Ca' Foscari University of Venice

Master Degree

in

Data Analytics for Business and Society

Final Thesis

Score Driven Model for Realized Volatility of Crypto Asset

Innovative strategies for Modelling Cryptocurrencies Market Dynamics

Supervisor
Dr. Prof. Dario Palumbo

Graduand
Alberto Bersan
879597

Academic Year
2024 / 2025

Index

Introduction

Finance is one the most thrilling fields of economics in terms of econometrics data analysis and digital tools application. This is one of the reasons why I decided to develop my Master Degree thesis in Data Analytics for Business and Society from the econometrics basis provided by the course Predictive Business and Finance.

This thesis aims to reduce the level of complexity required to analyse Realized Volatility by developing a sophisticated tool which can apply Dynamic Conditional Score Models (DCS Models).

Realized volatility represents a fundamental measure of the price variation of a financial asset over time. This concept is important for past market dynamics and crucial for making future forecasts and informed decisions.

In recent years cryptocurrencies have amplified the importance of measuring and managing volatility because of their decentralized nature and continuing technological innovations. Cryptocurrencies exhibit significantly higher volatility compared to traditional financial assets. Applying classic advanced analytical tools to realized volatility with cryptocurrencies' market is not enough considering the complexity of risk management and portfolio optimization of the sector.

This work is based on the theory of Harvey and Palumbo on Score-Driven Models for Realized Volatility.

This theoretical elaborate is followed by the practical application by computing R-code functions which allows to create and use score-driven models for realized volatility

The thesis is structured as follows:

Chapter I: Realized Volatility

The first chapter, Realized Volatility, introduces the concept of realized volatility and its significance in financial markets. It is structured into several sections. The Data Description section explores the complexity of financial data and its key characteristics. Market Microstructure examines the ways in which market microstructure impacts volatility. Lastly, Autocorrelation Analysis investigates autocorrelation in financial data, which is essential for understanding volatility dynamics.

Chapter II: Modelling Realized Volatility - Score Driven Models

The second chapter, Modelling Realized Volatility - Score Driven Models, explores the use of score-driven models for volatility modeling. It begins with the Statistical Framework, which introduces the statistical foundation underlying these models. The chapter then discusses Location/Scale Distributions (GB2), focusing on location and scale distributions with particular attention to the GB2 distribution. Following this, the EGB2 Distribution section examines the properties of the EGB2 distribution. The chapter also addresses Long Memory, analyzing the persistence of volatility patterns over time. Finally, the Leverage section explores the leverage effect in financial markets and its implications for volatility dynamics.

Chapter III: R Application - Score Driven Model for Crypto Realized Volatility

The third chapter, R Application - Score Driven Model for Crypto Realized Volatility, focuses on the development of R functions for applying score-driven models to realized volatility. This tool provides a practical platform for implementing and testing the models discussed in the previous chapters. The chapter begins with an overview of the Code Structure and Functioning, offering a brief explanation of its design and operation. It then presents a Case Study on the application of score-driven models to cryptocurrency markets, demonstrating their effectiveness in capturing volatility dynamics.

I. Realized Volatility

Volatility in asset returns is a fundamental part in studying numerous critical financial challenges, including asset allocation, risk management, and the pricing of options. Specifically, in risk assessment, the increasing reliance on the Value-at-Risk approach underscores the growing importance of accurately measuring and forecasting short-term volatility. Volatility essentially quantifies the extent of fluctuation experienced by a random process.

The concept of volatility remains elusive, lacking a single universally accepted definition, though multiple interpretations exist (see Ch. 1 Measuring and Modelling Realized Volatility: from tick by tick to long memory, Fulvio Corsi, 2005).

Actual Historical Volatility is measured over a specified period but with the last observation on a date in the past. A near synonym for this is realized volatility, which will be explored in detail in the first chapter of this thesis. Actual Future Volatility represents volatility over a specific period starting at the present moment and ending at a future date. It can be further classified into different types. Historical Implied Volatility refers to the implied volatility observed from historical prices of the financial instrument. Current Implied Volatility is derived from current prices of the financial instrument. Lastly, Future Implied Volatility is based on implied volatility observed from future prices of the financial instrument.

To sum up Realized Volatility is a model free measure of historical volatility, in other terms is the measure of the actual movement in the price of a financial asset, over a specified period.

From an econometrics perspective, volatility is defined as the standard deviation of a sequence of random variables, each of which is the return of the fund over some corresponding sequence of times. The generalized volatility σ for time horizion T in years is expressed as:

$$\sigma_T = \sigma_{annually}\sqrt{T}$$

the annualized volatility is:

$$\sigma_{annually} = \sigma_{daily}\sqrt{P}$$

Where σ_{daily} is the daily logarithmic returns of a stock and P is the time period of returns in trading days.

So,

$$\sigma_T = \sigma_{daily}\sqrt{PT}$$

In practice, the methods for empirically measuring volatility are broadly categorized into two main approaches: estimation of parametric models and direct nonparametric methods.

Historically, most research has gravitated toward parametric methods considering volatility as an unobservable variable within a fully specified model to predict ex-ante expected volatility. This consideration led to all ARCH-GARGH and stochastic volatility models. This kind of models suffer from a twofold weakness: first they are not able to replicate the empirical features of financial data; secondly, the estimation procedures required are often too complex.

Direct Nonparametric methods are certainly less applied and less mentioned by the literature.

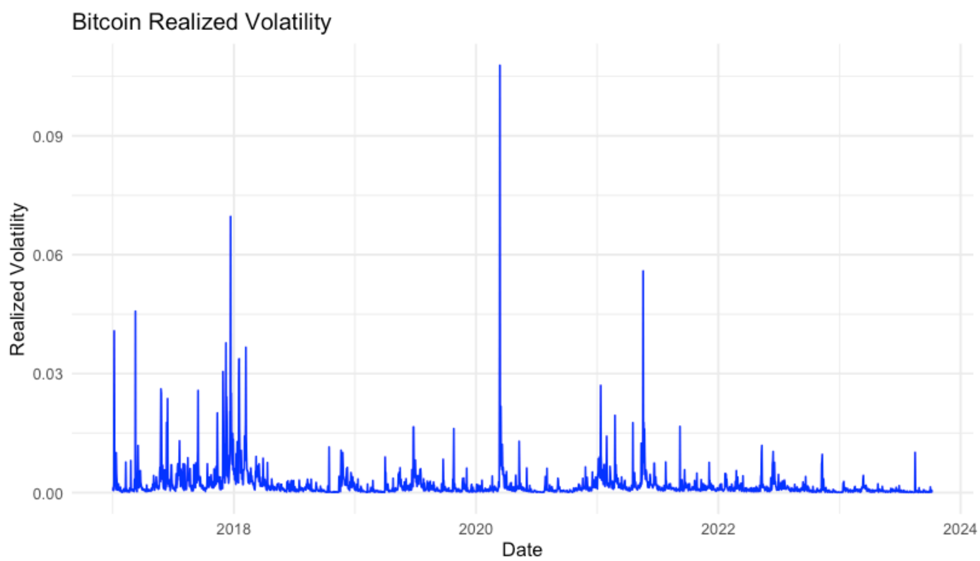
RV is calculated using past intra-day returns, capturing the extent to which the asset's price has fluctuated during that period.

The RV estimator can be computed as:

$$RV_t = \sum_{i=1}^{m} {r_{t_i}}^2$$

Where 't' is the daily interval, 'm' is the sub-interval (5-min, 1-min) and 'r' is the return defined as the difference between log prices

Fig. 1.1 Bitcoin Realized Volatility series



Source: R plot

As cited before, the concept of RV (Realized Volatility) is a daily measure of ex-post returns volatility based on high-frequency intra-day squared returns introduced by *Andersen et al. (2003).* The theory of quadratic variation suggests that, under suitable conditions, realized volatility is an unbiased and highly efficient estimator of return volatility, as discussed in *Andersen, Bollerslev, Diebold*, and *Labys (2001)* (henceforth ABDL) as well as in concurrent work by *Barndorff-Nielsen and Shephard (2002a, 2001).* Similarly to the instantaneous volatility measures, RV can be classified according to whether the estimation of the notional volatility only exploits returns observations falling in the internal [t – h, t] (*local*) or also incorporates return outside [t – h, t].

Daily volatility is classically measured with daily absolute return. However, *Andersen and Bollerslev's 1998* research highlights the significant noise associated with this measure, demonstrating some limitations about it. This is why there is the necessity of using, if possible, intraday data to derive more precise volatility estimates.

The concept of realized volatility is obtained from the sum of squared high-frequency returns within a specific time interval [t – h, t], which is equivalent to calculating the second uncentered sample moment of these returns.

This approach was enhanced by *Robert Merton's* work in 1980 where he demonstrated that the integrated variance of Brownian motion could be estimated with great accuracy by summing the squared returns recorded at higher frequencies throughout the day. He proposed that while these higher frequencies might not contribute significantly to the calculation of the mean, they are very important for determining the variance. In a random walk, the mean can be effectively captured just by the starting and ending points, whereas the volatility requires a full analysis of all the incremental steps.

An error-free estimation of volatility would allow to treat realized volatility as an observable, rather than a latent estimated quantity as with a GARCH(1,1) model. This opens the possibility to optimize models and forecast and to better extract the real underlying signal.

The assumptions which supports these theories may be fragile when using empirical data which are very far from theory. For example, a continuous record of prices is often unavailable and as the time scale decrease the assumption that log asset prices evolve as a diffusion process becomes less realistic. Using a very short time series causes the estimator of the daily volatility to be biased and inconsistent.

Because of the widely available daily data, the most used time horizon in literature is one day, agents and research are not interested in very short time intervals (1 minute, 5 minute horizon). Using these kinds of details is due to the improvement of the estimation of volatility, rather than being interested in risk existing at the extremely short time frames. It is important to remember that the bias of high-frequency data can be measure by the distance between the theoretical horizontal line and the empirical one. It is higher with lower financial liquidity specifically futures and individual stock show positive bias, stock indices show negative bias and FX present a strong positive bias

1.1 Characteristics of Crypto Financial Data

Crypto finance and volatility are among the most prolific subject of the econometrics literature, theories often deviate from the reality because of some intrinsic characteristics of the data.

1.1.1 Heavy Tails Distribution

From a statistical point of view, "heavy tails" refer to the phenomenon where the tails of a probability distribution are thicker than those of an exponential distribution. This implies that extreme values (far from the mean) have a higher probability of occurring. More precisely, distributions with fat tails have higher kurtosis and have a Probability Density Function (PDF) which decrease more slowly distancing the mean.

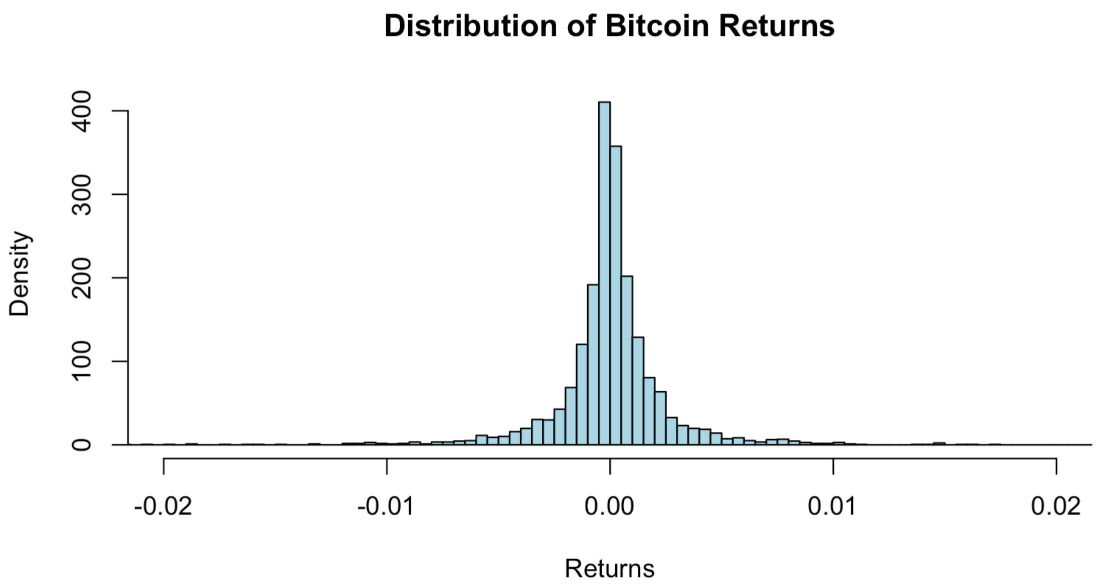
(1.1)
$$\lim_{x \to \infty} e^{\lambda x} \bar{F}(x) = \infty \qquad \forall \lambda > 0$$

Where $\bar{F}(x) = 1 - F(x)$ is the survival function (tail distribution) and F(x) is the cumulative distribution function.

This aspect is due to the fact that financial markets are susceptible to sudden and significant events like economic crises, political instability, or major corporate scandals and the human behaviour of investor which can deviate from rationality. These events can cause extreme price movements that are far more frequent than a normal distribution would predict.

It is important to remember that from a risk management perspective, fat tails imply that the standard deviation (a measure of risk) underestimates the probability of extreme losses. Therefore, models that assume normality might underestimate the risk of extreme events.

Fig. 1.2 Bitcoin Returns Distribution



Source: R plot

1.1.2 Non-Normality of the distribution

The distribution of Bitcoin (Fig. 1.2) suffers from high level of skewness and kurtosis which decrease with higher temporal aggregation. Monthly data presents less skewness, but it is very far from being normal. This can cause the underestimation of the Value at Risk because the measurement usually assumes normality. In addition to that, hypothesis tests are invalid because non-normality cause the wrong estimation of p-values and incorrect decision.

1.1.3 Long Memory

Autocorrelation of both squared and absolute returns will have a strong and persistent long-run behavior. Autocorrelation in both cases will have a strong and persistent long-run behavior and is simply referred to as "volatility clustering" with high-volatility period following high-volatility period and low-volatility period following low-volatility period.

The behaviour of memory is described with the autocorrelation function:

$$(1.2) \qquad \rho(k) = \frac{Cov(X_t, X_{t+k})}{\sqrt{Var(X_t)Var(X_{t+k})}}$$

Where k is lag and X_t the time series.
Long memory processes show a slower decay of correlations:

$$(1.3) \qquad \rho(k) \sim k^{-d}, \quad 0 < d < 0.5$$

Where d is the long-memory parameters

Unlike non-normality, such a property is not localized in a single observation interval but a strong property for realised volatilities at any aggregation interval, for example, hourly, daily, weekly, and even monthly observations.

The realization and consideration of such behavior is important in developing correct models and reliable forecasts, most particularly in settings that require an in-depth analysis of behavior in a marketplace and price trends over a duration, such as significant financial cryptocurrencies markets.

1.2 Market Microstructure

Maureen O'Hara, one of the most famous researchers in the market microstructure field, gives this definition "Market Microstructure is the study of the process and outcomes of exchanging assets under explicit trading rules. While much of economics abstracts from the mechanics of trading, microstructure literature analyses how specific trading mechanisms affect the price formation process". The standard theory regarding Market Microstructure implies some difficulties using ultra-high frequency data due to the unobservability of the efficient price. Considering the bid-ask spread as:

$$observed\ price = efficient\ price + noise$$

The noise composition depends on the frequency of the data. Considering moderate-frequency, bias can be filter out from microstructure noise using two scales estimator designed by *Zhang, Mykland* and *Ait-Sahalia*. However, they base their work on the assumption that microstructure noise is an i.i.d. process, which clearly does not meet real data.

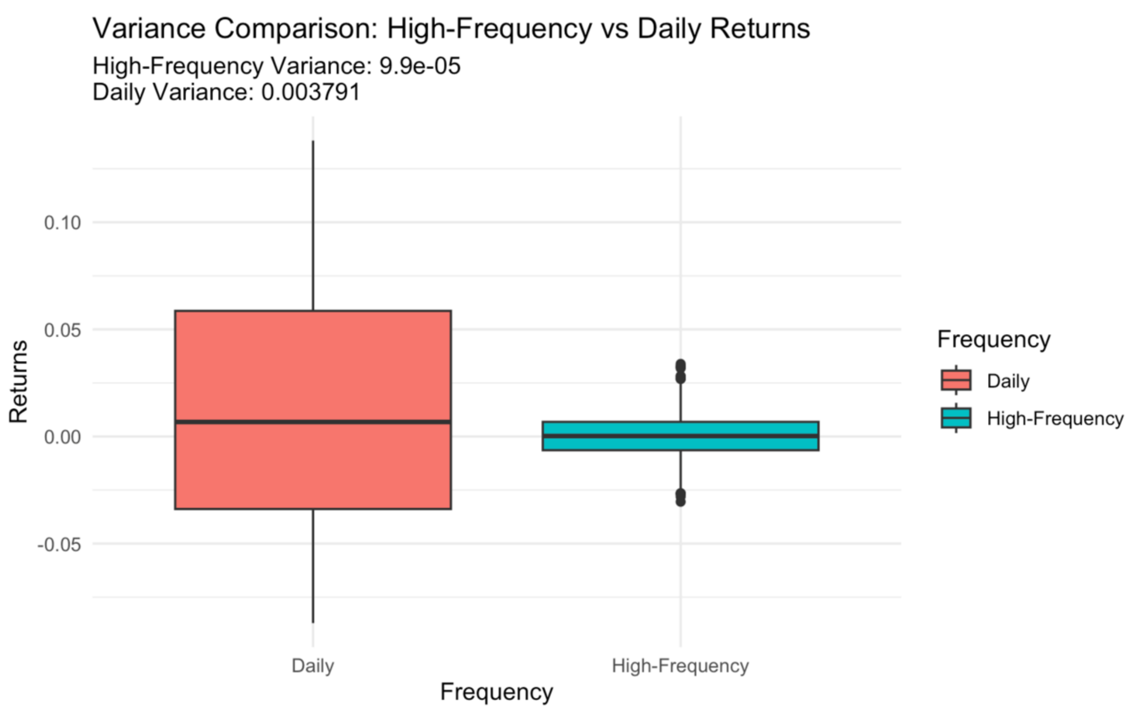
Let p_t be the observed price and u_t be the microstructure noise then:

$$p_t = m_t + u_t$$

Usually, u_t is assumed to be stationary because the observed price cannot deviate too much from the fundamental price.

This effect can be empirically verified by studying the unconditional expectation of the squared returns $E[r^2(\Delta)]$ as a function of the return's frequency (Δ). The unconditional expectation of the variance computed with returns taken at small time intervals is not equal to the volatility obtained with daily returns (higher frequency).

Fig 1.3 Returns Variance Comparison High vs Daily Frequency



Source: R plot

The variance estimator computed at a short time interval is strongly biased as compared to the mean squared daily return. These biases are generated by the market microstructure therefore tolerating this bias would led to distorted daily risk measures when using high frequency data.

The size of this bias is the distance between the theoretical line which shows the expected volatility of an i.i.d diffusion process and the empirical one.
In order to develop a microscopic model for the price, the real structure of the market must be taken into account as the return autocorrelation which will be analysed in the next chapter.

1.3 Autocorrelation analysis

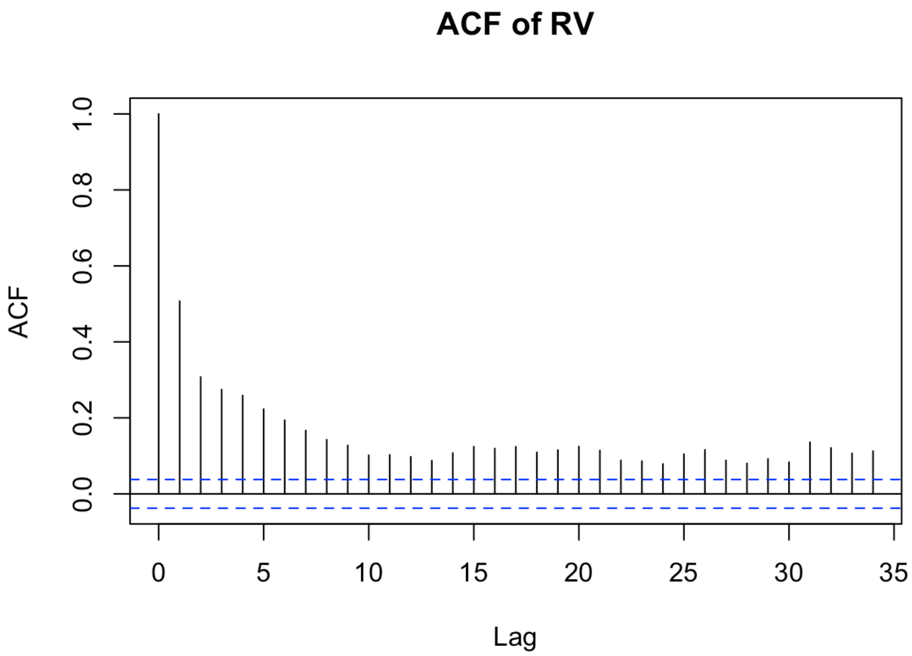
With $\rho(k)$ being the autocorrelation function of the return the second moment can only be explained by a non-zero autocorrelation when $\rho(k) \neq 0$ for some lags k. Then the variances computed with return at scale Δ_1 and Δ_2 (where Δ_1 is the return at high frequencies and Δ_2 the return at time scale m $\Delta_2 = m\,\Delta_1$)

$$\sigma^2(\Delta_2) = \sigma^2(\Delta_1)\frac{1}{m}\sum_{k=1}^{m-1}\sum_{l=i}^{m-1}\rho(k-l)$$

$$= \sigma^2(\Delta_1)[1 + 2\sum_{k=1}^{m-1}(1-\frac{k}{m})\,\rho(k)$$

$\rho > 0$ implies negative anomalous scaling (stock indices), $\rho < 0$ implies 8 positive anomalous scaling (FX, stocks and futures) and $\rho = 0$ implies no anomalous scaling.

Fig. 1.4 Autocorrelation plot of Bitcoin



Source: Yahoo Finance (R plot)

To conclude, a statistically significant negative correlation induces a positive sign bias (RV computed with high frequency returns is larger that the volatility of the true process), on the other side a long-lasting persistence of stock index returns increase the slope of the scaling function. Such biases increase with the sampling frequency.

The presence of a bias variance trade-off: bias is generated by microstructure effects which impose a reduction of the sample while the variance reduction suggest to use high frequency returns.

The choice of the sampling frequency is a key aspect when measuring and modelling RV, this choice is affected by the index considered and the frequency available. For the scope of this thesis, 5 min frequency is the best option because as suggested by Bollerslev et al. (2016), it "provides a simple way of mitigating the contaminating influences of market microstructure "noise" arising at higher intra-day sampling frequencies".

A possible solution is to choose the right interval at which the volatility it is not too much affected by the bias. Another solution is to correct the microstructure effects at the tick by tick level with some specific treatments. For example it is possible to apply an EMA filter or a discrete sine transformation, these methods are presented in "Measuring and Modelling Realized Volatility: from tick by tick to long memory" Fulvio Corsi (2005).

II. Modelling Realized Volatility – Score Driven Models

2.1 Score Driven Models

Modelling financial data can be quite complex and sometimes the estimation of the parameters can be wrong. One major issue is that one of the most used models are linear models, which are a staple in econometrics but don't capture the non-linear features of financial data. When studying economic phenomena, it's important to consider innovations, human behaviour, and both internal and external shocks; all of these external factors cannot be taken into consideration with linearity.

These factors add complexity when you try to model this kind of phenomena, in particular the difficulty comes from jumps: sudden, discontinuous changes in asset prices that deviate from the smooth diffusion processes. These require adjustments to accurately capture the true nature of volatility.

To face this challenge, this thesis proposes the use of models based on fat tailored distributions models because they better capture extreme events and high kurtosis. Traditional models, which rely on normal distributions, often underestimate the frequency and impact of large price volatility movements.

To better handle these complexities, economists often enhance linear models by allowing parameters to change over time rather than staying fixed. *Plackett* (1950) used recursive least squares methods involving matrix algebra to derive these time-varying parameters. Later, *Kalman* (1960), suggested that every parameter might follow a linear dynamic process, challenging once more the notion that parameters should remain constant.

Kalman's work, initially focused on control and system theory, had a profound impact on econometrics, especially through the contributions of *Andrew C. Harvey* (1970). *Harvey* demonstrated how the Kalman filter could be used for linear regression, ARMA, and other dynamic models, demonstrating that all linear dynamic models could be represented as state space models. These models consist of two equations: observation and updating the state vector with time-varying parameters. This representation allows for easy application of the Kalman filter, facilitating the estimation of these parameters and evaluation of the likelihood function through prediction error decomposition.

Complete accuracy in predicting time-varying parameters remains a challenging task even with historical and current data. These are the parameter-driven in state space models. Contrarily, in observation-driven models, time-varying parameters are regarded as functions of past dependent and exogenous variables that make them fully predictable from the history up to the present time. Instances of observation-driven models include GARCH models by *Engle* (1982) and *Bollerslev* (1986), *Nelson's* EGARCH model (1991) as well as *Engle* and *Russell's* ACD and ACI models (1998).

Score-driven models have been first presented by *Creal* et al. (2011, 2013) and *Harvey* (2013). In these models, conditional observational density score is used to update time-varying parameters. The score represents the first derivative of the log-likelihood function with respect to the parameter which supplies a natural updating mechanism for time varying parameters that takes advantage of complete predictive density structure. The score-driven approach is based on the modelling of parameters using all the available information without relying on the existence of higher moments.

According to *Blasques* et al. (2015), score-driven models can approximate the conditional observation density well despite imperfect specification of a model. In line with *Creal* et al.,(2013) and *Harvey* (2013), these were also observed in such models.

To summarise, when modelling realized volatility (RV), it is essential to choose a general distribution that can flexibly accommodate the observed properties of financial data, such as asymmetry, heavy tails, and heteroscedasticity. The Generalized Beta of the Second Kind (GB2) as a location/scale distribution and its exponential variant (EGB2) as a location and scale distribution are the perfect fit for this purpose. The GB2 distribution is highly flexible, capable of representing a wide range of shapes and capturing the skewness and heavy-tailed nature of financial data. Moreover, the EGB2 provides additional insights into the structure of RV, as it has the normal distribution as a limiting case, bridging the gap between traditional models and those better suited for capturing real-world phenomena.

2.2 Statistical Framework

The score driven model applied for Realized Volatility can be described by a multiplicative error structure:

$$y_t = \xi_t \mu_{t|t-1} \tag{2.1}$$

Where ξ_t is a IID random variables. In the multiplicative error model (MEM) this variable is a conditionally gamma distributed with conditional means $u_{t|t-1}$ which follows a GARCH-type process.

The GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model is a statistical approach used to estimate the volatility of financial time series data, such as stock prices or returns. Its primary purpose is to capture the time-varying volatility often observed in financial markets, where periods of high volatility are followed by high volatility and periods of low volatility follow low volatility. The model comprises two main components: the conditional mean equation and the conditional variance equation.

It is possible to define an equivalent model but in terms of scale $\alpha_{t|t-1}$:

$$y_t = \varepsilon_t \alpha_{t|t-1} \tag{2.2}$$

Where ε_t distributed with a location/scale distribution and α_t =exp($\lambda_{t|t-1}$)

Using the score driven approach dynamic volatility can be modelled giving dynamics directly to $\lambda_{t|t-1}$ with a first order DCS (dynamic conditional score) model:

$$\lambda_{t+1|t} = \omega(1-\phi) + \phi\lambda_{t|t-1} + \kappa u_t \tag{2.3}$$

Where μ_t is the score of the conditional distribution of y_t

This model is a DCS EGARCH with | ϕ | < 1 which ensures stationarity.

Taking the logarithms of y_t:

$$log\ y_t = x = \lambda_{t|t-1} + log\varepsilon_t \tag{2.4}$$

Where $log\varepsilon_t$ is distributed with a location scale distribution with 0 location and $\lambda_{t|t-1}$ is a location parameter.

2.3 Location / Scale Distributions (GB2)

In the previous section, the model in terms of scale $\alpha_{t|t-1}$ was described as:

(2.5) $$y_t = \varepsilon_t \alpha_{t|t-1}$$

ε_t is distributed as a GB2 conditional distribution.

The GB2 (Generalized Beta of the Second Kind) distribution is a flexible and versatile distribution used in various fields, including finance and economics, for modelling income distribution, wealth distribution, and other financial data. When used as a conditional distribution, it provides a way to model the conditional behaviour of a variable given certain conditions.

The usual form of the GB2 density is defined as:

(2.6) $$f(y) = \frac{v(\frac{y}{\alpha})^{v\xi-1}}{\alpha\beta(\xi,\varsigma)[\left(\frac{y}{\alpha}\right)^{v} + 1]^{\xi+\varsigma}} \quad \alpha, v, \xi, \varsigma > 0,$$

Parameter α is the scale. This parameter adjusts how spread out or compressed the distribution is. If α is larger, the distribution becomes wider, while a smaller α makes the distribution narrower. The value of α sets the units for the distribution, essentially scaling the entire distribution up or down.

v, ξ, ς are shape parameters, specifically: ξ – Reflects the decay rate for small values. ς - Reflects the decay rate for large values. v - Controls the overall shape of the distribution. And $\beta(\xi, \varsigma)$ is the beta function.

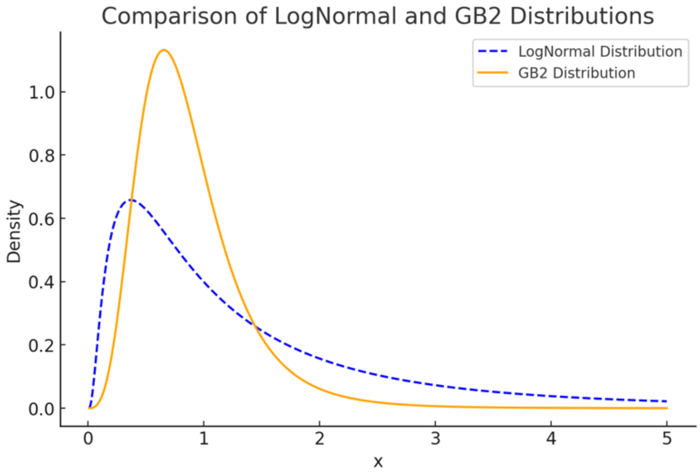
The score with respect to the logarithm of scale is:

(2.7) $$\frac{\partial logf_t}{\partial \lambda_{t|t-1}} = u_t = v(\xi + \varsigma)b_t(\xi, \varsigma) - v\xi$$

Where:

$$b_t(\xi, \varsigma) = \frac{(y_t e^{\lambda_{t|t-1}})^{v}}{(y_t e^{\lambda_{t|t-1}})^{v} + 1} \tag{2.8}$$

Fig 2.1 GB2 Distribution vs LogNormal



Source: Example random data (R plot)

It is possible to set these parameters to change the shape and the distribution of the GB2 family:

When $\xi = 1$ the distribution is a Burr (v, ζ) (Pareto IV)

$$f(x; \alpha, \nu, \zeta) = \frac{\nu\zeta}{\alpha}\left(\frac{x}{\alpha}\right)^{\nu-1}\left(1 + \left(\frac{x}{\alpha}\right)^{\nu}\right)^{-(\zeta+1)} \tag{2.9}$$

When $\xi = 1$, $\zeta \rightarrow \infty$ the distribution is a Weibull (v)

$$f(x; \alpha, \nu) = \frac{\nu}{\alpha}\left(\frac{x}{\alpha}\right)^{\nu-1} \exp\left(-\left(\frac{x}{\alpha}\right)^{\nu}\right) \tag{2.10}$$

When $\xi = 1$, $\zeta \rightarrow 1$ the distribution is a Log-logistic (v) (Pareto III)

$$f(x; \alpha, \nu) = \frac{\nu}{\alpha}\left(\frac{x}{\alpha}\right)^{\nu-1}\left(1 + \left(\frac{x}{\alpha}\right)^{\nu}\right)^{-2} \tag{2.11}$$

When $\xi = 1$ and $v = 1$ the distribution is a Lomax (ς)

$$f(x; \alpha, \varsigma) = \frac{\varsigma}{\alpha}\left(1 + \frac{x}{\alpha}\right)^{-(\varsigma+1)} \tag{2.12}$$

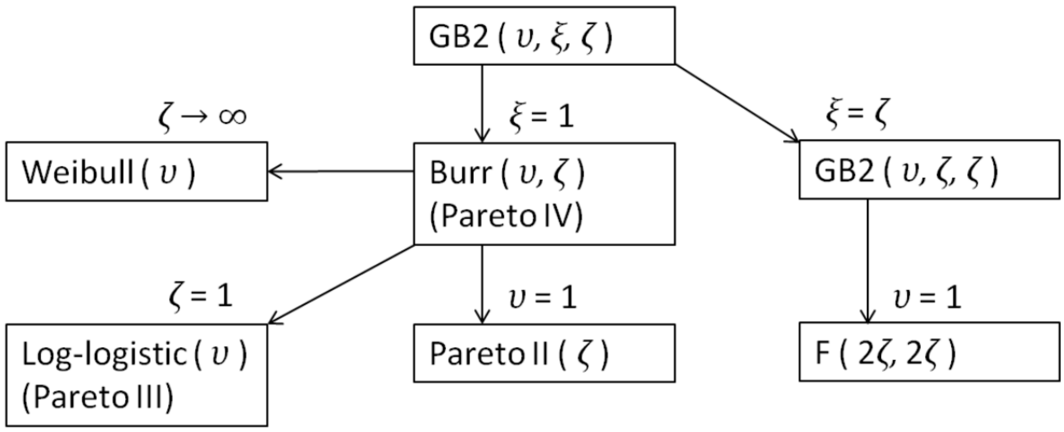
When $\xi = \zeta$ the distribution is a GB2 (v, ζ, ζ)

(2.13) $$f(x; \alpha, v, \zeta, \zeta) = \frac{v}{\alpha}\left(\frac{x}{\alpha}\right)^{v-1}\left(1 + \left(\frac{x}{\alpha}\right)^{v}\right)^{-2\zeta}$$

When $\xi = \zeta$ and $v = 1$ the distribution is a F $(2\zeta, 2\zeta)$

(2.14) $$f(x; 2\zeta, 2\zeta) = \frac{(2\zeta)^{\zeta} x^{\zeta-1}}{\beta(\zeta, \zeta)}(2\zeta + x)^{-2\zeta}$$

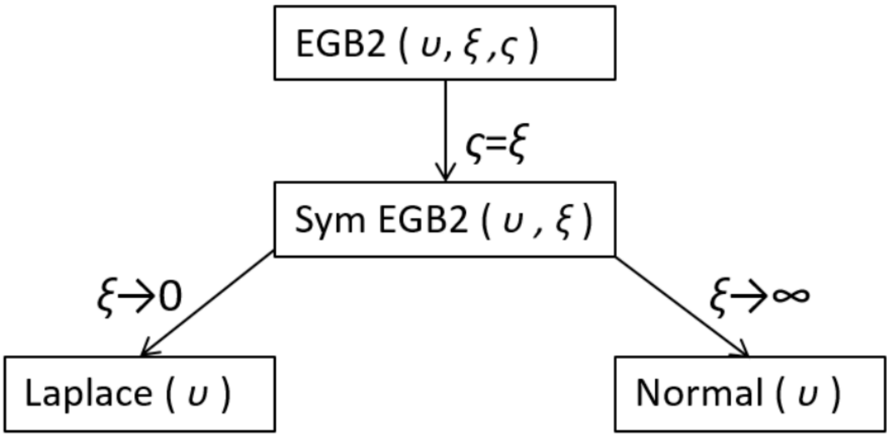
Fig. 2.2 Diagram of the distributions nested in the GB2 family



2.4 EGB2 Distribution

It is possible to create a similar structure of distributions taking the logarithm of the GB2 obtained the, so called, EGB2. When $\varsigma = \xi$ the distribution is a Symmetric EGB2 $(u, \xi\ )$ if $\xi \to 0$ the distribution is a Laplace (u), if $\xi \to \infty$ is a Normal (u).

Fig 2.3 Diagram of the distributions nested in the EGB2 family



The form of the EGB2 can be state as the logarithm of GB2. Considering x distributed as GB2$(\alpha, v, \xi, \varsigma)$ the density function:

(2.15) $$f(x; \lambda; v; \xi; \varsigma) = \frac{vexp\{\xi(x-\lambda)v\}}{B(\xi,\varsigma)(1+exp\{\xi(x-\lambda)v\})^{\xi+\varsigma)}}$$

The logarithm of scale in the GB2 $\lambda = \log{(\alpha)}$ now becomes a location parameter in the EGB2; v becomes a scale parameter and $\xi; \varsigma$ are still shape parameters which define skewness and kurtosis.

The mean is equal to

(2.16) $$\lambda + v^{-1}[\psi(\xi) - \psi(\varsigma)]$$

and the standard deviation is:

$$\sigma = \frac{h}{v}$$

where: $h = \sqrt{\psi'(\xi) + \psi'(\varsigma)]}$ and ψ; ψ' are the digamma and trigamma functions. The skewness of the distribution lies between -2 and 2 while the excess kurtosis ranges from 0 to 6 (see "Time series models with an EGB2 conditional distribution" Caivano – Harvey 2014)

$u_t$is defined as:

$$u_t = \sigma^2 \frac{\partial log f_t}{\partial \lambda} = \sigma h[(\xi + \varsigma) b_t(\xi, \varsigma) - \xi] \quad (2.17)$$

Where:

$$b_t(\xi, \varsigma) = \frac{e^{(x_t - \lambda_{t|t-1})h/\sigma}}{e^{(x_t - \lambda_{t|t-1})h/\sigma} + 1} \quad (2.18)$$

2.5 Long Memory

RV series show hyperbolic decay in their correlograms which it is one of the effects of long memory.

A long memory process can be defined as:

$$\Phi(L)(1-L)^d y_t = \Theta(L)u_t \qquad t = 1, \dots, T \tag{2.19}$$

Where L is the lag operator , u_t is a short memory, y_t is a fractionally integrated process to the order d, $I(d)$ where d represent the degree of 'long memory'.

For $|d| < \frac{1}{2}$ the process is stationary and invertible, for $\frac{1}{2} < |d| < 1$ the process does not have a finite variance. For $d \rightarrow \infty$ the process is given by:

$$y_t = \sum_{k=0}^{\infty} \psi_k u_{t-k}$$

Where: $E(u_t) = 0$; $E(U_t^2) = \sigma^2$; $E(u_t u_s) = 0. for\ s \neq t$

A long-term memory is a type of stochastic process in which observations exhibit strong relations with observations at a distance in terms of time. That property entails a decay in a specific observation's influence in future observations, in a less sharp manner in contrast with processes with short-term memory, in which relations decay with a sharp decay when lags become larger.

Long-run processes and long memory processes have to be understood and modelled in a manner that enables proper forecasts and effective management of risk. Traditional models like ARMA (AutoRegressive Moving Average) cannot model long-run dependencies and, therefore, have to use specific models like ARFIMA (p,d,q) (AutoRegressive Fractional Integrated Moving Average) that can better account for long-range correlations.

Another possible method to capture long memory is by HAR model for RV. The HAR model decomposes volatility into components representing different time periods, typically including daily, weekly, and monthly averages. By doing so, it captures the idea that market participants operate on various time scales, and that volatility is influenced by both short-term and long-term factors.

The basic HAR model can be expressed as:

$$(2.20) \qquad \bar{y}_{h,t+h} = u + \beta_d \bar{y}_t + \beta_w \bar{y}_{w,t} + \beta_m \bar{\bar{y}}_{m,t} + \epsilon_{h,t+h}$$

Where y_t is the RV series, $\bar{y}_{h,t} = (\sum_{i=1}^{h} y_{t-i}/h)$ for h =w=5 and h=m=22 and $\bar{y}_{h,t+h}$ is the h-days cumulative average for h=1,2,… and $\epsilon_{h,t+h}$ is the disturbance term

One of the key advantages of the HAR model is its simplicity and intuitive structure, making it easy to implement and interpret. It provides a more accurate representation of volatility dynamics by acknowledging that market behaviour is influenced by past volatility at multiple time scales.

A two-component structural time series model can be used to approximate the features of a long memory process. Thus:

$$\lambda_{t|t-1} = \omega + \lambda_{1,t|t-1} + \lambda_{2,t|t-1},$$
$$\lambda_{i,t+1|t} = \phi_i \lambda_{i,t|t-1} + \kappa_i u_{i,t}$$

The score curve from the RV model is rotated and, when the distribution of (mean-adjusted) returns is symmetric, the expectation of the composite variable driving the dynamic equation is zero. Identifiability requires $\phi_1 \neq \phi_2$ together with $\kappa_i \neq 0$ or $\kappa_i^* \neq 0$ and $\kappa_2 \neq 0$ or $\kappa_2^* \neq 0$. An attraction of the two-component model is that the leverage effect can differ in the long and short run.

2.6 Leverage

The leverage effect is an age-old reality in financial markets, with heightened volatility in reaction to a price fall (a down shock) deeper than in reaction to a price rise (an up shock) of similar value. As a financial modeling reality, such asymmetry is important, for it reflects real-life behavior in response to unfortunate occurrences in financial markets. Loss and bad information generate heightened uncertainty and a heightened awareness of peril, in contrast with gain and positive information.

The leverage effect arises for several reasons.

The Debt-to-Equity Relation matters. When a company's price per share declines, its debt-to-equity ratio will rise because the value of its equity declines but not that of its debt. Consequently, higher leverage conveys higher financial risk, and financial volatility will escalate.

Market Reactions contribute to the leverage effect, too. Negative events, such as poor earnings releases and recessions, have a propensity to generate a larger emotional reaction in investors. That reaction creates increased trading volumes and uncertainty, both of which drive market volatility upward.

Another factor is Volatility Feedback. In some financial constructs, heightened volatility can drive down prices even lower, creating a feedback loop whose effect is amplified through leverage.

Classic approaches, such as GARCH, have long assumed that volatility will react symmetrically to both positive and negative shocks. That, in reality, can make forecasting volatility inexact, not having regard for asymmetry in the leverage effect. With a model having incorporated the leverage effect, a truer picture can then be painted of behaviour in the marketplace, and forecasts for risk and volatility can become even more effective. For RV, the leverage term is governed by, $sgn(-r_t)$ where (r_t) is the mean-adjusted returns, thus:

$$\text{(2.21)} \qquad \lambda_{i,t+1|t} = \phi_i \lambda_{i,t|t-1} + \kappa_i u_{i,t} + \kappa_i^* sgn(-r_t)(u_t + 1)$$

The leverage effect can be incorporated in such models such as in the Dynamic Conditional Score (DCS) model in terms of sign and level of returns. For example, a leverage term such as sgn($-r_t$), with r_t denoting the return, can be included in a volatility equation. With such a term included, its estimate will increase when returns become negative, and in a way, it will replicate the leverage effect.

In conclusion, the leverage effect characterizes asymmetry in price fluctuations and its relation, and its consideration in financial times series modelling is a must. By such asymmetry consideration, such models can make more reliable and real-life forecasts, and become even more effective in forecasting, analysis of risk, and planning for investments.

III. R code: RV Modelling function

The RVModelling function developed for the purpose of this thesis is a general model for estimating, modeling, and analyzing realized volatility (RV) with a high level of accuracy, robustness, and adaptability. Realized volatility, derived from high-frequency financial observations, is a significant financial econometric variable, and its estimation is a necessity for financial and commercial use in, for example, risk management, portfolio maximization, derivatives pricing, and policy evaluation. The function's structure utilizes state-of-the-art methodologies for dealing with real-life datasets' complexity in terms of modeling changing volatility, including non-linearity, asymmetry, and fat tails in distributions.

The function RVModelling is a strong tool for fitting a variety of volatility models derived from a range of statistical distributions, including GB2 and its specific forms, such as Burr, F, and Exponential Burr distributions. All these distributions have a high suitability for financial data with a view to explaining the extreme events, asymmetries, and fat tails characteristically exhibited in RV data. With its property of allowing key parameters to be tuned, the function can adapt to a variety of requirements for modeling, including simple forms of linearity through to complex forms of score-driven (DCS).

Also included for less sophisticated model scenarios are specific cases of these distributions, such as Log-Logistic and Weibull.

The model describes volatility in terms of a dynamic mechanism, with significant time-varying behavior such as trends, seasonality, and cycling behavior, and with leverage effects included in its form. With such factors included, the model creates a rich characterization of realised volatility (RV) dynamics, with enhanced interpretability and forecasting performance.

The use of a variety of ingredients allows for a rich unscrambling of RV behavior, offering information regarding individual factors contributing to altering volatility.

The function can have estimation algorithms for its parameters in many forms, including Nelder-Mead, BFGS, and DEoptim, and such algorithms have been designed with a view towards delivering sound estimation and convergence of its parameters, even in a high-dimensional and complex scenario.

The function compares a variety of algorithms in a view towards selecting an effective resolution for a considered model configuration and a considered dataset, in an attempt towards maximising performance and accurately representing underlaying dynamics of volatility.

The function can be customised with numerous settings, such as selecting an applicable distribution, inserting in terms that vary dynamically, and transformation (e.g., log-transformation) of the observations. With such a high degree of malleability, the function can adapt to a variety of datasets and with a variety of model objectives, and is a general-purpose tool for use in analysis of volatility.
The algorithm first processes and transforms the input data, with provisions for logarithmic transformations in case variance is not constant. It estimates initial values via techniques such as static GB2 estimation and dynamically refines them via score-driven techniques. Users can include dynamic terms such as seasonality (e.g., weeklies), cyclicality (e.g., business cycles), and leverage effects in an attempt to model all types of volatility behaviour.

For robust output, iterative optimising routines minimize log-likelihood with additional constraints and penalties. Any one of a range of algorithms can be used, and one with best performance is determined automatically. Full diagnostics can also be generated, including residuals, tests for autocorrelation, and model checking plots. In conclusion, the RVModelling function is a powerful and flexible tool for modelers and researchers interested in modeling and forecasting for realized volatility. With its ability to handle complex structures in the data, its accommodation for real-life factors such as asymmetry and fat tails, and its output in a form useful for taking actions, it is a valuable tool for financial and economic modeling. For academic work and real-world application, it is a function with high-value, readable output for a range of challenging datasets.

3.1 Code structure and functionin

The R script is designed to model the Realized Volatility (RV) using advanced statistical and econometric techniques. The script begins by loading a custom function (RV_Modelling.R) that contains the core model and supporting functions necessary for volatility modeling. It also loads libraries for handling files and dplyr for data manipulation.

The script imports Crypto market data from a CSV file (i.e. Bitcoin.csv) and converts the dates column to the appropriate Date format. The dataset is then filtered to include only the wanted period, selecting the dates and RV columns for analysis. Any missing (NA) values in the RV column are replaced with the column's mean, and the script checks for and reports any remaining missing values. The cleaned RV data is then converted into a data frame for further processing.

Parameters for different statistical distributions are initialized with:

- Par = 3, D = 1 for GB2
- Par = 3, D = 2 for EGB2
- Par = 2, D = 1 for GB2 with $xi = \frac{1}{vega}$
- Par = 2, D = 2 for Burr
- Par = 3, D = 1 for F with different density function not from GB2 family
- Par = 2, D = 4 for GB2 with xi = zeta
- Par = 2, D = 5 for exponential Burr
- Par = 2, D = 6 for exponential EGB2 with xi=zeta
- Par = 1, D = 1 for exponential Log Logistic
- Par = 1, D = 2 for exponential F with equal density function
- Par = 1, D = 3 for exponential Generalized Pareto
- Par = 1, D = 4 for exponential Weibul
- Par = 1, D = 5 for exponential Lognormal from GB2
- Par = 1, D = 6 for exponential Gaussian from EGB2

The script contains numerous model parameters to control such factors as cyclical terms, seasonality, smoothing with splines, and autoregressive terms and quantities of dynamic terms to estimate. All these parameters enable sophisticated behavior in the data to be captured.

The settings for risk management are calibrated, with provisions for analysis through Probability Integral Transform (PIT) incorporated in them. The script then calls for function RVModelling, with all initiated parameters and prepared information being handed over to it. The function then estimates model parameters and estimates model fitting for the crypto information, offering graphical observations.

Notably, the function accepts numerous parameters, driving the model development process. Some of them include a dataset (XX), an options for optimizations (opt_method), and model options for seasonality, for cycles, and for terms for volatility. There are preliminary printing statements reporting about a format of a data and about distribution type estimation.

Required packages such as Deoptim, nloptr, lmtest, and stats are checked and installed in case of a necessity. Log-transformation of the data, in case of a necessity, is conducted according to a prescription. Preliminary scale factors and frequencies for seasonal and for cyclical components are computed.

An objective function for estimating probability is built with a distribution of one's preference. There can be supported in the function several algorithms for optimizations, including Nelder-Mead, NLOPT (SBPLX), BFGS, and DEoptim. Optimum answer is determined in terms of minimum objective value (probability).

The model parameters are then calculated, and its starting log-likelihood value is printed out. Model-fitting values and real values are contrasted with statistics including Mean Squared Error (MSE) and Quasi-Likelihood (QLike). Autocorrelation tests (Ljung-Box tests) for model residuals are performed in an attempt to evaluate model fitness.

The model, when run, performs tests for Probability Integral Transform (PIT). Goodness of fit statistics, including AIC and BIC, are calculated in order to assess model complexity and fit.

The function generates plots, including plots of actual and predicted values, model-implied and actual densities, and autocorrelated plots of residuals, useful for model diagnostics and model performance and fit checking.

As discussed earlier, such function implementations use a range of techniques for optimizations and choose the most optimized one to use, namely:

3.1.1 Nelder-Mead method (Optim)

The Nelder-Mead method, also known as the downhill simplex method, is a derivative-free optimization algorithm used for minimizing a function of several variables. It is especially useful when the gradient of the function is not available.

1. Initial Simplex Construction:
    - The algorithm starts by creating a simplex geometric shape with n+1 vertices in an n-dimensional space.

2. Simplex Operations: the algorithm iteratively modifies the shape using these operations:
    - Reflection: Moves away from the worst point to explore better solutions.
    - Expansion: Moves further in the reflection direction if improvement is observed.
    - Contraction: Moves closer to the better points if no improvement is found.
    - Shrinkage: Reduces the simplex size if no other operations yield improvement.

3. Stopping Criteria:
    - Convergence is achieved when the simplex becomes sufficiently small or the function values do not change significantly.

The method works without needing derivatives and is effective for noisy, discontinuous, or non-smooth functions. However, it can be slow for high-dimensional problems and may get stuck in local minima without guarantees of finding a global optimum.

3.1.2 COBYLA (nloptr)

COBYLA (NLOPT_LN_COBYLA) is a no-gradient nonlinear constraint algorithm, designed specifically for use in search guidance with linear constraint approximations.

1. Linear Approximation:
    - The algorithm approximates nonlinear constraints with a series of successive linear approximations.

    - A trust region scheme will make such approximations correct in a small region
2. Simplex-Based search
    - It iteratively maximizes/minimizes the objective function under constraint
3. Penalty-Based Handling:
    - Under penalty, when constraints are not adhered to, an objective function is moved towards feasible regions.

The method is useful when gradients won't suffice and handles inequality constraints effectively. However, it is not efficient for highly nonlinear constraints and requires careful fine-tuning of the trust region.

3.1.3 BFGS (optim)

The BFGS algorithm is a quasi-Newton algorithm for optimization, in that it approximates Hessian matrices of second derivatives but doesn't explicitly calculate them.

1. Gradient Estimation:
    - Compared to Nelder-Mead, BFGS requires an objective function gradient.
    - It utilizes gradient evaluations in its construction of an approximation of the Hessian matrix.
2. Updating the Hessian Approximation
    - The Hessian is updated iteratively with rank-one updates (according to gradient differences).
3. Step Direction and Size
    - The algorithm then calculates a search direction with the Hessian approximation and takes a move in that direction with a proper step size.

The method offers quicker convergence compared to simplex-based algorithms, is optimal for smooth, differentiable functions, and can be expanded to tackle larger issues. However, it requires gradient computation and is sensitive to poor starting values.

3.2 Case study: Score Driven Models for Crypto Currencies

3.2.1. Bitcoin (USD)

Bitcoins, designed in 2009 from an individual, or a group of persons, by an unknown individual, namely, Satoshi Nakamoto, is regarded to be the first global electronic, decentralized money. Unlike traditional currency, Bitcoin operates over a peer-to-peer network, in which value can pass between persons directly, excluding entities such as governments and banks. It is a secure, transparent, and unalterable form of global transactions.

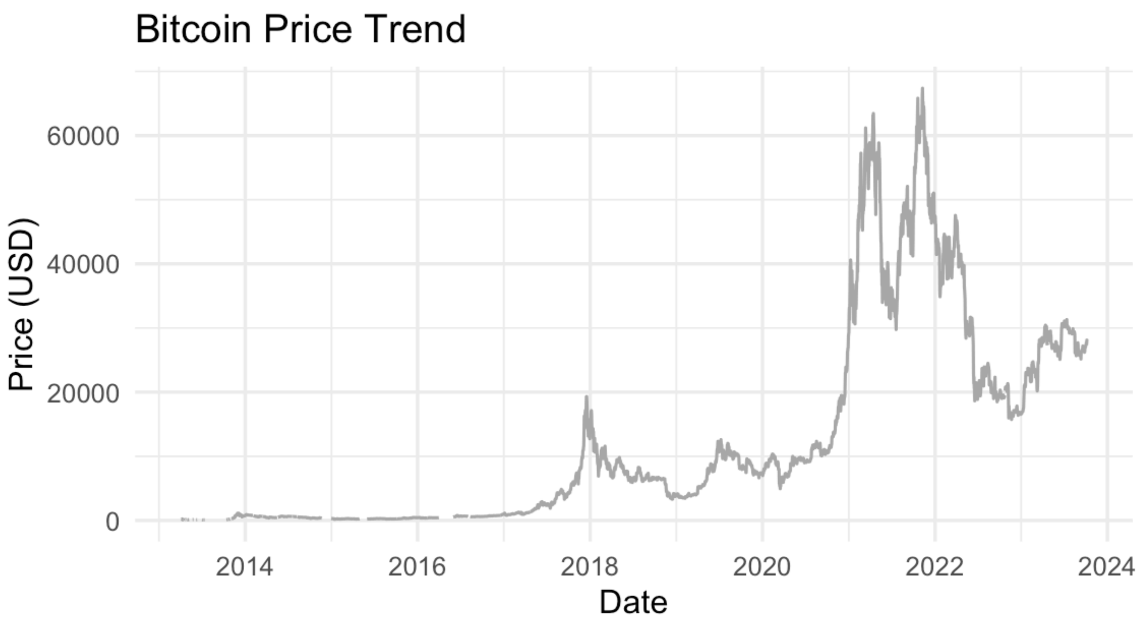
The backbone of Bitcoin is a technology, namely, a shared book record, a book in which all transactions are kept transparent and unchangeable. All transactions are kept in a collection in one single block in a book record, and when validated through a mechanism, mining, it is added irreversibly to a book record in a sequence. Mining involves employing computational powers in an attempt to solve complex algorithms in a form of a problem, and in return, miners receive new coins and a fee for each transaction in a form of new coins and fee for a transaction, respectively.

The total bitcoin supply stands at 21 million coins, and it is a deflationary asset simply for that fact alone. That scarcity is preserved through a programmed mechanism, "halving," that occurs approximately every four years. In both instances, miners' reward for validating transactions is reduced in half, effectively slowing new bitcoin issuance over time.

Where Bitcoin first emerged as a unit of exchange, over time, it grew into a store of value and an investment, and its lack of a controlling entity and its finite supply have invited comparisons with gold, and it is, therefore, sometimes "digital gold." Bitcoin price is determined by a variety of factors, including demand in the marketplace, general macroeconomics, technological advances, and political actions, but its extreme price volatility brings opportunity and peril for investors and speculators alike.

The analysis considers Bitcoin’s realised volatility (RV) over a daily basis, with observations between 2023 and 2017. That timeframe provides useful information about Bitcoin’s performance as a financial asset and its role in the overall cryptocurrency marketplace.

Fig. 3.1: BTC (USD) 2017-2023



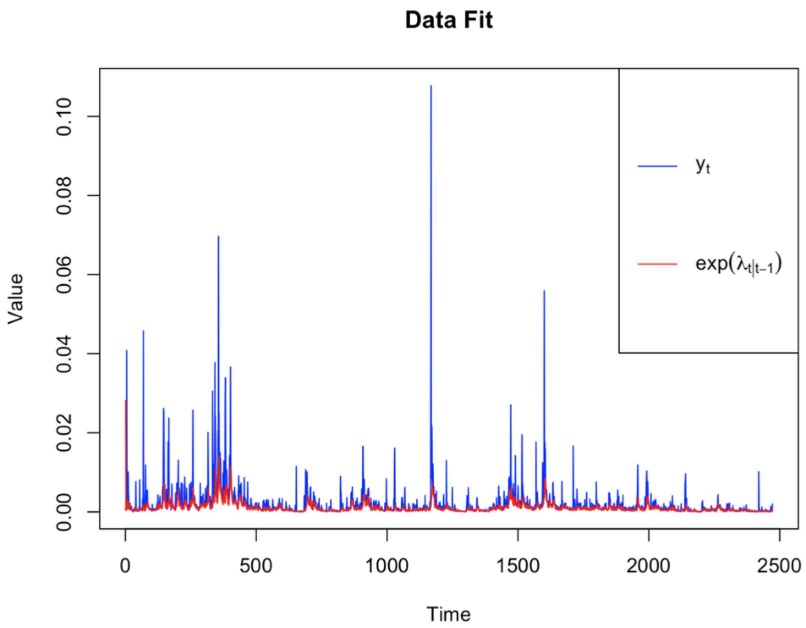
Source: R function output

Table 3.1: BTC code application

Distribution	ξ	υ	ζ	ω	ϕ_1	κ_1	ϕ_2	κ_2	*AIC*	*BIC*	*LOGL*
Exp Log Logistic	1	1.453	1	-3.549	0.975	0.004	-	-	-26642.38	- 26619.13	13325.19
GB2	1.388	2.172	0.943	-3.579	0.937	0.283	-	-	-28984.04	-28949.16	14498.02
GB2	2.221	0.845	1.646	-7.125	0.990	0.078	0.759	0.450	-28607.18	-28560.685	14311.59
EGB2	1.055	1	1075.1	-0.001	0.959	0.08	-	-	7324.39	7359.26	-3656.19
GB2 with xi=1/vega	0.1992	5.019	0.366	-3.3273	0.939	0.301	-	-	-28457.96	-28428.89	14233.98
GB2 with xi=1/vega	0.862	1.160	0.794	-7.547	0.990	0.415	1	0.4889	-26902.33	-26861.64	13458.16
Burr	1	2.654	0.725	-3.533	0.931	0.2713	-	-	-28982.99	-28953.93	14496.49
Burr	1	1.210	0.690	-7.599	0.999	0.029	1	0.01	-26414.37	-26373.68	13214.18
GB2 xi=vega	1.571	1.706	1.571	-3.507	0.931	0.271	-	-	-28947.42	-28918.35	14478.71
Exponential Burr	1	1	2.606	0.0001	1	0.074	1	1.1068	-943.15	-902.46	478.57
Exponential EGB2 xi=vega		5.128	0.22	0.0009	0.79	0.015	0	0	-5264.06	-5223.37	2639.03

The best distributions in terms of levels which is the scope of this first analysis are GB2 and Burr. The GB2 distribution appear to be slightly better, and we will continue the analysis with this one.

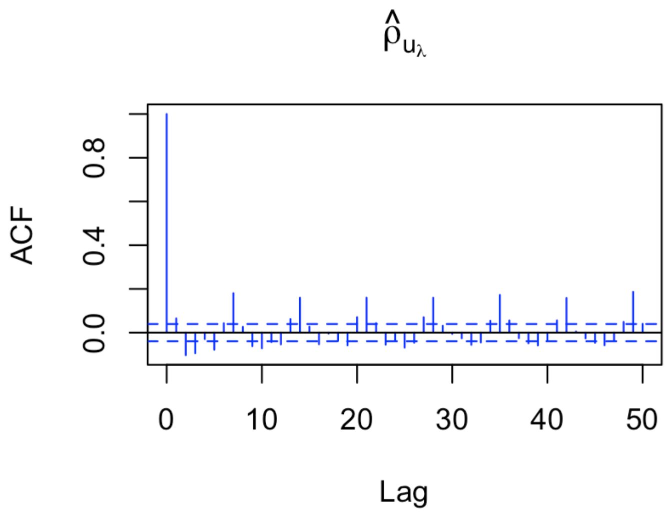
Fig. 3.2: BTC Data Fit



Source: R function output

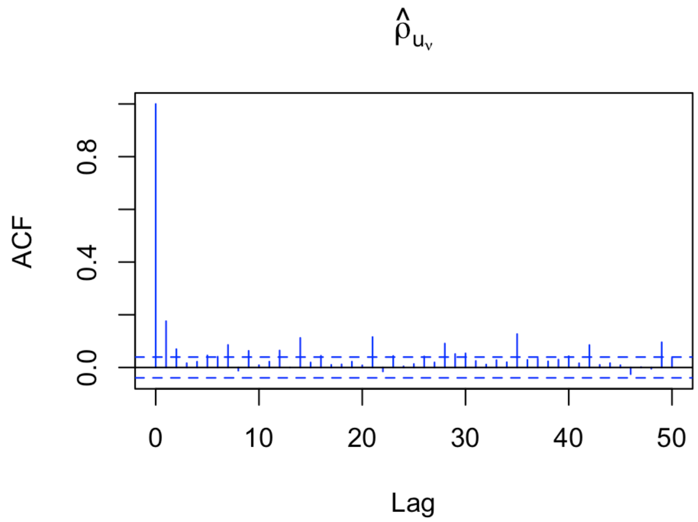
The Autocorrelation Function (ACF) plots illustrate the correlation of parameters λ and υ with their lagged values. Both plots show significant autocorrelation at lag 0, which is expected, but more importantly, the repeating pattern of autocorrelations at regular intervals suggests the presence of seasonality in the data.

Fig. 3.3: λ ACF



Source: R function output

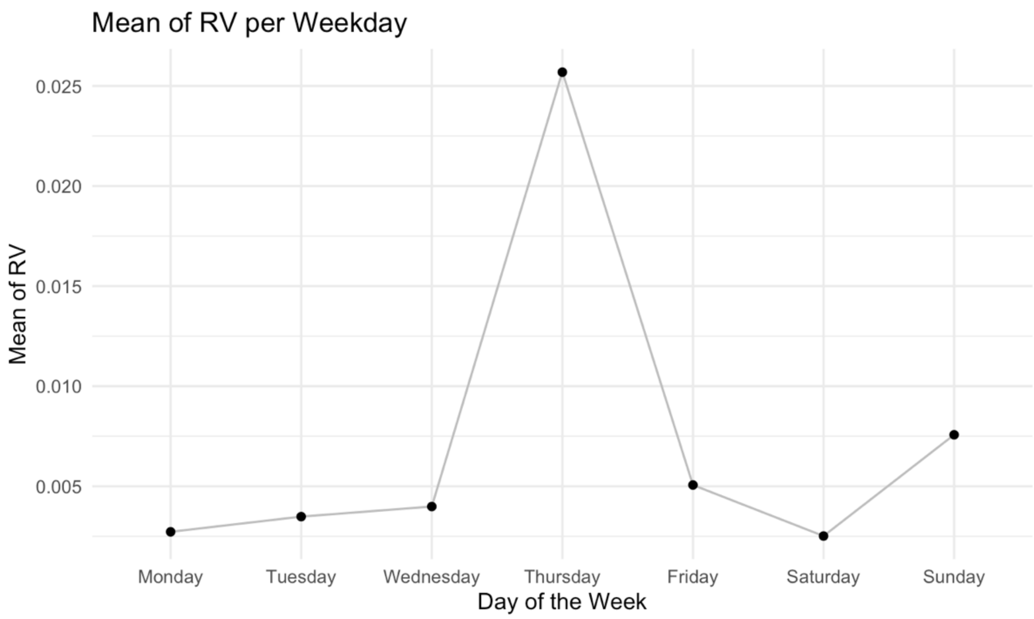
Fig. 3.4: υ ACF



Source: R function output

The presence of significant peaks at lag 7 in the ACF plots strongly indicates a seasonality of 7 periods. The next step will enhance the fit by adding seasonal components with a 7 period.

Fig. 3.5: Mean of RV per Weekday



Source: R function output

Fig. 3.6: ACF plot with seasonal component

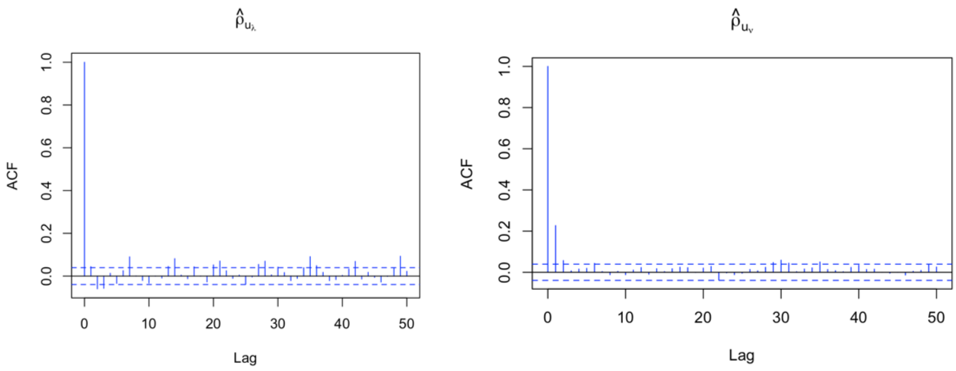


Fig. 3.7: Seasonal analysis

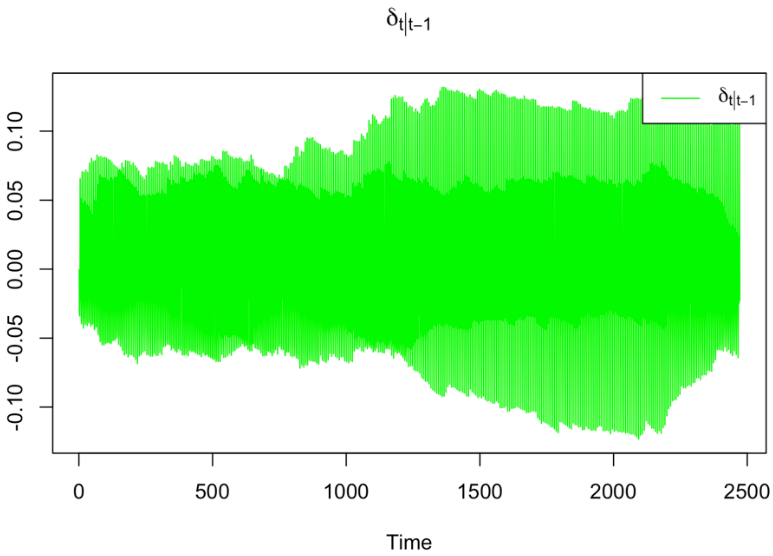


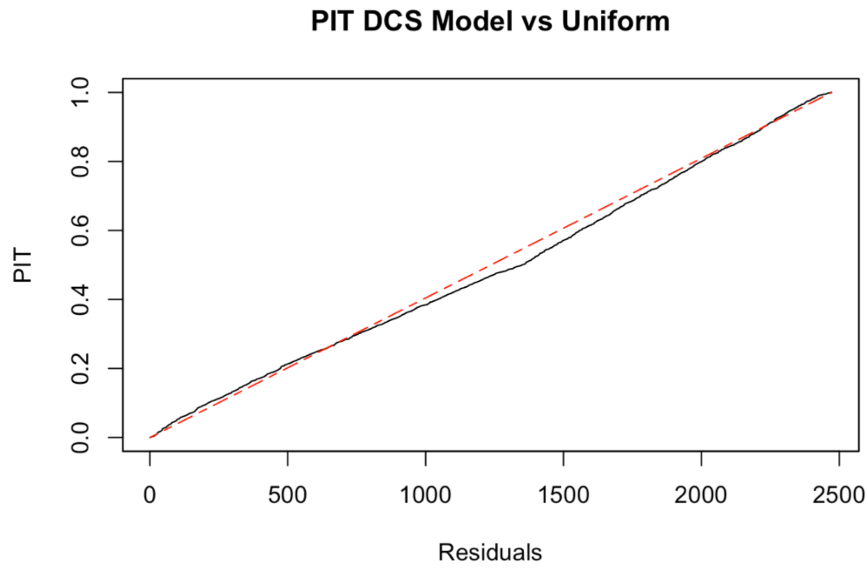
Table 3.2 Seasonal parameters

κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7
0.0039	-0.0329	0.0153	0.0234	0.03314	0.0149	0.0338

Source: R function output

To conclude, the code computes the PIT which is a technique used to assess the goodness-of-fit of a probabilistic model. It is based on the idea that if a model correctly captures the underlying distribution of observed data, then the transformed values should follow a uniform distribution on the interval [0,1].

Fig. 3.8: PIT Analysis



Source: R function output

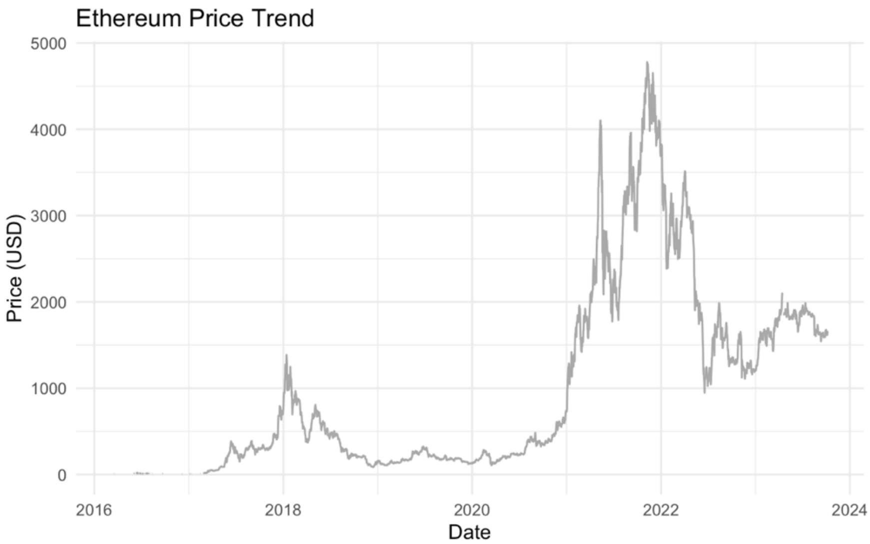
3.2.2. Ethereum (USD)

Ethereum is an open-source, decentralized platform for a blockchain that revolutionized the digital economy with smart contracts—programmatic, computerized contracts with terms coded directly into software. Established in 2015 with founder and head Vitalik Buterin, Ethereum not only birthed a cryptocurrency but a platform for developing decentralized programs (dApps) in a range of sectors, including finance, gaming, and supply chains.

Whereas Bitcoin is a virtual currency, Ethereum is a programmable blockchain. Ethereum’s native token, Ether (ETH), is both a store of value and a payment for computational and transaction assets in the network. Central to Ethereum’s operations is Ethereum’s virtual computation environment, Ethereum Virtual Machine (EVM), a virtual computation environment that runs smart contracts and hosts dApps. With such an arrangement, developers can develop complex programs with no intermediate servers, and innovation in decentralized technology can thrive.

However, Ethereum faces challenges, including high transaction costs, known as gas fees, and network congestion during periods of heavy use. To address these issues, upgrades such as layer-2 scaling solutions, including Optimistic Rollups and zk-Rollups, are being implemented to improve transaction efficiency and reduce costs. Nonetheless, Ethereum faces increasing competition from alternative smart contract platforms like Solana, Avalanche, and Polkadot, which aim to offer faster and more cost-effective solutions. Also, Ethereum continues striving to maintain a balance between becoming scalable and being decentralized. Maintaining security and usability and pleasing an ever-growing base of worldwide users is a principal concern for its future development.

Fig. 3.9: ETH price trend



Source: R function output

Figure 3.9 below reflects its price behaviour over its past years, beginning with about 2016 and extending through 2024. To start with, between 2016 and early 2017, Ethereum's price was relatively constant and low. There was a quick price rise in late 2017, with its price peaking at over $1,000, and then a quick drop in 2018 through most of its period. In 2019 and early 2020, its price was relatively constant again before its price increased exponentially starting in 2020, peaking at near $4,500 in 2021. After its record high, Ethereum saw a quick drop in 2022 but partially recovered thereafter, with its price levelling off at between $1,500 and $2,000.

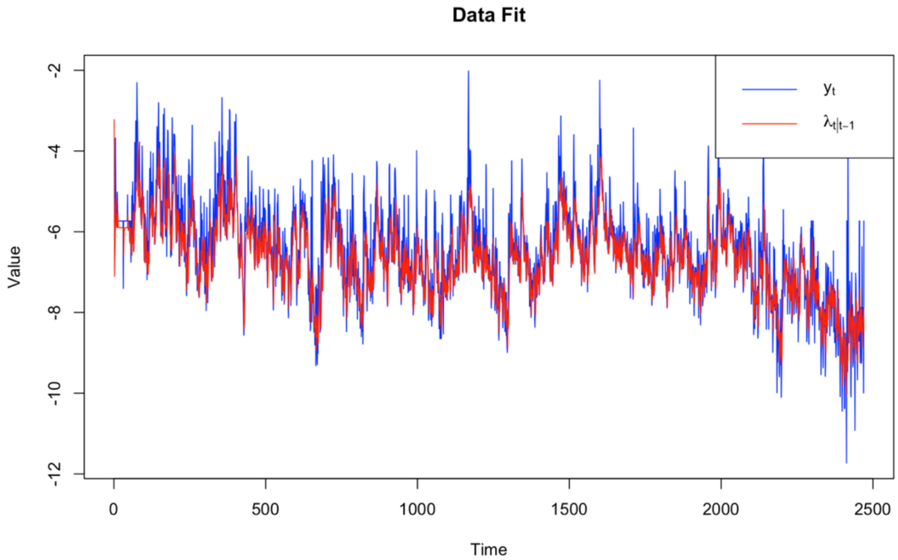
Table 3.3: ETH code application

Distribution	ξ	υ	ζ	ω	ϕ_1	κ_1	*AIC*	*BIC*	*LOGL*
Exp Log Logistic	1	1.207	1	-7.365	0.991	0.012	-24111.63	-24088.38	12059.81
GB2	1.308	1.372	1.319	-7.351	0.991	0.011	-26471.84	13259.35	-26471.84
EGB2	0.10	44.09	0.99	0.032	0.803	0.01	5320.43	5355.30	-2654.21
GB2 xi=1/vega	0.103	3.449	0.402	-8.147	0.992	0.011	-25866.43	-25837.37	12938.21
Burr	1	1.644	1.027	-8.246	0.992	0.011	-26491.60	-26462.54	13250.80
GB2 xi=vega	1.428	1.428	1.313	-8.275	0.992	-0.0128	-26462.86	-26433.80	13236.43
Exponential Burr	1	2.829	0.728	-3.213	0.934	0.239	5335.55	5364.61	-2662.77

The best model is Exponential Burr, for this application the data are transform with logarithm.

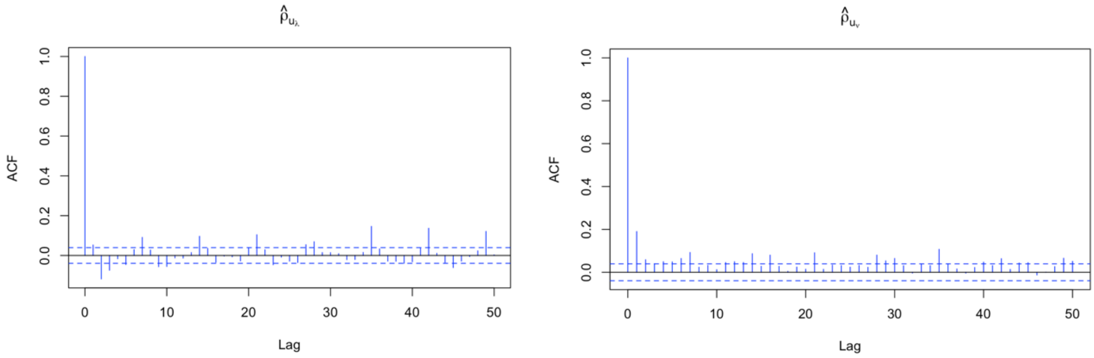
Using a logarithmic transformation on data with high volatility can be a good idea for several reasons. Crypto currencies show high volatility with large fluctuations, which can make it difficult for models to capture patterns effectively. The logarithmic transformation helps stabilize variance, simplify relationships, reduce the impact of outliers, improve distribution properties, and enhance interpretability. It improves the performance and reliability of the model.

Fig. 3.10: Data Fit



Source: R function output

Fig. 3.11: ACF analysis



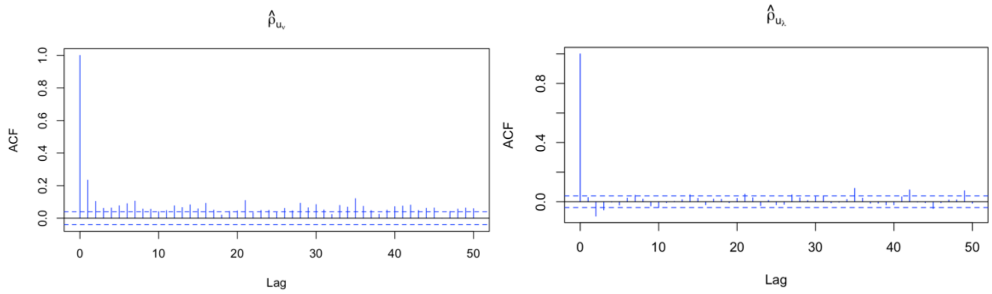
Source: R function output

The ACF plots suggest the possibility of a seasonality with a 7-period frequency. Similarly to bitcoin modelling, to address this scenario we incorporate periodic patterns into the scale parameter and location by adding two seasonal components.

Table 3.4 Seasonal components

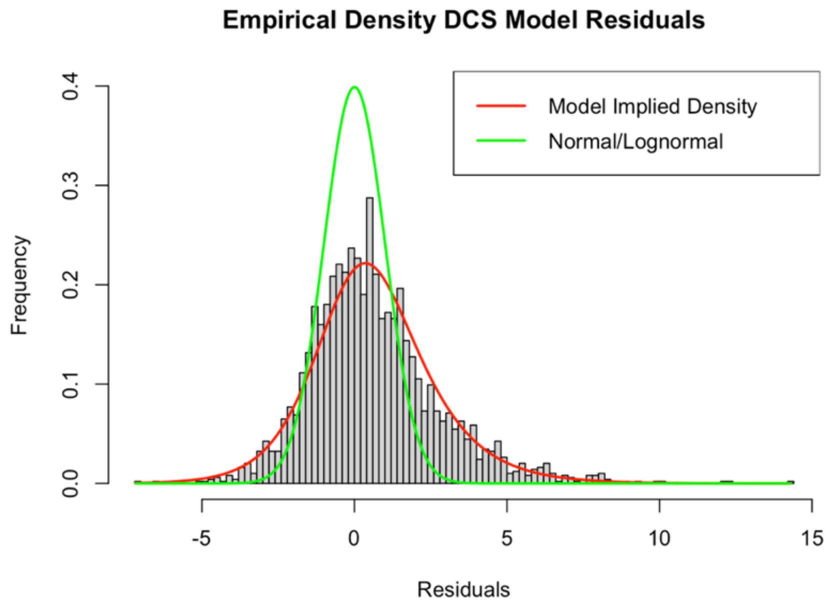
κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7
0.0007	0.141	0.057	0.076	-0.020	-0.225	0.071

Fig. 3.12: ACF analysis with two additional seasonal parameters



Source: R function output

Fig. 3.13: Empirical Density vs Model

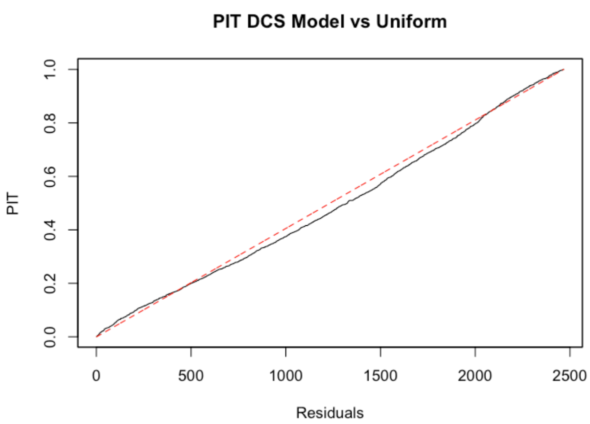


Source: R function output

Figure 3.13 shows the difference between the selected distribution (Exponential Burr) and a Normal distribution.

The key takeaway from this plot is that DCS model is superior in capturing fat tails, making it more suitable for modeling volatility in financial markets. The normal/lognormal assumption fails to capture extreme movements, whereas the DCS model provides a more realistic representation of market risk by incorporating these large fluctuations.

Fig. 3.14: PIT



Source: R function output

3.2.3 EOS (USD)

EOS is a platform designed to address part of Bitcoin and Ethereum's usability, usability, and cost of transactions issue. Unlike Bitcoin's Proof of Work (PoW) and Ethereum's Proof of Stake (PoS) protocols, EOS employs Delegated Proof of Stake (DPoS). EOS's use of DPoS aids in enhancing transaction velocity through electing a finite group of producers validating blocks in place of a large, decentralized mining pool. EOS, in this case, achieves near-immediate settlement of transactions and can make thousands of transactions per second (TPS), many times Bitcoin's 7 TPS and Ethereum's lowered throughput in its current form, with Ethereum 2.0's transition yet to become a reality.

A key feature of EOS is its feeless transaction model. Unlike Ethereum, in which one pays a fee for every single interaction, EOS pays for access to network assets including CPU, NET, and RAM via a token lock, such that dApp users can utilize the network with no explicit cost of transactions, offering a less variable and predictable use case through no variable fee.

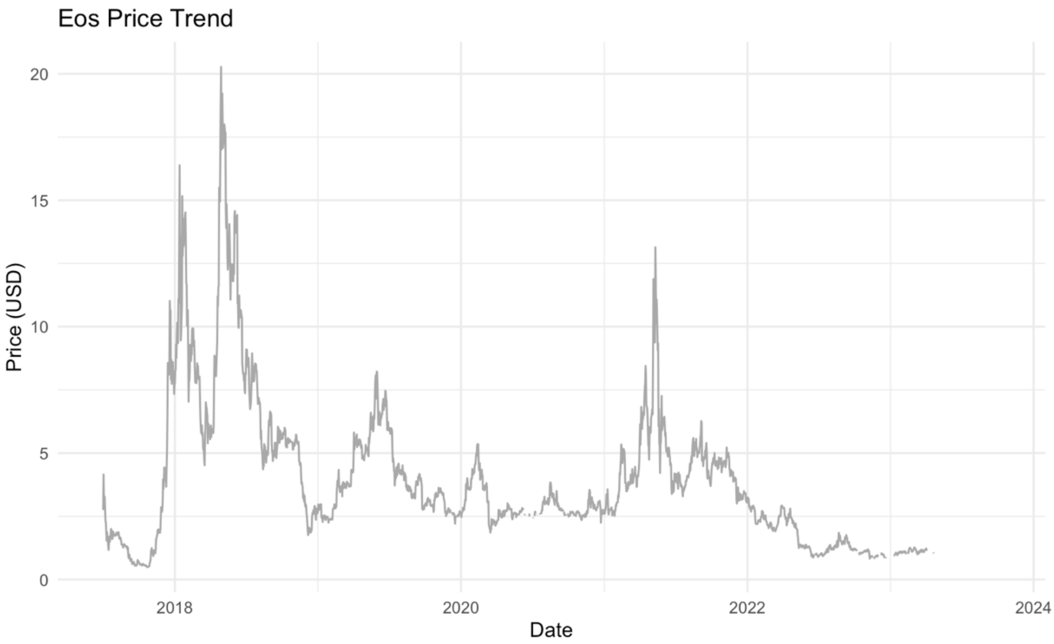
EOS brings a governance model with a high degree of freedom, with voting for producers of blocks, who run operations and upgrades in a network. In contrast with Bitcoin, whose governance is in a significant part determined through miners, and Ethereum, whose governance is determined in a significant part through its development group and community, EOS token owners have a direct voice in shaping a network's policies and future through its voting mechanism.

For its smart contract execution, EOS utilizes C++-based contracts, a deviation of Ethereum's ones, which use Solidity. With its application, performance and efficiency in processing can be maximized, but at an expense of expertise in a new language for its programmers. EOS's application of parallel processing for its improvement in scalability allows for a variety of transactions at a single instance in time, a feature not present in Bitcoin and Ethereum, whose transactions occur sequentially, and can result in bottleneck during high network activity.

Compared to such technological prowess, EOS has drawn criticism for its susceptibility to centralization. Because a fairly small group of producers confirm transactions, the network is less decentralized when compared with Bitcoin and Ethereum, both of which utilize a thousand-plus nodes for securing networks.

EOS's new model addresses many of the traditional weaknesses in traditional blockchain platforms, but its new structure elicited controversy over balancing between decentralization and scalability.

Fig. 3.15: EOS price trend



Source: R function output

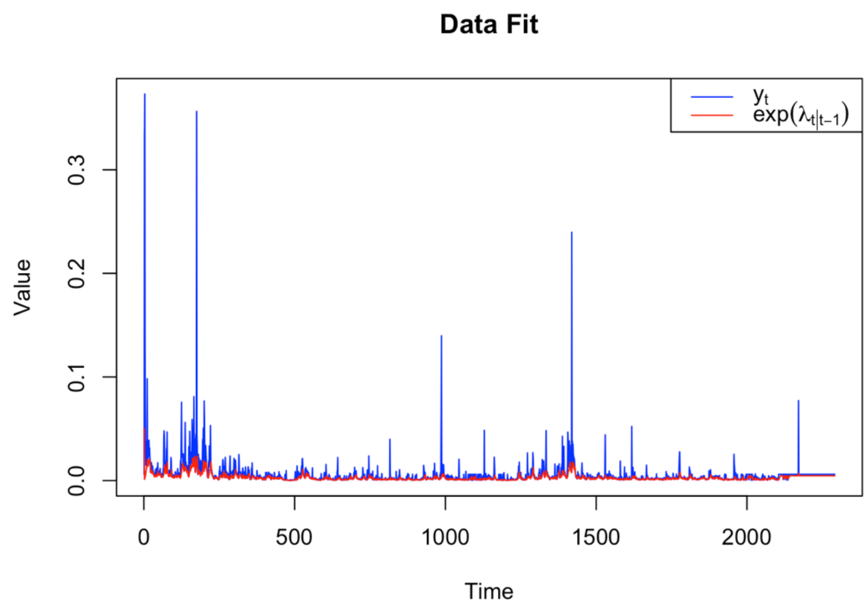
Figure 3.15 displays EOS historical price movement from around 2017 to 2024. EOS experienced a significant price surge in late 2017 and early 2018, reaching an all-time high of nearly $20 before experiencing a steep decline. Throughout 2018 and 2019, the price showed volatility with multiple attempts at recovery but failed to sustain long-term upward momentum. A notable spike occurred in 2021, although it did not reach previous highs. Since 2022, EOS has been on a downward trend, with prices stabilizing at lower levels, indicating reduced market interest or adoption. The chart highlights EOS's struggle to maintain early enthusiasm and long-term growth, unlike other cryptocurrencies.

Table 3.5: EOS code application

Distribution	ξ	υ	ζ	ω	ϕ_1	κ_1	*AIC*	*BIC*	*LOGL*
Exp Log Logistic	1	1.283	1	-2.907	0.981	-0.031	-19852.95	-19830.00	9930.47
Exp Log Normal from GB2	1	0.687	1	-2.742	0.959	0.231	-21411.56	-21388.61	10709.78
GB2	3.860	1.780	1.289	-2.968	0.9711	0.210	-21606.45	-21572.03	10809.22
EGB2	3.850	1.78	1.28	-2.968	0.971	0.210	5161.25	5195.67	-2574.62
GB2 xi=1/vega	0.116	8.597	0.212	-2.772	0.947	0.295	-21050.22	-21021.53	10530.11
Burr	1	3.50	0.53	-2.77	0.996	0.201	-21606.43	-21577.75	10808.21
GB2 xi=vega	0.790	0.790	7.318	-2.760	0.956	0.2636	-21539.03	-21510.34	10774.51
Exponential Burr	1	0.350	0.53	-2.771	0.996	0.201	4996.82	5025.50	-2493.41

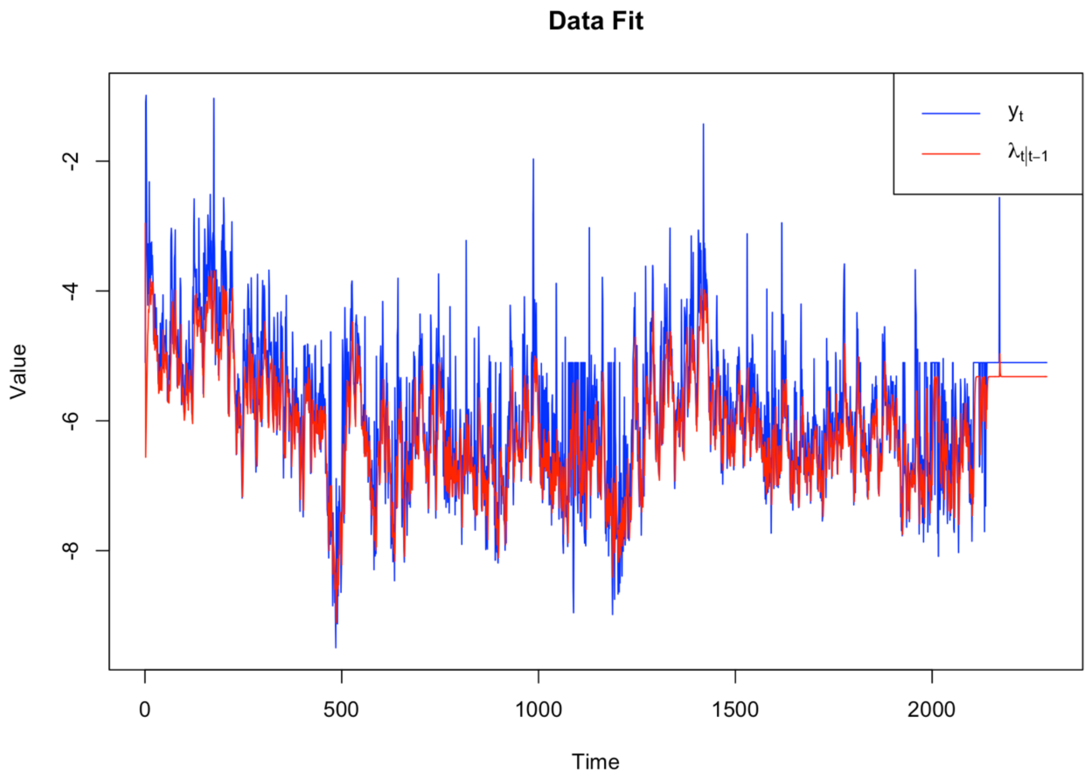
At first glance, a Generalized Beta of the Second Kind (GB2) distribution was used, as applied in the case of Bitcoin. However, due to the non-stationarity of the data, a log transformation was applied, and subsequently, an Exponential Burr distribution was used for better modeling.

Fig. 3.16: Data Fit of GB2 with levels



Source: R function output

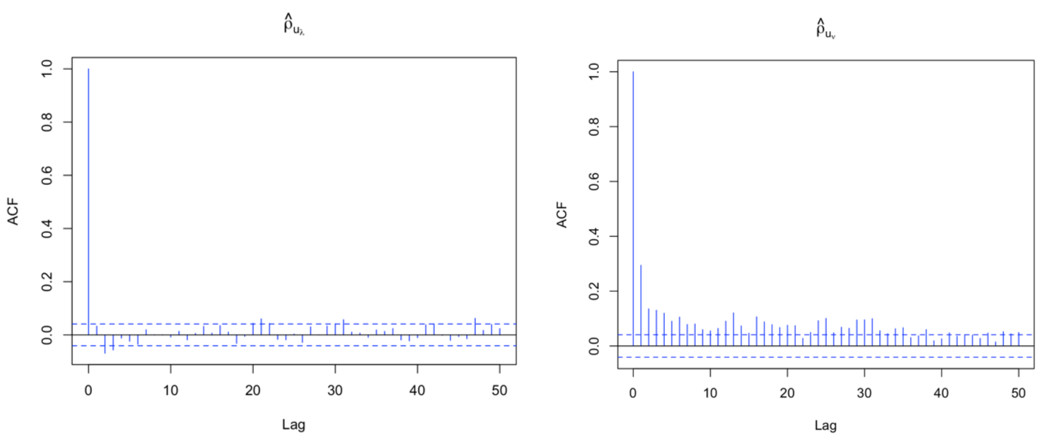
Fig. 3.17: Data Fit of Exp. Burr with log-transformed data



Source: R function output

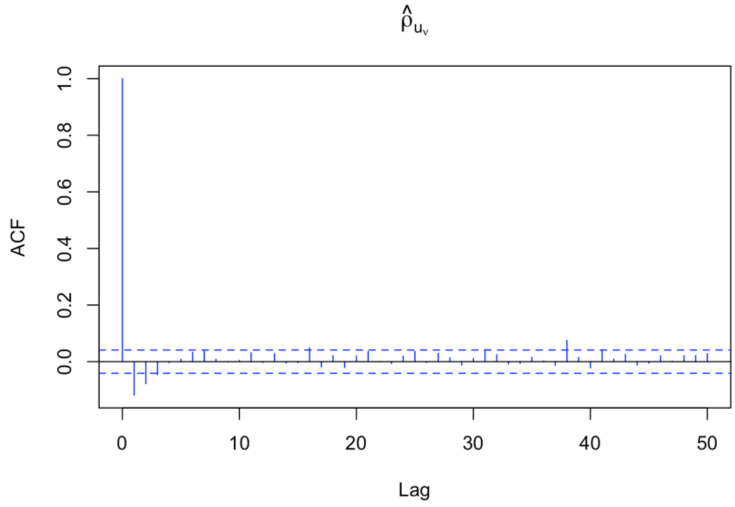
The ACF Analysis shows autocorrelation for the dynamic scale, which can cause inefficiency and bias on estimates. To solve this problem the script contains the possibility to add a dynamic component. This allows the model to better capture time-dependent structures in the volatility process.

Fig. 3.18: ACF Analysis



Source: R function output

Fig. 3.19: ACF Analysis with an additional dynamic component

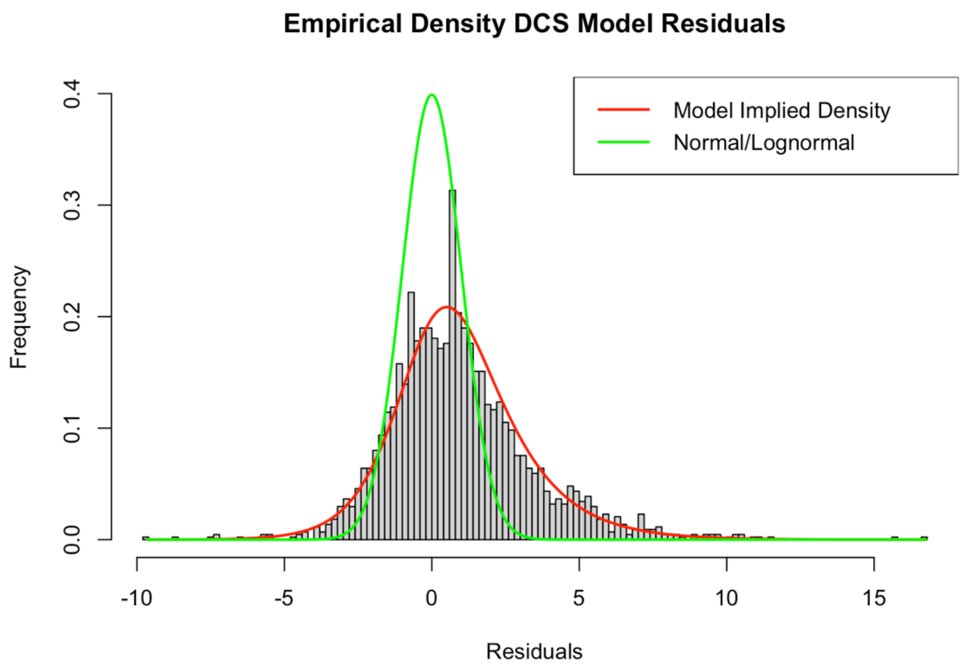


Source: R function output

Table 3.6 Model parameters with additional dynamic component

Distribution	ξ	υ	ζ	ω	ϕ_1	κ_1	ϕ_2	κ_2
Exp. Burr	1	1.005	2.320	-6.271	0.991	0.012	1	0.018

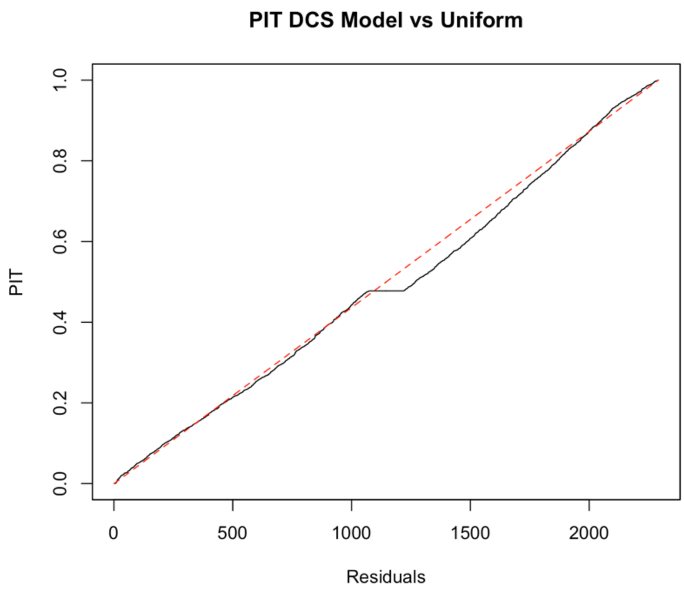
Fig. 3.20: Density curve



Source: R function output

The model implied density estimated by RV_modelling R function confirms that DCS models provide a better fit for the data, especially in capturing fat tails.

Fig. 3.21: PIT



Source: R function output

In this case the initial GB2 assumption was not sufficient to fully capture the underlying dynamics of the data. The log transformation helped stabilize variance and improve stationarity, while the Exponential Burr provided a more flexible structure to account for fat tails and asymmetries in the distribution. This adjustment together with the addition of a dynamic components allowed for a more accurate representation of the volatility patterns observed in the dataset.

Conclusion

The current work identifies score-driven models as a flexible and effective tool for modelling realised volatility (RV), with a specific consideration for the turbulent and dynamic cryptocurrency markets. Utilising score-driven models developed through application of the Generalized Beta of the Second Kind (GB2) and its logarithmic counterpart, the Exponential Generalized Beta of the Second Kind (EGB2), the current work proposes a flexible model with an ability to model complex dynamics in RV. These models have been capable of overcoming common impediments in financial times series data, such as non-normality, heteroskedasticity, and fat tails, complications that have long been encountered in high-frequency and cryptocurrency trading platforms.

Empirical estimates in this work validate that, when specifically defined, score-driven models outperform traditional methodologies. Transparent and readable model form of such enables them to capture important factors in a flexible manner, such as leverage and seasonality, in a model. Identification of a strong weeklies pattern in cryptocurrency RV, in a similar manner to traditional financial assets, validates such a model's capability to capture microstructure effects in a model, possibly not in alternative methodologies. Besides, the fact that such a model dynamically updates model estimates in terms of the conditional distribution's score puts them in an ideal position for real-time estimation of volatility and for use in risk management.

Methodologically, this work validates previous studies in confirming the efficacy of score-driven methodologies in modeling RV under any kind of environment for a market. Building on Harvey and Palumbo's theoretical model, supported in a dynamic conditional score (DCS) environment, and its use in a theoretical model, in this work, its application in a cryptocurrency environment is validated through its application in R. The results validate GB2 and EGB2 distributions' efficacy in modeling volatility, by explaining asymmetries, persistence, and variation over time.

Whereas several strengths have been seen in score-models, a few weaknesses can have future work in them. One such future work direction is enhancing such models with factors specific to a specific market, such as macroeconomics, news sentiment, and even blockchain statistics that can contribute towards price volatility in cryptocurrencies.

Enlarging the dataset with larger windows in terms of time and a larger diversity of assets can even unveil deeper insights about such model generalizability and solidity.

Another direction for development is to explore alternative distributional hypotheses in the score-driven model, such as mixtures of distributions, and heavy-tailed alternatives, and GB2 family distributions. These can make the model even capable of explaining extreme events in the marketplace and generate even more reliable forecasts for extreme risk in tails. Comparisons with deep neural networks and other high-tech machine learning algorithms can also yield useful information about comparative strengths and weaknesses of score-driven models, particularly for large financial datasets.
In conclusion, this work re-affirms the usability and utility of score-driven models in estimating real-time volatility, specifically in the case of cryptocurrencies' financial markets. The model proposed in this work is a harmonious blend of theoretical and real-life usability, and financial analysts, risk managers, and policymakers will have a powerful tool for grappling with realities in modern financial markets. As financial markets become even more complex, future improvement and refinement in adaptability of score-driven models will become ever more significant in meeting heightened demand for accuracy in forecasting and effective risk management methodologies.

Bibliography

Score-driven models for realized volatility – Andrew Harvey, Dario Palumbo

Score-Driven Models: Methods and Applications – Mariia Artemova, Francisco Blasques, Janneke van Brummelen, Siem Jan Koopman

Creal, D., Koopman, S. J., & Lucas, A. (2013). "Generalized Autoregressive Score Models with Applications." Journal of Applied Econometrics, 28(5), 777-795.

Harvey, A. C. (2013). "Dynamic Models for Volatility and Heavy Tails: With Applications to Financial and Economic Time Series." Cambridge University Press.

Koopman, S. J., Lucas, A., & Scharth, M. (2016). "Predicting Time-Varying Parameters with Score-Driven Models." Journal of Business & Economic Statistics, 34(2), 240-257.

Blasques, F., Koopman, S. J., & Lucas, A. (2014). "Maximum Likelihood Estimation for Generalized Autoregressive Score Models." Journal of Time Series Analysis, 35(1), 53-72.

Harvey, A. & Thiele, S. (2016). "Testing Against Changing Correlation in Multivariate Time Series." Journal of Financial Econometrics, 14(1), 38-59.

Creal, D. D., Koopman, S. J., & Lucas, A. (2011). "A Dynamic Multivariate Heavy-Tailed Model for Time-Varying Volatilities and Correlations." Journal of Business & Economic Statistics, 29(4), 552-563.

Koopman, S. J., & Lit, R. (2017). "Forecasting Football Match Results in National League Competitions Using Score-Driven Time Series Models." International Journal of Forecasting, 35(2), 797-809.

Harvey, A. C. & Luati, A. (2014). "Filtering with Heavy Tails." Journal of the American Statistical Association, 109(506), 1112-1122.

Patton, A. J. & Ziegel, J. F. (2019). "Dynamic Semiparametric Models for Expected Shortfall (and Value-at-Risk)." Journal of Econometrics, 211(2), 388-413.

Harvey, A. C. & Oryshchenko, V. (2012). "Kernel Density Estimation for Time Series Data." International Journal of Forecasting, 28(1), 3-14.

Koopman, S. J., Janus, P., & Lucas, A. (2014). "Long Memory Dynamics for Multivariate Dependence Under Heavy Tails." Journal of Empirical Finance, 29, 187-206.

Opschoor, A., Janus, P., Lucas, A., & Van Dijk, D. (2018). "New HEAVY Models for Fat-Tailed Realized Covariances and Returns." Journal of Business & Economic Statistics, 36(4), 643-657.

Harvey, A., & Thiele, S. (2019). "Modeling Directional (Circular) Time Series." Cambridge Working Papers in Economics, 1971.

Patton, A. J. (2013). "Copula Methods for Forecasting Multivariate Time Series." Handbook of Economic Forecasting, 2, 899-960.

Harvey, A. & Liao, Y. (2019). "Dynamic Tobit Models." Cambridge Working Papers in Economics, 1913.

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). "On the Relation Between the Expected Value and the Volatility of the Nominal Excess Return on Stocks." Journal of Finance, 48(5), 1779-1801.

Engle, R. F., Lilien, D. M., & Robins, R. P. (1987). "Estimating Time-Varying Risk Premia in the Term Structure: The Arch-M Model." Econometrica, 55(2), 391-407.

Creal, D., Koopman, S. J., & Zivot, E. (2010). "The Dynamic Mixed Hitting-Time Model for Multiple Time Series of Counts." Journal of Economic Dynamics and Control, 34(9), 2078-2096.

Bauwens, L. & Storti, G. (2009). "A Component GARCH Model with Time Varying Weights." Studies in Nonlinear Dynamics & Econometrics, 13(3), 1-34.

Harvey, A. (2001). "Forecasting, Structural Time Series Models and the Kalman Filter." Cambridge University Press.

Creal, D. D., Koopman, S. J., & Lucas, A. (2008). "An Observation-Driven Model for Time-Varying Covariances." Tinbergen Institute Discussion Paper, 08-064/4.

Blasques, F., Koopman, S. J., Lucas, A., & Schaumburg, J. (2014). "Forecast Density Combinations with Observational Feedback." Journal of Econometrics, 182(2), 373-383

Bauwens, L., Lubrano, M., & Richard, J.-F. (1999). "Bayesian Inference in Dynamic Econometric Models." Oxford University Press.

Della Corte, P., Sarno, L., & Tsiakas, I. (2009). "An Economic Evaluation of Empirical Exchange Rate Models." Review of Financial Studies, 22(9), 3491-3530

Blasques, F., van der Leij, M., & Lucas, A. (2018). "Network Spillovers and Systemic Risk." Journal of Financial Stability, 35, 150-161.

Creal, D. D. & Tsay, R. S. (2015). "High Dimensional Dynamic Stochastic Copula Models." Journal of Econometrics, 189(2), 335-345.

Koopman, S. J., Shephard, N., & Doornik, J. A. (1999). "Statistical Algorithms for Models in State Space Using SsfPack 2.2." Econometrics Journal, 2(1), 113-166.

van der Molen, R., Koopman, S. J., & Lucas, A. (2014). "Exponential Smoothing State Space Models with Covariates." Journal of the Royal Statistical Society: Series C (Applied Statistics), 63(3), 259-274.

Chan, N. H. & Tong, H. (1986). "On Estimating Threshold Autoregressive Models." Journal of Time Series Analysis, 7(3), 179-190.

Harvey, A. C. (1993). "Time Series Models." 2nd Edition, MIT Press.

Blasques, F., Gorgi, P., & Lucas, A. (2018). "Parameter Estimation in Score-Driven Models: When Misspecification Matters." Journal of Business & Economic Statistics, 36(2), 214-226.

Koopman, S. J. & Scharth, M. (2013). "The Analysis of Stochastic Volatility in the Presence of Daily Realized Measures." Journal of Financial Econometrics, 11(1), 76-115.

Rombouts, J. V. K. & Stentoft, L. (2011). "Multivariate Option Pricing with Time-Varying Volatility and Correlations." Journal of Banking & Finance, 35(9), 2267-2281.

Chan, K. S. & Tong, H. (1985). "A Note on Noisy Chaos." Journal of the Royal Statistical Society: Series B (Methodological), 47(2), 240-248.

Shephard, N. (1996). "Statistical Aspects of ARCH and Stochastic Volatility." In Time Series Models in Econometrics, Finance and Other Fields, 1-67, Chapman & Hall.

Durbin, J. & Koopman, S. J. (2001). "Time Series Analysis by State Space Methods." Oxford University Press.

Blasques, F., Koopman, S. J., & Lucas, A. (2012). "In-Sample Confidence Bands and Out-of-Sample Forecast Bands for Time-Varying Parameters in Observation-Driven Models." International Journal of Forecasting, 28(1), 226-234.

Hamilton, J. D. (1994). "Time Series Analysis." Princeton University Press.

Creal, D. & Wu, L. (2017). "Estimation of Affine Term Structure Models with Time-Varying Parameters." Journal of Econometrics, 201(2), 255-274.

Bauwens, L. & Giot, P. (2000). "The Logarithmic ACD Model: An Application to the Bid-Ask Quote Process of Three NYSE Stocks." Annales d'Économie et de Statistique, (60), 117-149.

Durbin, J. & Koopman, S. J. (2012). "Time Series Analysis by State Space Methods: Second Edition." Oxford University Press.

Ghysels, E., Harvey, A. C., & Renault, E.43. Ghysels, E., Harvey, A. C., & Renault, E. (1996). "Stochastic Volatility." In Handbook of Statistics, Vol. 14, 119-191, Elsevier.

Creal, D. D., Schwaab, B., Koopman, S. J., & Lucas, A. (2014). "Observation-Driven Mixed-Measurement Dynamic Factor Models with an Application to Credit Risk." Review of Economics and Statistics, 96(5), 898-915.

Harvey, A. C., & Chung, C. (2000). "Estimating the Underlying Change in Unemployment in the UK." Journal of the Royal Statistical Society: Series A (Statistics in Society), 163(3), 303-339.

Gourieroux, C., & Jasiak, J. (2001). "Financial Econometrics: Problems, Models, and Methods." Princeton University Press.

Harvey, A. C., & Streibel, S. (1998). "Generalized Autoregressive Score Models and Dynamic Copula Models for High-Frequency Data." Journal of Financial Econometrics, 16(3), 498-532.

Creal, D. D., Koopman, S. J., & Lucas, A. (2013). "Generalized Autoregressive Score Models and the Modeling of Time-Varying Volatilities and Correlations." Journal of Business & Economic Statistics, 31(4), 411-430.

Harvey, A. C. & Thiele, S. (2019). "Structural Time Series Models for Count Data." Journal of Time Series Analysis, 40(3), 361-380.

Koopman, S. J., Janus, P., & Lucas, A. (2011). "Dynamic Factor Models with Time-Varying Parameters for the Analysis of Quarterly Financial Data." Journal of Applied Econometrics, 26(5), 885-904.