Do Google Trends Forecast Bitcoins? Stylized Facts and Statistical Evidence

Argimiro Arratia[1] and Albert X. López Barrantes[2]

[1] Universitat Politècnica de Catalunya,
Dept. of Computer Science, Barcelona, SPAIN
argimiro@cs.upc.edu
[2] Universitat Autónoma de Barcelona, SPAIN
albertxavier.lopez@gmail.com

Abstract. In early 2018 Bitcoin prices peaked at US$ 20,000 and, almost two years later, we still continue debating if cryptocurrencies can actually become a currency for the everyday life or not. From the economic point of view, and playing in the field of behavioral finance, this paper analyses the relation between Bitcoin prices and the search interest on Bitcoin since 2014. We questioned the forecasting ability of Google Bitcoin Trends for the behavior of Bitcoin price by performing linear and nonlinear dependency tests, and exploring performance of ARIMA and Neural Network models enhanced with this social sentiment indicator. Our analyses and models are founded upon a set of statistical properties common to financial returns that we establish for Bitcoin, Ethereum, Ripple and Litecoin.

Keywords: Google Trends, Bitcoin, causality, ARIMA, Neural Networks

1 Introduction

Bitcoin is the most popular and prominent cryptocurrency in the world. The first design was published in 2008 under the pseudonym of Satoshi Nakamoto [13]. A currency where everyone from anywhere can execute transactions with no need of a traditional financial institution involved in the process generated a huge expectation around the world rising Bitcoin prices to a peak of US$ 20,000 in January 2018. However, the low number of transactions per second that are able to support and the non-recognition as a currency by most companies and governments, caused a big drop in the price to the current US$ 3,000 price. A new economic bubble in the exact sense of the term: a huge increase on expectations on a short period of time, often coming from the irrational decisions we do as human beings. We can indeed observe that a classical financial bubble have formed on the Bitcoin price time series (Figure 1 left). And with the intention to build upon research on the use of big data in social media to construct predictors for various economic variables or social events as in, for example, [1, 3, 8] and most relevant for this project the paper [6], we looked at the trend of the topic "bitcoin" in Google Trends for the last four years, and obtained the time series depicted in Figure 1 (right).

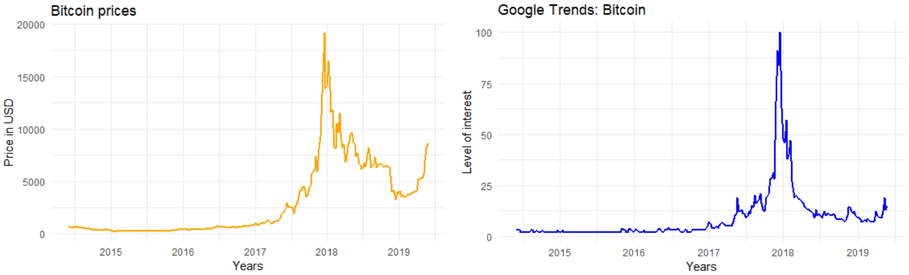

Fig. 1. Bitcoin prices and Google Trends on topic "bitcoin" series

The strong resemblance among both time series may lead one to think of the Google Bitcoin Trends (GBTrends) as predictor of Bitcoin price behavior. But being cautious about drawing conclusion from pictures and aware of the failure of the Google Flu Trends [11], we turn to traditional statistical methods to prove hypothesis.

2 Stylized empirical facts and statistical issues

Financial time series have a common set of characteristics that should be studied and considered before any further analysis, modelling or conclusions. These are commonly known as stylized empirical facts [4], and knowing which of these hold will guide on proper modeling of the financial asset. However, a basic underlying hypothesis is that of stationarity, and more often than not, these data sets are non stationary which means that we have to transform them, usually by considering returns, in order to recognize some statistical properties of the data which remain invariant over time, and so that modeling is possible.

2.1 Data gathering

We obtained Bitcoin prices from *CoinMarketCap.com*. The data set contains Bitcoin daily prices since 2014 with access to Open and Closing prices as well as volume traded and the total market capitalization over time. On the other hand, Google Trends on the word "Bitcoin" can be obtained through the R package *gtrendsR* where one can query everything with the same parameters over time and geography as in the web page of Google Trends. One has only to take into account the minimum time scale one can get from Google Trends, which for this paper we got a weekly time scale since 2014.

2.2 Stationarity

Kendall [10] was the first one to realize financial time series are seldom stationary. To get close to be stationary, or at least to be *second-order stationary*, a common technique is to apply successive differences to the series. Hence, it is recommendable to work instead with the series logarithmic returns: $r_t = \log\left(\frac{P_t}{P_{t-1}}\right)$.

There are several tests for stationarity and a few for second order stationarity; a survey of the former kind of stationarity and a proposal for the latter can be seen in [15]. In this work we applied two tests for second order stationarity: the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test with null hypothesis that an observable time series is stationary around a deterministic trend; and the Priestley-Subba-Rao (PSR) test that investigates how "non-constant" the time-varying Fourier spectrum of the series is. The KPSS is implemented in the R-package `tseries` test, and PSR is implemented in `fractal` package.

Table 1. Priestley-Subba-Rao (PSR) test

series	**p-value for T**
Bitcoin returns	3.74812e-05
Ethereum returns	0.001743593
Ripple returns	2.279732e-12
Litecoin returns	4.931523e-08
GBTrends returns	0

We ran both tests for all four cryptocurrencies and GBTrends returns, and for all the null hypothesis of stationarity was rejected. Hence modeling is in principle in this context not statistically well-founded.

2.3 Aggregational Gaussianity

One of the most used conventions when working with financial data is the assumption that returns are log-normally distributed, which is equivalent to the assumption that log-returns are normally distributed. Early in 1953 Kendall, Mandelbrot in 1960 or Fama in 1965, among other researchers, signaled the non normal distribution of asset returns and the heavy tails [10, 5]. However, it is remarkably important that distributions become closer to normal when the timescale increases. This convergence in distribution towards normality as timescale increases is called Aggregational Gaussianity and is widely documented across assets around the world. In our case, we checked for this phenomenon in all cryptocurrencies returns series.

For Bitcoin daily returns we found the typical leptokurtic distribution, sharp peaked and heavy tailed, which is far from being normally distributed. However, once we increase the timescale to weekly samples, the distribution gets closer

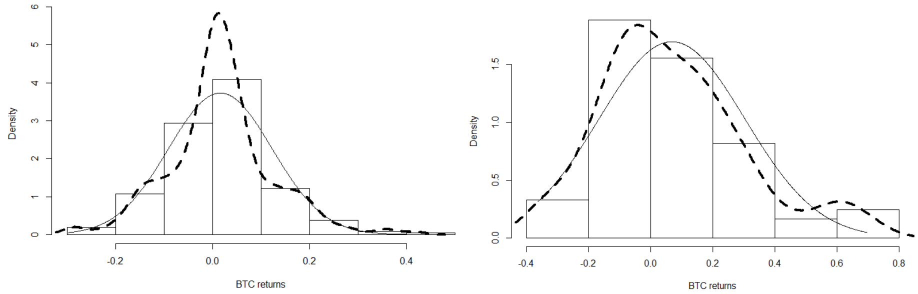

Fig. 2. Weekly (left) and monthly (right) bitcoin returns estimated density (dashed line) and normal density fit (solid line)

to a normal distribution. It is no until we get the monthly returns distribution when we closer to a normal distribution. On the timescale of weekly returns, if we try to fit the GBTrends series we get a similar shape as the weekly Bitcoin prices, far from a normal shape with considerable fat tails and a considerable peak, which can be observed also in the skewness comparative at the end.

However, a visual analysis is not enough to say if the aggregational gaussianity is happening on cryptocurrency's returns. To properly check this, we will use Shapiro-Wilk and Jarque-Bera normality tests on different time scales of Bitcoin, Ethereum, Ripple and Litecoin returns. The null hypothesis of Shapiro-Wilk normality test is data is normally distributed, and the null hypothesis of Jarque-Bera test is a joint hypothesis of $skewness = 0$ and excess $kurtosis = 0$.

Overall, Bitcoin returns is in line with literature on Aggregational Gaussianity, the more we increase the timescale, the closer we get to have log normal returns, specifically when we reach a monthly times scale. For the other three cryptocurrencies results are more extreme, since we only get normality once we reach yearly returns in Jarque-Bera test. Since our data starts at 2014 for all cryptocurrencies and GBTrends, yearly returns are calculated on 5 observations, which invalidates any results of these tests of this timescale.

The work by Chan et al [2] goes a step further by fitting non-normal distributions to each of the cryptocurrencies. They find that for Bitcoin and Litecoin, the generalized hyperbolic distribution gives the best fit, while the other cryptocurrencies return distributions are better fitted by the normal inverse Gaussian, generalized t and Laplace distributions.

2.4 Autocorrelations

We now check the possibility of self linear association of the cryptocurrencies return time series, by computing the ACF and PACF. The auto-correlation function (ACF) gives us the values of autocorrelation between the time series and its lagged values. The partial auto-correlation function (PACF) gives the correlation values of the residuals with the next lag value.

Table 2. Normality tests

Data	**Shapiro-Wilk**	**Jarque-Bera**
	p-value	*p*-value
Weekly returns GBTrends	0.00001	0.00001
Daily returns Bitcoin	0.00001	0.00001
Weekly returns Bitcoin	0.00006	0.00001
Monthly returns Bitcoin	0.03808	0.07850
Quarterly returns Bitcoin	0.00633	0.01168
Yearly returns Bitcoin	0.00126	0.20780
Daily returns Ethereum	0.00001	0.00001
Weekly returns Ethereum	0.00001	0.00001
Monthly returns Ethereum	0.00002	0.00001
Quarterly returns Ethereum	0.00004	0.00001
Yearly returns Ethereum	0.00103	0.40190
Daily returns Ripple	0.00001	0.00001
Weekly returns Ripple	0.00001	0.00001
Monthly returns Ripple	0.00001	0.00001
Quarterly returns Ripple	0.00001	0.00001
Yearly returns Ripple	0.00002	0.16900
Daily returns Litecoin	0.00001	0.00001
Weekly returns Litecoin	0.00001	0.00001
Monthly returns Litecoin	0.00001	0.00001
Quarterly returns Litecoin	0.00001	0.00001
Yearly returns Litecoin	0.00009	0.17430

Visualizing the ACF and PACF for Bitcoin and Google Bitcoin Trends returns, we can not observe any significant auto-correlations in both series (Figure 3). These results are in line with financial literature, where it is a common characteristic the lack of auto-correlation, and widely supported by "The Efficient Market Hypothesis". In a competitive market where participants use all available information, market prices should be very close to the intrinsic value of the company, leaving very few opportunities to arbitrage [5].

Computing ACF and PACF for the other top three cryptocurrencies, we see that Ripple and Litecoin show no significant autocorrelation, while Ethereum have some positive autocorrelations around lag 3 (Figure 4).

2.5 Volatility clustering

Volatility clustering refers to the observation, first noted by Mandelbrot [12], that "*large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes.*" This is tested computing the ACF and PACFs of the squared of returns. It is generally expect that log returns to be serially uncorrelated, but the squared log returns to show significant autocorrelations. This is the case for the four cryptocurrencies but not for Google

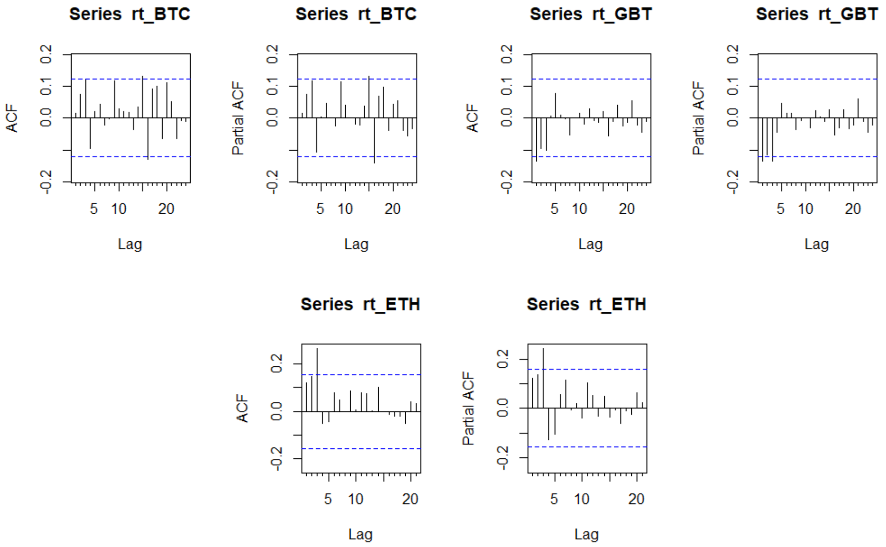

Fig. 3. ACF and PACF of Bitcoin, GBTrends and Ethereum returns

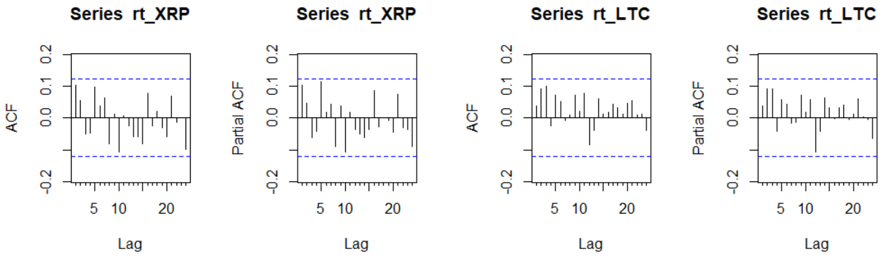

Fig. 4. ACF and PACF of Ripple and Litecoin returns

Bitcoin Trends: while Bitcoin, Ethereum, Ripple and Litecoin squared log returns show significant autocorrelations, GBTrends squared returns does not.

2.6 Causality

When analyzing relationships between time series, correlation only captures the linear dependency between two variables, but one would like to know in which direction the information flows from one series to the other. In other words, if Bitcoin prices and GBTrends are correlated we want to know which one causes the move of the other. In this context potential outcomes could be:

a) The interest in Bitcoin around the world and captured by Google Trends is actually a good proxy for measuring the social interest on the cryptocurrency triggering a Bitcoin price movement.
b) Big swings on Bitcoin prices create media and social attention triggering an increase in the interest of people around Bitcoin.
c) A combination of both events, were both series move at the same time and direction, up or down.
d) There is no relation at all between them and none of them influences the other.

We measure causality using Granger causality test [7]. The basic idea of Granger causality is that X *causes* Y*, if* Y *can be better predicted using the histories of both* X *and* Y *than it can by using the history of* Y *alone.* Formally one consider a bivariate linear autoregressive model on X and Y, making Y dependent on the history of X and Y, together with a linear autoregressive model on Y, and then test for the null hypothesis of "X does not causes Y", which amounts to test that all coefficients accompanying the lagged observations of X in the bivariate linear autoregressive model are zero. Then, we can evaluate the null hypothesis through an F-test. To perform this test on our time series we have used the R package `MSBVAR` which provides methods for estimating frequentist and Bayesian Vector Autoregression (VAR) models and other tools such as the Bivariate Granger Causality Test.

We perform this test at four different epochs marked by full calendar years 2015, 2016, 2017, 2018, and sampling weekly returns. Lag lengths considered to compute this test were the first 4 lags, to cover any autocorrelation through the month. We run the causality test for GBTrends, Bitcoin, Ripple and Litecoin returns (Ethereum has not been included in this analysis because of the lack of historical data, since it started to trade in late 2015, and low volume of trade).

Table 3 shows the results for GBTrends and Bitcoin for 2017. We see significant causality from GBTrends to Bitcoin in all four lags. This is also the case for the year 2015 (although only for lag 1), but not for 2016 and 2018 where rather contemporaneous correlation (causality in both directions) has been shown at all lags. A quick explanation of these results is that as the hype for Bitcoin was building up, both GBTrends and Bitcoin were moving each other, but as the Bitcoin prices started to rise exponentially news started to anticipate the next jump until the peak in 2017-2018. At that point the curiosity on the topic was globally widespread, leading to an increase in searches in Google on this cryptocurrency due to its incredibly high price.

Table 3. Granger Tests results for 2017

Lag length used	**1 lag**	**2 lags**	**3 lags**	**4 lags**
Causal relations	*p*-val	*p*-val	*p*-val	*p*-val
GBTrends to BTC	0.2740	0.1156	0.1052	0.1985
BTC to GBTrends	0.0030	0.0130	0.0168	0.0342

Table 4 shows causality results for GBTrends, Ripple and Litecoin for year 2017. Here we observe causality in both directions (GBTrends to and from the cryptocurrencies). This situation is more or less the same on the other epochs considered. A plausible explanation could be that these two cryptocurrencies do not induce massive searches, and are somewhat surrogates for Bitcoin.

Table 4. Granger Tests results for 2017

Lag length used	**1 lag**	**2 lags**	**3 lags**	**4 lags**
Causal relations	*p*-val	*p*-val	*p*-val	*p*-val
GBTrends to XRP prices	0.1687	0.3184	0.2668	0.3486
GBTrends to LTC prices	0.9368	0.7929	0.8024	0.9205
BTC prices to XRP prices	0.3262	0.6521	0.6598	0.3956
BTC prices to LTC prices	0.3862	0.6928	0.8505	0.8956
XRP prices to GBTrends	0.8432	0.7538	0.8005	0.8803
LTC prices to GBTrends	0.3827	0.7749	0.2705	0.4101

2.7 Neglected nonlinearity

A multivariate test of nonlinearity to ascertain if two time series are nonlinearly related can be achieved with the neural network test for neglected nonlinearity developed by White [14]. The basic idea is to perform a test of the hypothesis that a given neural network defines a perfect mapping between its input and output and that all the errors are due to randomness. For our experiments we use the Teräsvirta linearity test, presented in [16] and based on Whites neural network test for neglected nonlinearity. An implementation of this algorithm is available in the `tseries` R library.

Again we only have space to show results for 2017 in Table 5. The null is the hypotheses of linearity in mean. Then, results from Teraesvirta test indicate existence of a non-linear map from GBTrends to Bitcoin. The presence of a non-linear relation justifies the use of non-linear models such as neural networks that profit from all those existing relations between time series.

Table 5. Terasvirta Neural Network Tests results

2017	**X-squared**	***p*-value**
Bitcoin to GBTrends	5.8703	0.0531
GBTrends to Bitcoin	2.8758	0.2374
Bitcoin to Ethereum	0.002	0.9989
Bitcoin to Ripple	2.111	0.3479
Bitcoin to Litecoin	0.9126	0.6336
GBTrends to Ethereum	0.7732	0.6794
GBTrends to Ripple	0.1757	0.9159
GBTrends to Litecoin	3.0942	0.2129

3 Modelling

To sum up, we have seen the stylized facts for Bitcoin prices (and other three cryptocurrencies) and Google Bitcoin Trends. As expected, both of them share most characteristics seen in stock market literature:

- Both time series are non stationary, so further statistical analysis are performed on their returns time series. As well as Bitcoin, Ethereum, Ripple and Litecoin are not stationary.
- Bitcoin returns series converge to normality when increasing the sampling period. The other three major cryptocurrencies Ethereum, Ripple and Litecoin do not converge to normality.
- Bitcoin prices and GBTrends returns time series do not experience autocorrelation with their first few lags.
- GBTrends has a significant causal effect on Bitcoin price changes at certain epochs (2015 and more strongly at 2017), so there could be some value in GBTrends as predictor. Other epochs, and for most of the cryptocurrencies we observe contemporaneous correlations, among themselves and with GBTrends.

With this information, our approach in this section is to test predictability on top of some models. To do so we will be using a control model, only using past information from Bitcoin returns and comparing errors against models using GBTrends series as external variables to predict. We used Bitcoin price changes on a weekly timescale. This is forced by the fact that we could only get Google Trends data on a weekly time scale.

3.1 ARIMA back-testing

First, we start using an ARIMA model to predict Bitcoin prices only using past observations, and an additive model including GBTrends data as external variable. ARIMA models are the most general class of models for forecasting time series, so it is going to be good as a base model for later iterations. To fit the ARIMA models we have used the auto.arima() function from the R library *forecast* developed by Hyndman [9]. This function returns the best ARIMA model according to either AIC, AICc or BIC value. The function conducts a search over possible model within the order constraints provided. To proceed with the model comparison, we fit a base model only using Bitcoin returns and later a second one adding GBTrends as an external variable. We fitted the data year by year from 2015 to 2018 on different models, and averaged the error measures.

Table 6. arima model comparison

models	**AIC**	**RMSE**	**MAE**
ARIMA	-90.23	592.31	343.41
ARIMA + GBTrends	688.22	525.61	342.58

Using ARIMA models, we see a decrease in errors when adding GBTrends as external regresor, although the model fit is not as good according to AIC.

3.2 Neural Networks

We modelled with feed-forward neural networks to profit from non linear relations we observed in the neglected nonlinearity section. As we are dealing with time series data, lagged values of them can be used as inputs to a neural network. This particular version of models are called neural network autoregression or NNAR models. We used the function nnetar() from the R package *forecast* which fits feed-forward neural networks with a single hidden layer and lagged inputs as well as external variables. In our case, we choose lags 1 to $p = 5$ in order to consider the first 5 lags. Notation on this models is NNAR(p, k) which declares a model that uses up to the last p observations and k neurons in the hidden layer. We fixed the size of the hidden layer to $k = 4$. We note that although the function NNAR allows for tuning the p and k parameters, we do not use that feature as we are interested in fixing a base neural network model and test if that model improves forecasting with the addition of GBTrends. Hence, as in previous experiment, we repeat the procedure fitting a base model only using Bitcoin returns and later a second one adding GBTrends as an external variable. We fit the data year by year on different models, as we did in the stylized facts, and finally we averaged the error measures:

One can observe that using neural networks we get considerably lower errors than modelling with ARIMA. One can also see that when introducing GBTrends

Table 7. neural networks model comparison

NNAR(p,k)	RMSE	MAE
NNAR(5,4)	392.7452	210.73
NNAR(5,4) + GBTrends	285.38	151.78

to predict Bitcoin prices, errors decrease even more. This is in line with the results obtained in Teraesvirta test of existence of a non-linear map from GBTrends to Bitcoin. Hence the NNAR model enhanced with GBTrends shows better forecasting power for Bitcoin than NNAR alone and ARIMA based models.

4 Conclusions

In the past years, Bitcoin and the world of cryptocurrencies have gained some level of establishment. Some people see them as the new world currency far from the intervention of governments and central banks. For others, cryptocurrencies are considered as a new type of commodity like gold which can offer some protection against inflationary cycles. And for the rest, just another creation from the digital age we are living right now, as many other trends we have seen through the past in many other fields. The main motivation for this work was to explore the idea of using Google Trends data to forecast Bitcoin prices, under the hypothesis that Google Trends could be a good proxy of the interest of people around a specific topic in internet. In order to properly face this question we built a proper study around the main statistical issues to be covered in order to build a solid basis prior to any modeling and give more consistency to our results.

From our analysis of stylized empirical facts, the most relevant fact we found is the causal relation from GBTrends to Bitcoin in some years, which signaled the potential for using this source of data to improve accuracy when forecasting Bitcoin prices. After a careful consideration of relations found in the study of linear and non linear dependencies, we fitted a linear model (ARIMA) and a non linear model (Neural Network) trying to profit from the initial idea of using GBTrends as an alternative source of data from a social sentiment point of view. Doing so, we found a significant reduction in the out-of-sample error in the neural network model when introducing this new variable. There is no much improvement in the case of forecasting with autoregressive linear models. All in all we found evidence to believe that Google Bitcoin Trends could serve as predictor of Bitcoin prices, although not consistently through time.

From an economic perspective, these results could be explained following the supply and demand model of price determination in a competitive market. If supply is relatively fixed in the long-term then demand should be the largest single contributor to Bitcoin prices, given there are no actual quarterly earnings or interest rates associated with Bitcoin. In this case demand, or at least the interest from the demand side, can be observed from the search interest in the biggest search engine of internet, Google.

Acknowledgments. A. Arratia acknowledge support by grant TIN2017-89244-R from MINECO (Ministerio de Economía, Industria y Competitividad) and the recognition 2017SGR-856 (MACDA) from AGAUR (Generalitat de Catalunya)

References

1. Arias, M., Arratia, A. & Xuriguera, R. (2013). Forecasting with Twitter Data. ACM Transactions on Intelligent Systems and Technology, (TIST) 5 (1).
2. Chan, S., Jeffrey, C., Nadarajah, S., & Osterrieder,J. (2017) *A Statistical Analysis of Cryptocurrencies.* Journal of Risk and Financial Management, 10(2), 12.
3. Ciulla, F., Mocanu, D., Baronchelli, A., Gonalves, B., Perra, N., & Vespignani, A. (2012). Beating the news using social media: the case study of American Idol. EPJ Data Science, 1(1), 8.
4. Cont, R. (2000) *Empirical properties of asset returns: stylized facts and statistical issues.* Quantitative Finance.
5. Fama, E. (1965). The behavior of stock market prices. The Journal of Business.
6. Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. Nature, 457(7232), 1012.
7. Granger, C.W.J. (1969). Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. Econometrica 37:424-438.
8. Grcar, M., Cherepnalkoski, D., Mozetic, I., & Novak, P. K. (2017). Stance and influence of Twitter users regarding the Brexit referendum. Computational social networks, 4(1), 6.
9. Hyndman, R.J. & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. Journal of Statistical Software, 26(3).
10. Kendall, M. (1953). The Analysis of Economic Time Series, Part I: Prices. Journal of the Royal Statistical Society, 96.
11. Lazer,D., Kennedy, R., King, G. & Vespignani, A. (2014) *The Parable of Google Flu: Traps in Big Data Analysis.* Science (vol 343).
12. Mandelbrot, B. (1963). The Variation of Certain Speculative Prices. The Journal of Business.
13. Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.*
14. Lee, T.H., White, H., Granger, C.W.J. (1993) Testing for neglected nonlinearity in time series models, Journal of Econometrics 56, 269–290.
15. Nason, G. (2013). A test for secondorder stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series. J. R. Stat. Soc. B, 75: 879-904.
16. Teraesvirta, T., Lin, C.F., Granger, C.W.J. (1993) Power of the Neural Network Linearity Test, Journal of Time Series Analysis 14, 209–220.