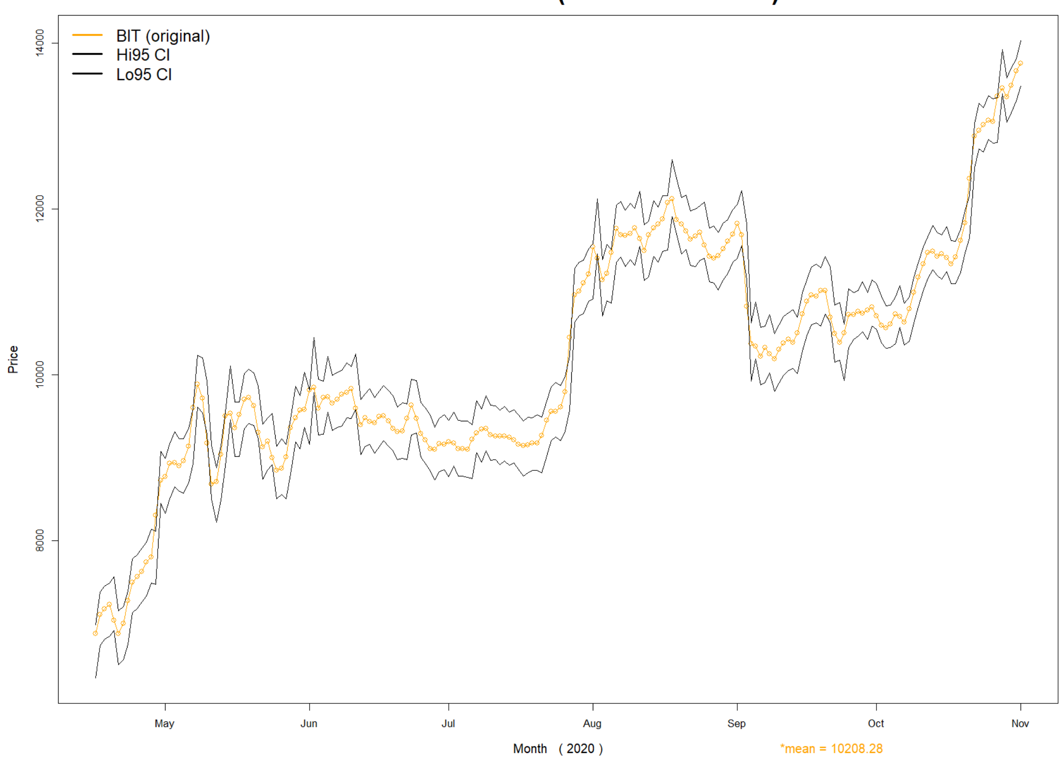


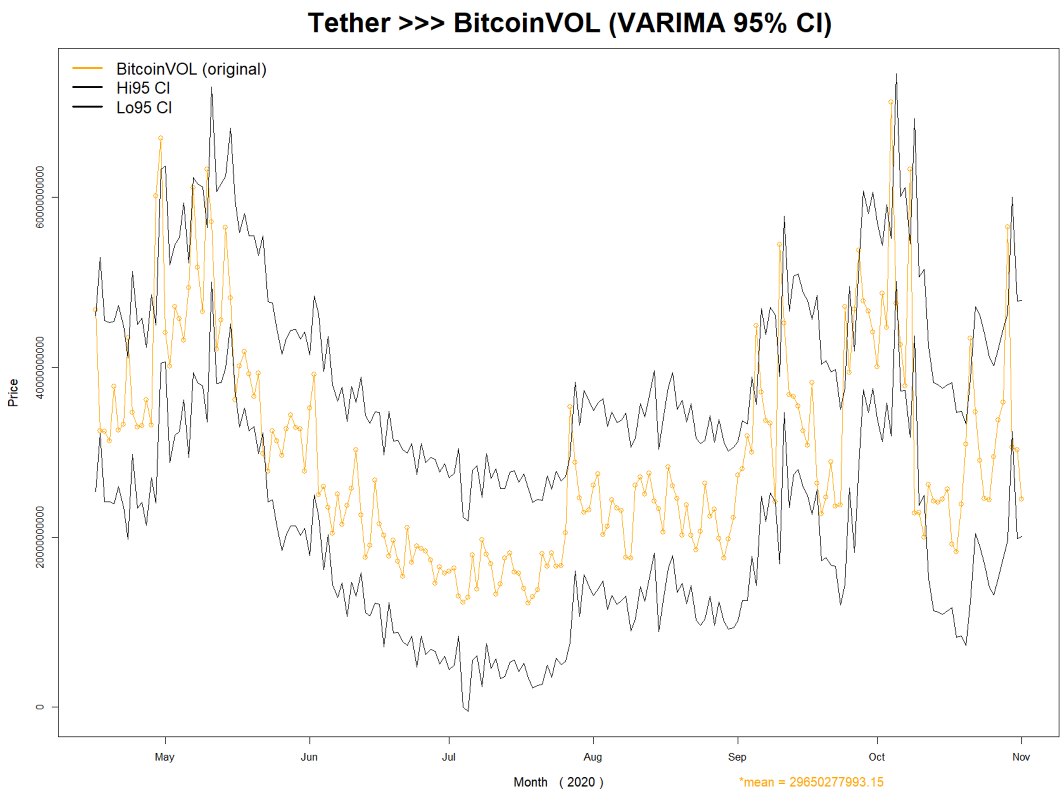
pc1 >>> BIT (VARIMA 95% CI)

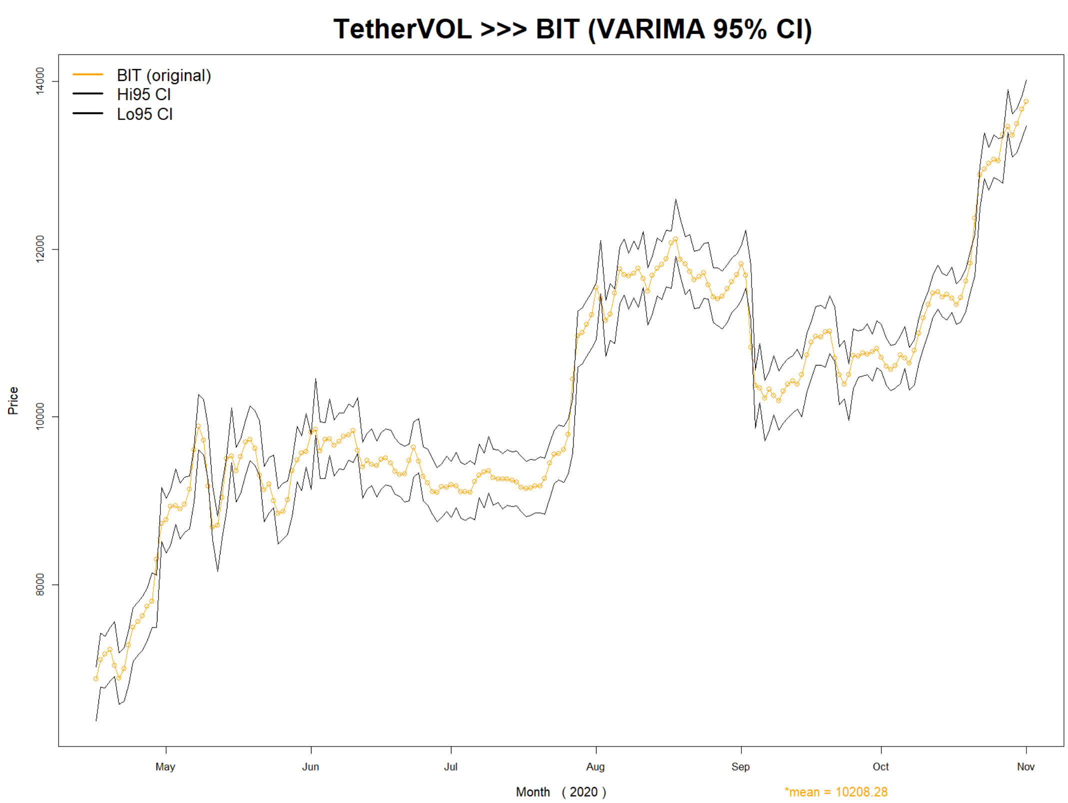
pc1 >>> BitcoinVOL (VARIMA 95% CI)

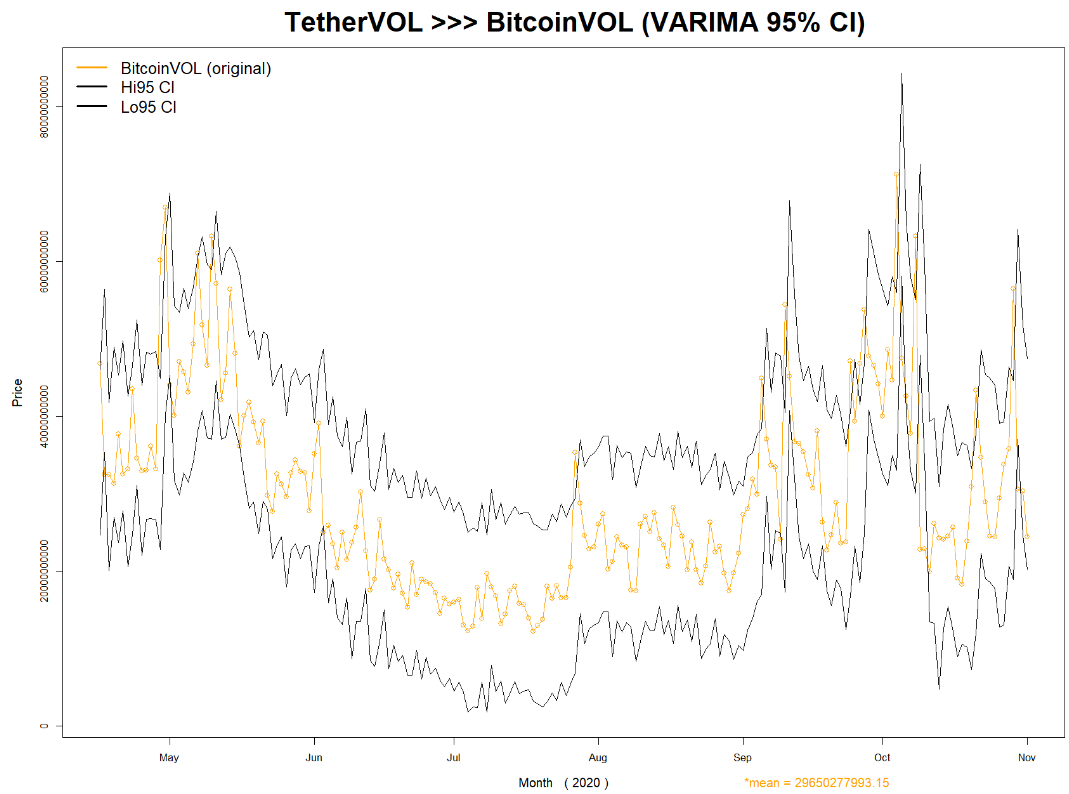


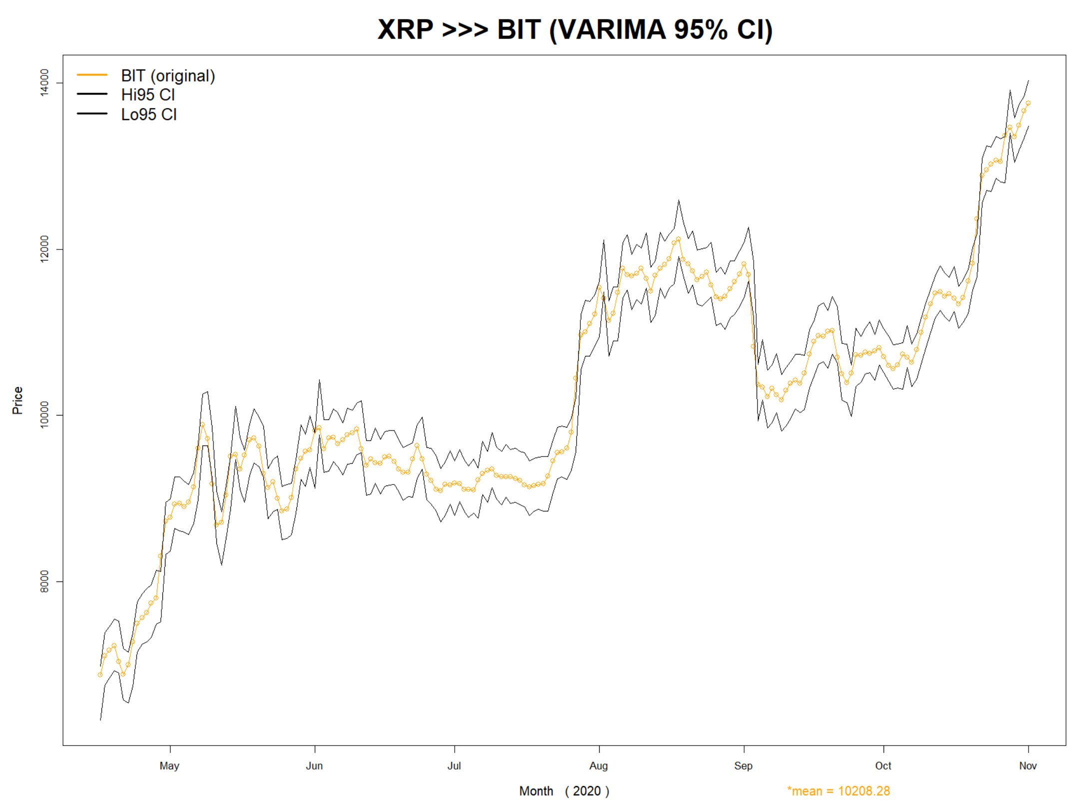
Tether >>> BIT (VARIMA 95% CI)



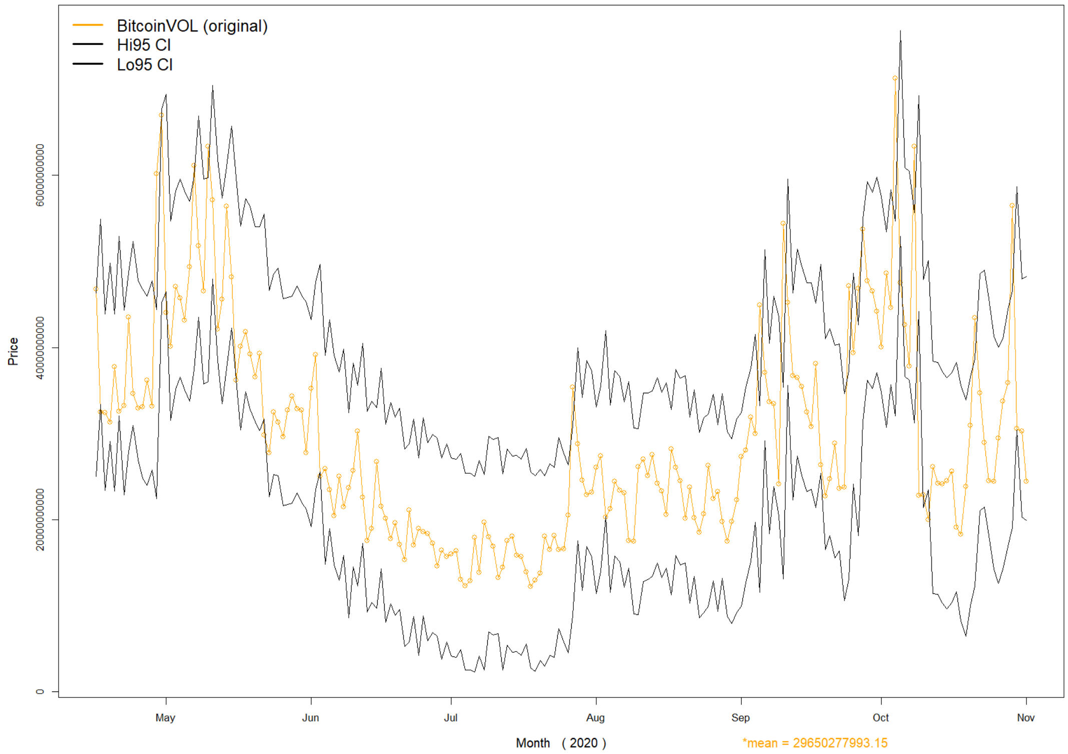

Tether >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15


TetherVOL >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

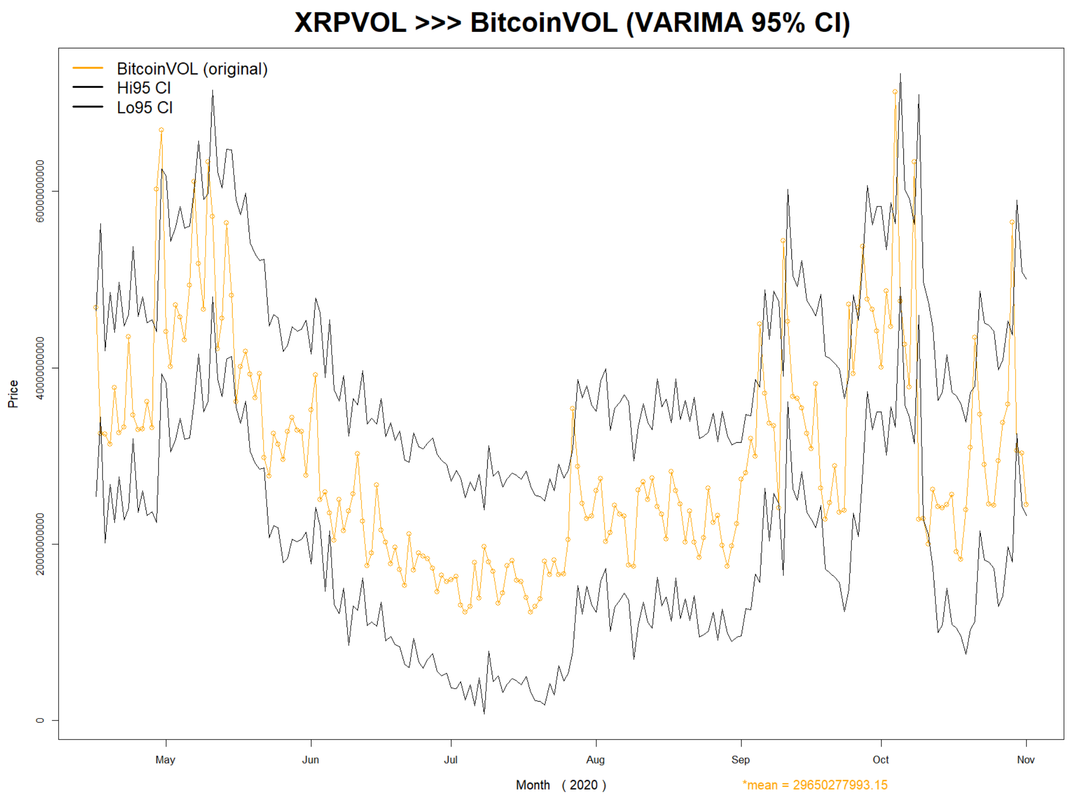

TetherVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
80000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15


XRP >>> BIT (VARIMA 95% CI)
BIT (original)
Hi95 CI
Lo95 CI
Price
8000
10000
12000
14000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 10208.28

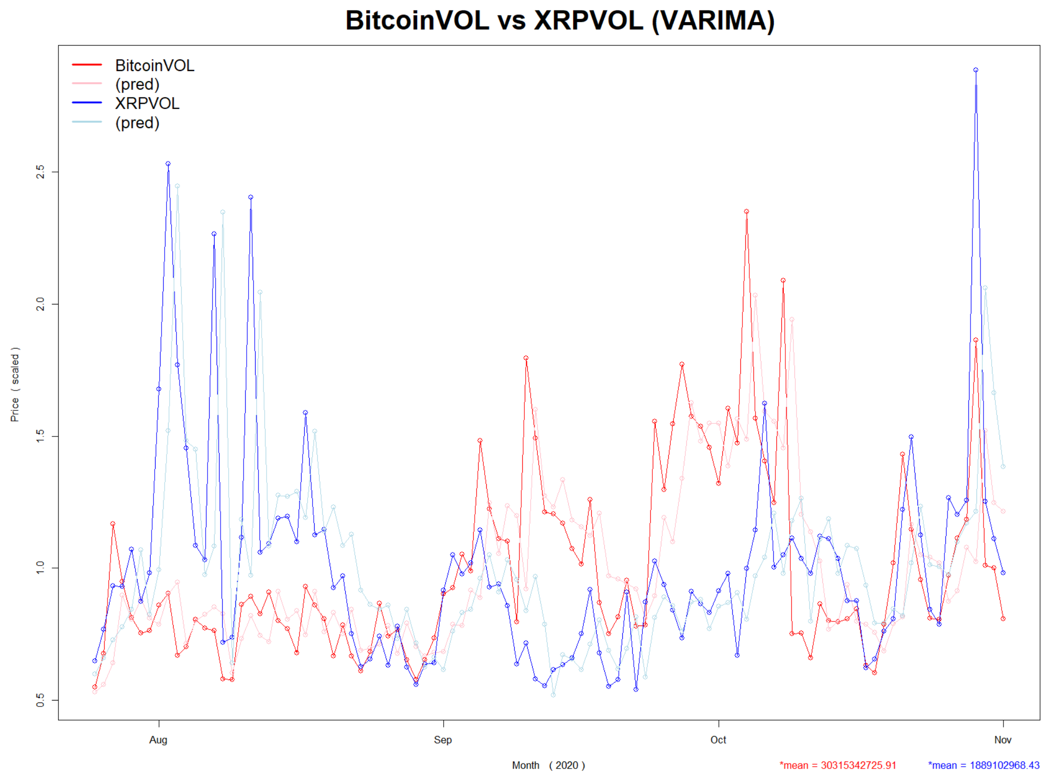
XRP >>> BitcoinVOL (VARIMA 95% CI)



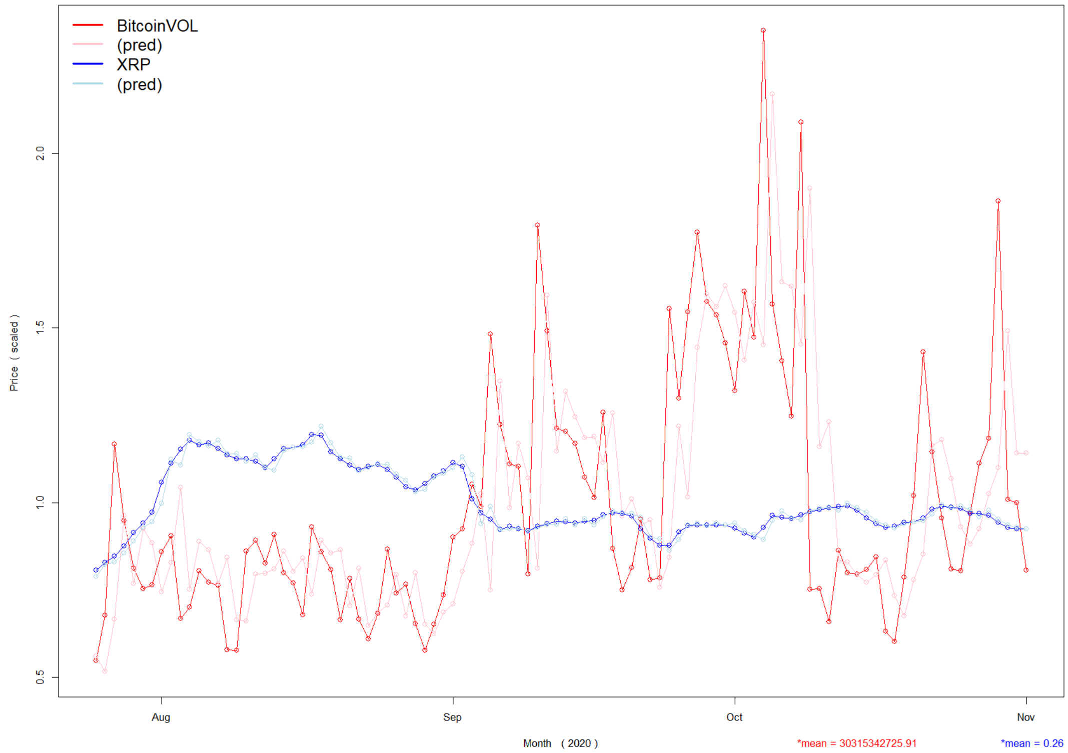
XRPVOL >>> BIT (VARIMA 95% CI)


XRPVOL >>> BitcoinVOL (VARIMA 95% CI)
BitcoinVOL (original)
Hi95 CI
Lo95 CI
Price
0
20000000000
40000000000
60000000000
May
Jun
Jul
Aug
Sep
Oct
Nov
Month (2020)
*mean = 29650277993.15

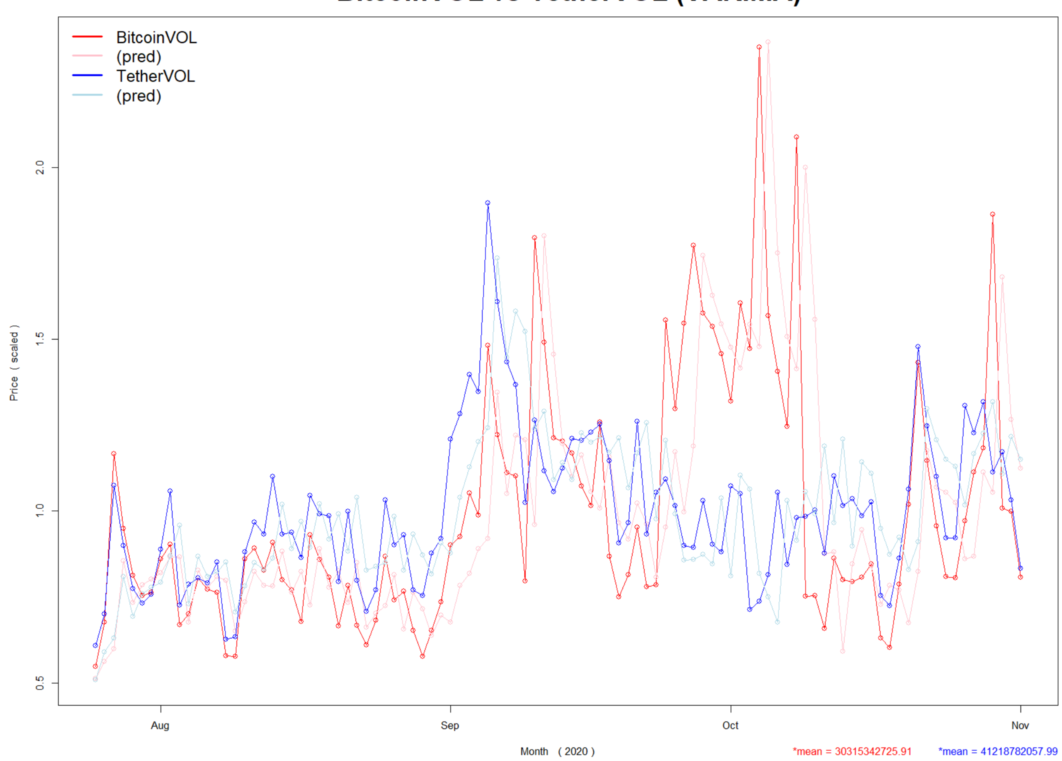
A.4 Forecasted means for ARIMA and VARIMA



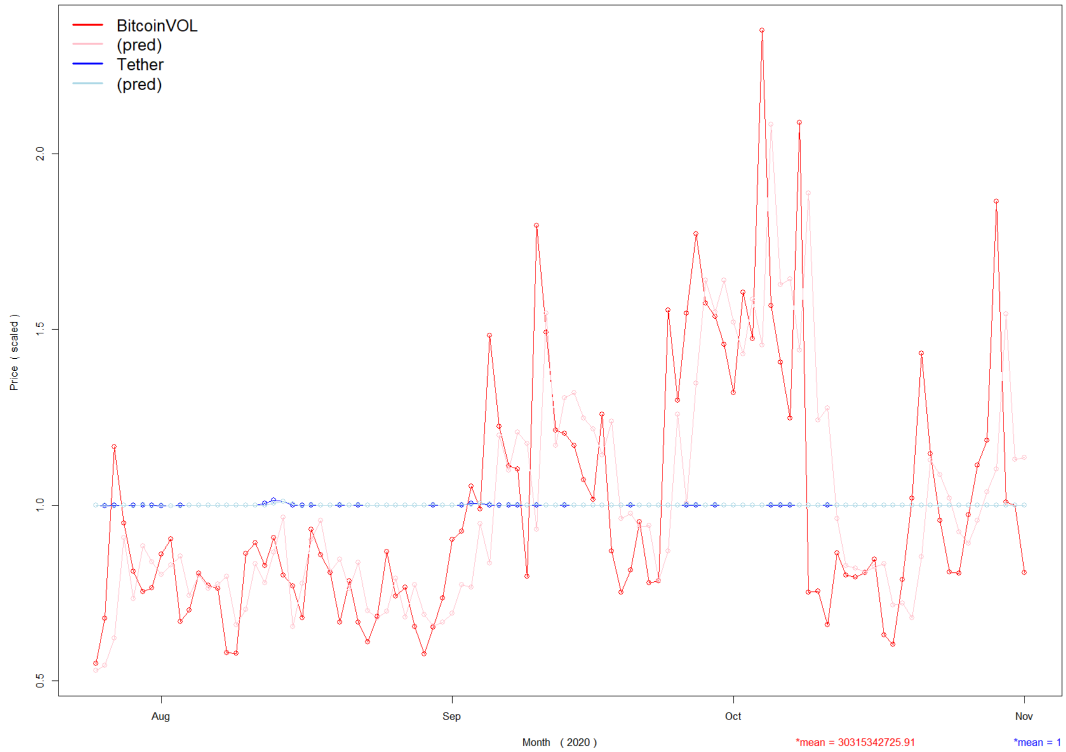
BitcoinVOL vs XRP (VARIMA)



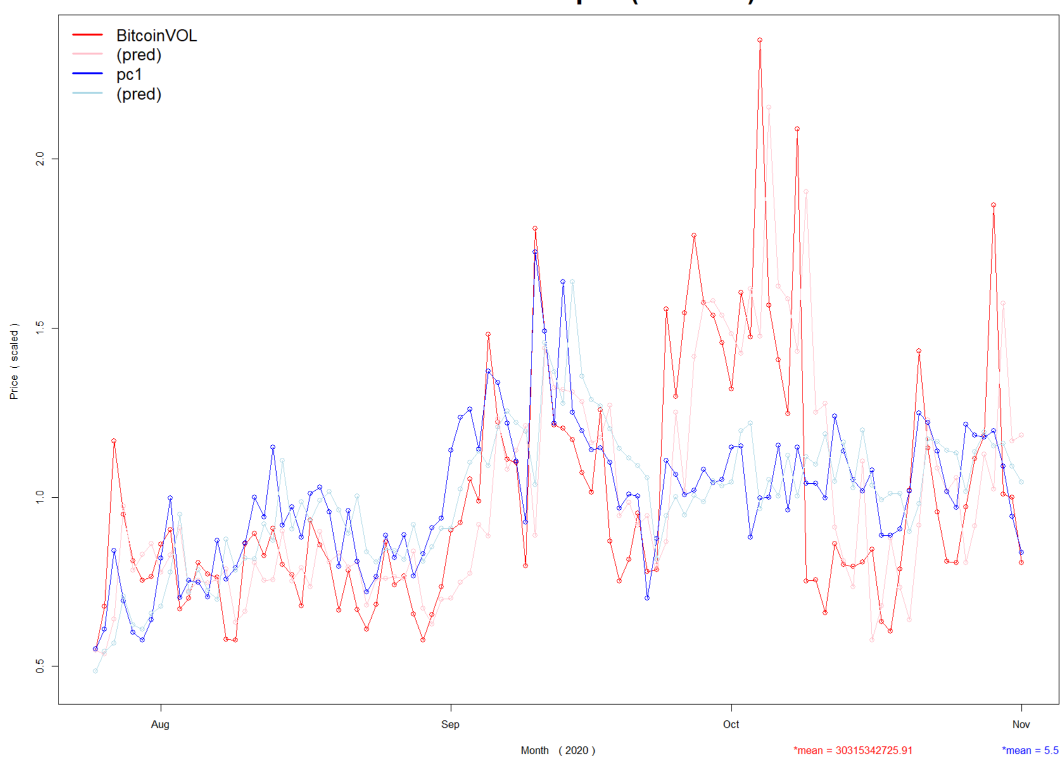
BitcoinVOL vs TetherVOL (VARIMA)

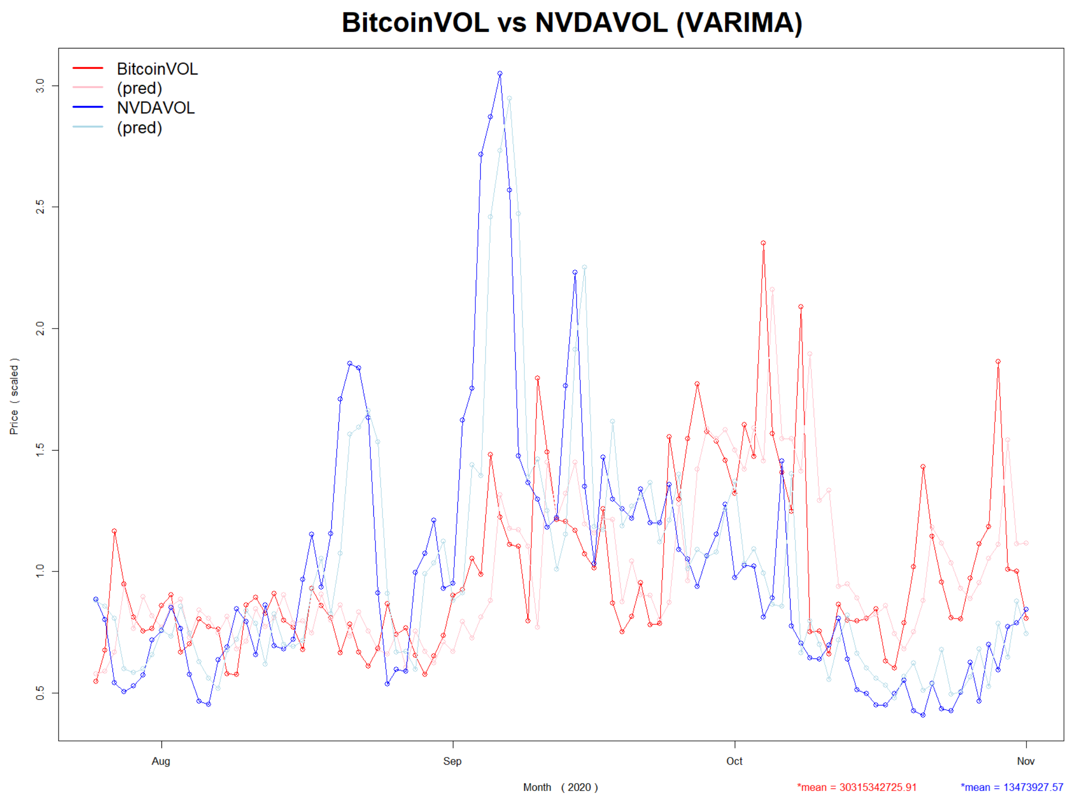


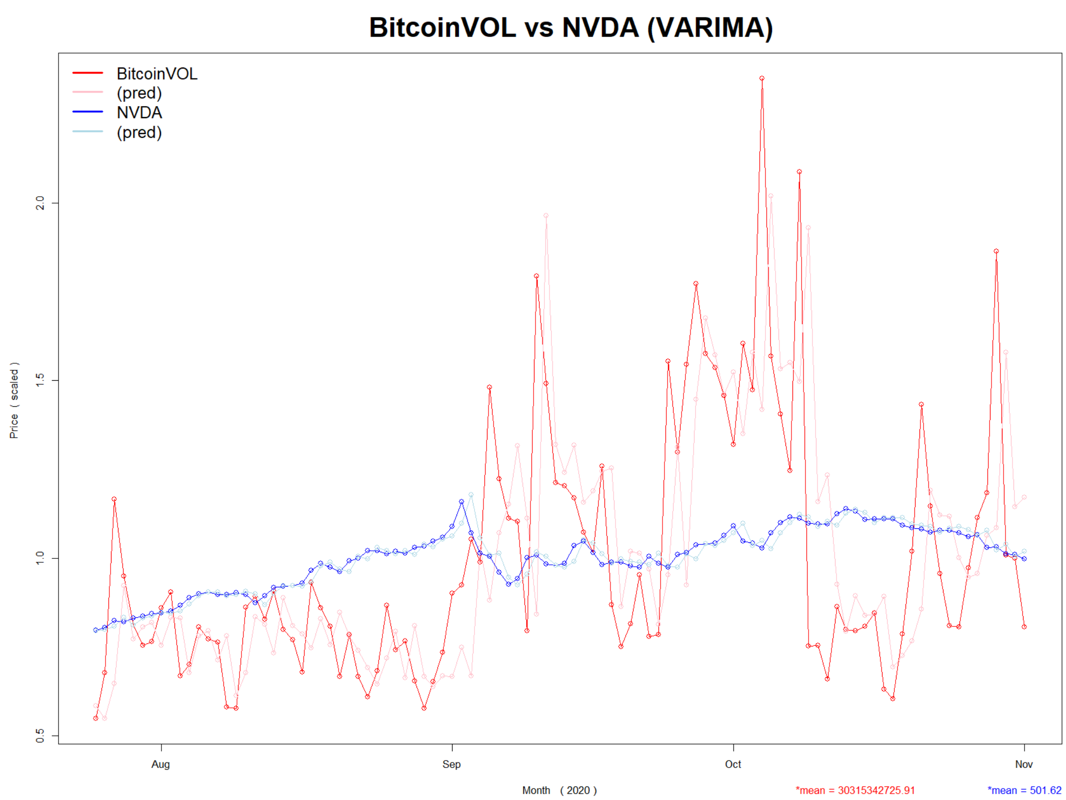
BitcoinVOL vs Tether (VARIMA)

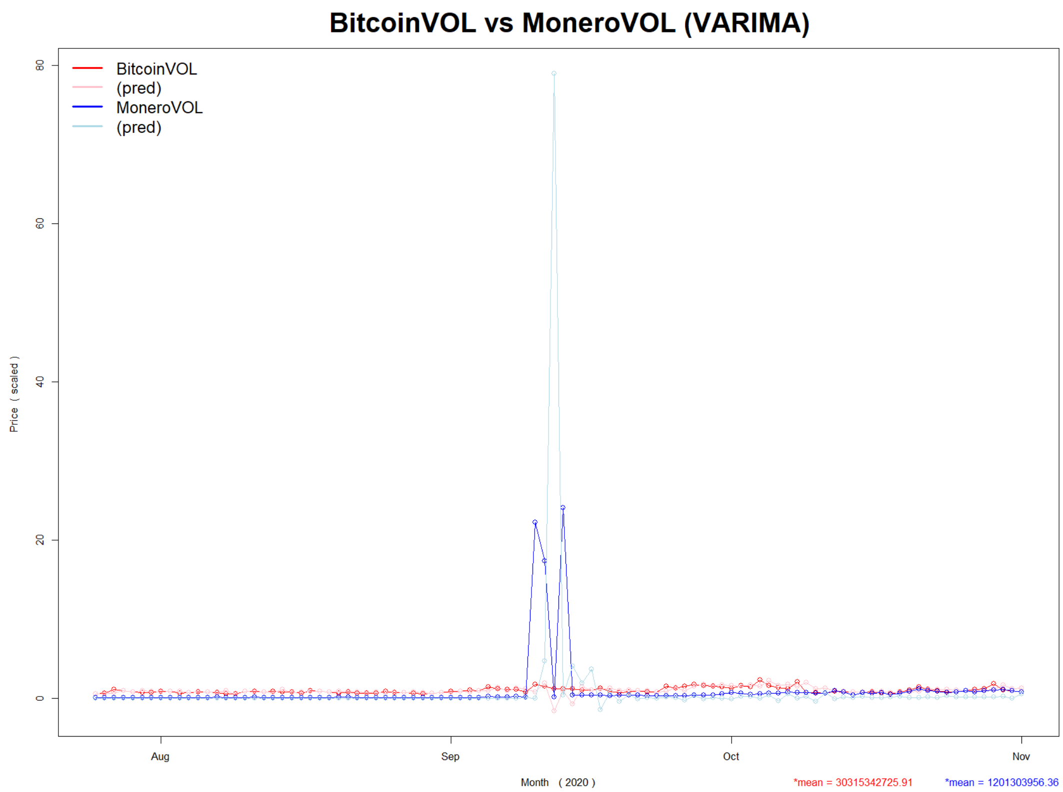


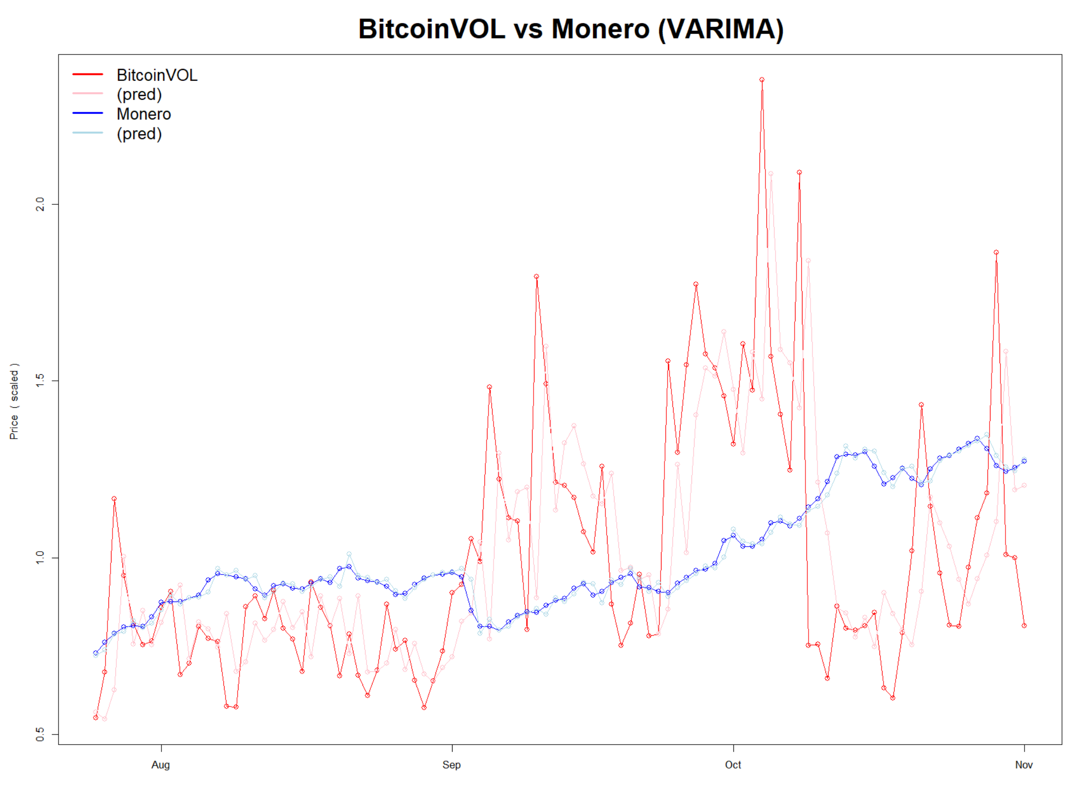
BitcoinVOL vs pc1 (VARIMA)

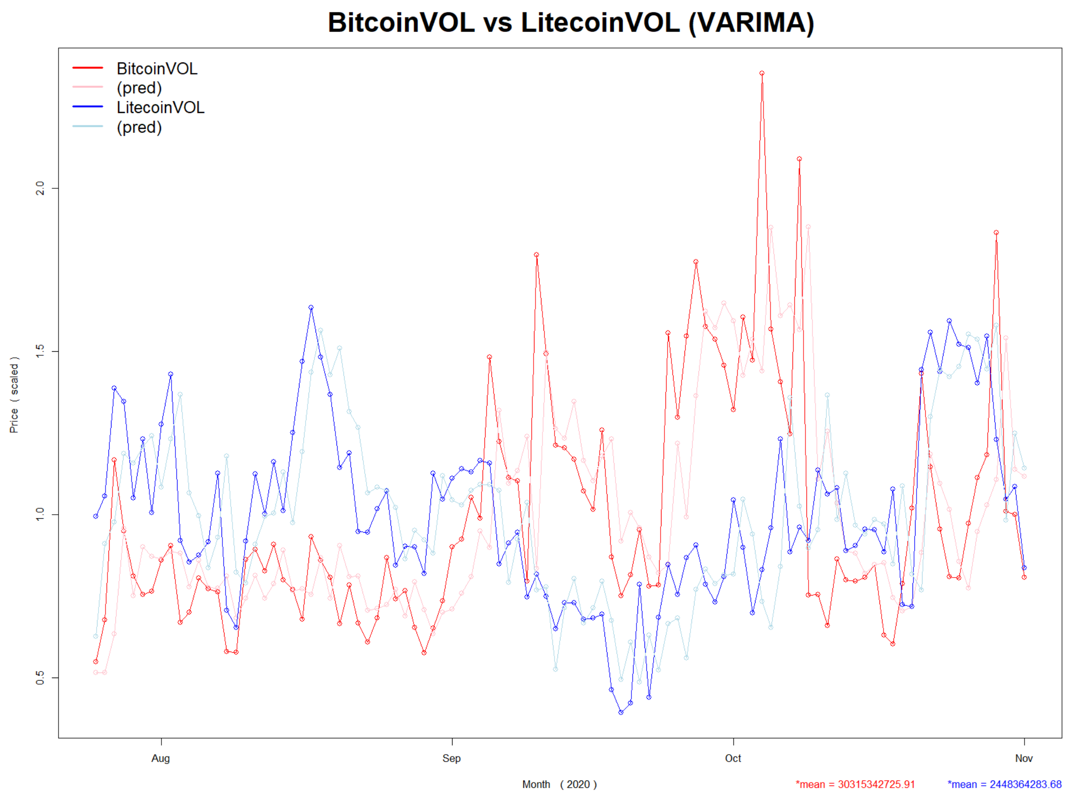


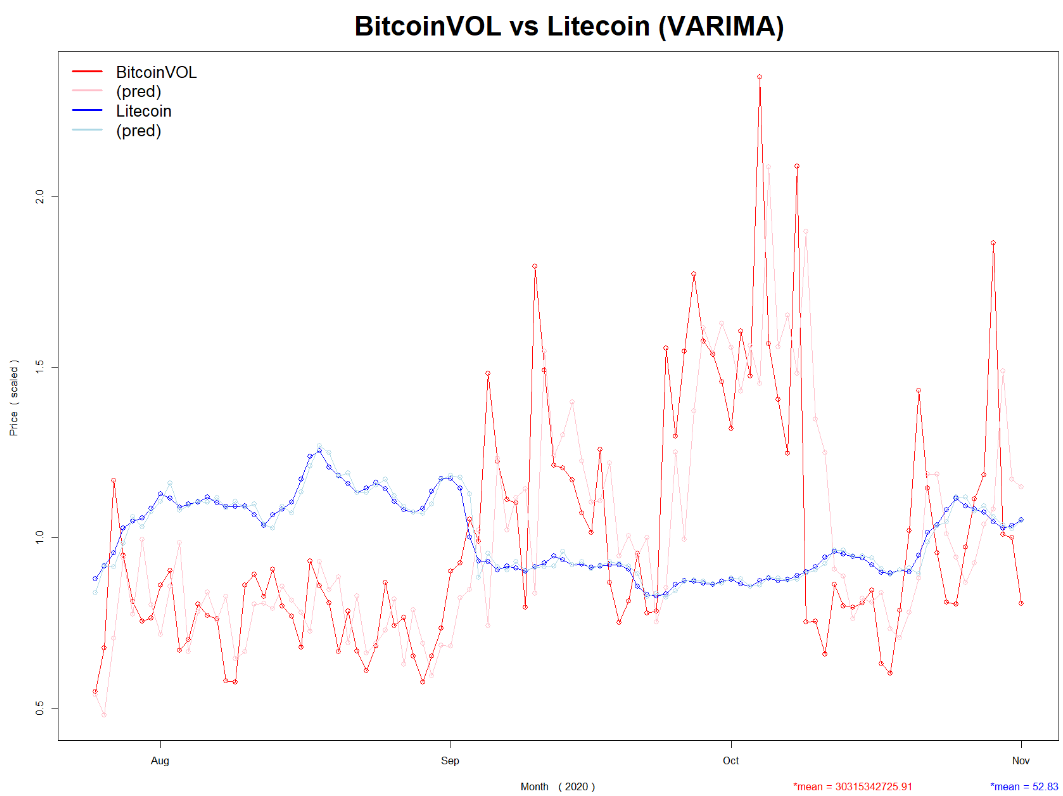

BitcoinVOL vs NVDAVOL (VARIMA)
BitcoinVOL
(pred)
NVDAVOL
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 13473927.57

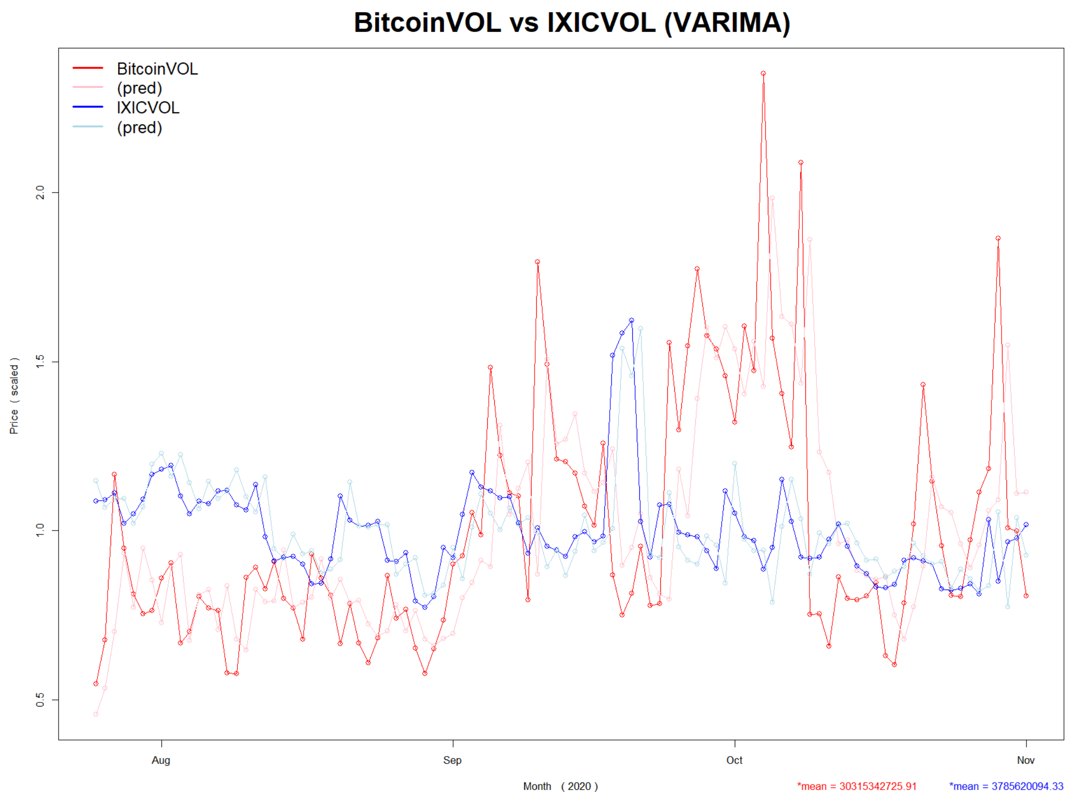

BitcoinVOL vs NVDA (VARIMA)
BitcoinVOL
(pred)
NVDA
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 501.62


BitcoinVOL vs MoneroVOL (VARIMA)
BitcoinVOL
(pred)
MoneroVOL
(pred)
Price (scaled)
Month (2020)
Aug
Sep
Oct
Nov
*mean = 30315342725.91
*mean = 1201303956.36

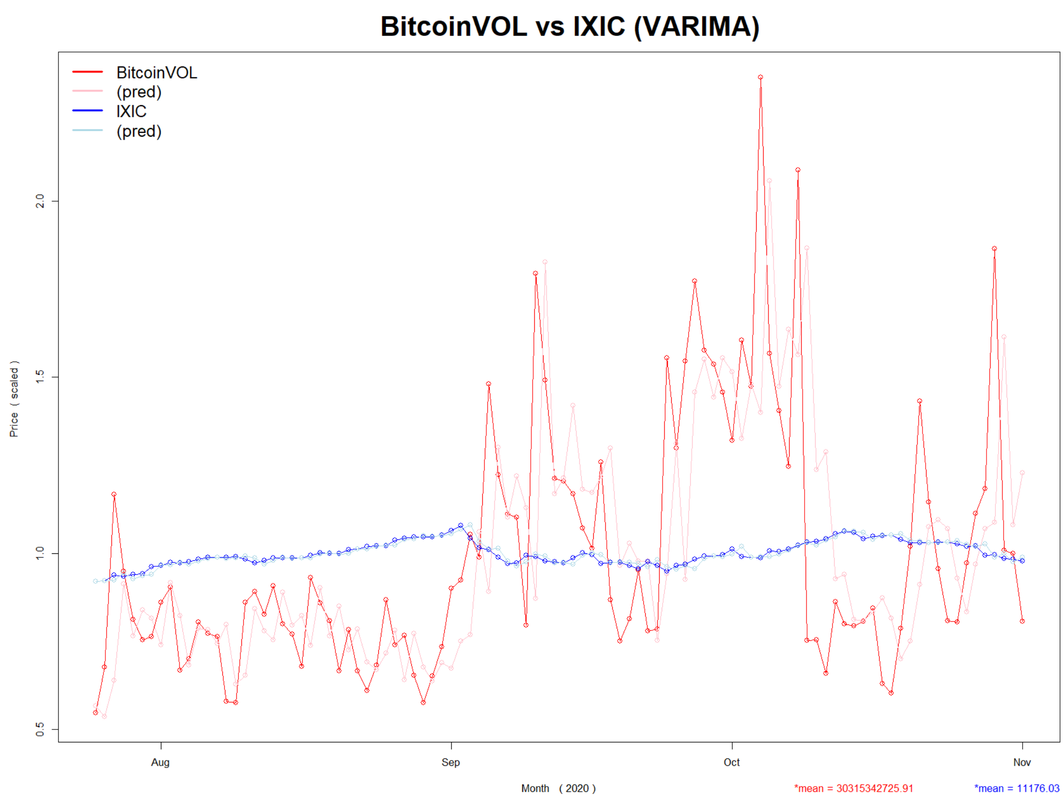

BitcoinVOL vs Monero (VARIMA)
BitcoinVOL
(pred)
Monero
(pred)
Price (scaled)
Month (2020)
Aug
Sep
Oct
Nov
*mean = 30315342725.91
*mean = 99.42


BitcoinVOL vs LitecoinVOL (VARIMA)
BitcoinVOL
(pred)
LitecoinVOL
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 2448364283.68

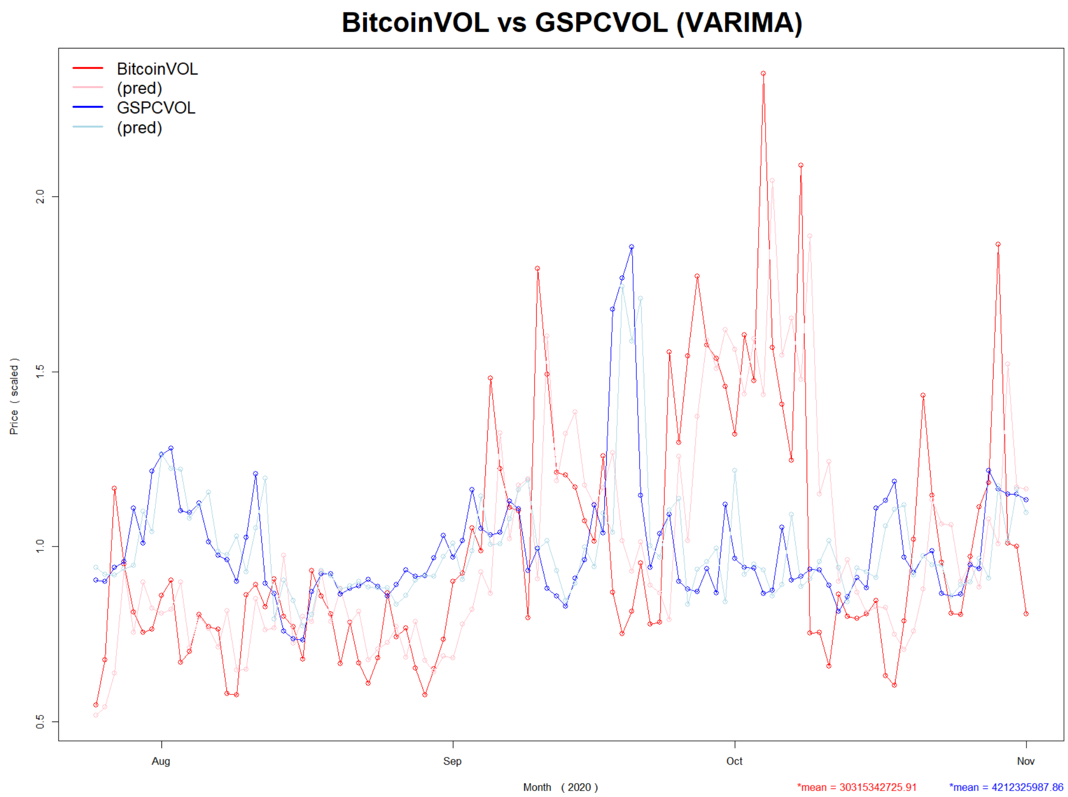

BitcoinVOL vs Litecoin (VARIMA)
BitcoinVOL
(pred)
Litecoin
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 52.83

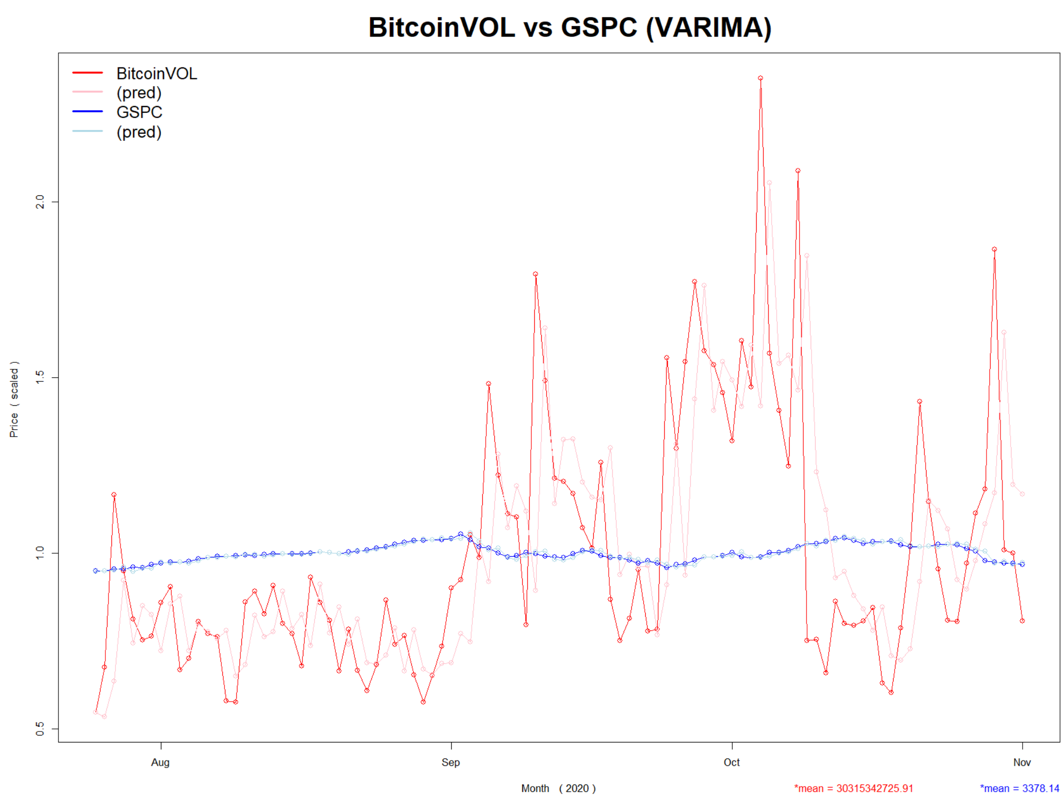


BitcoinVOL vs IXICVOL (VARIMA)
BitcoinVOL
(pred)
IXICVOL
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 3785620094.33

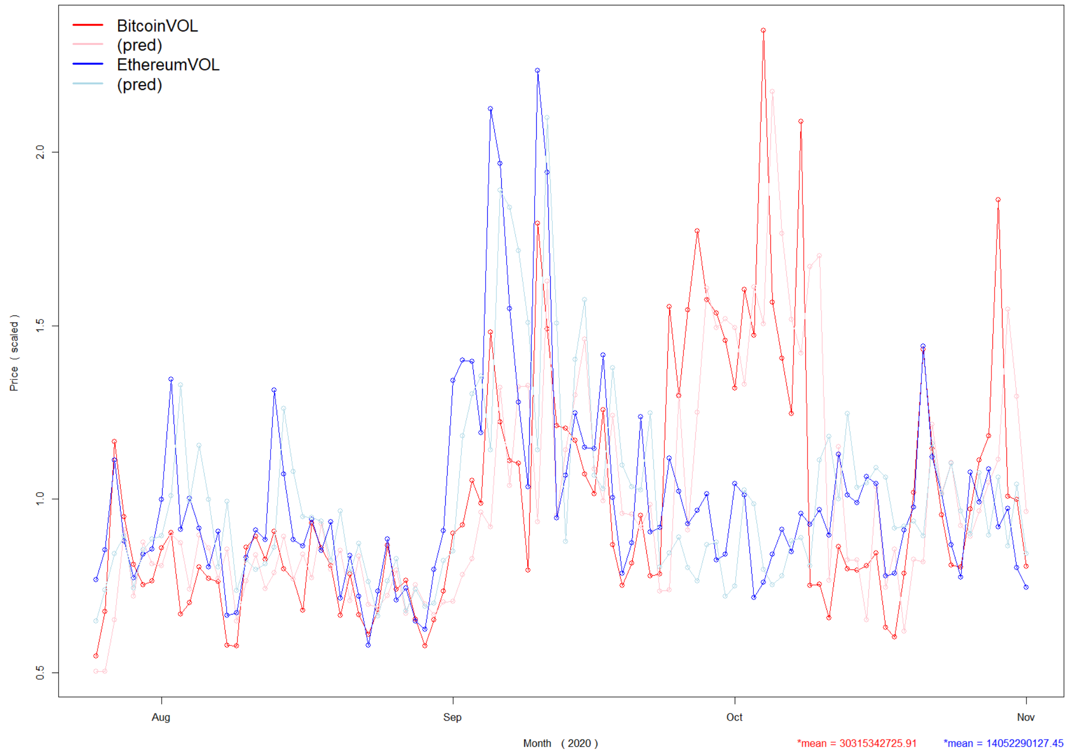


BitcoinVOL vs IXIC (VARIMA)
BitcoinVOL
(pred)
IXIC
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 11176.03

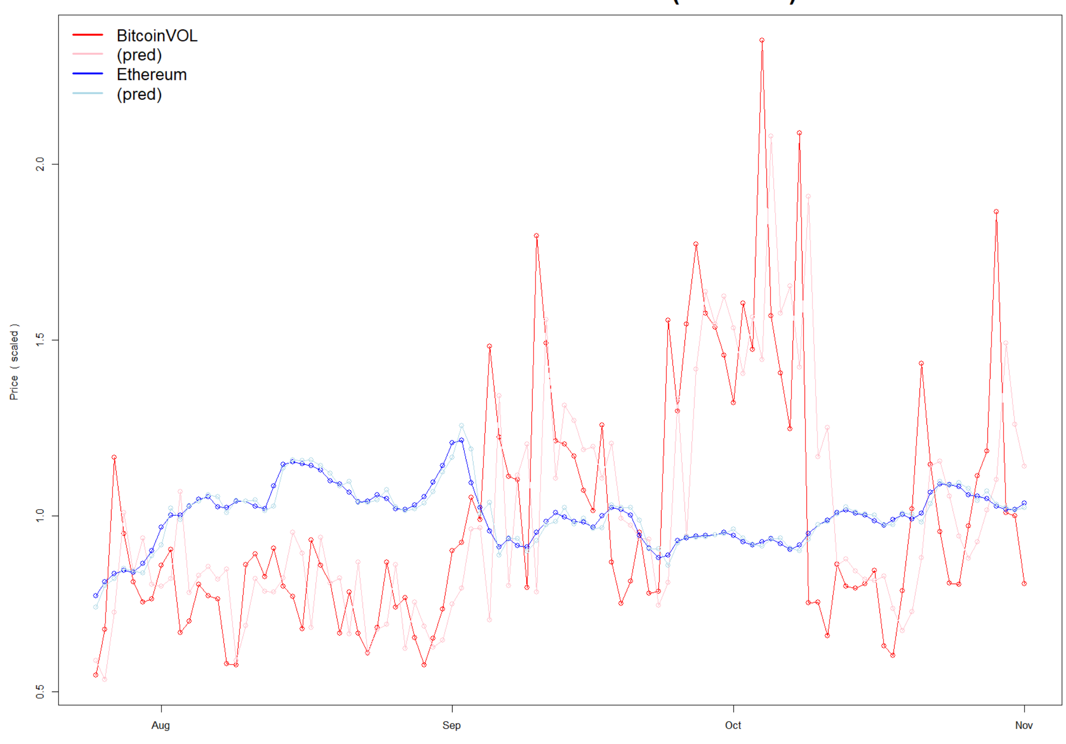

BitcoinVOL vs GSPCVOL (VARIMA)
BitcoinVOL
(pred)
GSPCVOL
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 4212325987.86

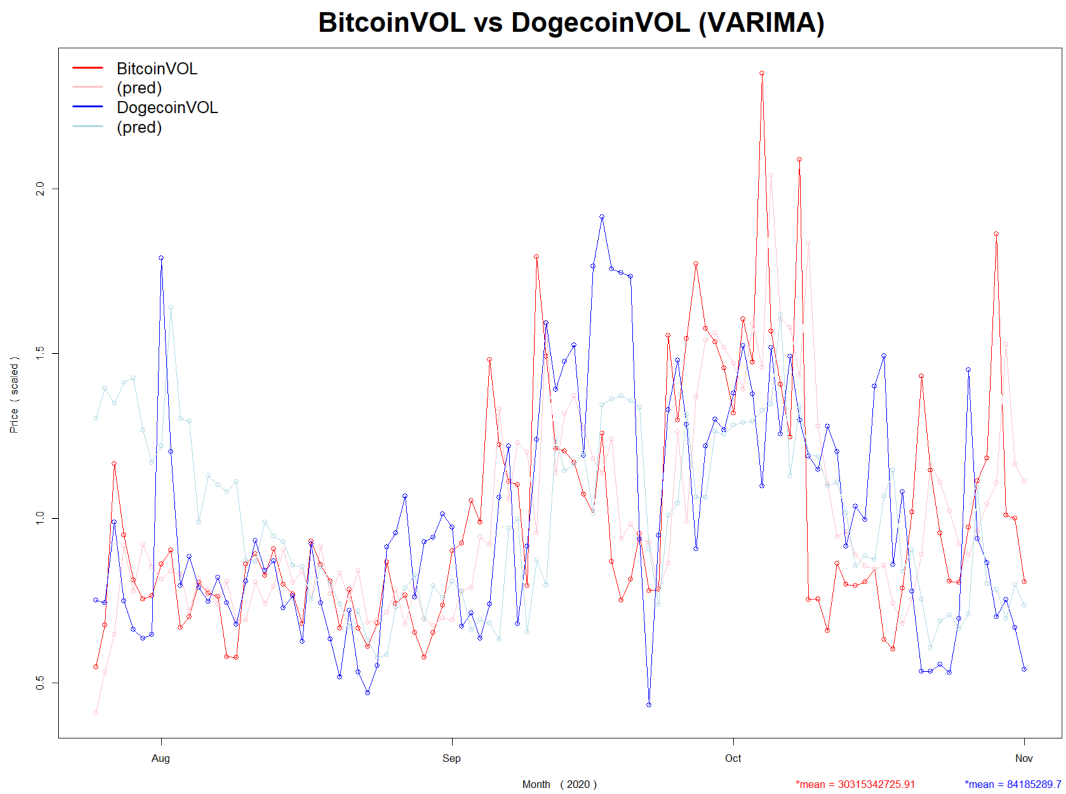

BitcoinVOL vs GSPC (VARIMA)
BitcoinVOL
(pred)
GSPC
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 3378.14

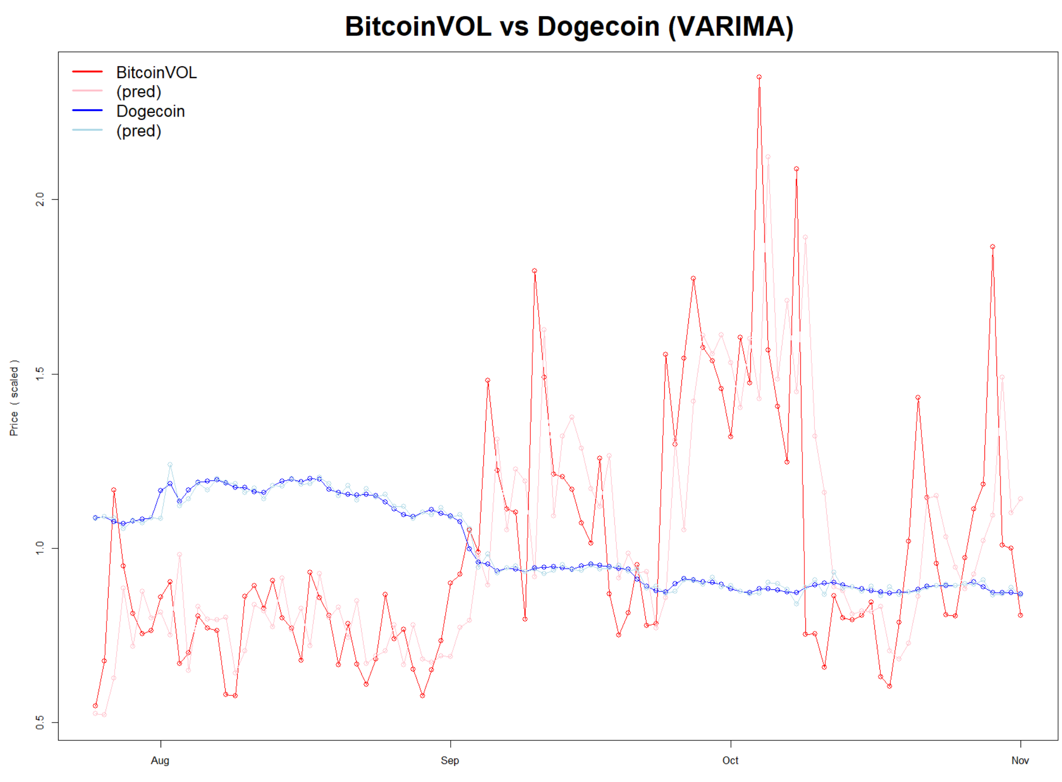
BitcoinVOL vs EthereumVOL (VARIMA)

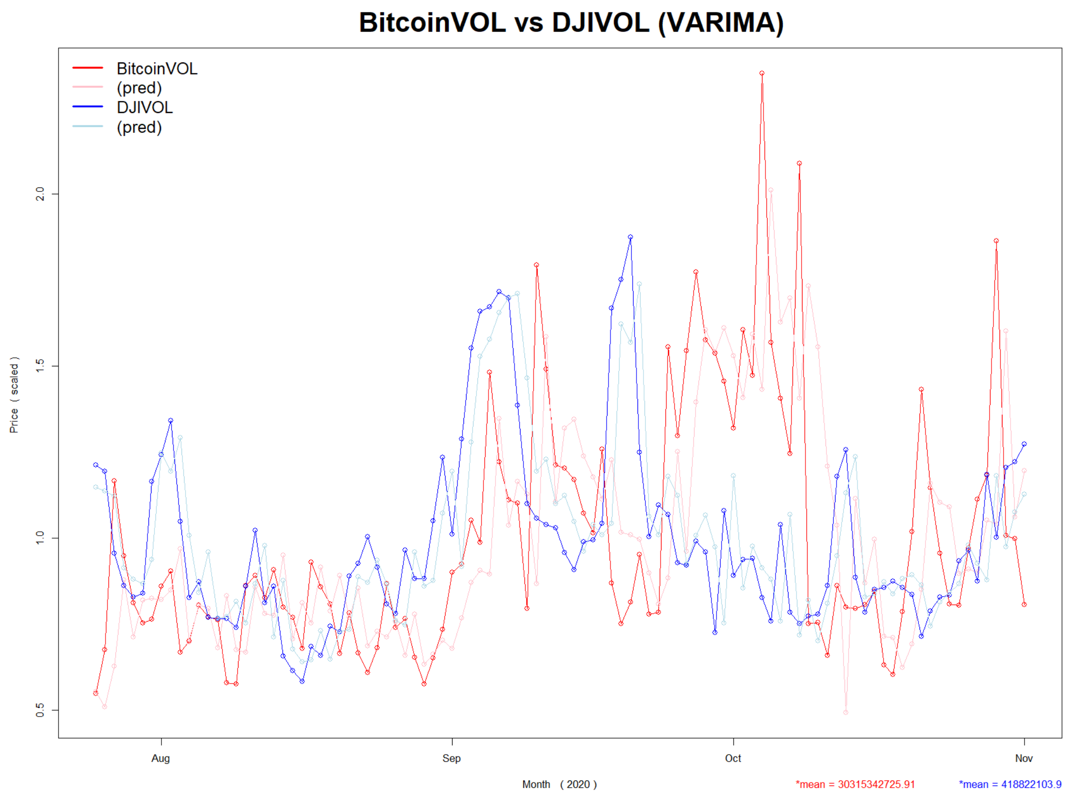


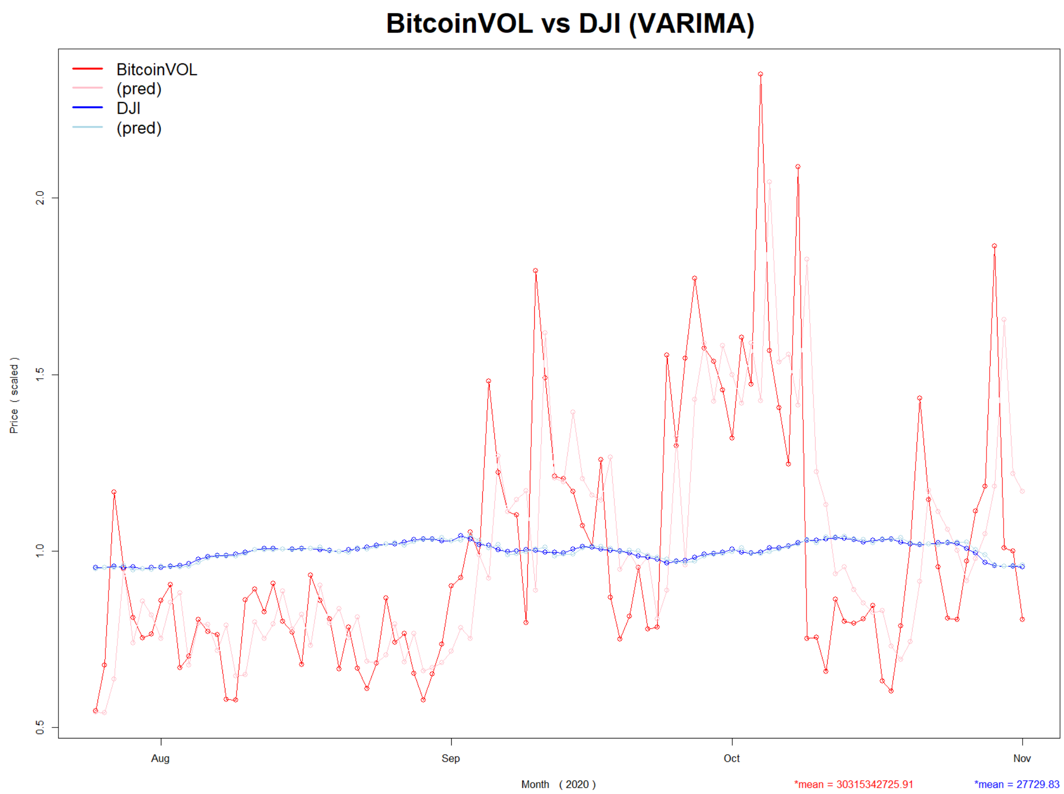
BitcoinVOL vs Ethereum (VARIMA)



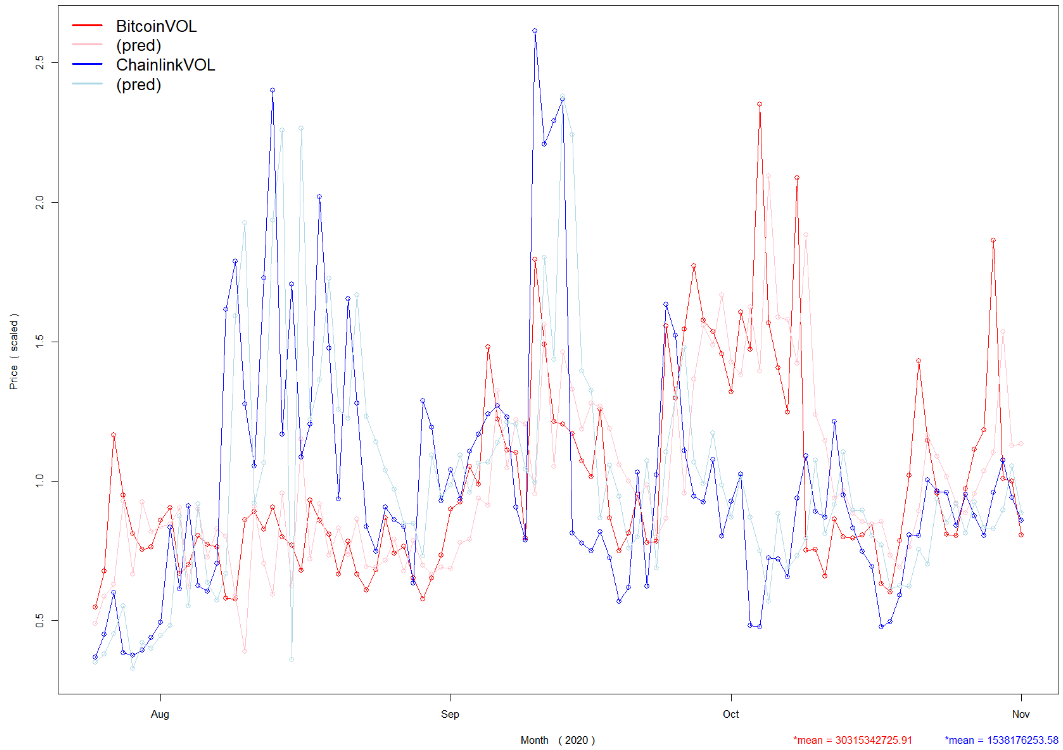

BitcoinVOL vs DogecoinVOL (VARIMA)
BitcoinVOL
(pred)
DogecoinVOL
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 84185289.7


BitcoinVOL vs Dogecoin (VARIMA)
BitcoinVOL
(pred)
Dogecoin
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 0

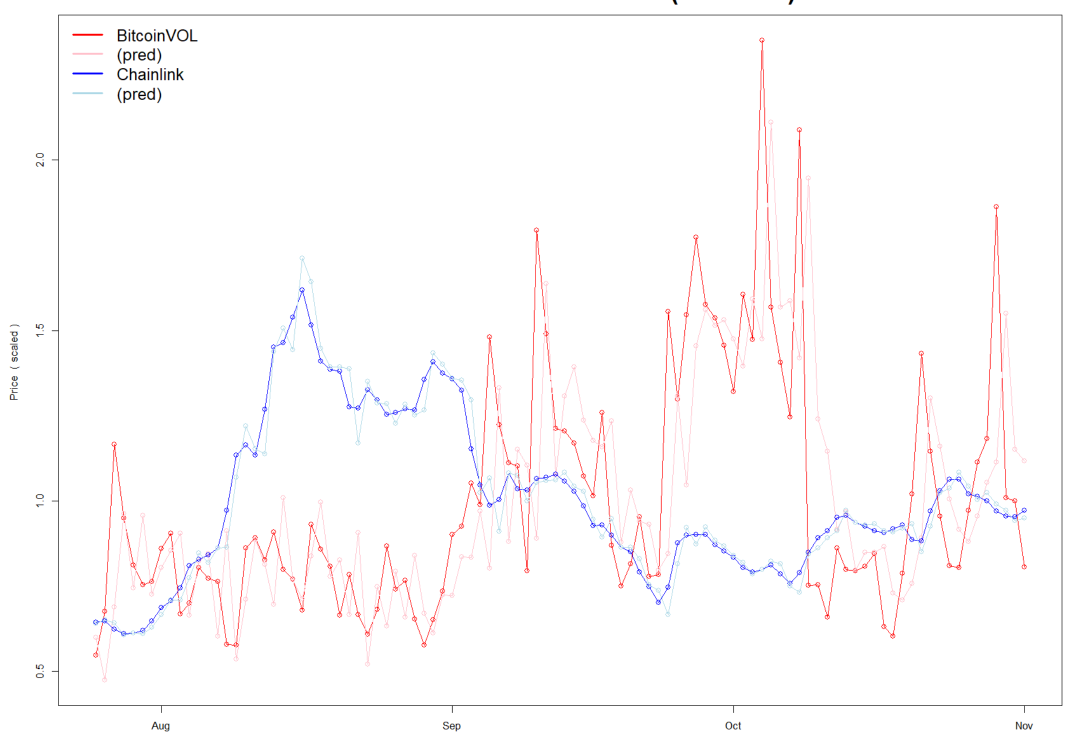

BitcoinVOL vs DJIVOL (VARIMA)
BitcoinVOL
(pred)
DJIVOL
(pred)
Price (scaled)
Month (2020)
Aug
Sep
Oct
Nov
*mean = 30315342725.91
*mean = 418822103.9


BitcoinVOL vs DJI (VARIMA)
BitcoinVOL
(pred)
DJI
(pred)
Price (scaled)
Month (2020)
Aug
Sep
Oct
Nov
*mean = 30315342725.91
*mean = 27729.83

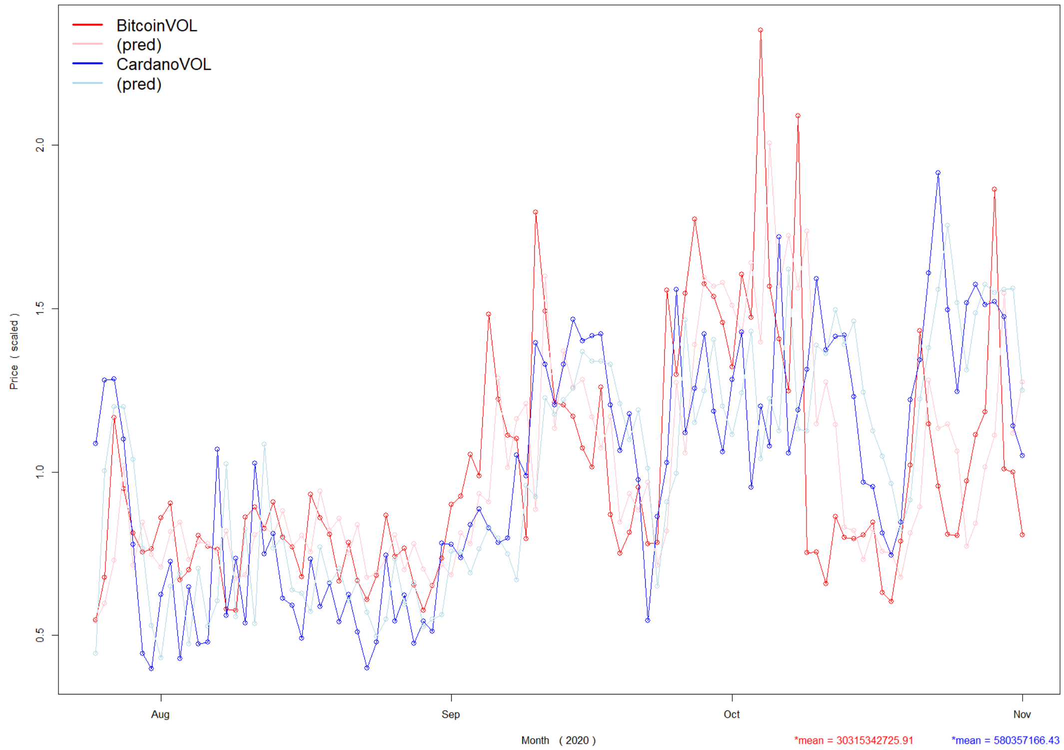
BitcoinVOL vs ChainlinkVOL (VARIMA)



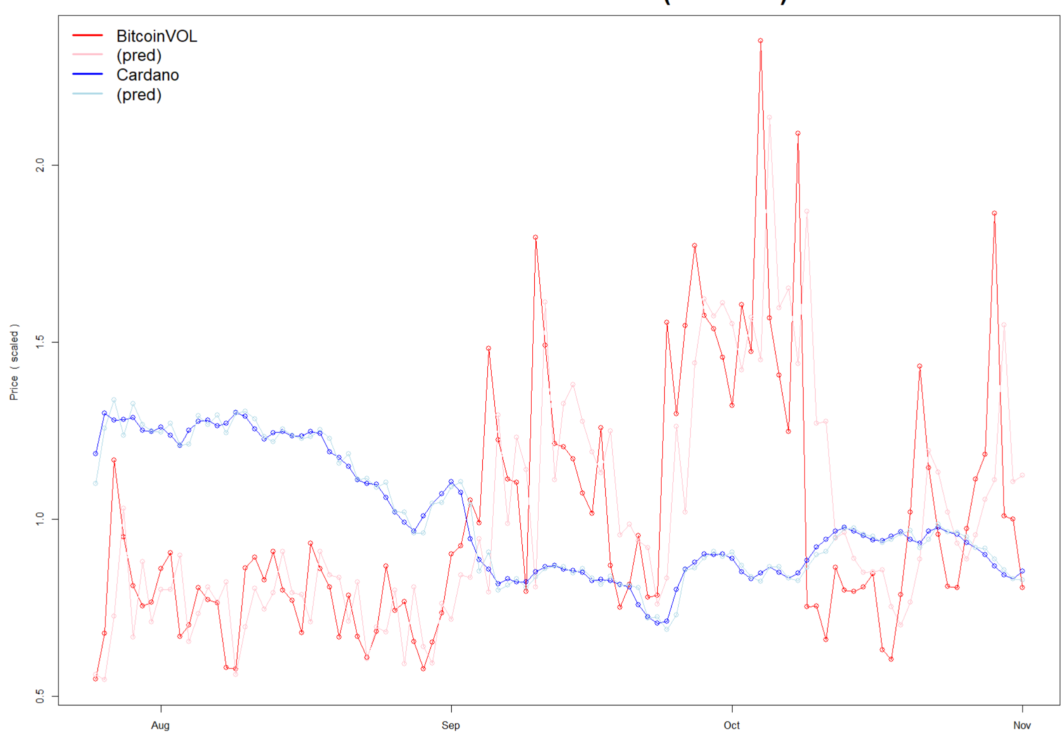
BitcoinVOL vs Chainlink (VARIMA)

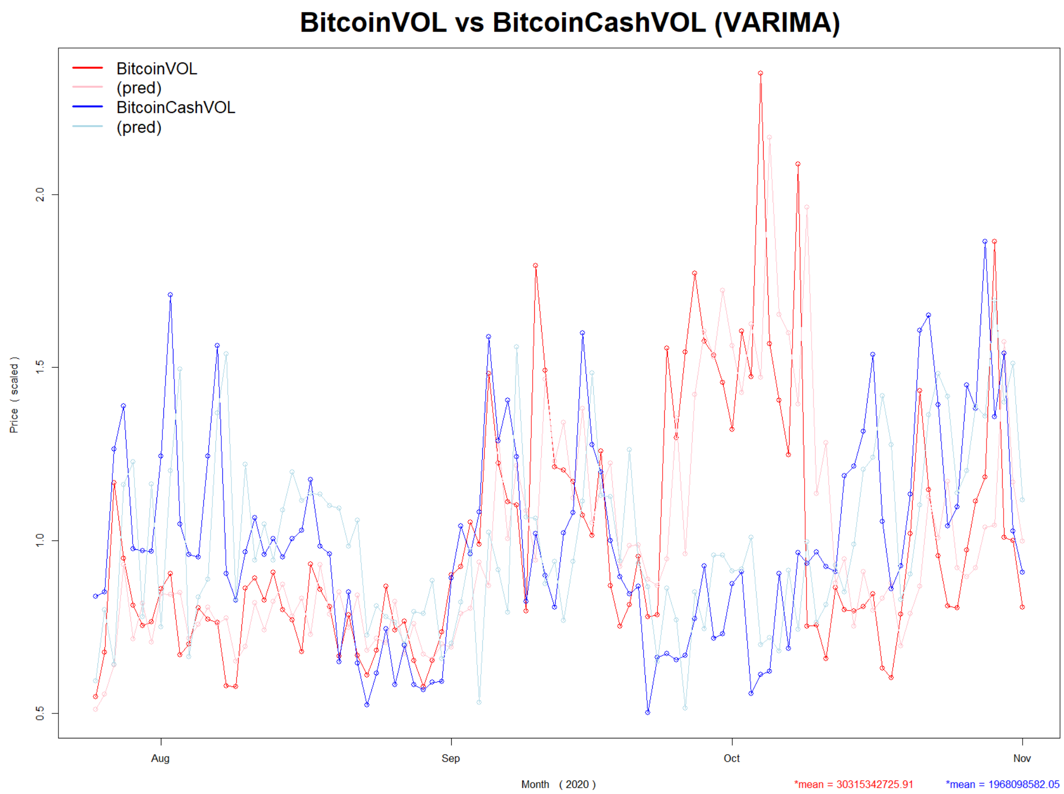


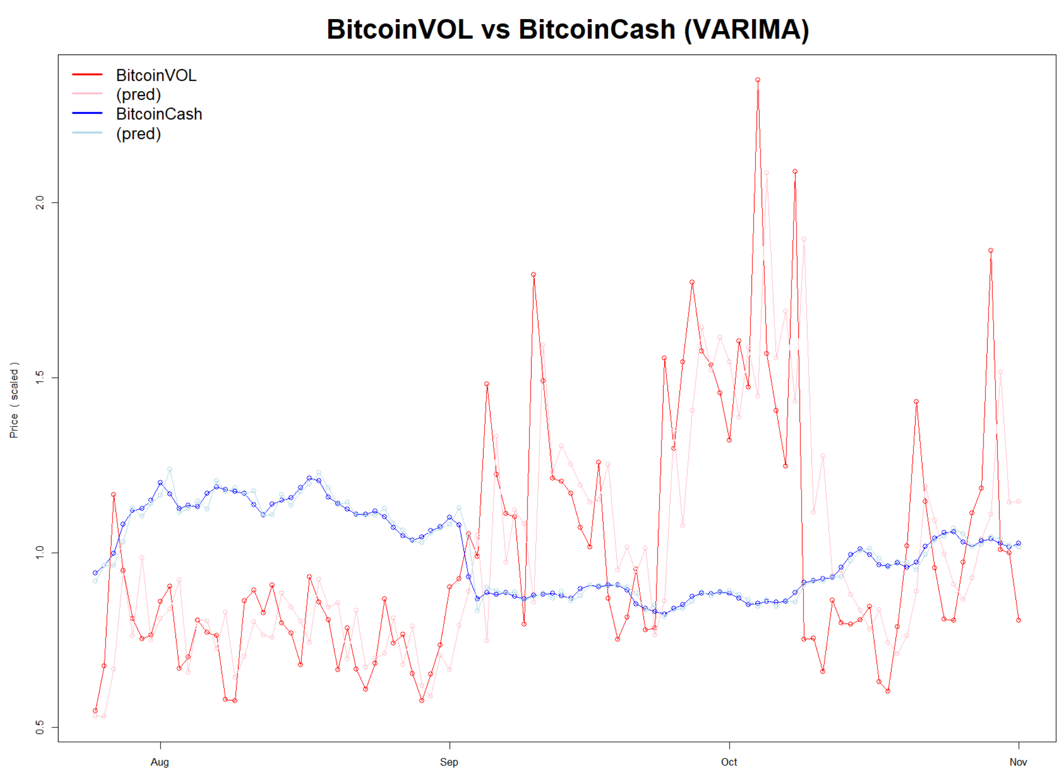
BitcoinVOL vs CardanoVOL (VARIMA)



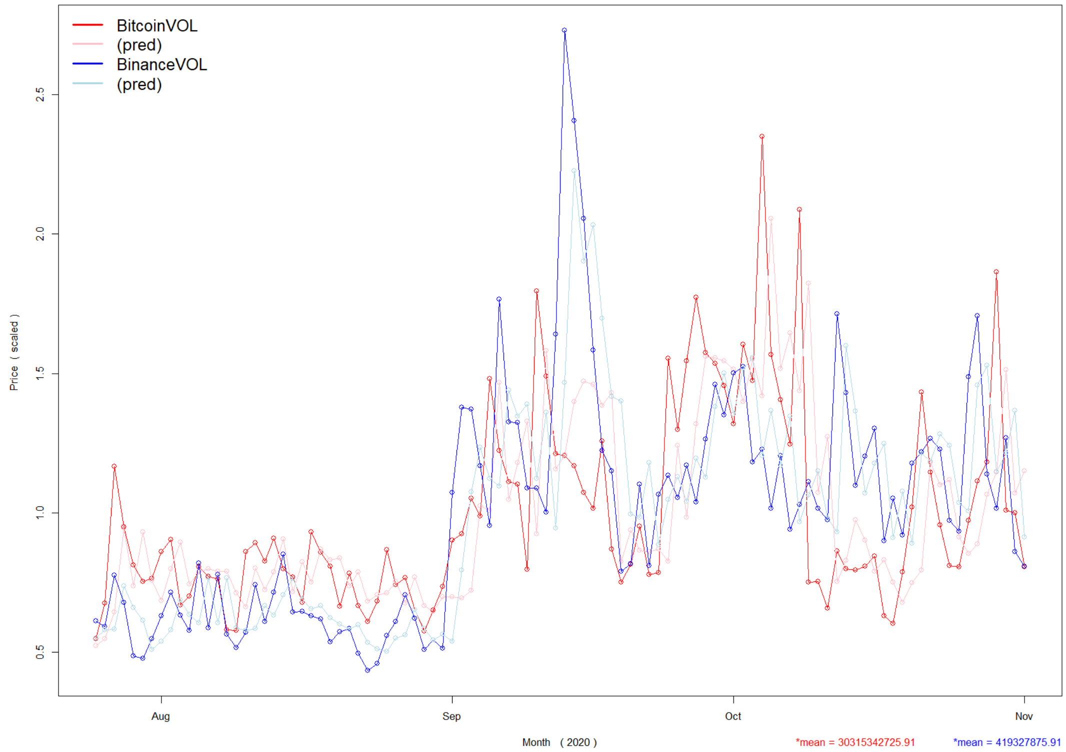
BitcoinVOL vs Cardano (VARIMA)



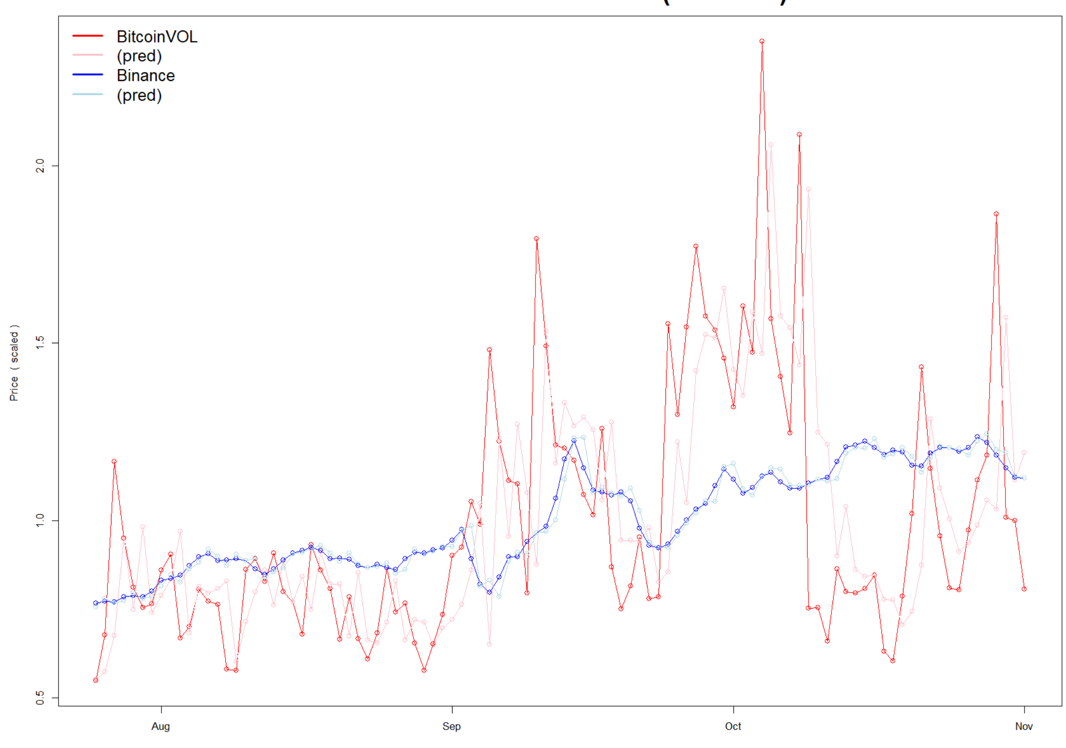

BitcoinVOL vs BitcoinCashVOL (VARIMA)
BitcoinVOL
(pred)
BitcoinCashVOL
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 1968098582.05

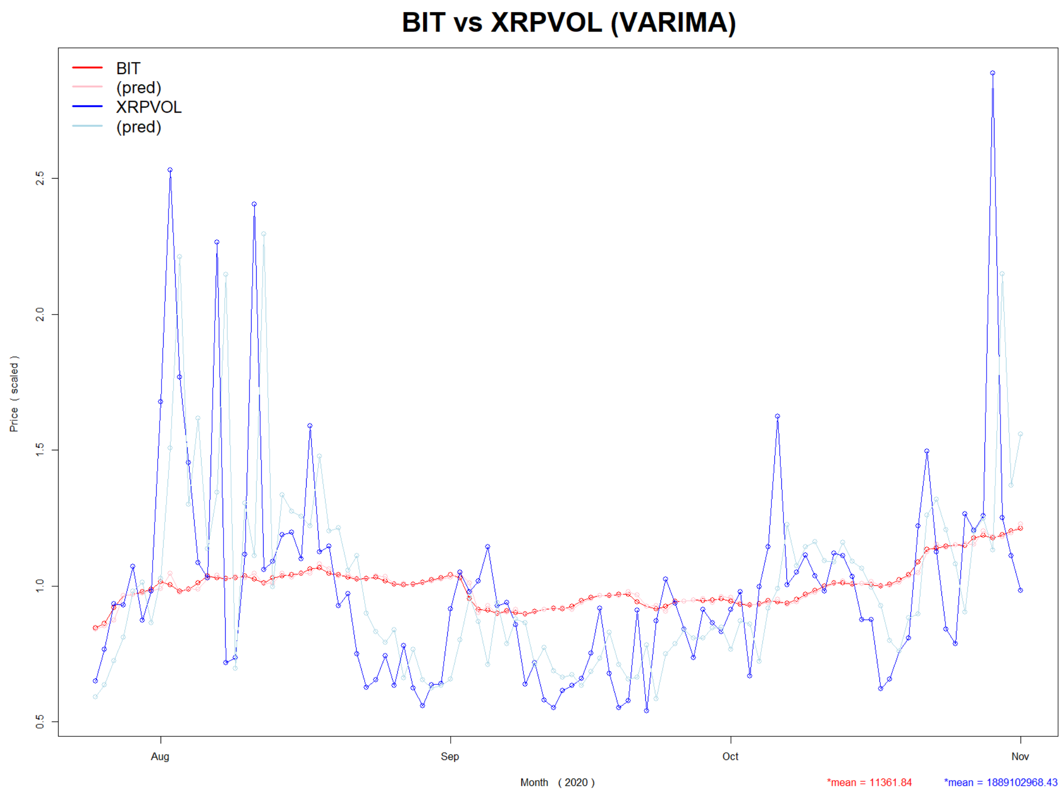

BitcoinVOL vs BitcoinCash (VARIMA)
BitcoinVOL
(pred)
BitcoinCash
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 30315342725.91
*mean = 257.77

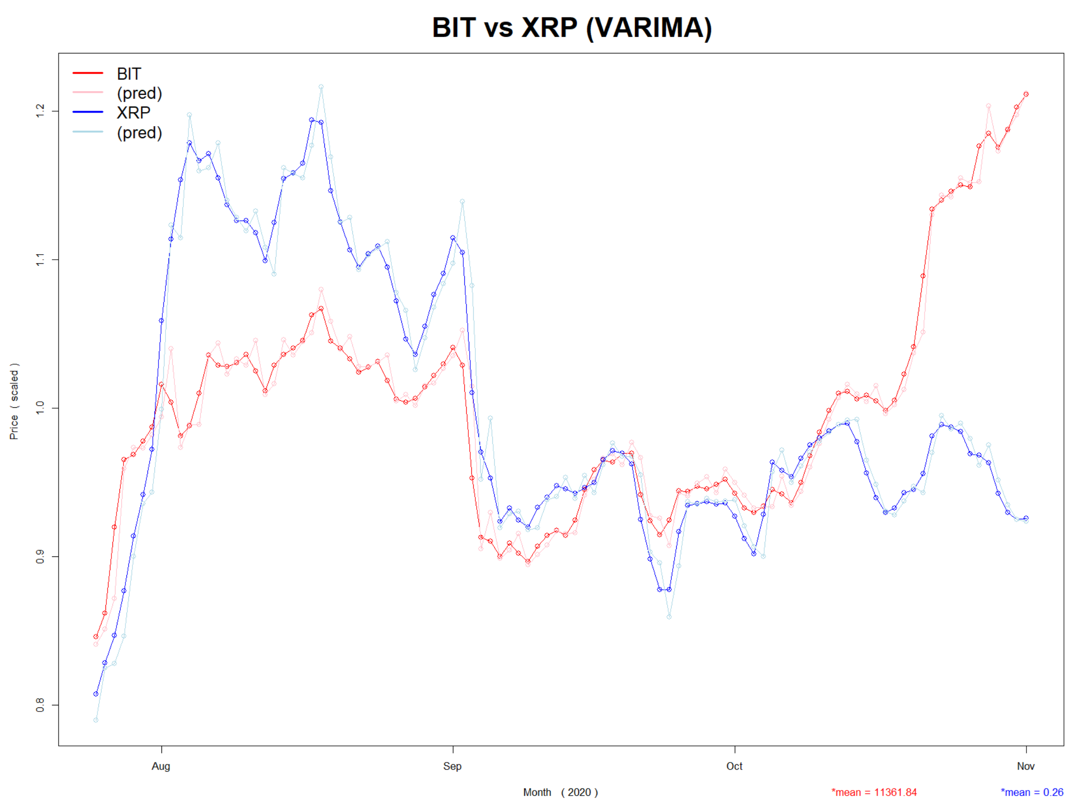
BitcoinVOL vs BinanceVOL (VARIMA)

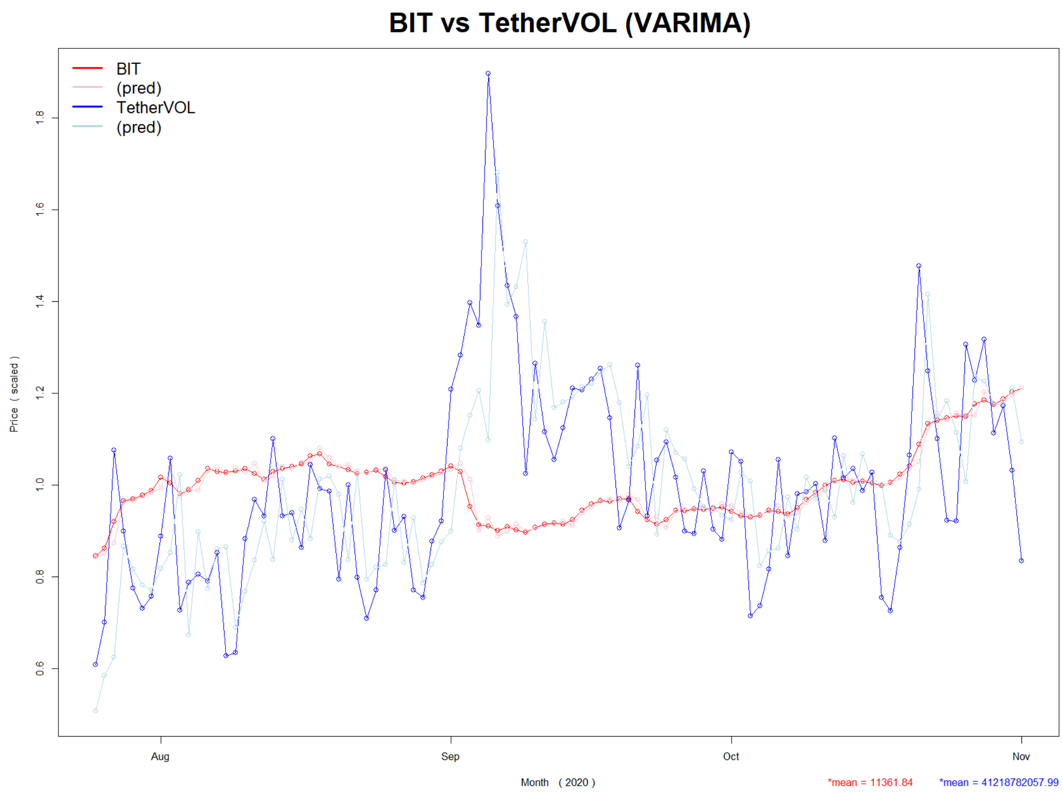


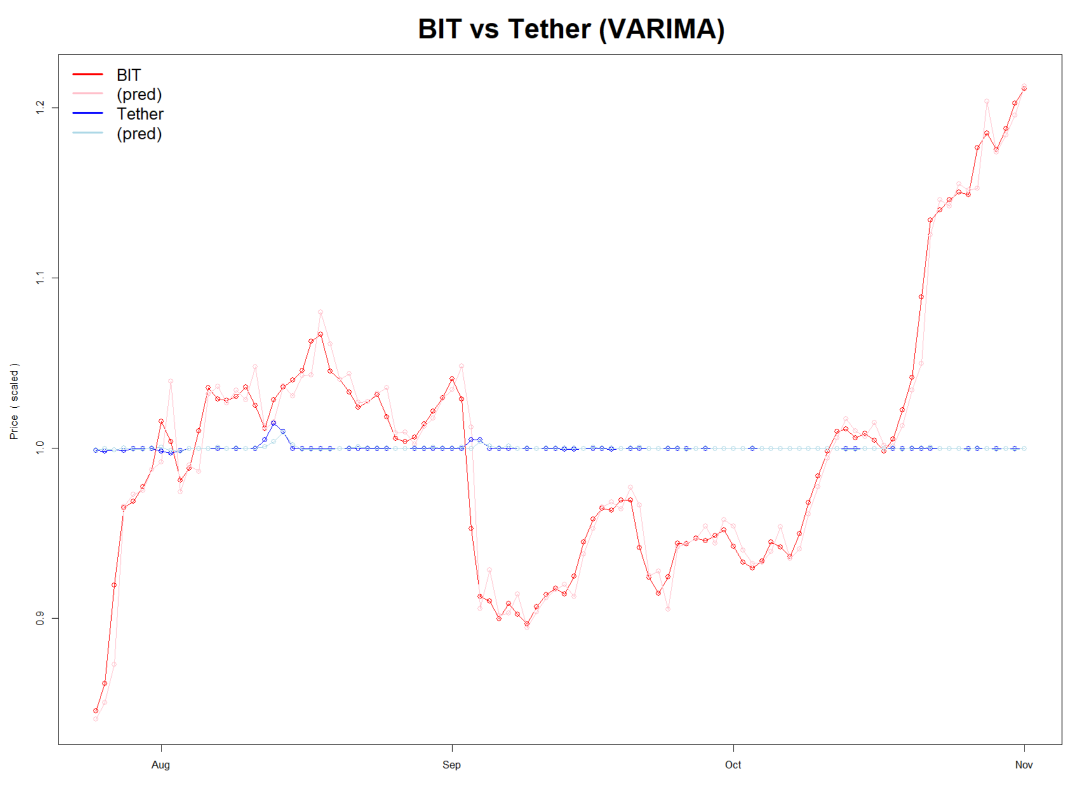
BitcoinVOL vs Binance (VARIMA)

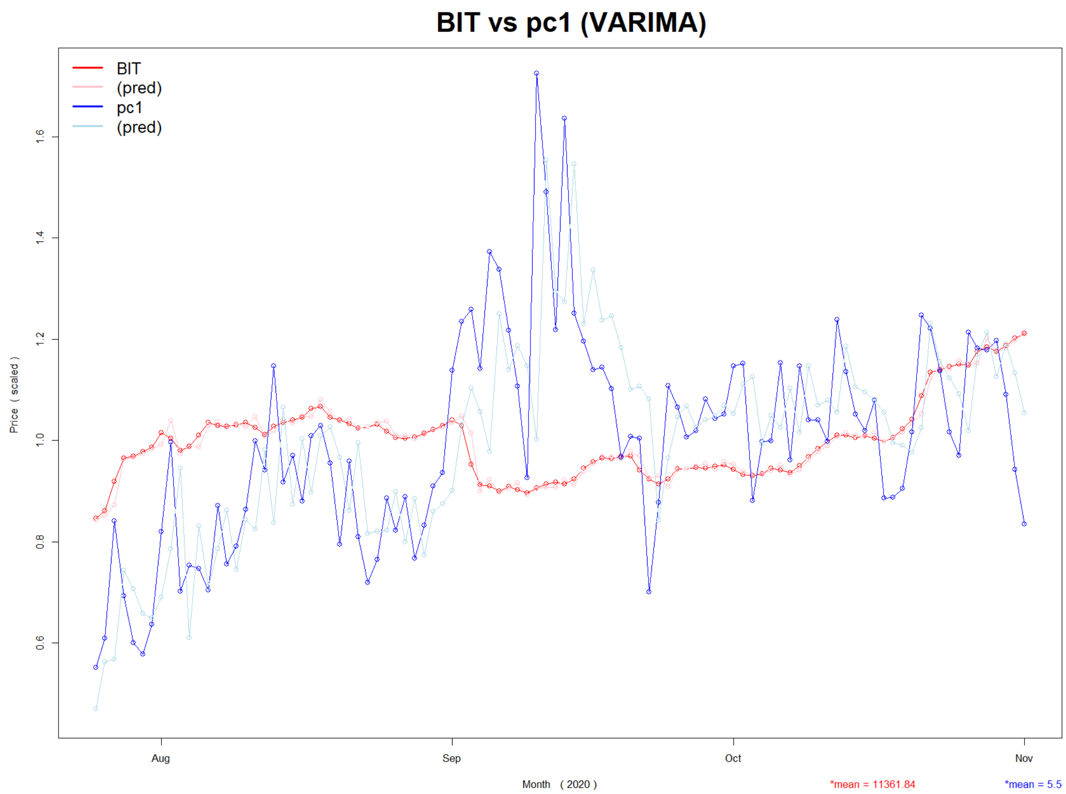


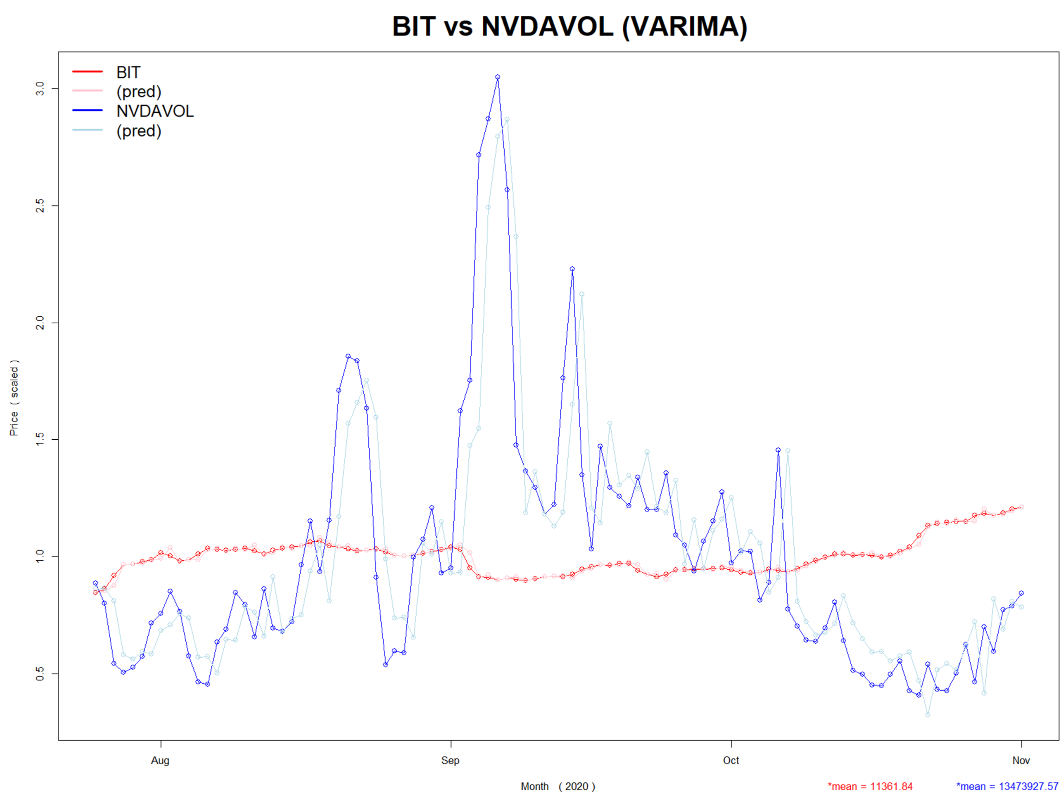

BIT vs XRPVOL (VARIMA)
BIT
(pred)
XRPVOL
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
2.5
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 1889102968.43


BIT vs XRP (VARIMA)
BIT
(pred)
XRP
(pred)
Price (scaled)
0.8
0.9
1.0
1.1
1.2
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 0.26


BIT vs TetherVOL (VARIMA)
BIT
(pred)
TetherVOL
(pred)
Price (scaled)
1.8
1.6
1.4
1.2
1.0
0.8
0.6
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 41218782057.99

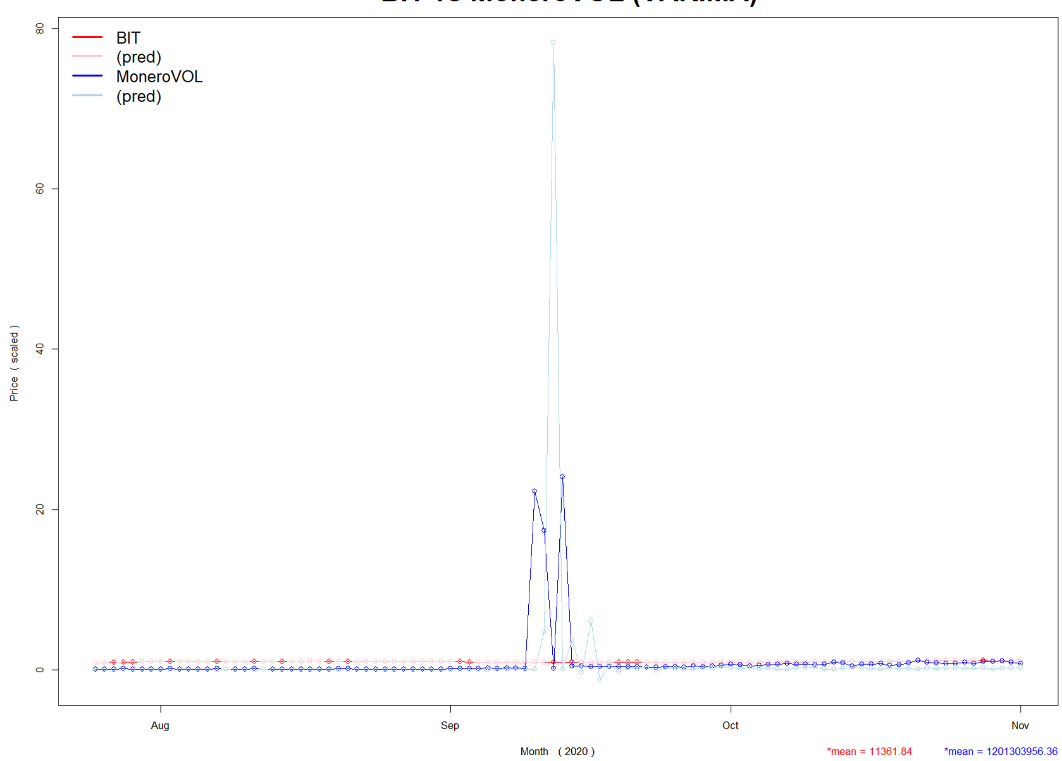

BIT vs Tether (VARIMA)
BIT
(pred)
Tether
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 1

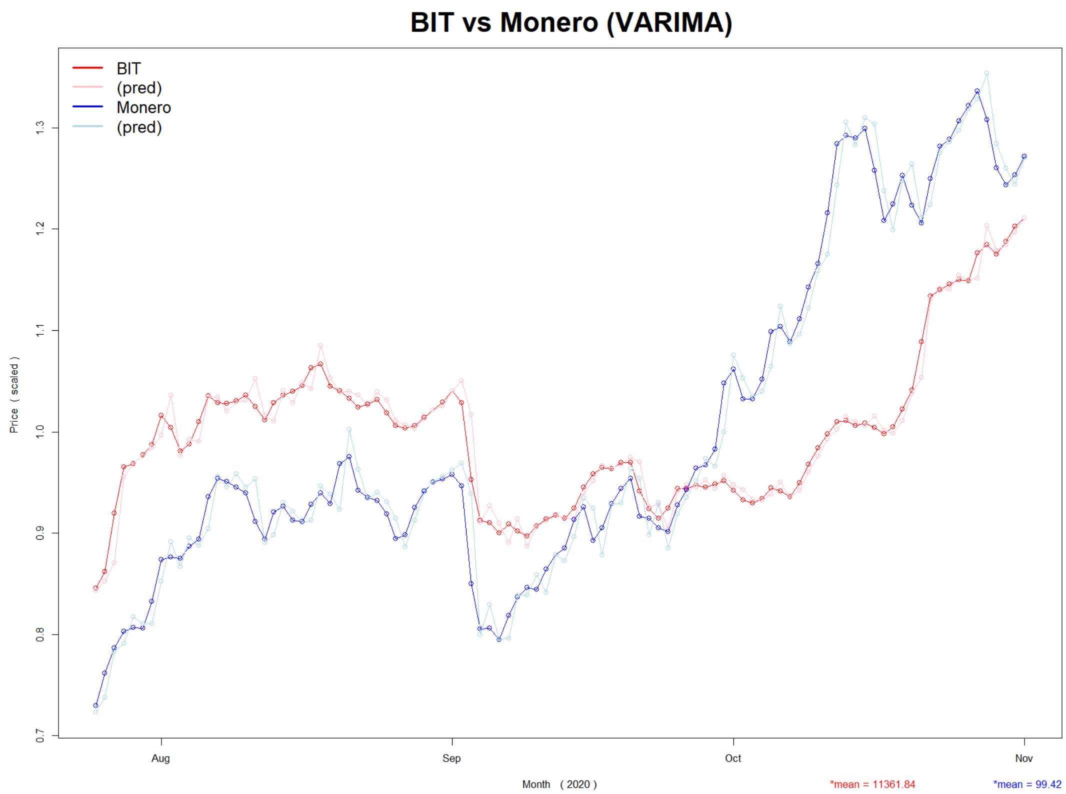

BIT vs pc1 (VARIMA)
BIT
(pred)
pc1
(pred)
Price (scaled)
1.6
1.4
1.2
1.0
0.8
0.6
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 5.5

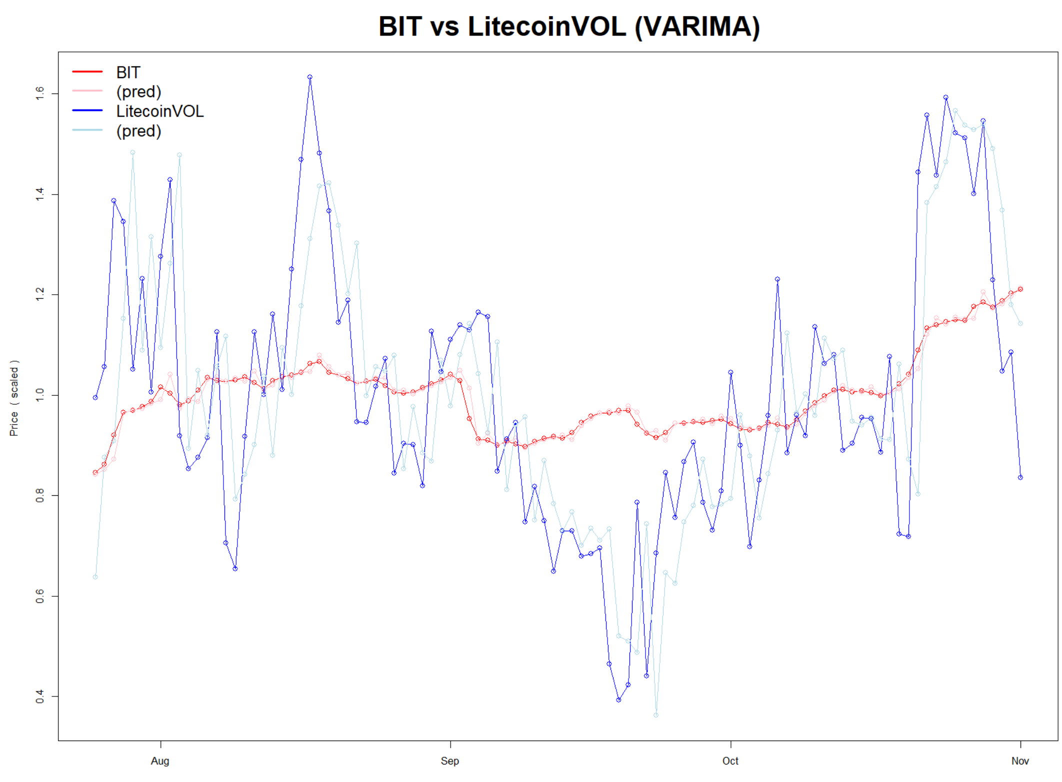

BIT vs NVDAVOL (VARIMA)
BIT
(pred)
NVDAVOL
(pred)
Price (scaled)
3.0
2.5
2.0
1.5
1.0
0.5
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 13473927.57

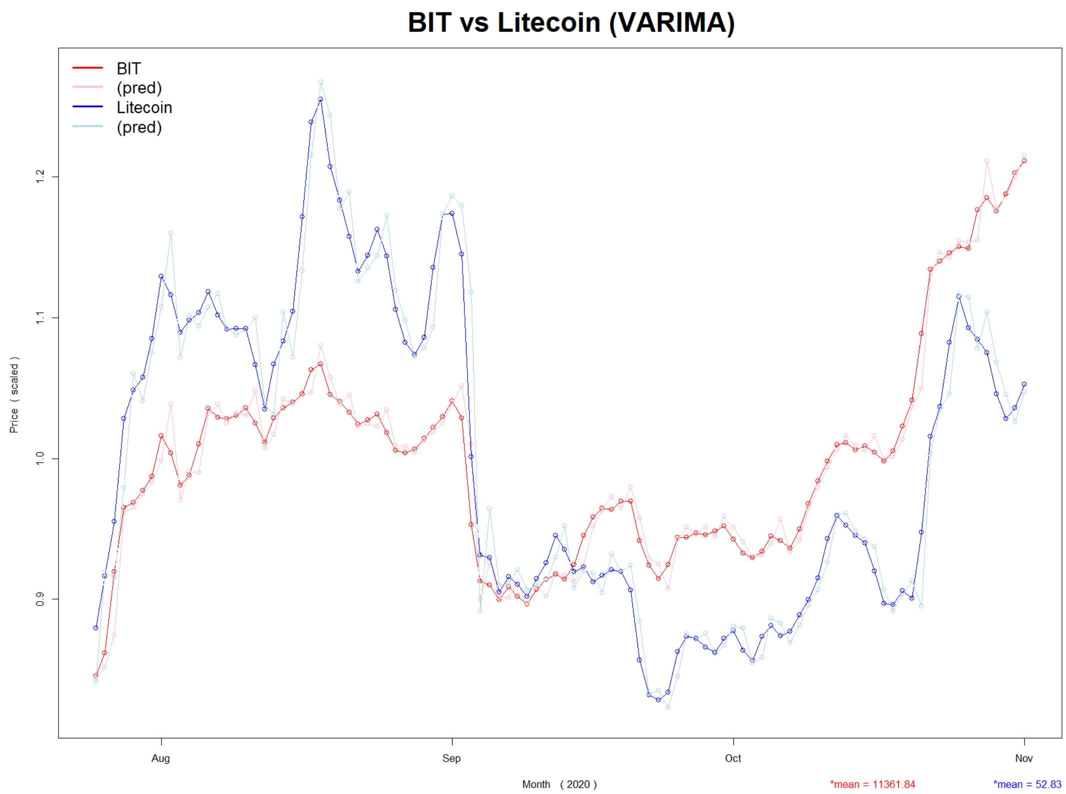
BIT vs NVDA (VARIMA)

BIT vs MoneroVOL (VARIMA)

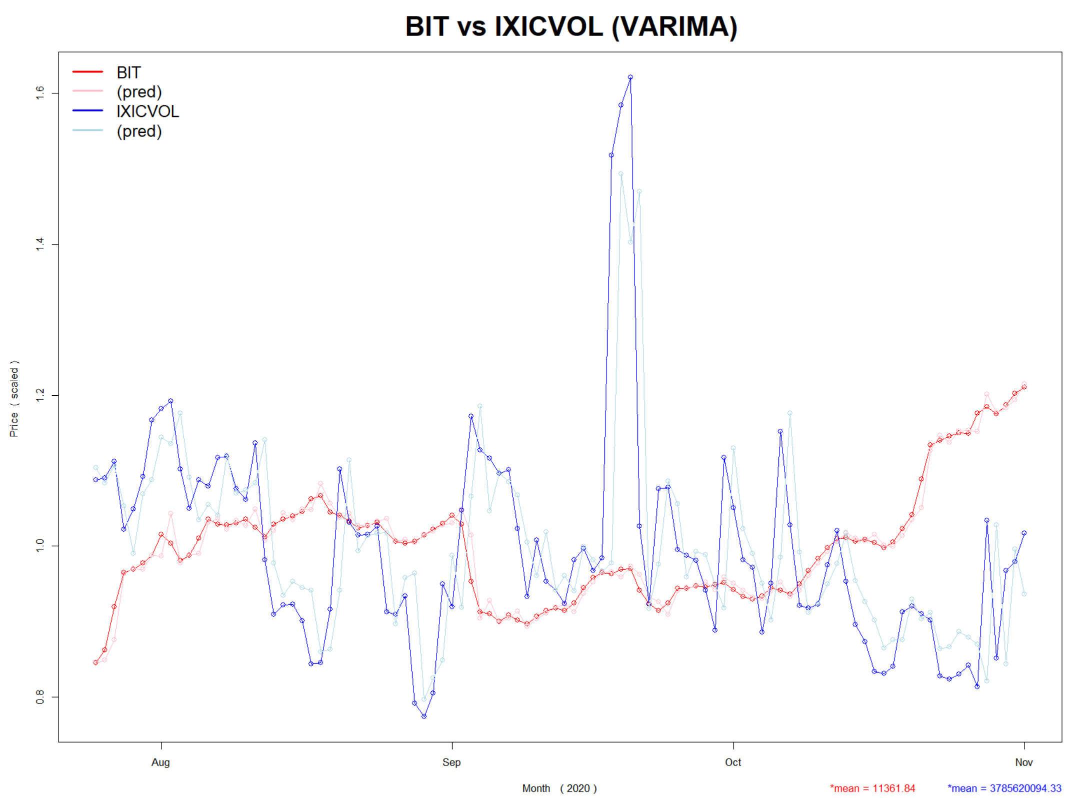



BIT vs Monero (VARIMA)
BIT
(pred)
Monero
(pred)
Price (scaled)
0.7
0.8
0.9
1.0
1.1
1.2
1.3
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 99.42

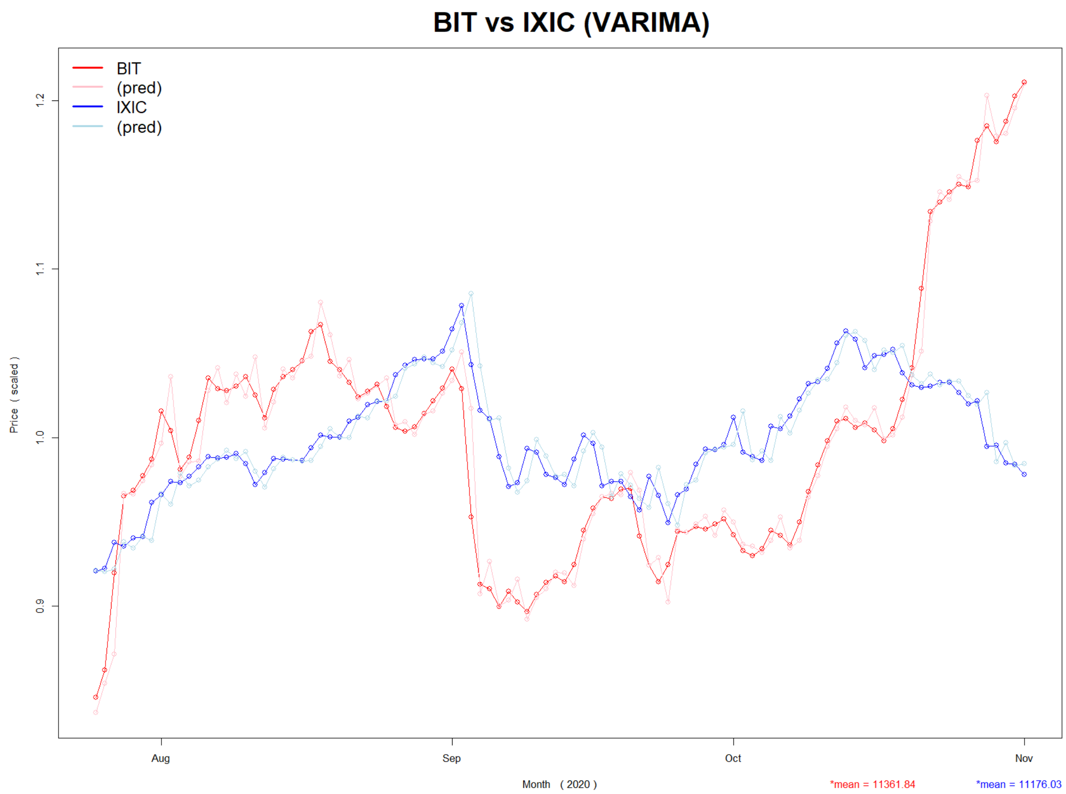

BIT vs LitecoinVOL (VARIMA)
BIT
(pred)
LitecoinVOL
(pred)
Price (scaled)
0.4
0.6
0.8
1.0
1.2
1.4
1.6
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 2448364283.68

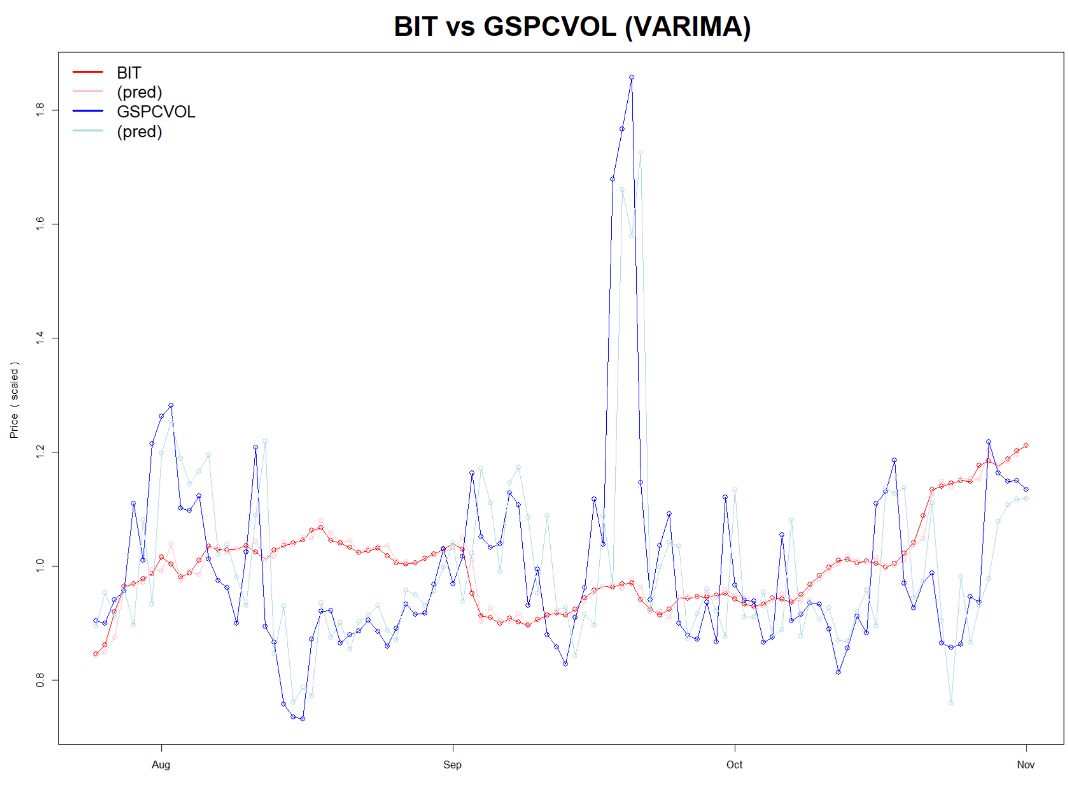


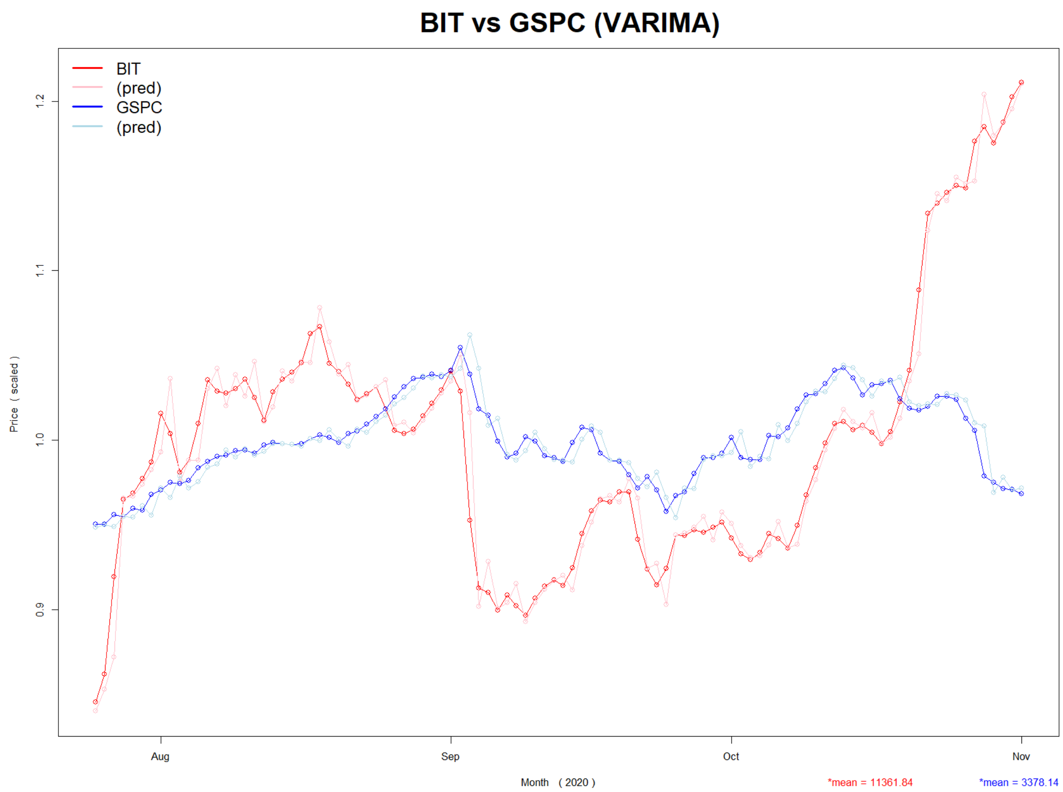
BIT vs Litecoin (VARIMA)
BIT
(pred)
Litecoin
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 52.83

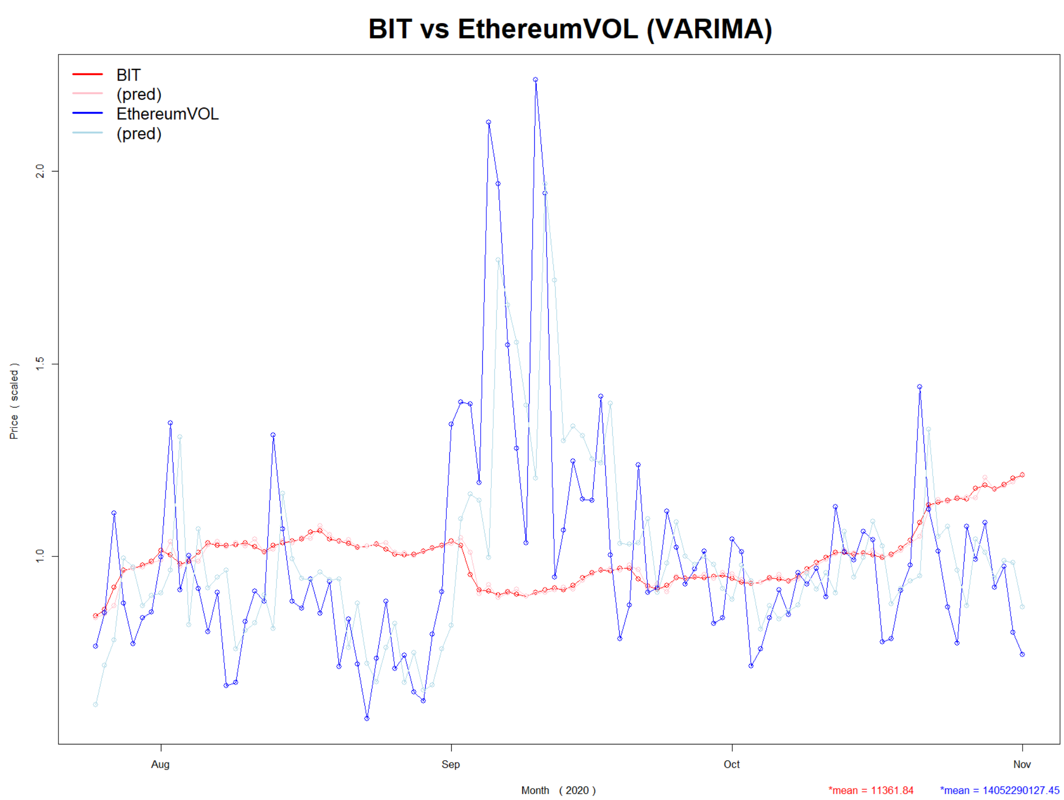


BIT vs IXICVOL (VARIMA)
BIT
(pred)
IXICVOL
(pred)
Price (scaled)
1.6
1.4
1.2
1.0
0.8
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 3785620094.33

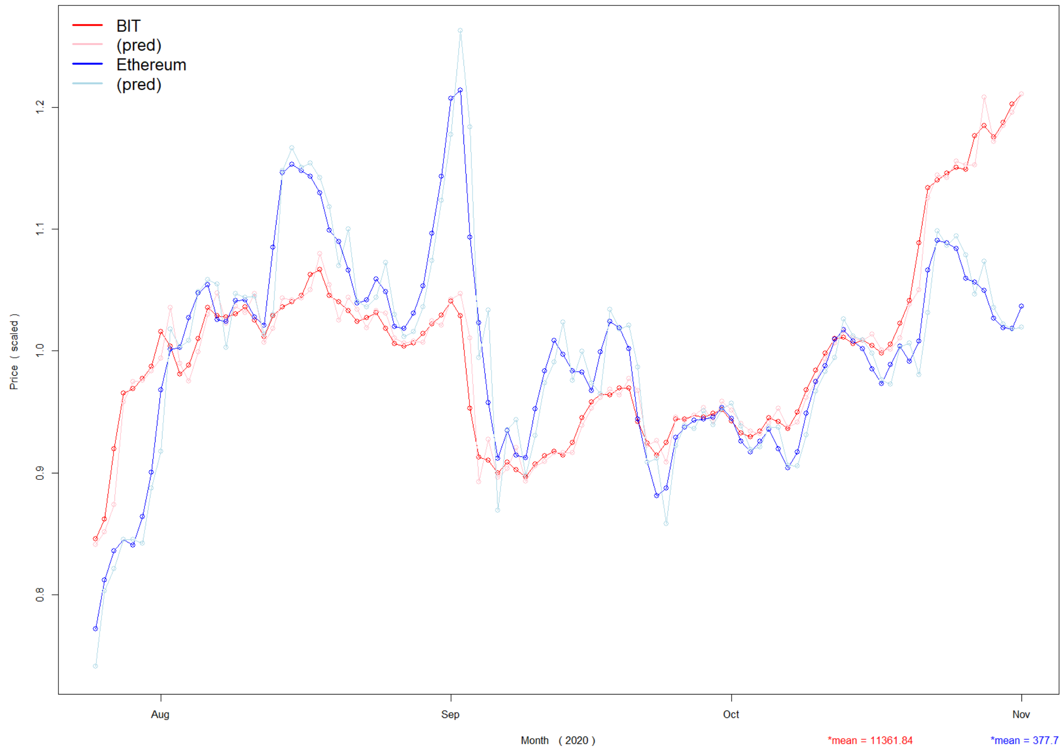

BIT vs IXIC (VARIMA)
BIT
(pred)
IXIC
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 11176.03


BIT vs GSPCVOL (VARIMA)
BIT
(pred)
GSPCVOL
(pred)
Price (scaled)
1.8
1.6
1.4
1.2
1.0
0.8
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 4212325987.86

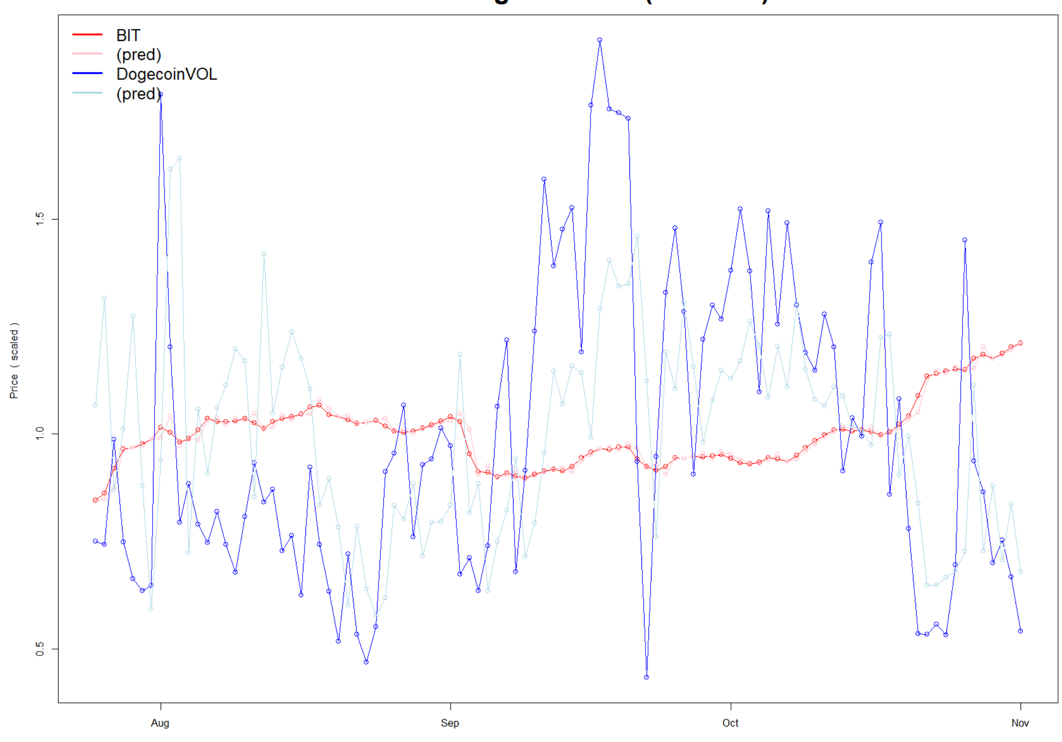

BIT vs GSPC (VARIMA)
BIT
(pred)
GSPC
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 3378.14

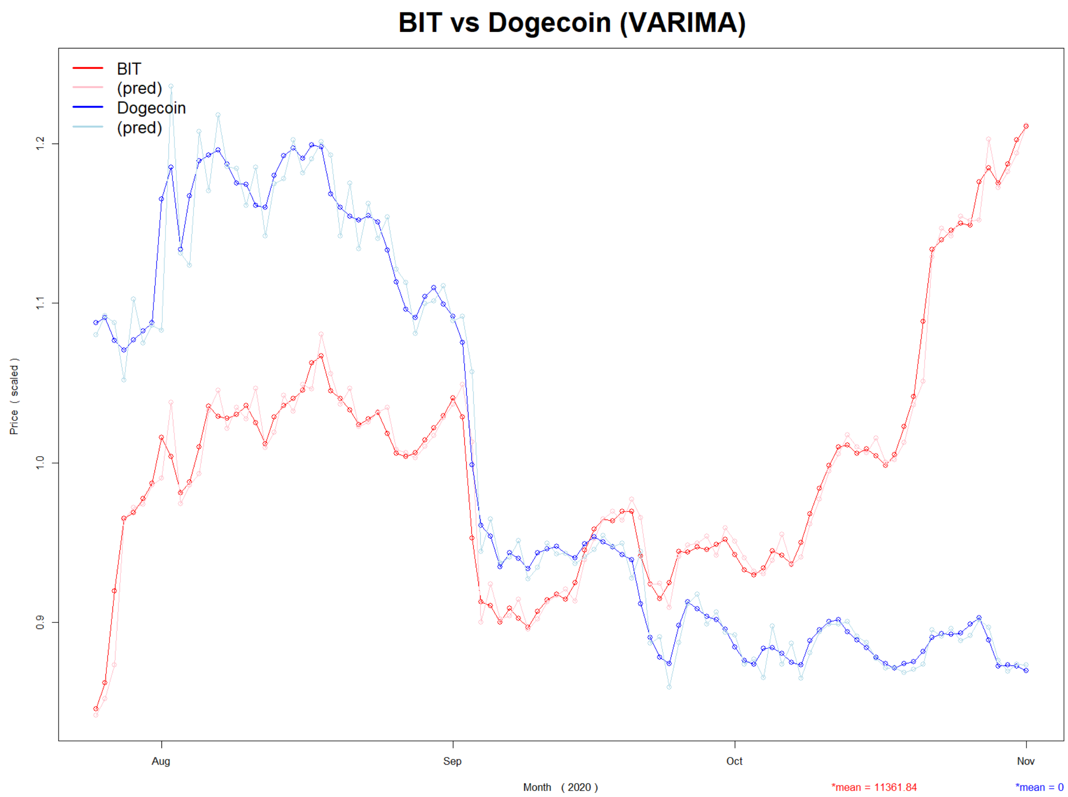

BIT vs EthereumVOL (VARIMA)
BIT
(pred)
EthereumVOL
(pred)
Price (scaled)
2.0
1.5
1.0
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 14052290127.45

BIT vs Ethereum (VARIMA)

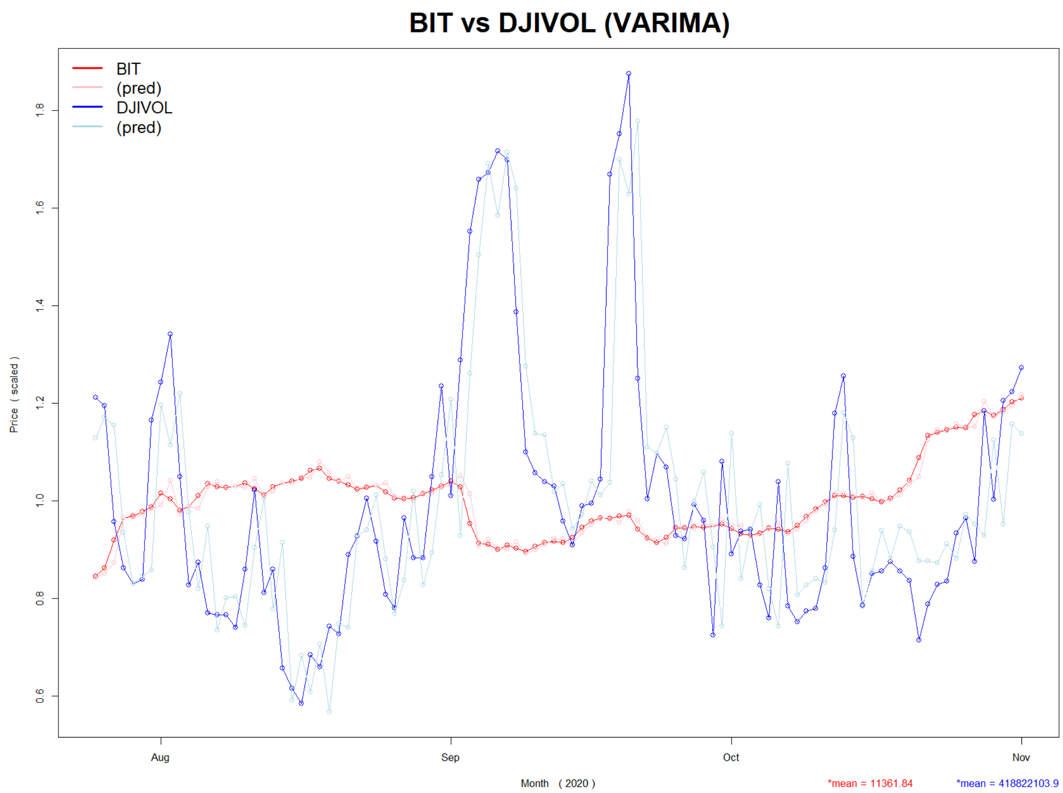


BIT vs DogecoinVOL (VARIMA)

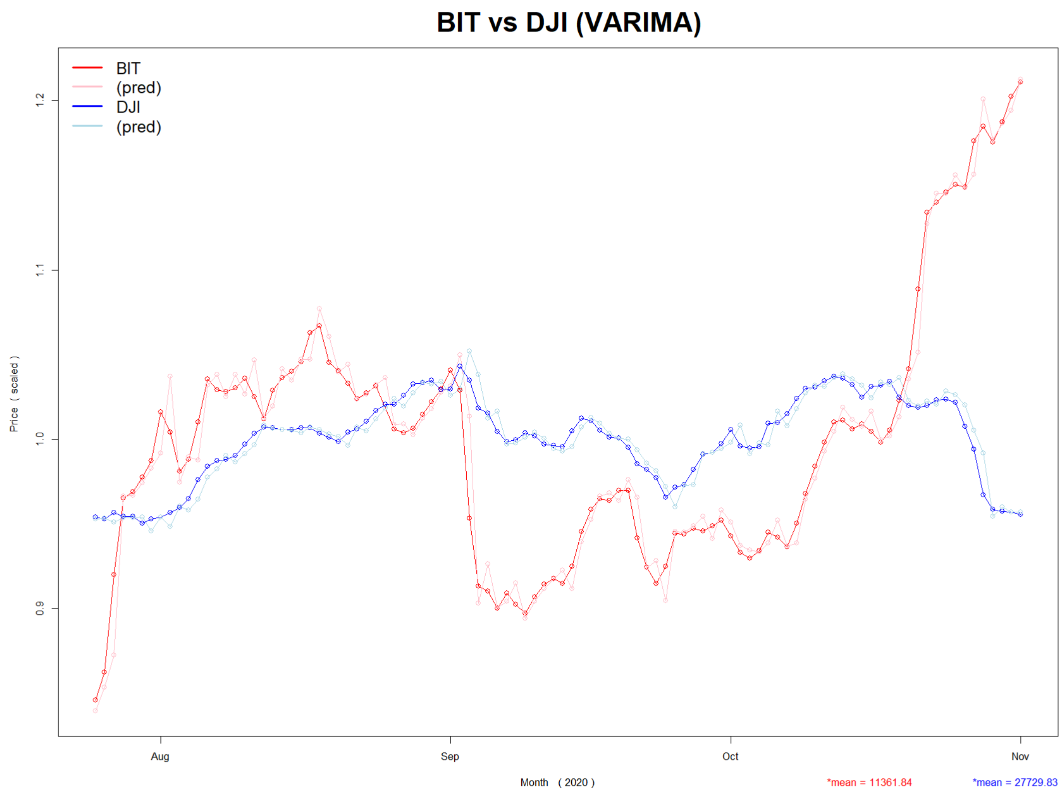


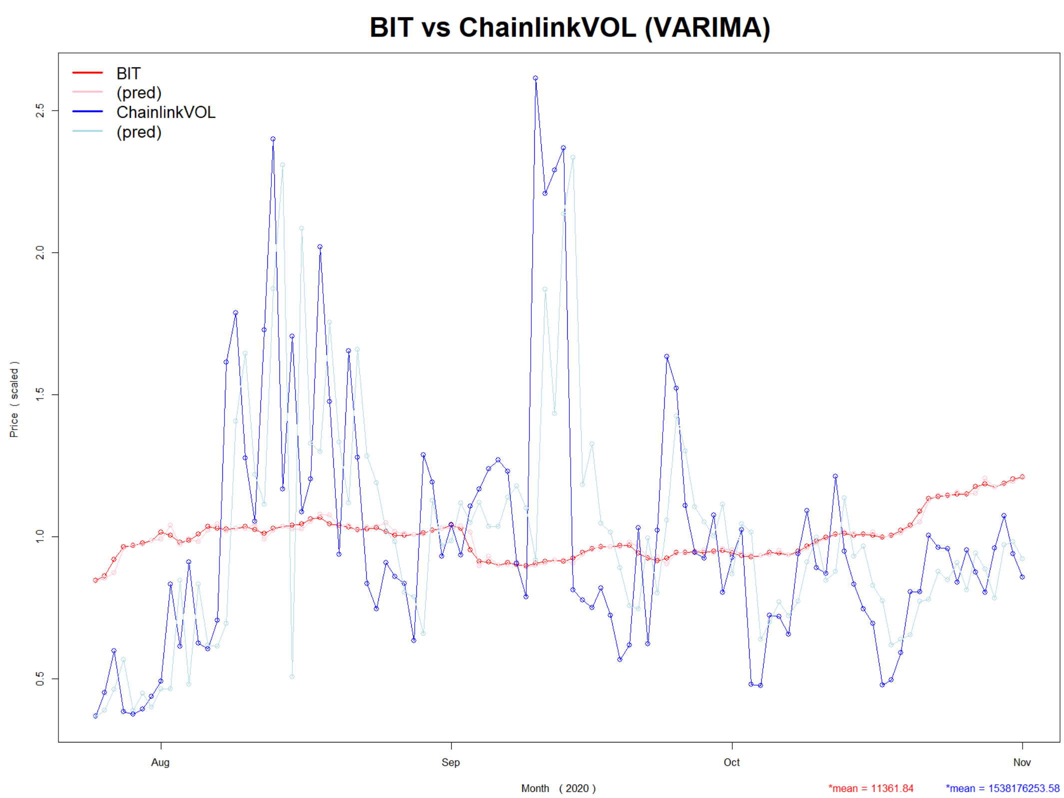


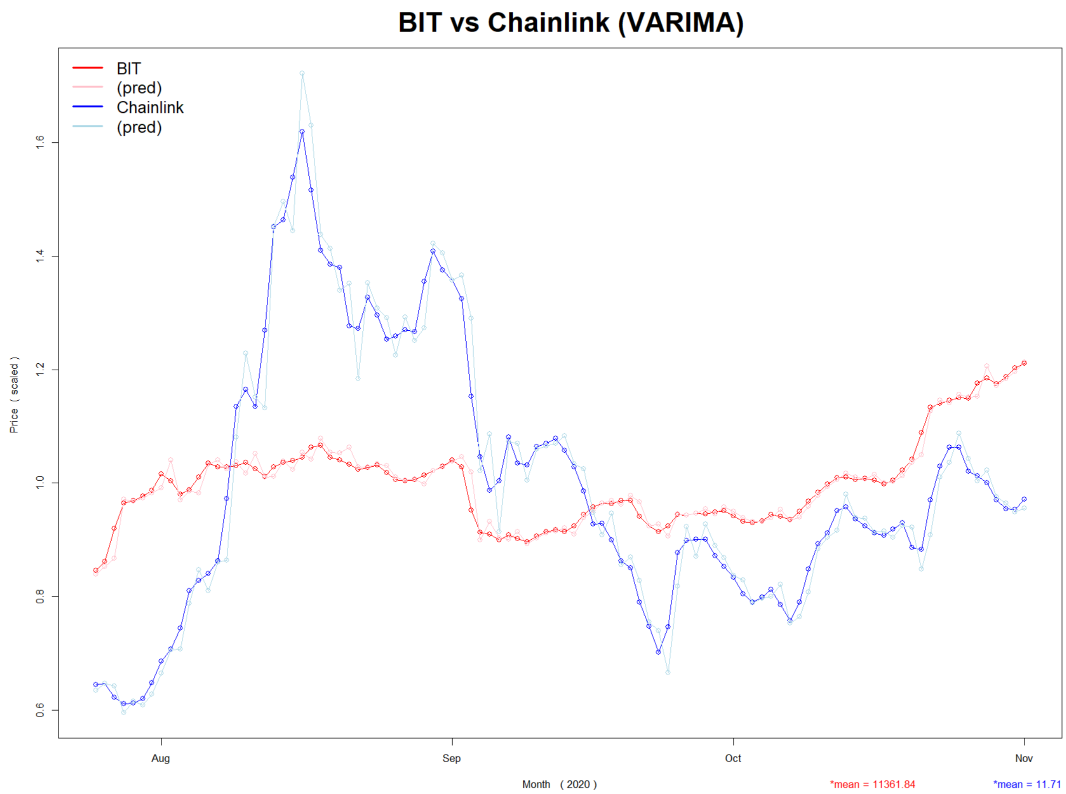
BIT vs Dogecoin (VARIMA)
BIT
(pred)
Dogecoin
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 0

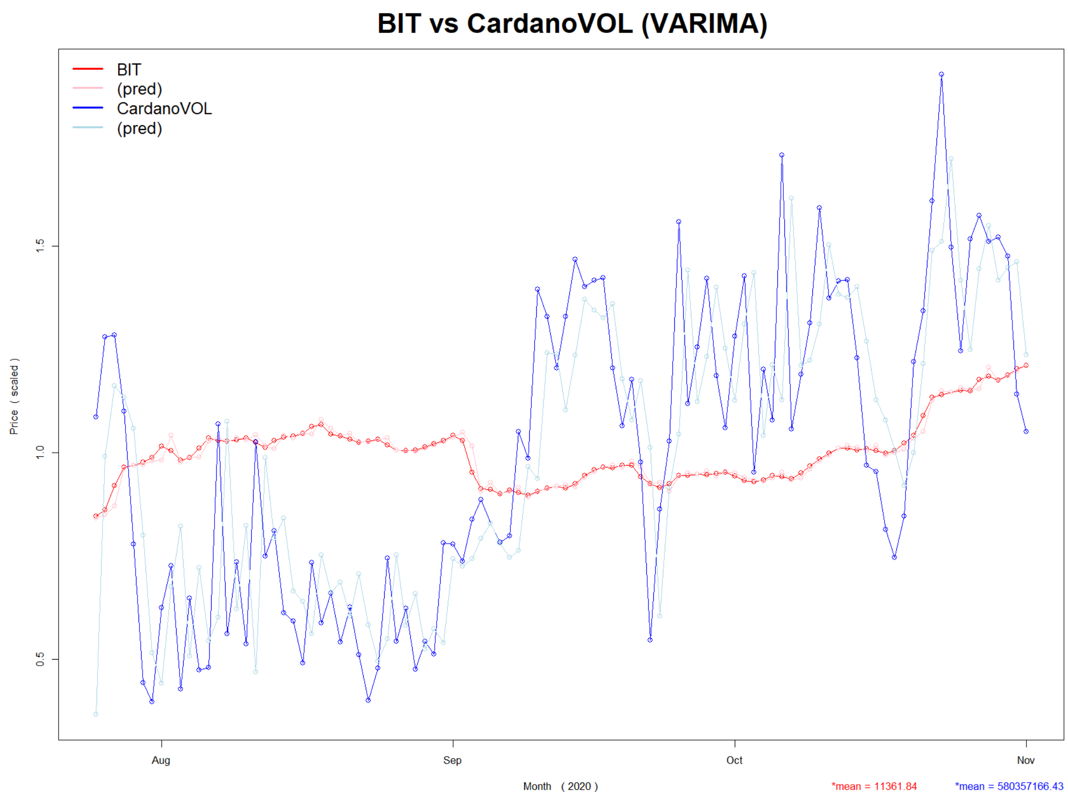


BIT vs DJIVOL (VARIMA)
BIT
(pred)
DJIVOL
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 418822103.9

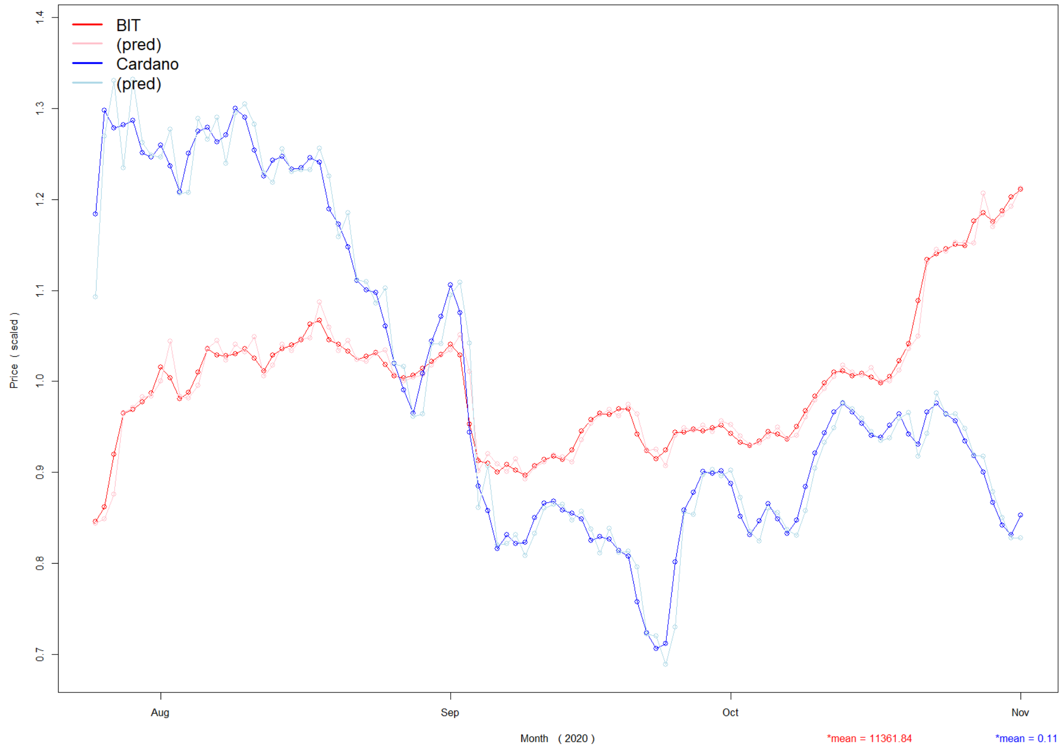

BIT vs DJI (VARIMA)
BIT
(pred)
DJI
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 27729.83


BIT vs ChainlinkVOL (VARIMA)
BIT
(pred)
ChainlinkVOL
(pred)
Price (scaled)
2.5
2.0
1.5
1.0
0.5
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 1538176253.58

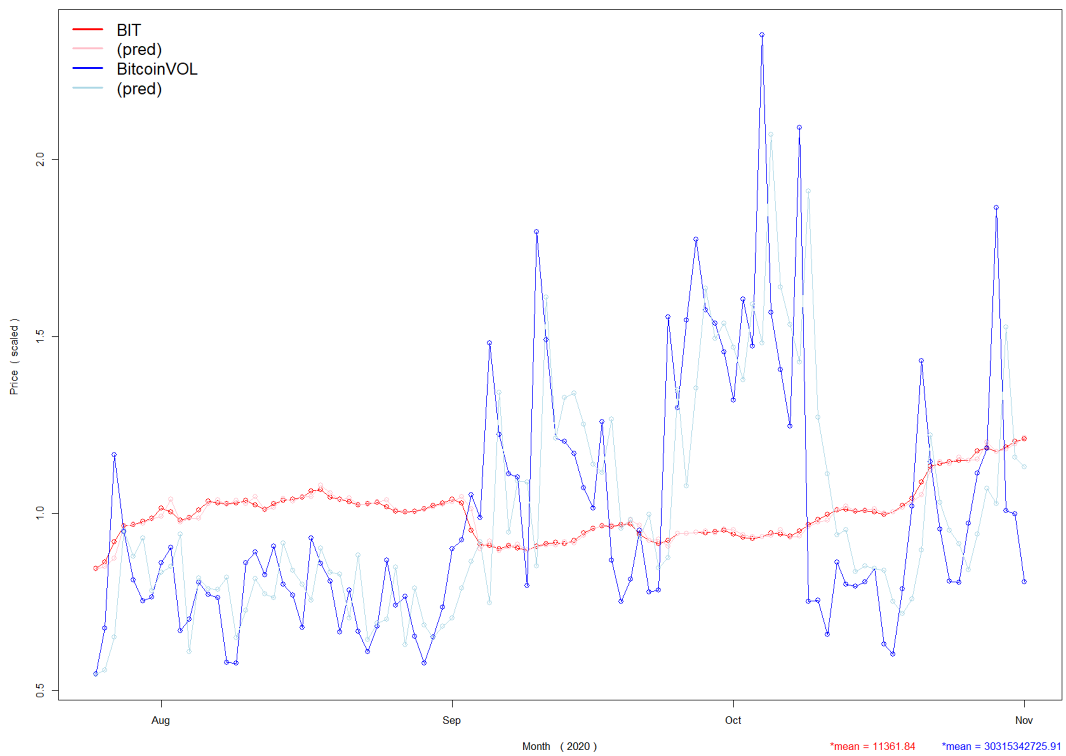

BIT vs Chainlink (VARIMA)
BIT
(pred)
Chainlink
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 11.71

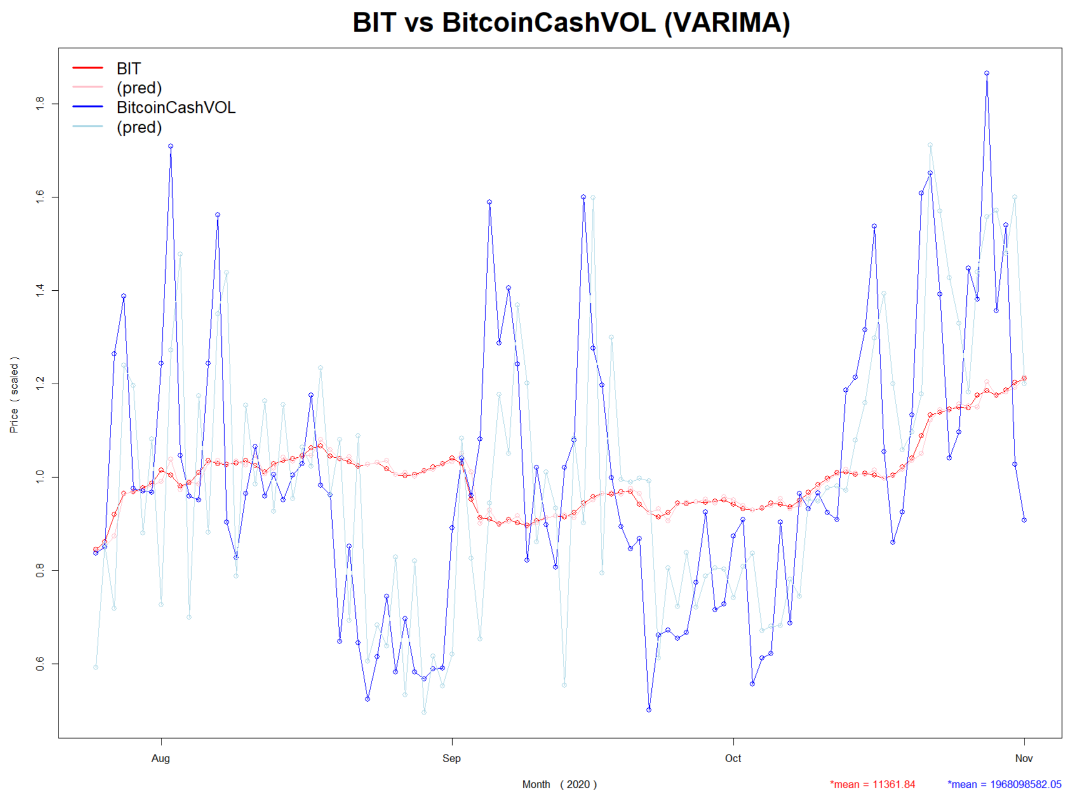

BIT vs CardanoVOL (VARIMA)
BIT
(pred)
CardanoVOL
(pred)
Price (scaled)
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 580357166.43

BIT vs Cardano (VARIMA)

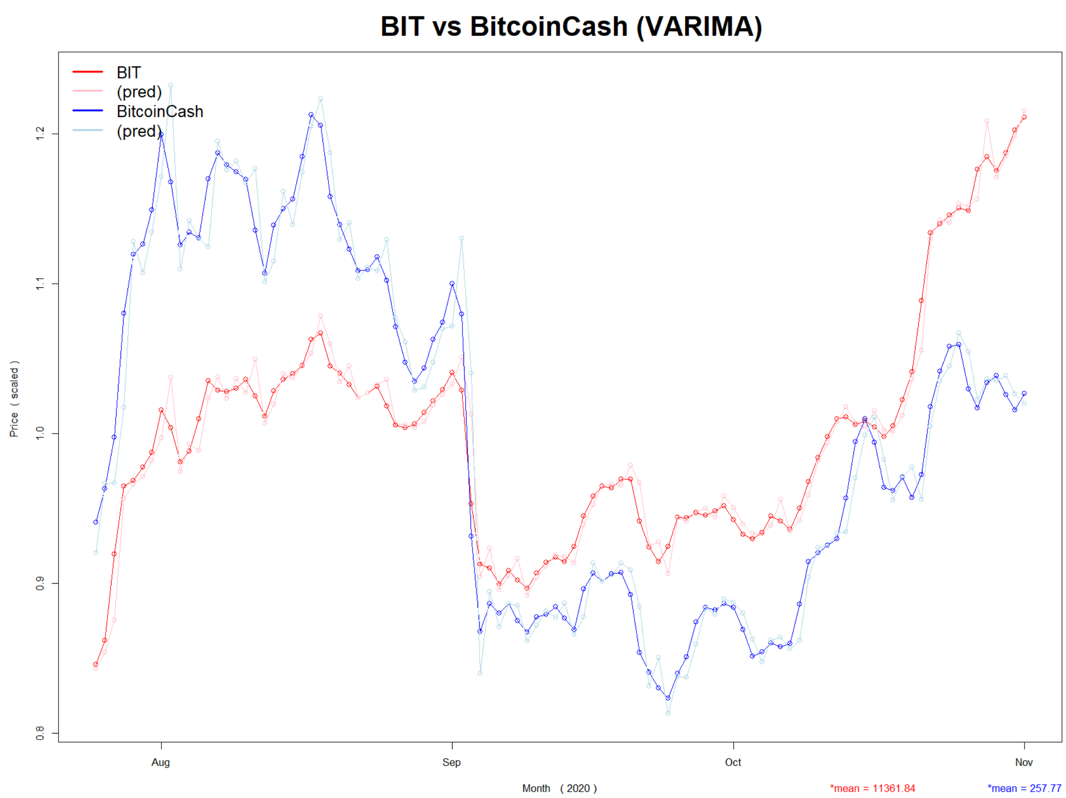


BIT vs BitcoinVOL (VARIMA)

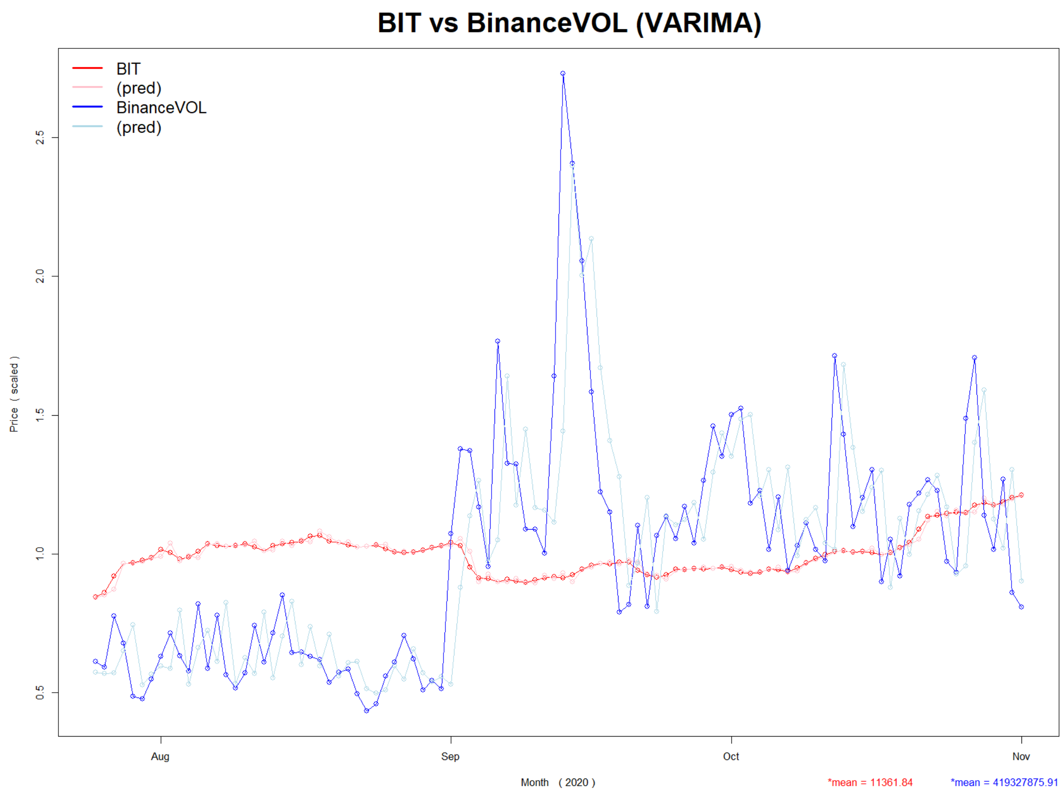


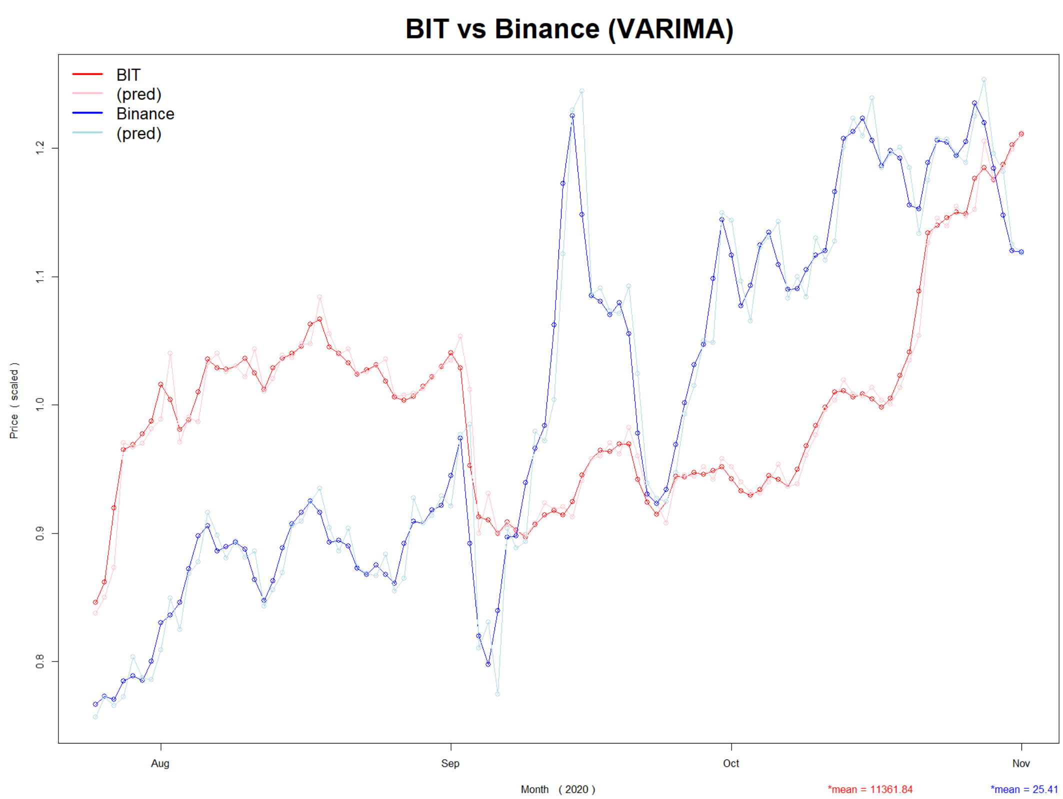


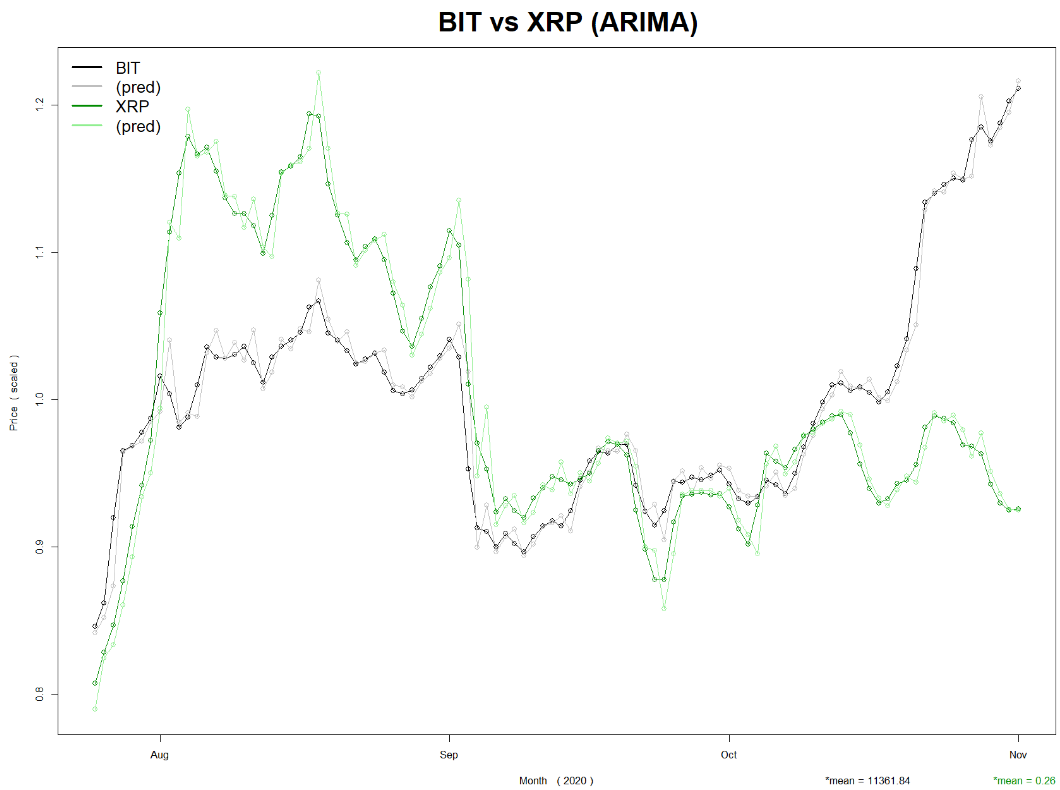
BIT vs BitcoinCashVOL (VARIMA)
BIT
(pred)
BitcoinCashVOL
(pred)
Price (scaled)
0.6
0.8
1.0
1.2
1.4
1.6
1.8
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 1968098582.05

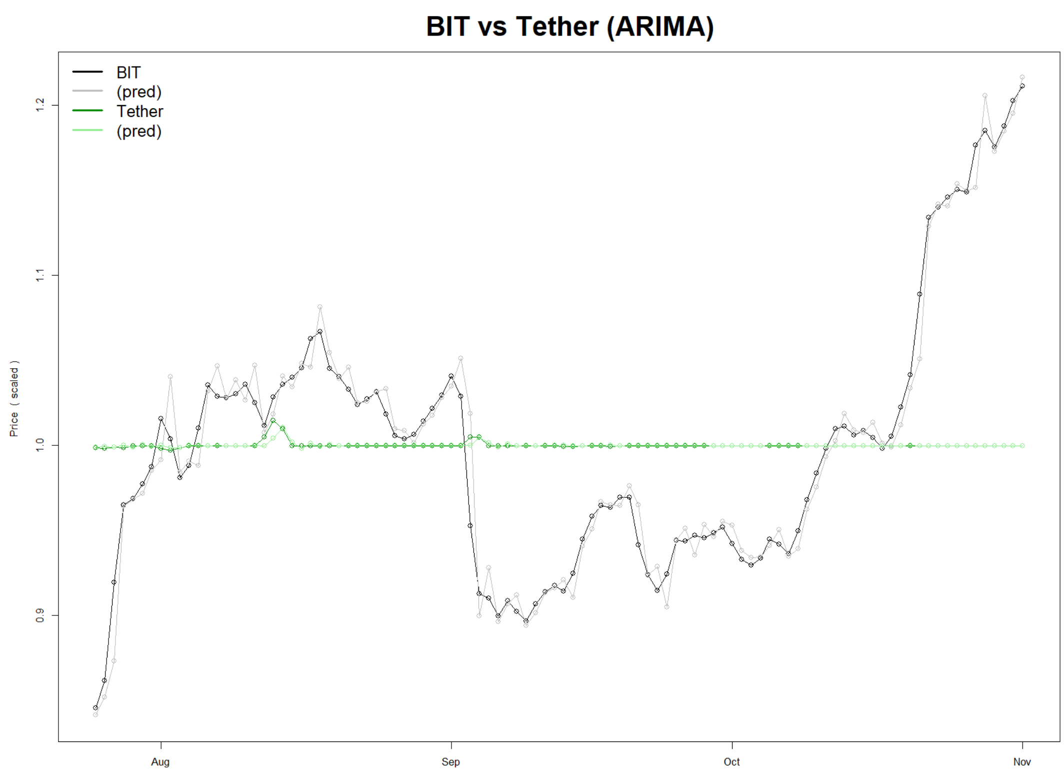


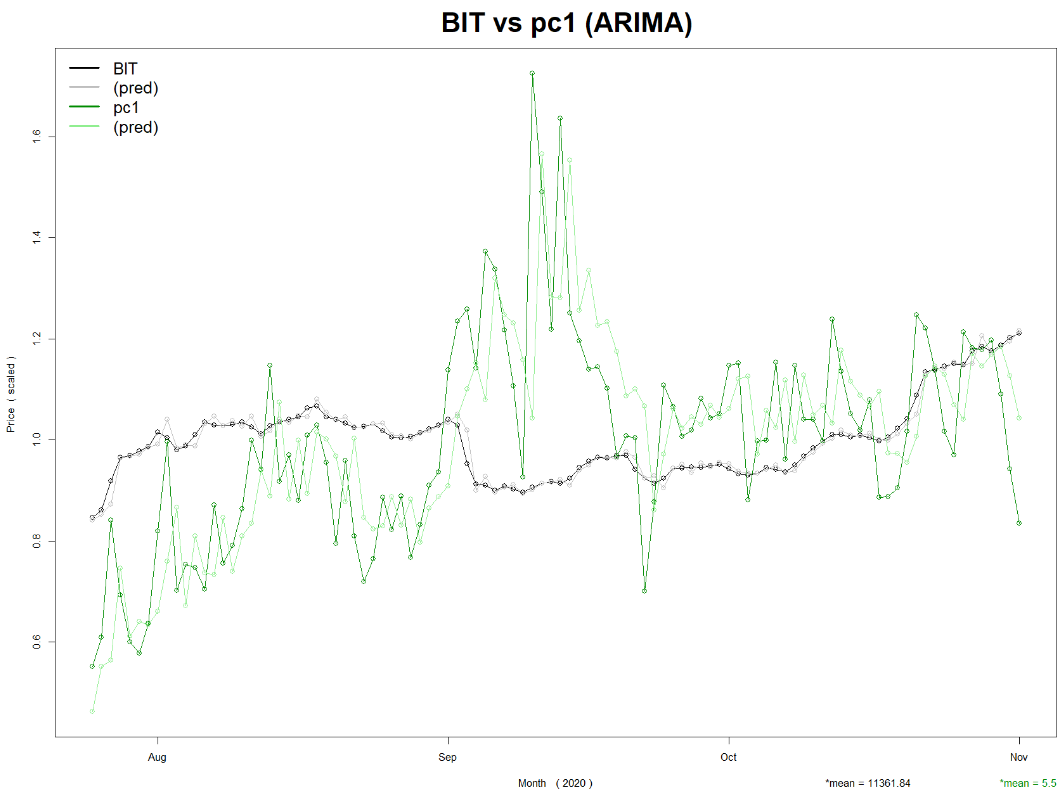
BIT vs BitcoinCash (VARIMA)
BIT
(pred)
BitcoinCash
(pred)
Price (scaled)
0.8
0.9
1.0
1.1
1.2
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 257.77

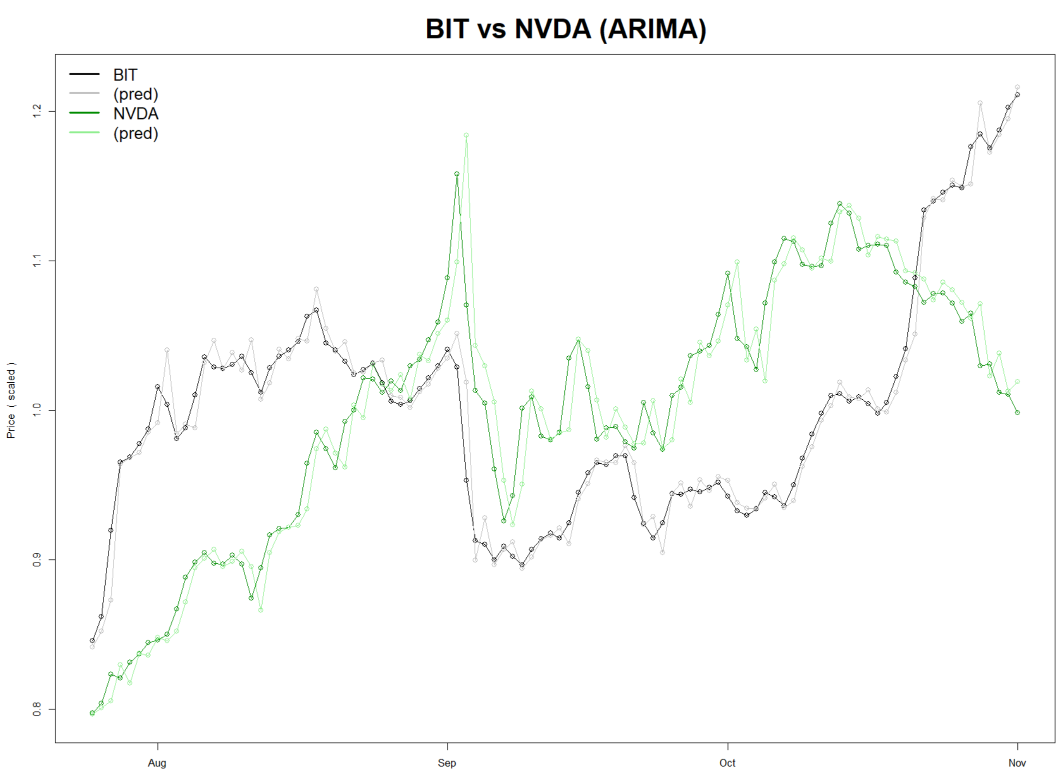

BIT vs BinanceVOL (VARIMA)
BIT
(pred)
BinanceVOL
(pred)
Price (scaled)
0.5
1.0
1.5
2.0
2.5
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 419327875.91

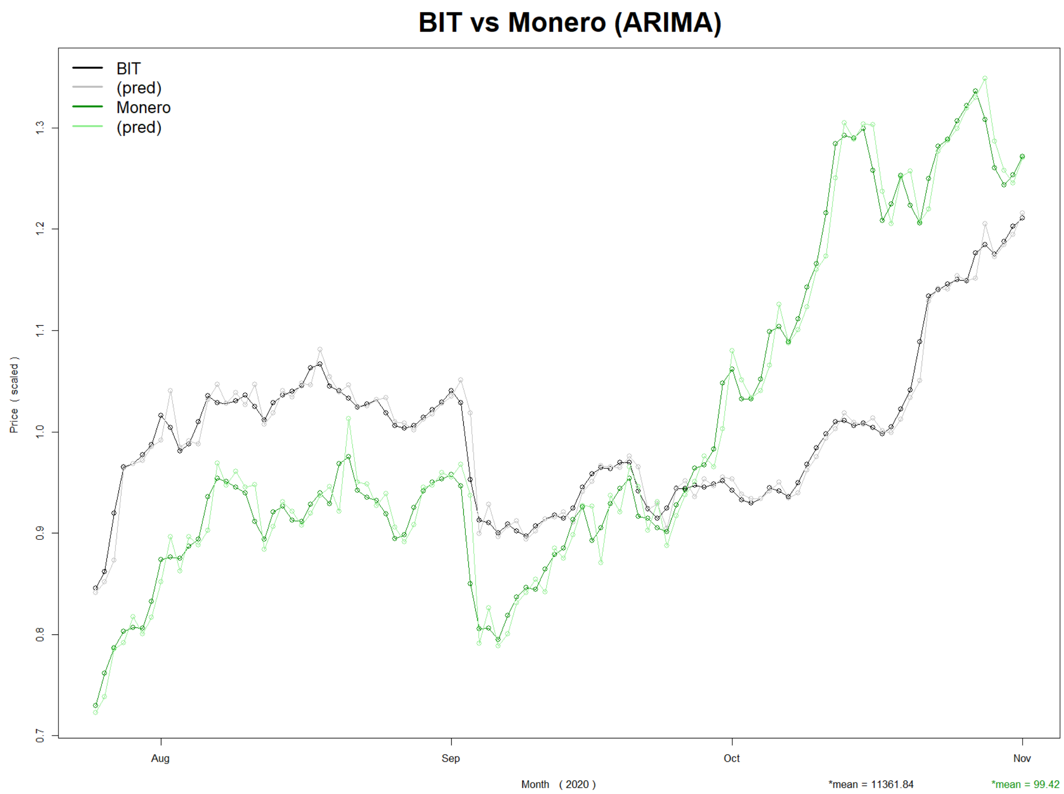

BIT vs Binance (VARIMA)
BIT
(pred)
Binance
(pred)
Price (scaled)
0.8
0.9
1.0
1.1
1.2
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 25.41

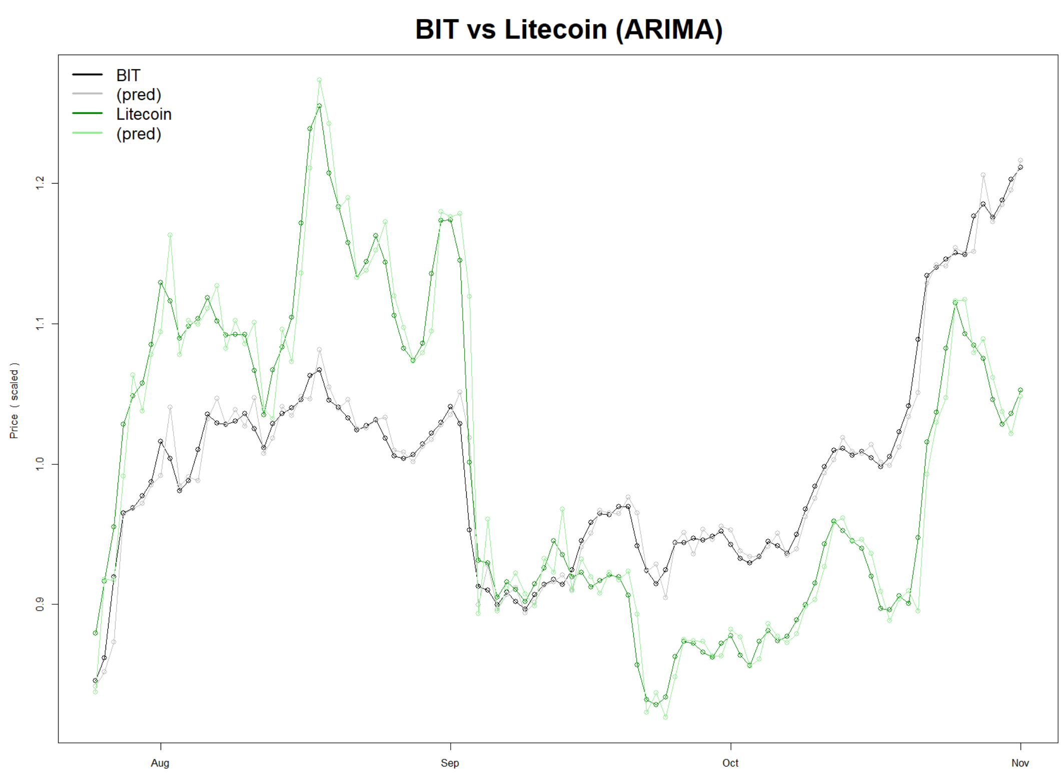

BIT vs XRP (ARIMA)
BIT
(pred)
XRP
(pred)
Price (scaled)
0.8
0.9
1.0
1.1
1.2
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 0.26

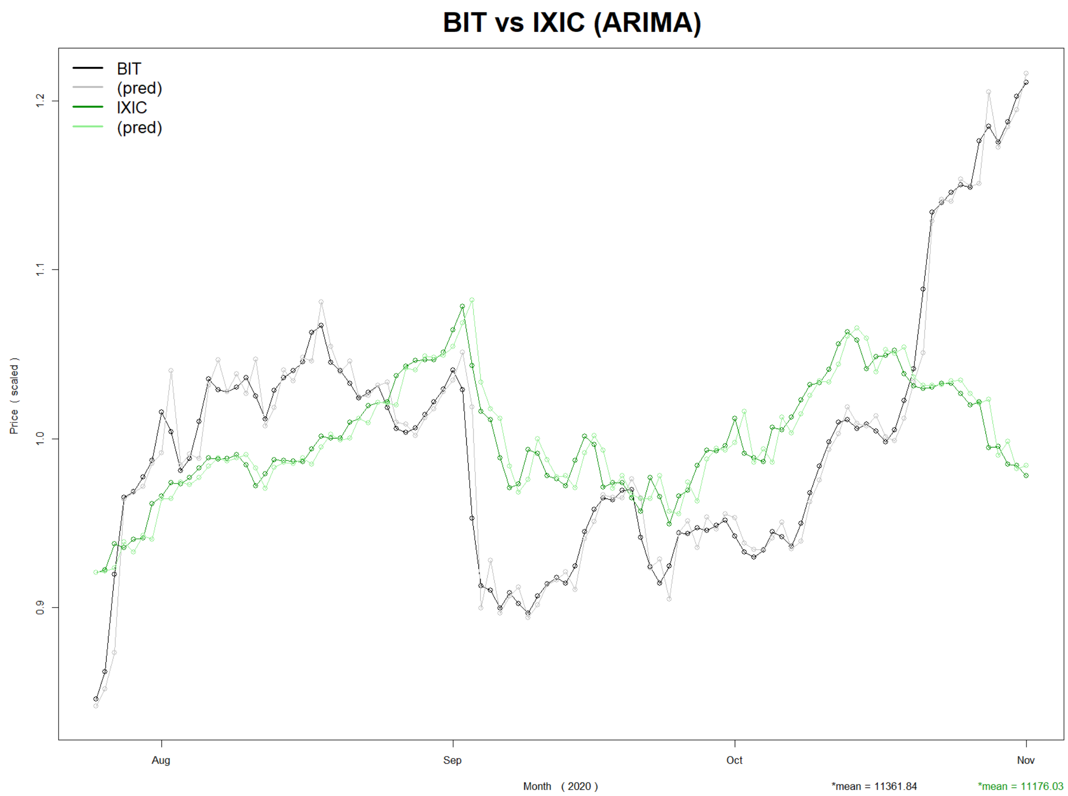

BIT vs Tether (ARIMA)
BIT
(pred)
Tether
(pred)
Price (scaled)
0.9
1.0
1.1
1.2
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 1

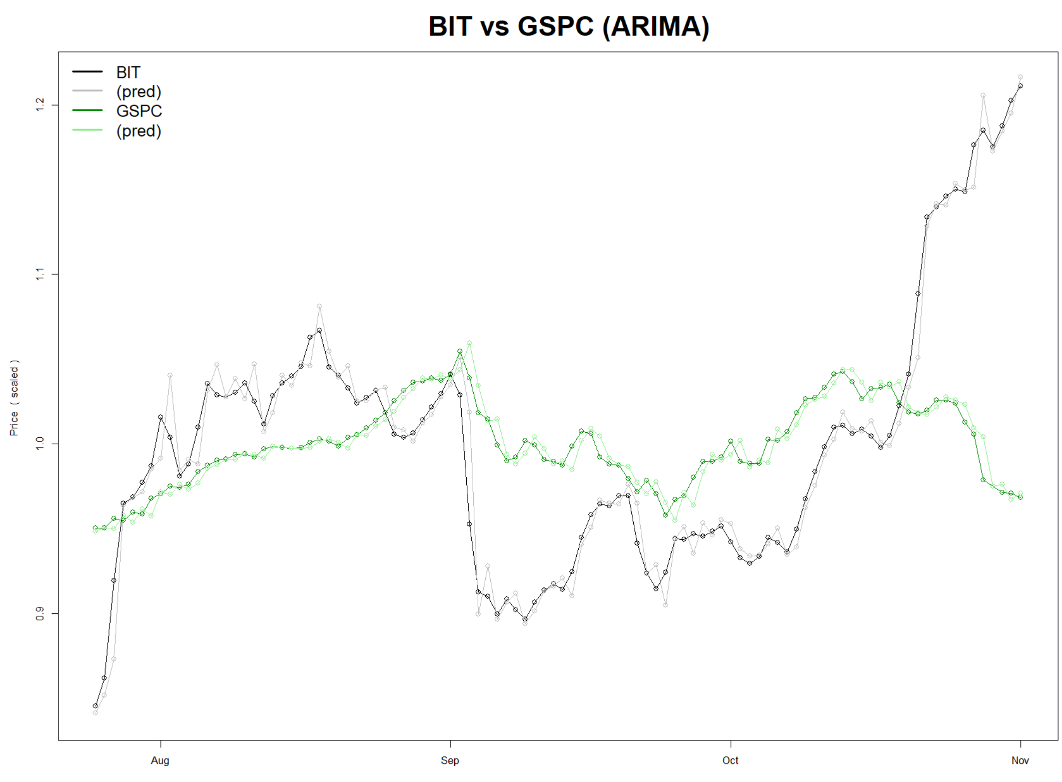

BIT vs pc1 (ARIMA)
BIT
(pred)
pc1
(pred)
Price (scaled)
1.6
1.4
1.2
1.0
0.8
0.6
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 5.5


BIT vs NVDA (ARIMA)
BIT
(pred)
NVDA
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
0.8
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 501.62

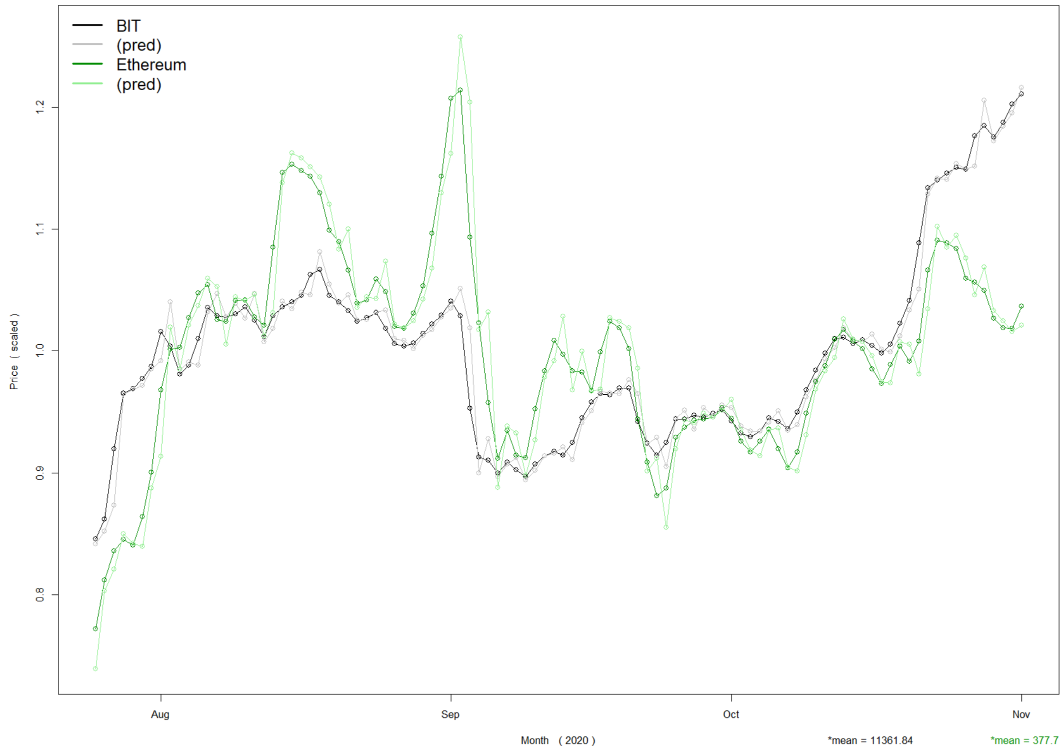

BIT vs Monero (ARIMA)
BIT
(pred)
Monero
(pred)
Price (scaled)
0.7
0.8
0.9
1.0
1.1
1.2
1.3
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 99.42


BIT vs Litecoin (ARIMA)
BIT
(pred)
Litecoin
(pred)
Price (scaled)
0.9
1.0
1.1
1.2
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 52.83

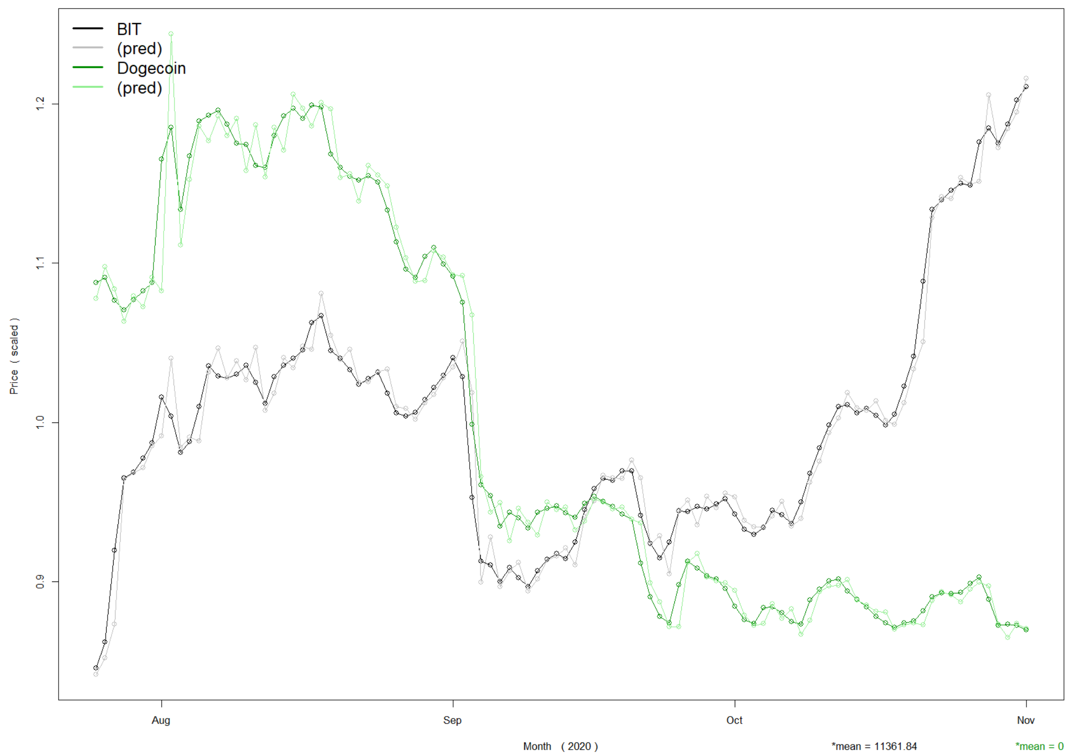

BIT vs IXIC (ARIMA)
BIT
(pred)
IXIC
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 11176.03

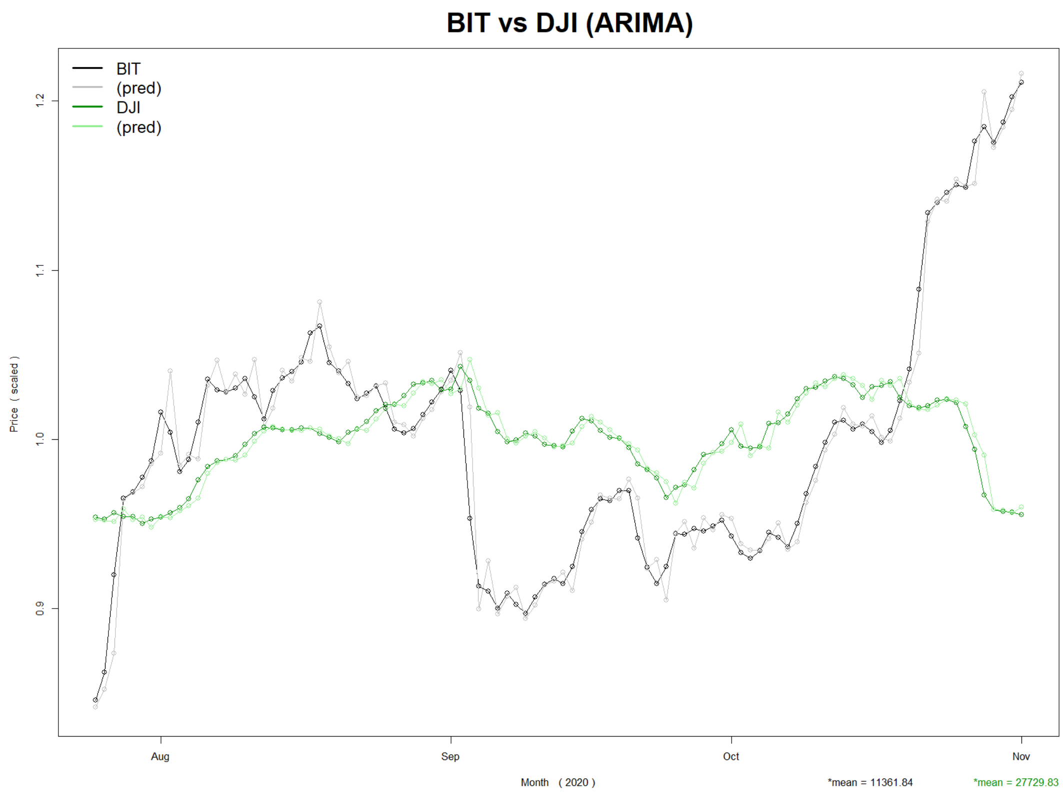

BIT vs GSPC (ARIMA)
BIT
(pred)
GSPC
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 3378.14

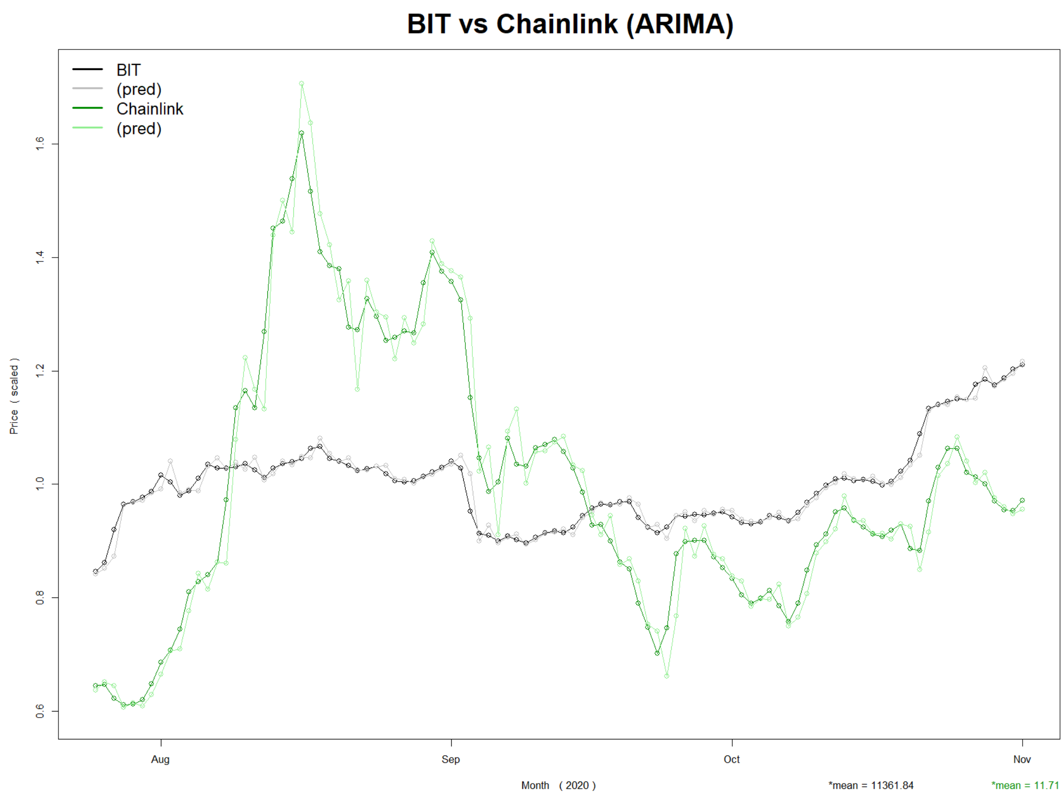
BIT vs Ethereum (ARIMA)

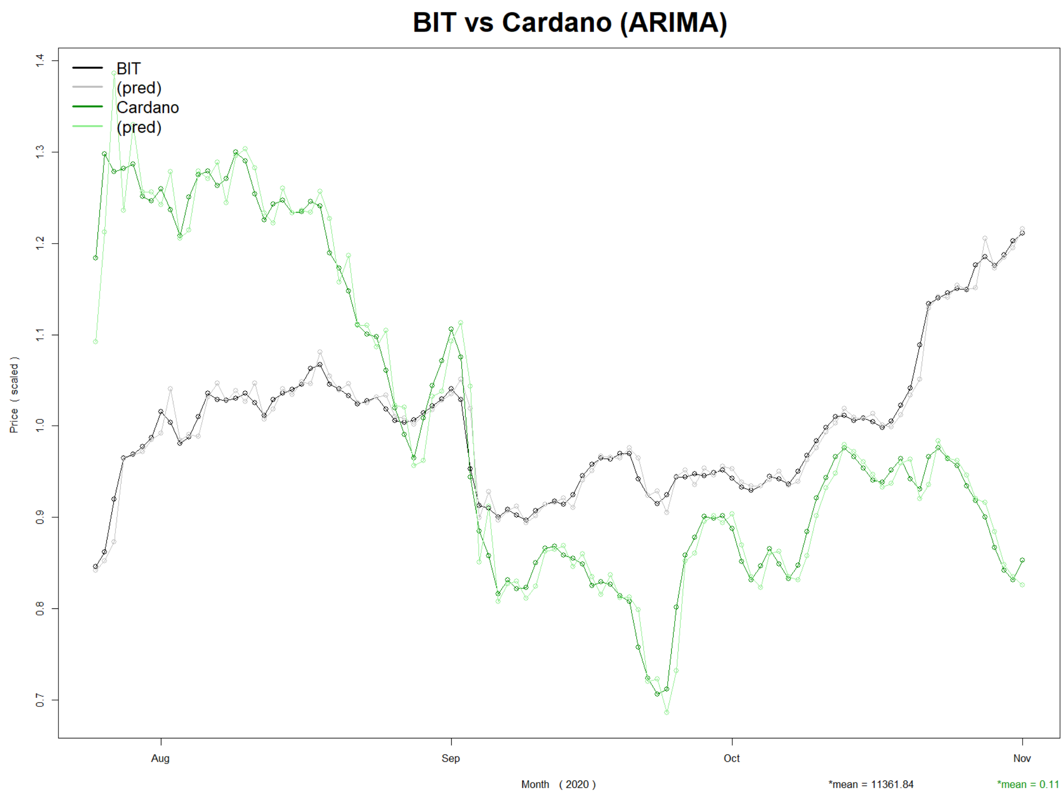


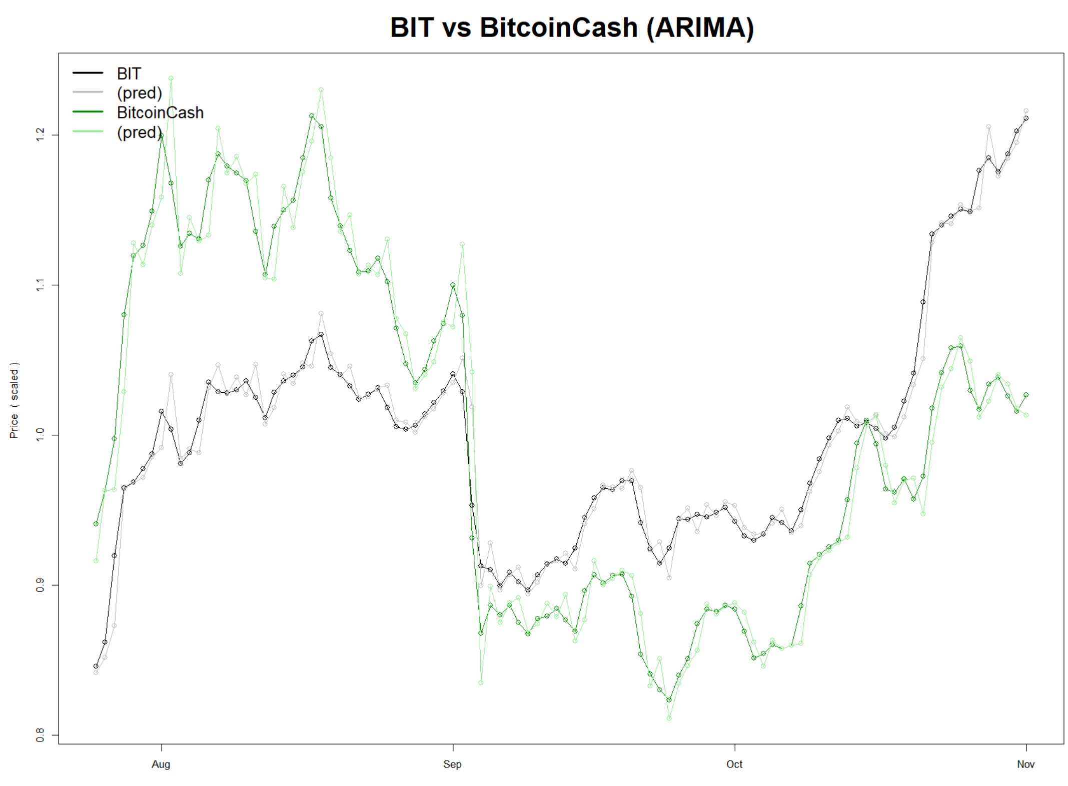
BIT vs Dogecoin (ARIMA)

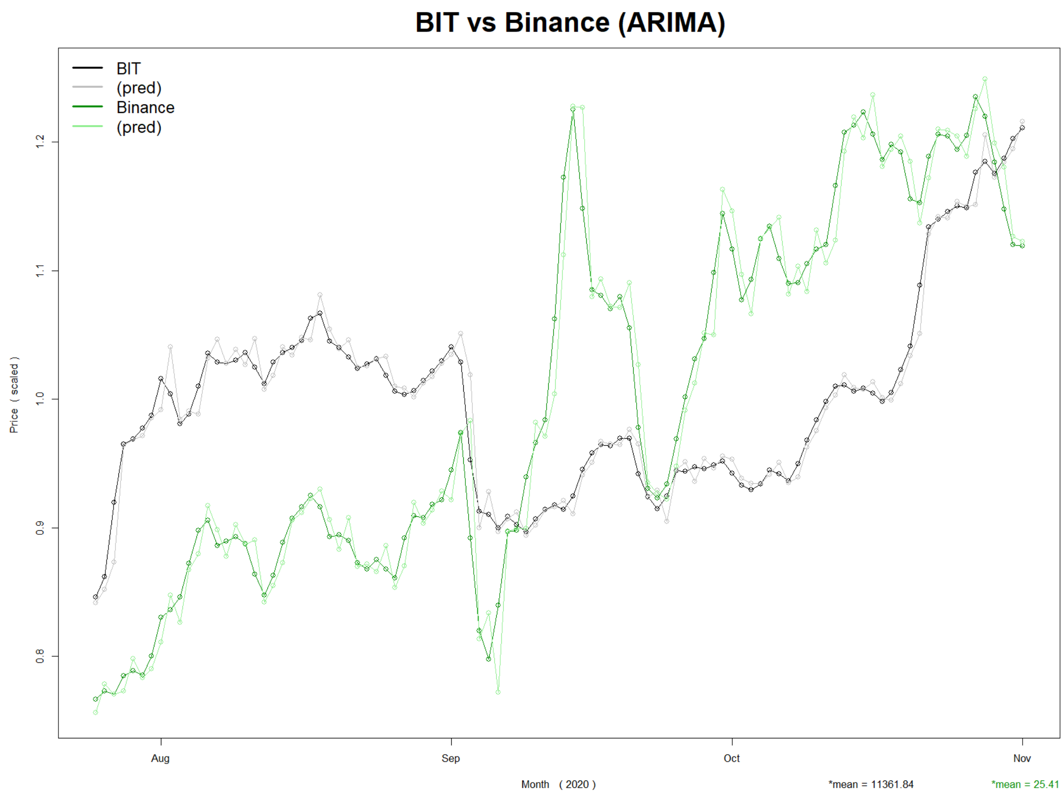



BIT vs DJI (ARIMA)
BIT
(pred)
DJI
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 27729.83


BIT vs Chainlink (ARIMA)
BIT
(pred)
Chainlink
(pred)
Price (scaled)
1.6
1.4
1.2
1.0
0.8
0.6
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 11.71


BIT vs Cardano (ARIMA)
BIT
(pred)
Cardano
(pred)
Price (scaled)
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 0.11


BIT vs BitcoinCash (ARIMA)
BIT
(pred)
BitcoinCash
(pred)
Price (scaled)
1.2
1.1
1.0
0.9
0.8
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 257.77


BIT vs Binance (ARIMA)
BIT
(pred)
Binance
(pred)
Price (scaled)
0.8
0.9
1.0
1.1
1.2
Aug
Sep
Oct
Nov
Month (2020)
*mean = 11361.84
*mean = 25.41