University of Dublin

TRINITY COLLEGE

Multiresolution Analysis of High Frequency Financial Time Series Data

Daniel Lavelle
14334496

Supervisor: Dr. Khurshid Ahmad

Submitted in partial fulfilment of the requirements for the degree of M.A.I. to the University of Dublin, Trinity College, April 2019

School of Computer Science and Statistics

O’Reilly Institute, Trinity College, Dublin 2, Ireland

Declaration of Authorship

I, Daniel Lavelle, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signed:

Date:

Summary

This dissertation explores the usefulness of multiresolution analysis when dealing with high frequency financial time series data. The method is firstly compared to traditional autoregressive methods used to model high frequency time series data. It is then used to examine the risk of Bitcoin, Ryanair and S&P 500 across different timescales.

Multiresolution analysis involves decomposing a signal into its component parts such that each component part reflects activity in the signal at a different scale. This allows for features on different scales to be separated from one another and examined in isolation. Scale in this case refers to scale in time. For example, a signal could be decomposed into component signals where one of the component signals reflects daily activity, another weekly activity, another monthly activity and so on. It is a nonparametric modelling approach and as such does not require the creator of the model to make specifications to the model that are specific to the domain the model is being created for.

It is compared to the autoregressive modelling approach due to the fact that the autoregressive approach relies heavily on the correct selection of its parameters. Although the autoregressive method has achieved very significant results, it is limited by the fact that is a parametric approach. Whether or not the residuals in the autoregressive model have a strong relation to the results of the multiresolution analysis model was investigated. This was done with a view to the multiresolution analysis approach possibly providing a better alternative for isolating the noise in a signal. The results however were inconclusive.

Multiresolution analysis was also used to measure the risk of Bitcoin, Ryanair and S&P 500 across multiple scales. It showed that, as expected, Bitcoin is very risky over time periods greater than one day. However, the analysis showed that Ryanair was at least as risky as Bitcoin over periods less than one day. S&P 500 was shown to be have the lowest risk over time periods less than one day.

Acknowledgements

I would first like to acknowledge my supervisor, Dr. Khurshid Ahmad. He managed to create time for me where there was none to help explain some of the more difficult concepts involved in this dissertation. Some concepts he had to explain more than once and I appreciate the patience he showed in those instances.

I would also like to acknowledge the part that my family played throughout this year, I have had to call off my attendance to events on many occasions and they were very understanding of this. My cousin Matthew and sister Sarah provided great assistance also in the proof reading of my dissertation.

Table of Contents

1. Introduction

1.1. Complex Systems

When dealing with signal output from a system, separately the signal from the noise is an infamously difficult task. There are two main reasons for why this is the case:

1. The complexity of the system can often make it tough to make the distinction; there might be many factors that influence the system, however it might be difficult to determine what those factors are exactly.
2. If factors are known to affect the system, it might be difficult to measure them.

This dissertation deals with financial time series. Financial time series are certainly subject to these two limitations. Much work has gone into trying to explain the fluctuations of the stock market and the currency exchange however it is most likely an impossible task. The system that generates these prices is so complex, it is difficult to say for certain that any given variable doesn't influence the system due to the wide range of unpredictable agents that influence the system. However, that does not mean that it is not worth creating approximations of what influences the system as these can give a greater understanding of how these systems work if not a complete one.

1.2. Measuring the Unmeasurable

One of the pieces of research that influenced this dissertation was based on sentiment being a factor that influences financial systems (Tetlock 2007). Sentiment in this case is an influence on one's activity in the market that is not based on information about the market. Sentiment is known to be a big contributor in markets, the prominence of the field of marketing is a good example of this. Marketing is an effort to increase exposure of a product but also to increase positive sentiment about a product so that people buy that product over other similar products. However, sentiment can be quite difficult to measure; how would the effect of an ad on public sentiment or the effect of a harshly written article on public sentiment be measured? It is a difficult task.

1.3. High Frequency Series

The usefulness of data is becoming more and more relevant in this day and age, all big companies now look for a data driven approach to decision making to ensure objectivity. The increased prevalence of machine learning and artificial intelligence has certainly contributed to this given that data is the fuel that determines the effectiveness of these algorithms; generally, the more data available, the more potential there is for an effective machine learning model.

If the same logic is applied to financial time series models, more data would be desirable in order to increase the effectiveness of the model. There are two ways of doing this:

1. Adding more features to the model.
2. Increasing the sampling rate of the data in the model.

High frequency data is a product of this approach of increasing the sampling rate in the hope that it will improve the effectiveness of models. Sampling of financial time series has increased to such an extent that traders often operate on models that are built on data that samples with a regularity in the region of milliseconds.

The high frequency data dealt with in this dissertation is sampled every 5 minutes and has 288 values per day in the case of Bitcoin and 78 values per day in the case of Ryanair and S&P 500 (The difference in the sizes of the datasets is due to the trading hours being longer for Bitcoin). Having 288 values per day as opposed to one value per day, as is the case in traditional modelling, changes the range of the model significantly.

1.4. Overcoming the Limitations of Linear Modelling

There are two ways of modelling these financial assets, one of them is trying to gather as much data about the agents that contribute to the system and inputting them as factors that contribute to the price of these assets. Another way is modelling it by trying to model the signal itself. That is the approach multiresolution analysis takes.

Multiresolution analysis is built upon a signal transform called the wavelet transform. The wavelet transform allows for the ability to split a time series into multiple component time series, where each of those component time series represents features of a certain scale. Scale in this case could also be expressed as time horizon. The lowest scale resulting series of this multiresolution analysis will contain features that occur on the shortest possible time horizons and the highest scale series will contain features that occur on the longest possible time horizons.

This allows for examination of features at different scales and allows for comparisons at these different scales that could perhaps help with further understanding of the dynamics of these systems.

1.5. Flow of the Dissertation

The assets chosen for analysis in this dissertation are Bitcoin, Ryanair and S&P 500. S&P 500 was chosen due to the fact it is relatively stable, Bitcoin was chosen as a point of comparison due to it being an interesting innovation in finance and Ryanair was chosen as it provides an interesting point of comparison from the perspective of innovative business practices. This dataset will be further explored in section 3.1.

The wavelet and multiresolution analysis model is explored in section 3.2 and the autoregressive model is explored in section 3.3. These provide frameworks for nonparametric analysis and parametric analysis respectively. Ordinary least squares is introduced in section 3.4 as it is used in fitting trendlines to the data.

The residuals of the autoregressive model are compared to the results of the multiresolution analysis in section 3.5.1. in an effort to find out if the noise in the autoregressive model can be extracted using this nonparametric method.

Multiresolution analysis is then used to compare the risk of the Ryanair stock, the S&P 500 index and Bitcoin across multiple scales by comparing the variance of their volatility at these scales. The variance of volatility series of financial assets is used as a proxy for risk and allows for a case to be made for how risky these assets are at different scales and how they compare to one another. This is done in section 3.5.2.

The flow of this analysis can be seen in figure 1.5.

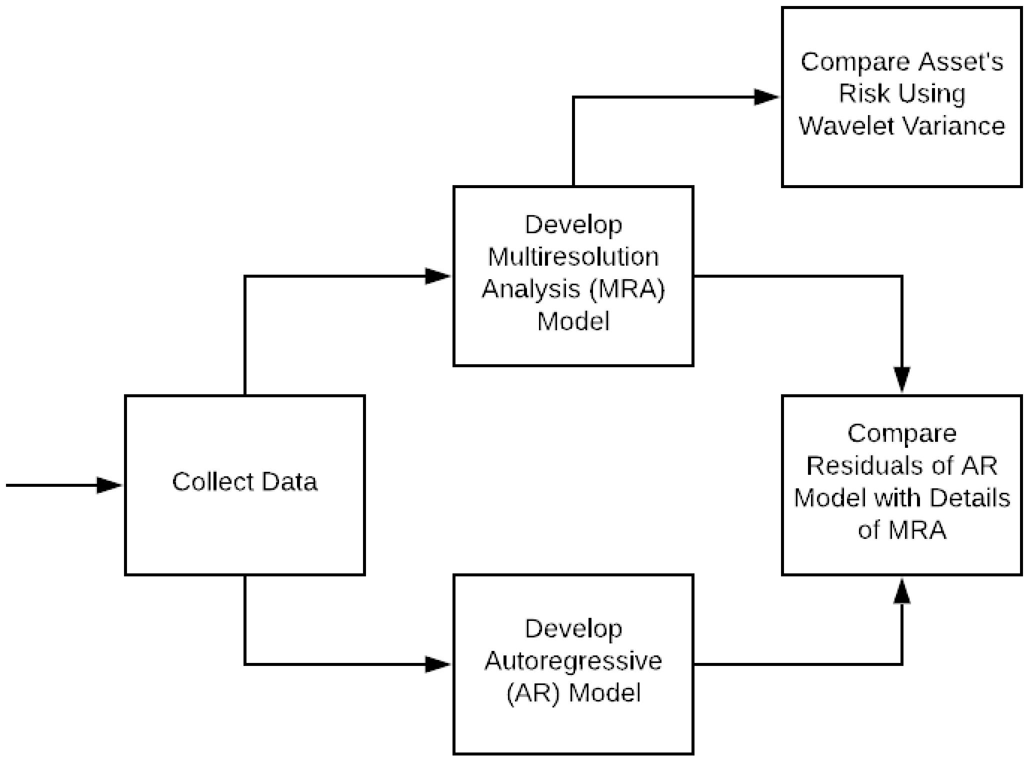


Figure 1.5. The flow of analysis for this dissertation.

2. Related Work

As implied by the title of this dissertation, multiresolution analysis is at the core of the research that was conducted for this dissertation. As the name suggests, multiresolution analysis involves analyses that break an input signal down into components of different resolution or scale (scale is the terminology used in the literature of this subject). Analysing an input series at a particular scale in the sense that it is used here involves extracting certain frequencies from the input series. The tool used to split an input series up into different scales is called the wavelet transform.

2.1. History of the Wavelet Transform and Multiresolution Analysis

The history of the Wavelet Transform cannot be discussed without discussing the Fourier Transform. The Fourier Transform was first developed by Joseph Fourier in 1807 (Polikar 1999) when he discovered that all periodic functions could be expressed as a weighted sum of basic trigonometric functions, this opened the door for a field of research exploring the possibility of representing signals in frequency rather than time. The Fourier Transform's prominence grew greatly when (Cooley and Tukey 1965) came up with the Fast Fourier Transform algorithm, this allowed the transform to be computed in $O(Nlog_2N)$ rather than $O(N2)$ time.

The main limitation of the Fourier Transform however is that it does not give resolution in time, only resolution in frequency. It shows the strength of the frequencies present in the signal but not if and when the strength of these frequencies change, thus it is only suitable for analysing stationary series. Dennis Gabor tried to solve this problem with the short time Fourier transform (STFT) (Gabor 1946). In order to capture localisation in time and in frequency he split the signal up into time segments, computing the Fourier transform of each of these segments. This technique however relied on picking the right size of segments to split the signal into. If the task is to extract the lowest frequencies in the signal, a large segment is needed however with a large segment, changes in the high frequencies will not be captured well. With the STFT, a choice must be made to either have good resolution of low frequencies or good resolution of high frequencies.

Morlet came up with the idea to use different segment, otherwise known as window, sizes for different frequencies (Grossmann and Morlet 1984). These windows were generated by dilation or compression of a function now known as a wavelet. Whereas the basis functions of the Fourier Transform are Cosine and Sine functions, the basis functions for the Wavelet Transform are dilations and compressions of this wavelet function. They did not know at the time however that the wavelet transform they developed was a rediscovery of work done by Alberto Calderon (Calderón 1964).

It turned out that the basic functions that Morlet proposed had much redundancy; they could have been expressed in a simpler form. Orthonormal wavelet basis functions had been developed by Alfred Haar in 1909 which did not have this same redundancy (Polikar 1999). Due to the fact that the Haar basis functions are orthonormal, they do not have the same redundancy as the Morlet basis functions. However, they are of little practical use due to their poor frequency localisation.

The crown jewel of these wavelet functions was developed by Ingrid Daubechies in 1988, these are the wavelets most commonly used today (Daubechies 1988). She developed a set of orthonormal basis functions that allow for a trade-off to be made between redundancy and frequency localisation depending on what is required.

It was in 1989 that Mallat then published the framework for Multiresolution analysis (Mallat 1989). He described a technique of decomposing a discrete signal into its dyadic frequency bands by a series of low pass and high pass filters, this process of feeding a signal through a low pass and high pass filter is in its essence the discrete wavelet transform where the filters are extracted from the wavelet function used. This combination of Daubechies' and Mallat's work created the framework for how multiresolution analysis has been carried out since.

As can be seen in figure 2.1a, the growth in the use of the wavelet transform has coincided with a decline in the use of the Fourier transform, possibly due to the fact that it overcomes the limitations of the Fourier transform.

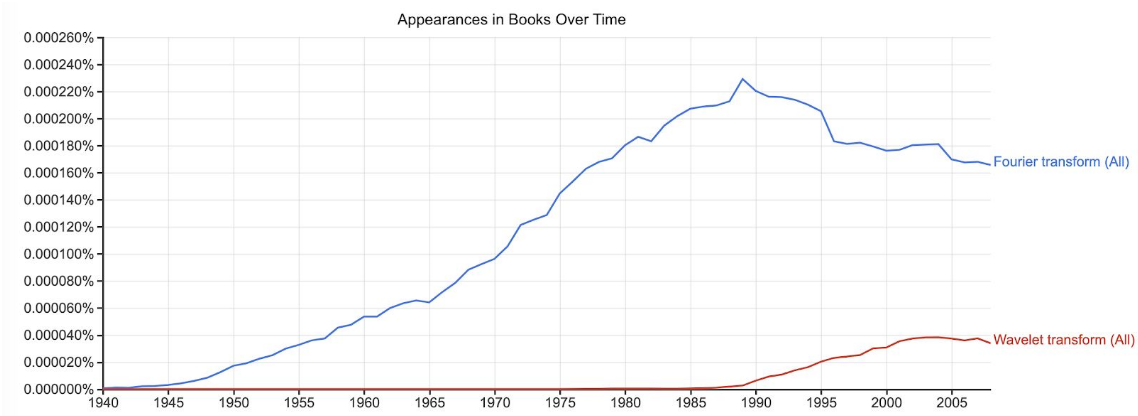


Figure 2.1a A comparison of the appearances in books over time of the Fourier transform and the wavelet transform. The data is taken from Google Books Ngram Viewer.

Due to the fact that Google Books Ngram Viewer only provides data up to 2008, this trend could not be compared up to present day. However, Google Trends was used to measure whether or not the interest in wavelet transforms has risen since, the results of this can be seen in figure 2.1b. Interestingly, search interest in the wavelet transform has been declining since 2004.

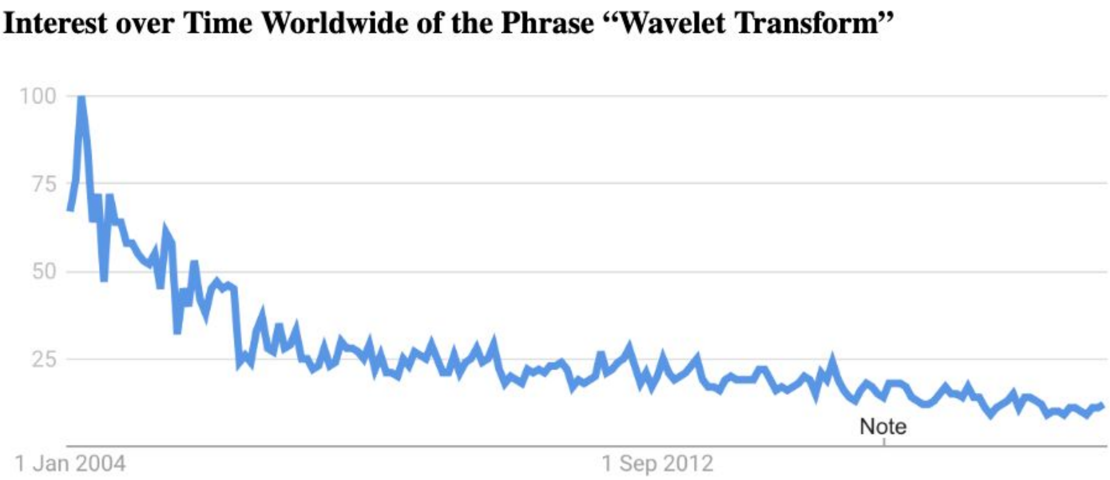


Figure 2.1b Interest over time worldwide in the phrase "Wavelet Transform" taken from 2004 to present day (2019). The above results are taken from Google trends. The following definition is Google's description of what the metric represents: "Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means that there was not enough data for this term."

2.2. Wavelet Methods in Finance

In complex systems such as geophysical systems, stock markets or currency exchanges, the patterns that emerge are complex. It is difficult to decipher what periods these patterns obey by simply looking at the series; a more sophisticated tool is needed. The Fourier transform has often been that tool to provide further insight into the frequencies that are present in the input series. Given an input time series, the Fourier transform will return the associated strength of a range frequencies present in the signal. The problem with this is that it assumes that the strength of these frequencies doesn't change over time; it assumes a stationary series.

However, when dealing with time series data that is a response variable of these complex systems, the series is invariably non-stationary and thus the usefulness of Fourier transform is quite limited in this circumstance. The wavelet transform is a potential alternative to the Fourier transform, it is not limited by the stationary assumption. It allows for resolution in frequency and time, it allows for analysis that accounts for these potential changes in frequency over time.

The usefulness of wavelets in the world of finance was brought forward by Ramsey just over 20 years ago, he used wavelet decomposition as a tool to analyse U.S. stock price behaviour over multiple scales (Ramsey, Usikov, and Zaslavsky 1995) . Since then, there have however been several prominent papers using this method of wavelet decomposition to analyse financial time series, the papers discussed in the following section will be ones that have influenced the field of research that this dissertation is based on.

This dissertation was heavily influenced by Ramazan Gençay, Faruk Selçuk and Brandon Whitcher. They produced multiple papers using wavelet decompositions to analyse financial time series. (Gençay, Selçuk, and Whitcher 2001b) investigated the scaling properties of foreign exchange volatility using wavelet decomposition. They showed that foreign exchange rate volatilities follow different scaling laws at different horizons. Particularly, there is a smaller degree of persistence in intraday volatility as compared to volatility at one day and higher scales.

(Gençay, Selçuk, and Whitcher 2001a) introduces an interesting method to isolate intraday seasonalities from high frequency time series. It uses the wavelet decomposition method to extract the highest frequency components from the series and remove them so that they don't interfere with the lower frequency components of the series (inter day components). As always with wavelet methods, it is a non-parametric approach and as such provides a general method for isolating the intraday seasonalities from the series.

(Gençay, Selçuk, and Whitcher 2005) introduces a method of estimating the systematic risk of an asset using a multiresolution approach. They argue that the CAPM of Sharpe (Sharpe 1964) and Lintner (Lintner 1965) is flawed as it assumes a linear relationship between scale and risk. However, they found that the empirical results from different economies show that the relationship between the return of a portfolio and its "beta" becomes stronger as the wavelet scale increases. Beta is a proxy for risk that is calculated by measuring the volatility of an asset or portfolio in relation to the overall market.

Another interesting use case of wavelet decomposition was shown by Vuorenmaa (Vuorenmaa, T.A., 2005) who used wavelet decomposition to analyse the scaling laws and long-memory in stock market volatility. Similarly to Gençay, they found that the scaling laws may not be time-invariant.

2.2.1. Wavelet Methods to Determine Interdependence between Variables

In & Kim conducted multiple studies to analyse the correlation between variables across different scales. For example, (In and Kim 2006) used wavelet analysis to examine the relationship between the stock and futures markets in terms of lead-lag relationship, correlation, and the hedge ratio. Among other conclusions, their empirical results show wavelet correlation between these two markets vary over investment horizons but remains very high.

Gallegati and Gallegati use wavelet methodology to analyse the industrial production index of the G-7 countries (Gallegati and Gallegati 2007). The analysis is performed using a multi-scaling approach which decomposes the variance of the industrial production index and the covariance between the industrial production indices of two countries on a scale-by-scale basis.

Gallegati used the maximal overlap discrete wavelet transform to compute the wavelet variance and cross-correlation between stock market returns and economic activity (Gallegati 2008). Their results showed that stock market returns tend to lead the level of economic activity, but only at the highest scales (lowest frequencies) corresponding to periods of 16 months and longer, and that the leading period increases as the wavelet time scale increases.

2.2.2. Wavelet Methods for Forecasting

Wavelet methods for forecasting seem to involve a common idea: break the input series down into its components using a wavelet decomposition, model each of those components individually to make a forecast on each scale and finally, sum up the forecasts of all scales to form the final forecast. One method for doing this is to use an ARIMA model to forecast these individual components, this method was examined by Conejo for forecasting electricity prices . They found this method to be an effective method of forecasting (Conejo et al. 2005). Renaud devised a similar model using an AR model instead of an ARIMA model and found positive results also (Renaud, Starck, and Murtagh 2003).

2.3. Other Areas of Influence

Wavelets are the main tool used in this dissertation and as such provide the most amount of relevant theory and background. Given that this dissertation is for an engineering masters and not a finance masters, the emphasis will rightly be on the theory that approaches the problem from a signal processing and mathematical standpoint. However, there are some concepts central to this dissertation that are outside the realm of signal processing and mathematics but nonetheless must be explored and explained such that the reader understands the motivation for this research. In the following piece, the economics and finance material that has influenced this research will be discussed so that the reader can have a deeper understanding of the importance of this work.

Part of this dissertation centres around the concept of noise, what does the noise represent and how does it manifest itself in mathematical models. The model of noise, in the context of financial markets, that is assumed to be true for this dissertation is that of Black (Black 1986). He describes financial markets as follows:

> *In my basic model of financial markets, noise is contrasted with information. People sometimes trade on information in the usual way. They are correct in*

expecting to make profits from these trades. On the other hand, people sometimes trade on noise as if it were information. If they expect to make profits from noise trading, they are incorrect. However, noise trading is essential to the existence of liquid markets.

Essentially, there are two forces that control the price of an asset in a market, one is the force of the noise trader, influenced by noise that they believe to be information. The effect the noise trader has on the market is to increase the gap between the price of asset and the value of the asset. The other force is the force of the information trader. The effect of the information trader is to reduce the gap between the price of an asset and the value of an asset. The two forces fight against one another to produce something of an equilibrium.

One of the interesting by-products of this hypothesis is that if one accepts it to be true, one must also accept that the noise that noise traders mistake for information has a similar effect on the market as the actual information. Traditionally, models that try and predict the price of an asset will only take into account information about the asset or information about the market. However, if the Black hypothesis is true, it is a very naive approach as it only accounts for one of the two aforementioned market forces. In reality, if one could effectively quantify this noise that influences these noise traders, one could in theory better predict the patterns in the market.

This is exactly what Tetlock did in his paper which is one of the motivators for this research (Tetlock 2007). Tetlock developed a sentiment proxy for the market using text analytics to look for words associated with certain sentiments. For example, words such as ‘profit’ or ‘increase’ would be associated with positive sentiment and words such as ‘loss’ or ‘decrease’ would be associated with negative sentiment. He developed this using a popular Wall Street Journal article; “Abreast of the Market”. He added this sentiment proxy as a variable in his VAR model (VAR models will be explained later in this dissertation) to see if it would have a positive effect on his predictions of the market returns.

His results conclusively show a relationship between sentiment and the residuals in the model. He found that high media pessimism predicts downward pressure on market prices followed by a reversion to fundamentals. This fits with Black's model of the market; the noise traders mistake this Wall Street Journal article for information and as a result sell their stock, forcing the value down. Soon after, the information traders notice that as a result of the noise trader's activity, there is now value to be found in the market, they then buy stock forcing the value back up to its original price. It is a clear example of Black's equilibrium in practice and provides a convincing case for the truth of this hypothesis.

Given that the autoregressive model involves input of so many parameters such as how many lags to account for and which other variables to account for (for example, Tetlock includes market volume as a variable in his autoregressive equation), it is quite reliant on domain knowledge and can be subject to the biases of the creator of the model. In an ideal world, a method would be used to isolate the residuals that is nonparametric in order to avoid these problems.

3. Method

3.1. Data Used

Three high frequency time series were chosen for analysis in this project. There were two criteria based on which the data was picked. Firstly, 5-minute interval data needed to be available for the series in question, 5-minute interval data is classified as being high frequency data for this dissertation. Secondly, series were picked that had different properties so that differences in results could be examined as a function of these different properties.

The first series chosen was the Bitcoin series. Bitcoin is an interesting candidate to model given how new a concept it is and also given how volatile the series is. It seems to be as unpredictable as any asset and as such, further inspection could deliver useful insights. The data was taken from Kaggle.com where Bitstamp (a well renowned Bitcoin exchange platform) data had been collected for roughly the last 7 years up until November 2018. The high frequency data was available however there were a lot of missing values in the data as can be seen in figure 3.1. Based on figure 3.1, June 2017 was chosen to be the cut-off for when the high frequency data became reliable.

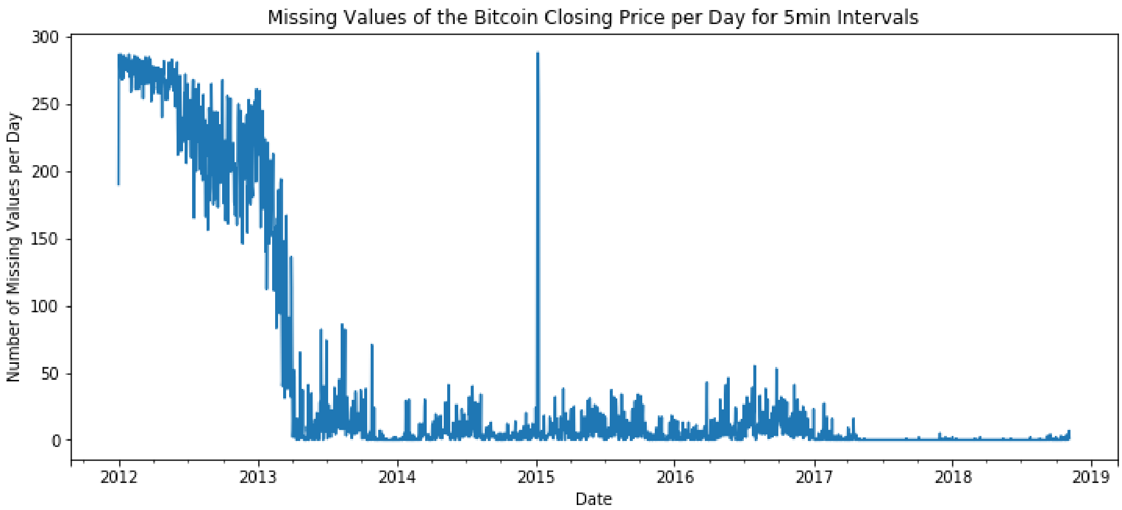


Figure 3.1 The missing values in the Bitcoin dataset sampled at 5-minute intervals from 2012 until 2019. The graph shows the consistency of the dataset improving significantly around mid 2017.

To complement this dataset, a series that was not so volatile or better understood was sought after. The S&P 500 index was used as it provides a good benchmark to compare against given that it is considered to be a reliable proxy for the movement of the market. The Ryanair data was also chosen due to its interesting feature of being an Irish company that trades on the NASDAQ stock exchange. The data for S&P 500 and Ryanair was generously provided by !QCL Solutions.

Although the data for S&P 500 and Ryanair was available for the last 2 years, in order to standardise this research, data was only chosen from 1st June 2017 until 11th November 2018 as this was the time period in which the Bitcoin data was reliable.

3.1.1. Dealing with Missing values

A complete dataset for Ryanair and S&P 500 was available, so missing values weren't a problem in that respect however there were a considerable number of missing values in the Bitcoin dataset and as such a method had to be decided on to deal with those missing values.

The choice was between removing all missing data points from the set or interpolating so as to replace the missing values with values that best approximate what the real values might have been. When it comes to removing values: it is the safest approach as no assumptions are made about the data. However, when dealing with time series data and in particular, when dealing with splitting the time series data in terms of frequency, the sampling rate must be consistent. If a missing value was dropped in a time series sampled every 5 minutes, there would be a 10-minute gap between the values either side of that which would be inconsistent with the 5-minute sampling rate. This inconsistent sampling rate would have repercussions that would make it hard to be convinced of the results of the analysis given the input data does not obey the assumption of a constant sampling rate.

The alternative to dropping the missing values is to use interpolation. In this dissertation, linear interpolation was chosen as the method for dealing with missing values. Linear interpolation involves fitting a line to the two values either side of the missing values and setting the value of the missing values to whatever the corresponding point is on the line for that index. For example, if there was just one missing value x_t, the value of x_t would be given as the average of x_{t-1} and x_{t+1}. This was used for all missing values of the series and allowed for use of a time series with a consistent sampling rate in this analysis.

3.1.2. Returns and Volatility

Throughout this dissertation, the two metrics analysed are log returns and volatility. The log returns (often referred to as just "returns") of the series were calculated as follows:
$r_t = log(P_t) - log(P_{t-1})$

where,

- P_t is the price of the price of the series at time t
- r_t is the returns at time t.

Returns is a metric that measures whether or not the price is moving up or down. Volatility is taken to be the absolute value of the returns for this dissertation similar to the literature in (Gençay, Selçuk, and Whitcher 2001c) and is a metric that measures how much the price of the series is moving.

3.1.3. Descriptive Statistics

Descriptive statistics were calculated for the volatility and returns of each of the assets. The statistics calculated were mean, median, standard deviation, minimum, maximum, kurtosis and skew. While the mean, median, standard deviation, minimum and maximum are very common metrics, kurtosis and skew are less commonly known. Kurtosis is defined in the Oxford English Dictionary as the sharpness of the peak of a frequency-distribution curve. A high kurtosis value is related to a sharply peaked distribution. While skew is defined as the state of not being symmetrical. Negative values of skew indicate that the distribution is skewed left while positive values of skew indicate that the distribution is skewed right.

3.2. Wavelet Model

The theory behind the wavelet model and multiresolution analysis will be presented in this section. This section was heavily influenced by (Percival and Walden 2006) and (Gençay, Selçuk, and Whitcher 2001c). The algorithms are described using linear filtering operations rather than matrix manipulations.

3.2.1. The Frequency Domain

When dealing with time series data, it is tough to understand the patterns of the series fully when looking purely in the time domain. It is difficult to detect and identify any seasonalities that are combinations of more than one sinusoidal function by merely a visual inspection. It would be possible to identify the period of a sine or cosine wave with the aid of a graphical representation but figuring out the individual periods of summations of these sinusoidal functions is very difficult without additional tools to help. In reality, when dealing with data such as stock market data, the patterns tend to be better modelled by the summation of multiple sinusoidal functions rather than just one. As such, tools are needed to help analyse these trends. The idea of looking at the series in the frequency domain is a useful idea and one that does give greater insight into these sinusoidal patterns.

The standard way to view time series data is in the time domain with time on the x-axis and the value of the function of the y-axis. However, in the frequency domain, instead of time on the x-axis, frequency is on the x-axis and on the y-axis, the associated amplitude of these frequencies is represented. One of the tools that allows for representation of a time series in the frequency domain is the Fourier transform.

3.2.2. Fourier Transform

The Fourier Transform decomposes a time series into its component frequency parts. The continuous Fourier Transform is defined as follows:

$$F\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}dt$$

where G(f) is the continuous Fourier transform of g(t).

When dealing with discrete time series, as is the case in this dissertation, a transform that takes a discrete series as an input is needed. The discrete Fourier transform fits this criterion. It transforms a sequence of N complex numbers

$$\{x_n\} := x_0, x_1, ..., x_{N-1}$$

into another sequence of complex numbers, the discrete Fourier transform of x_n:

$$\{X_k\} := X_0, X_1, ..., X_{N-1}$$

Which is defined by:

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-i2\pi}{N}kn}$$

where,

- k is the frequency index, range from 0 to N-1

3.2.3. Short time Fourier Transform

One of the limitations of the Fourier Transform is that it assumes a stationary series. A stationary series is one where the frequencies are constant, it does not account for changing frequencies. It simply returns one value of amplitude for each frequency but no resolution in time. As long as the signal being analysed is a stationary series, this is a perfectly adequate tool but when dealing with non-stationary time series, the Fourier transform does not suffice.

One way of dealing with non-stationary time series is to use the short time Fourier transform. The short time Fourier transform splits the series up into equal chunks and computes the Fourier transforms for each chunk so as to get resolution in time as well as frequency. However, the problem with the short-time Fourier transform is it can be difficult to select the size of the chunks the original series is split into. The wavelet transform provides a more general method of getting resolution in frequency and time, using different window sizes for extracting different frequencies.

The comparison between the domains of the input time series, Fourier transforms and wavelet transform is best captured in figure 3.2.3. Note that the wavelet transform has higher resolution for higher frequencies and lower resolution for lower frequencies as opposed to the short time Fourier transform which has the same resolution for all frequencies.

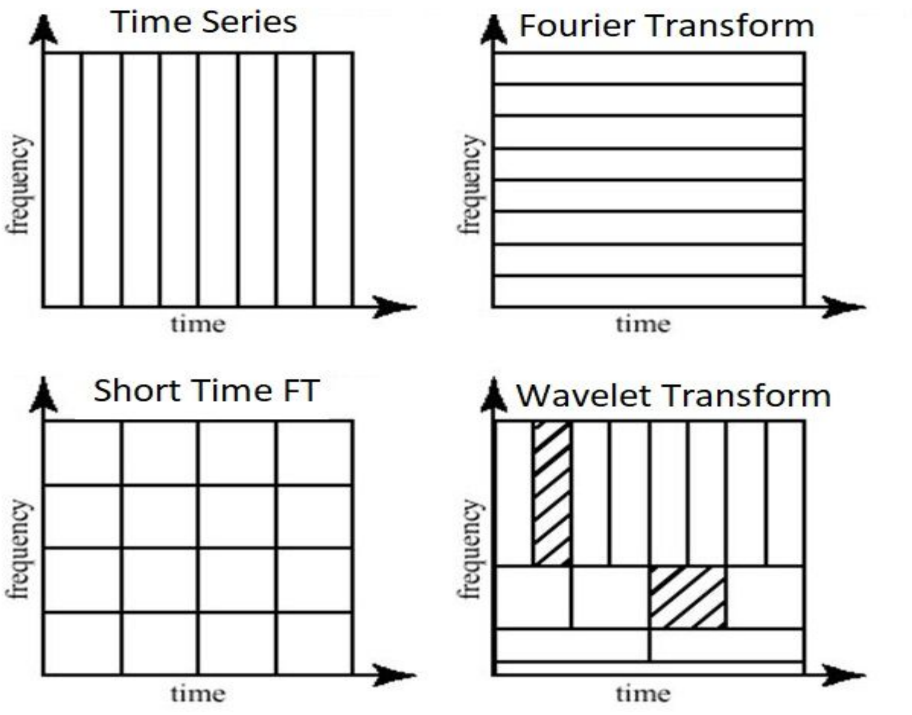


Figure 3.2.2.3, figure taken from (Gençay, Selçuk, and Whitcher 2001c) showing the localisation of a signal in time and frequency for the Fourier transform, the short time Fourier transform and the wavelet transform. Note the changing window size in the wavelet transform.

3.2.4. Wavelets

In order to describe the wavelet transform, wavelets must first be described. Wavelets are compact oscillations in time, the most obvious way in which they differ from sine and cosine functions is that they have compact support. A function has compact support if it is zero outside of a compact set (Rowland, Todd and Weisstein, Eric W). Sine and cosine waves do not have compact support as the signals continue oscillating no matter what the time frame chosen is. Wavelets can vary in properties such as length and symmetry but have a common property of resembling a brief oscillation in time. Figure 3.2.4 shows an example of multiple wavelets for reference.

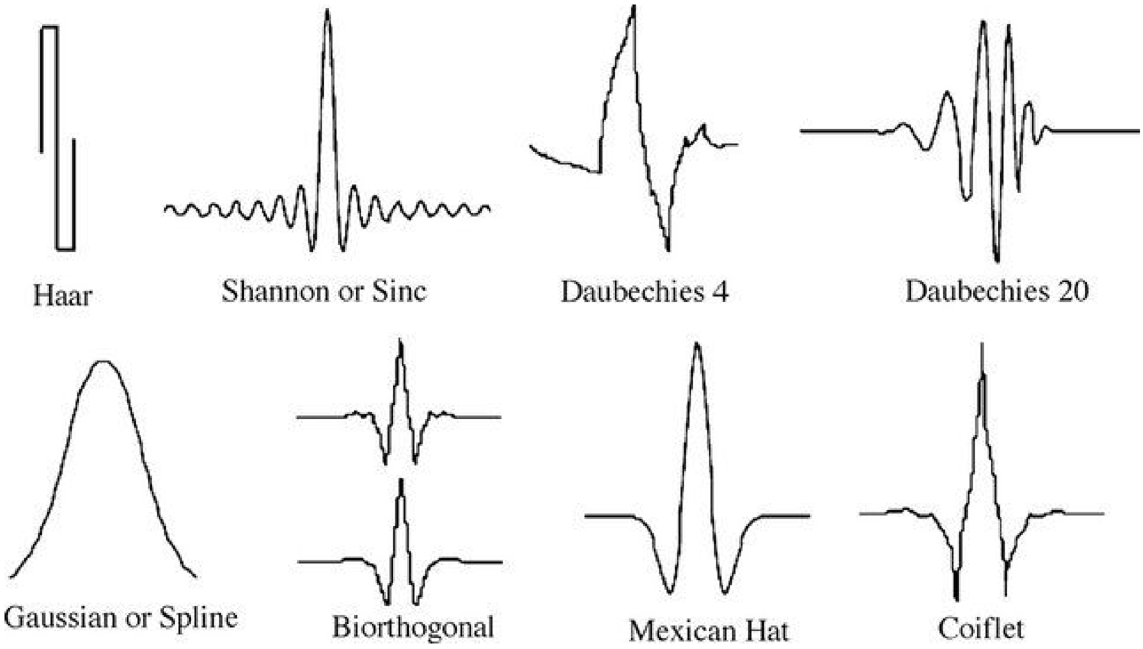


Figure 3.2.4 An illustration taken from (Godfrey et al. 2008) showing 8 examples of wavelet functions. Notice the compact support of each of these functions.

In order to qualify as a wavelet, the function $\psi(t)$ must obey the *admissibility condition*:

$$C_\psi = \int_0^\infty \frac{|\Psi(f)|}{f} df < \infty$$

where,

- Ψ is the Fourier transform of ψ

To guarantee that $C_\psi < \infty$, $\Psi(0) = 0$ must be imposed, which is equivalent to:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0$$

Equation 3.2.4a

A secondary condition imposed on a wavelet is function is unit energy where the energy of a function is defined to be the squared function integrated over its domain:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1$$

Equation 3.2.4b

Effectively what equations 3.2.4a and 3.2.4b imply is that the entries of the function must sum to zero but also have non zero entries.

3.2.5. Wavelet Transform

The wavelet transform is performed by convolving a wavelet with the input series at various different scales and translations. Each of these different scales of wavelets looks to isolate different frequencies present in the signal and each of these different translations looks to extract these frequencies at different points in time. The result of the transform is an output series that has high resolution for high frequencies and low resolution for low frequencies.

The formula for the continuous wavelet transform is as follows:

$$X_w(a,b) = \frac{1}{|a|^{\frac{1}{2}}} \int_{-\infty}^{\infty} x(t)\psi(\frac{t-b}{a})dt$$

where,

- a is a scaling value, $a > 0, a \in \mathbb{R}^+$
- b is a translational value, $b \in \mathbb{R}^+$
- ψ is a continuous function in both the time domain and the frequency domain called the mother wavelet.

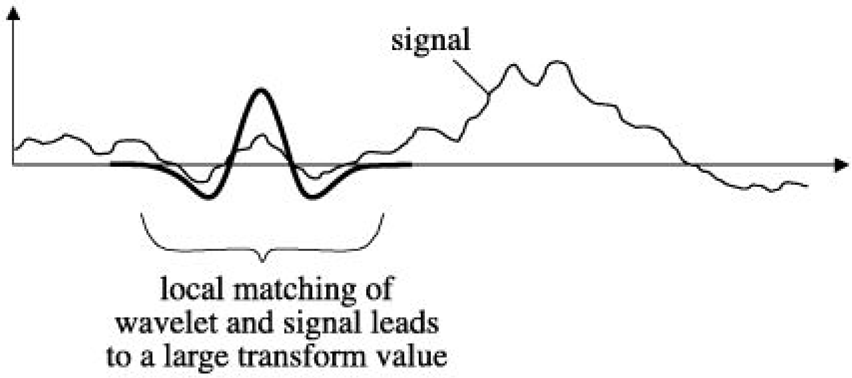


Figure 3.2.5a Illustration taken from (Pinello 2011) visualising the concept of convolving two signals.

Figure 3.2.5a is an example of a wavelet of a certain scale and translation being convolved with the input series; this would return a single value. In order to obtain the rest of the values in time for that particular wavelet scale, the wavelet would be translated across the series, convolving with the series for each translation. In order to obtain the rest of the values for the various other scales, the scale of the wavelet would be made larger to capture lower frequencies and smaller to capture higher frequencies.

Similarly to the Fourier transform, a discrete version of this formula is needed to work with real world discrete time series data.

3.2.6. Discrete Wavelets

In order to develop a model for a discrete wavelet transform, a model for discrete wavelets is needed. The discrete wavelet is defined in terms of a wavelet filter, h, and a scaling filter, g. These two filters act as high pass and low pass filters respectively. A high pass filter is a method of filtering in digital signal processing that has the effect of not allowing frequencies lower than a certain threshold through the filter; it is a method of extracting frequencies above a certain threshold. Conversely, a low pass filter is one that extracts frequencies below a certain threshold. The constraints for the wavelet filters are as follows:

$$\sum_{l=0}^{L-1} h_l = 0$$

$$\sum_{l=0}^{L-1} h_l^2 = 1$$

$$\sum_{l=0}^{L-1} h_l h_{l+2n} = 0$$

where,

- L is the length of the wavelet filter,
- n is any nonzero integer.

In words, the wavelet filter must sum to zero, have unit energy and must be orthogonal to even shifts. The scaling filter g must obey the second two properties specified however it must sum to either $\sqrt{2}$ or $-\sqrt{2}$:

$$\sum_{l=0}^{L-1} g_l = \sqrt{2} \quad \text{or} \quad \sum_{l=0}^{L-1} g_l = -\sqrt{2}$$

3.2.7. Discrete Wavelet Transform

The discrete wavelet transform (DWT) of a signal is calculated by passing the signal through these low pass and high pass filters. The coefficients of these low and high pass filters are extracted from the wavelet bases used. The output of the low pass filter in the DWT is commonly called the smooth coefficients (v) and the output of the high pass filter in the DWT is commonly called the detail coefficients (w). In order to capture the effect of the doubling of the scale of the wavelet, the series is down sampled by 2 during the wavelet transform.

The formula for the calculating the values w and v is as follows:

$$w_t = \sum_{l=0}^{L-1} h_l x_{2t+1-l mod N}$$

$$v_t = \sum_{l=0}^{L-1} g_l x_{2t+1-l mod N}$$

where,

- h is an array of high pass filter coefficients,
- g is an array of low pass filter coefficients,
- L is the length of the array of the filter coefficients, also known as the size of the wavelet used,
- x is the input series,
- t = 0, 1, … , N/2 - 1.

Note that the down sampling of the series is captured in the indexing of x_t.

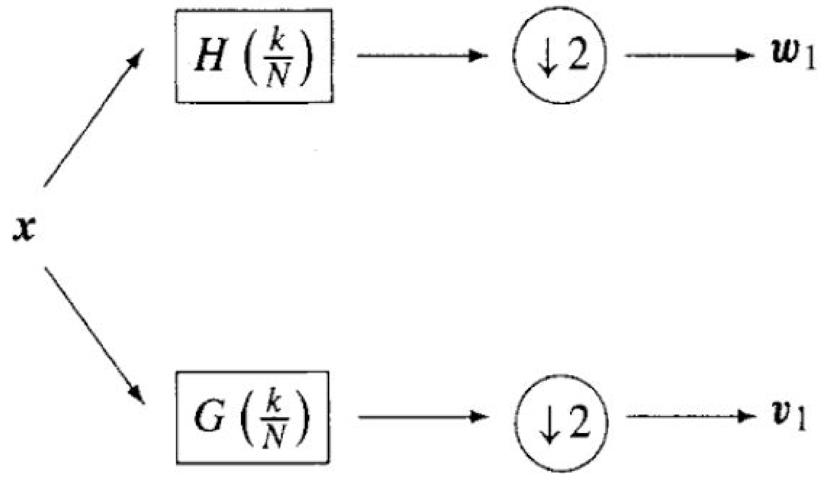


Figure 3.2.7 A diagram taken from (Gençay, Selçuk, and Whitcher 2001c) showing the high pass and low pass filter effect of the DWT. H(k/N) and G(k/N) are the high pass and low pass filters respectively. ↓2 signifies a down sampling of the signal by 2. For a given discrete signal x, down sampling the signal by two would result in dropping every second value in the signal.

Figure 3.2.7 demonstrates this process of passing the series through low and high pass filters and then down sampling to give v and w where w contains the features of the higher frequencies present in x and v contains the features of the lower frequencies present in x. It is important to note however that there is no such thing as an ideal high pass or low pass filter and as such the frequencies that pass through the filter aren't exactly what is desired, this will be further examined in section 3.2.11.

3.2.8. Maximal Overlap Discrete Wavelet Transform

The maximal overlap discrete wavelet transform (MODWT) is another type of wavelet transform that is computed by not down sampling the output from the low pass and high pass filters. The MODWT gives up the orthogonality of the DWT to gain features that the DWT does not possess. The following properties are important in distinguishing the MODWT from the DWT (Percival and Mofjeld 1997):

- The MODWT can handle any sample size N, while the $J_p th$ order partial DWT restricts the sample size to a multiple of 2^{j_p}
- The MODWT is invariant to circularly shifting the original time series. Hence, shifting the time series by an integer unit will shift the MODWT detail coefficients and smooth coefficients the same amount. This property does not hold for the DWT.

In the case of the DWT, the effect of the wavelet scaling is achieved by down sampling the signal at each iteration. However, in the case of the MODWT, the scaling effect is achieved by the up sampling of the vector of wavelet and scaling coefficients each time.

The detail coefficients and smooth coefficients of the MODWT are commonly referred to as $\tilde{W}$ and $\tilde{V}$ respectively. In order to compute $\tilde{W}$ and $\tilde{V}$ for the wavelet transform at decomposition j, the following equations are used:

$$\tilde{W}_{j,t} = \sum_{l=0}^{L-1} \tilde{h}_l \tilde{V}_{j-1,t-2^{j-1} l mod N}$$

Equation 3.2.8a

$$\tilde{V}_{j,t} = \sum_{l=0}^{L-1} \tilde{g}_l \tilde{V}_{j-1,t-2^{j-1} l mod N}$$

Equation 3.2.8b

where,

- $\tilde{h}$ is an array of high pass filter coefficients rescaled by dividing h by $\sqrt{2}$,
- j is the decomposition number of the transform,
- $\tilde{g}$ is an array of low pass filter coefficients rescaled by dividing g by $\sqrt{2}$,

- L is the length of the array of the filter coefficients,
- $\tilde{V}_j$ is the vector of smooth coefficients at decomposition j,
- $\tilde{W}_j$ is the vector of detail coefficients at decomposition j,
- t = 0, 1, … , N - 1,
- $\tilde{V}_0$ is the original series x_t.

The MODWT is quite desirable as it does not require the signal to be of length 2^n, this is quite a restrictive limitation of the DWT and the MODWT overcomes this. However, compensating for this is the fact that there is much redundancy in the filter coefficients used due to the fact that they are up sampled by the transform. This trade-off was made for this dissertation and as such, the MODWT was used in place of the DWT for all analyses.

The inverse maximal overlap discrete wavelet transform (IMODWT) is calculated using both the smooth coefficients and detail coefficients:

$$\tilde{V}_{j-1,t} = \sum_{l=0}^{L-1} \tilde{h}_l \tilde{W}_{j,t+2^{j-1}l mod N} + \sum_{l=0}^{L-1} \tilde{g}_l \tilde{V}_{j,t+2^{j-1}l mod N}$$

where the notation is the same as equations 3.2.8a and 3.2.8b.

3.2.9. The Pyramid Algorithm

In order to extract information about the input series across more than one scale, multiple DWTs, or MODWTs must be applied. Applying the DWT or MODWT only once will only result in acquiring the detail coefficients and smooth coefficients for one range of frequencies or one scale which doesn't give the resolution in frequency that is desired. In order to obtain this resolution in frequency, the transforms must be applied more than once. Each decomposition uses the smooth coefficients from the previous decomposition to produce the results for that decomposition level. The name of this algorithm is called the pyramid algorithm. The flow of the decomposition can be seen in figure 3.2.9.

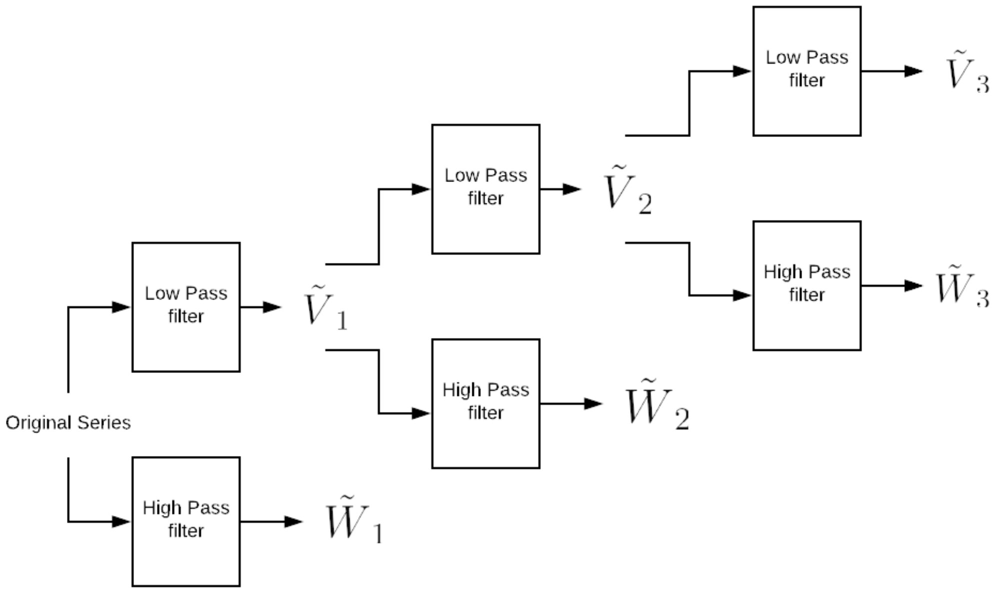


Figure 3.2.9 A diagram illustrating the process of the pyramid algorithm. Each operation where the signal, be it $\tilde{V}$ or the original series, is passed through a pair of filters, this is equivalent to the MODWT being performed. $\tilde{W}$ and $\tilde{V}$ represent the detail coefficients and smooth coefficients respectively

where,

- $\tilde{W}_1$, $\tilde{W}_2$ and $\tilde{W}_3$ reflect the detail coefficients at different scales,
- $\tilde{V}_1$, $\tilde{V}_2$ and $\tilde{V}_3$ reflect the smooth coefficients at different scales.

3.2.10. Multiresolution Analysis

Using the DWT or the MODWT, a decomposition can be formulated such that the sum of the decompositions equals the series decomposed. In this model, for a given point t, the value of x at point t will be given as follows:

$$x_t = \sum_{j=1}^{J} D_{j,t} + S_{j,t}$$

where,

- $D_{j,t}$ is the value of the wavelet detail at decomposition j and time t,
- $S_{J,t}$ is the value of the wavelet smooth at decomposition J and time t.

Similarly, the whole series can be decomposed as follows:

$$x = \sum_{j=1}^{J} D_j + S_j$$

where,

- D_j is the wavelet detail at decomposition j,
- S_j is the wavelet smooth at decomposition j.

In order to get to the details and the smooths from the detail coefficients and smooth coefficients, the following formulae are used:

$$D_{j,t} = \sum_{l=0}^{N-1} h^o_{j,l} \tilde{W}_{j,(t+l) modN}$$

$$S_{j,t} = \sum_{l=0}^{N-1} g^o_{j,l} \tilde{V}_{j,(t+l) modN}$$

where,

- N is the length of the original series x,
- $h^o_{j,l}$ is obtained through periodizing $h_{j,l}$ to length N,
- $\tilde{W}_{j,t}$ is the detail coefficients at decomposition level j and time t,
- $g^o_{j,l}$ is obtained through periodizing $g_{j,l}$ to length N,
- $\tilde{V}_{j,t}$ is the smooth coefficients at decomposition level j and time t,

In effect, this is equivalent to applying the IMODWT numerous times with vectors of zeros (Percival and Walden 2006), this flow is shown in figure 3.2.10.

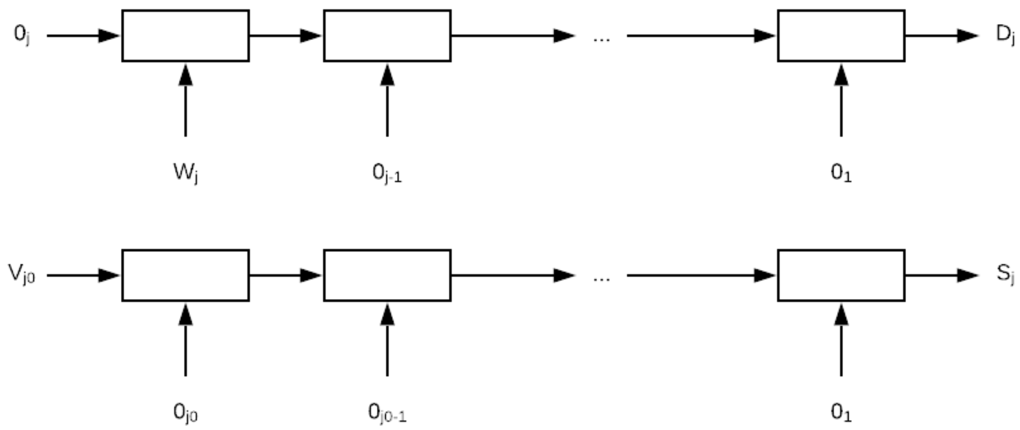


Figure 3.2.10 The flow of producing the details. Each box represents an IMODWT. J_0 represents the maximum decomposition reached in the process. 0_k, k=1, …, j, is a vector of N zeros.

3.2.11. Making Sense of the Results of Multiresolution Analysis

The results of a multiresolution analysis can be difficult to interpret given that this concept of scale or decomposition is not easy to relate to other concepts. It can be difficult to understand what the details describe for a given scale. In this section, the meaning of the details at a given scale will be explored. The meaning of the scales is described in (Gençay, Selçuk, and Whitcher 2001c) and goes as follows:

If a series is sampled with a sampling rate of period k seconds, the first detail accounts for features of the series with period $2k$ to $4k$, the second detail accounts for features of the series with period $4k$ to $8k$, the third detail accounts for features of the series with period $8k$ to $16k$ and so on. The upper and lower bounds for the band of frequencies described by the details at decomposition j are given as follows:

$$Lower\ bound = k(2^j)$$

Equation 3.2.11a

$$Upper\ bound = k(2^{j+1})$$

Equation 3.2.11b

where,

- k is the period of the sampling rate,

- j is the decomposition number.

Figure 3.2.11 gives an example of this band pass concept in the context of the details. Using the MODWT, the original signal of Bitcoin returns sampled hourly was decomposed three times. The power spectrum of the original signal and each of its composite series was calculated. The power spectrum is calculated by calculating the Fourier transform of the series and then squaring the values of the Fourier Transform. It describes the strength of the frequencies present in the signal.

As can be seen from figure 3.2.11, the first decomposition $D1$ includes the frequencies in the original signal that range from having period two hours to period four hours. $D2$ includes frequencies that range from having period four hours to period eight hours. $D3$ includes frequencies that range from having period eight hours to period sixteen hours. $S3$ accounts for the rest of the frequencies. It is interesting to note however that these aren't strict thresholds.

It is quite clear in all four of the composite series that the frequencies in the series aren't confined to just the frequencies within the lower and upper bounds. This is a result of the wavelet transform not being a perfect band pass filter and this should be kept in mind when looking at the results.

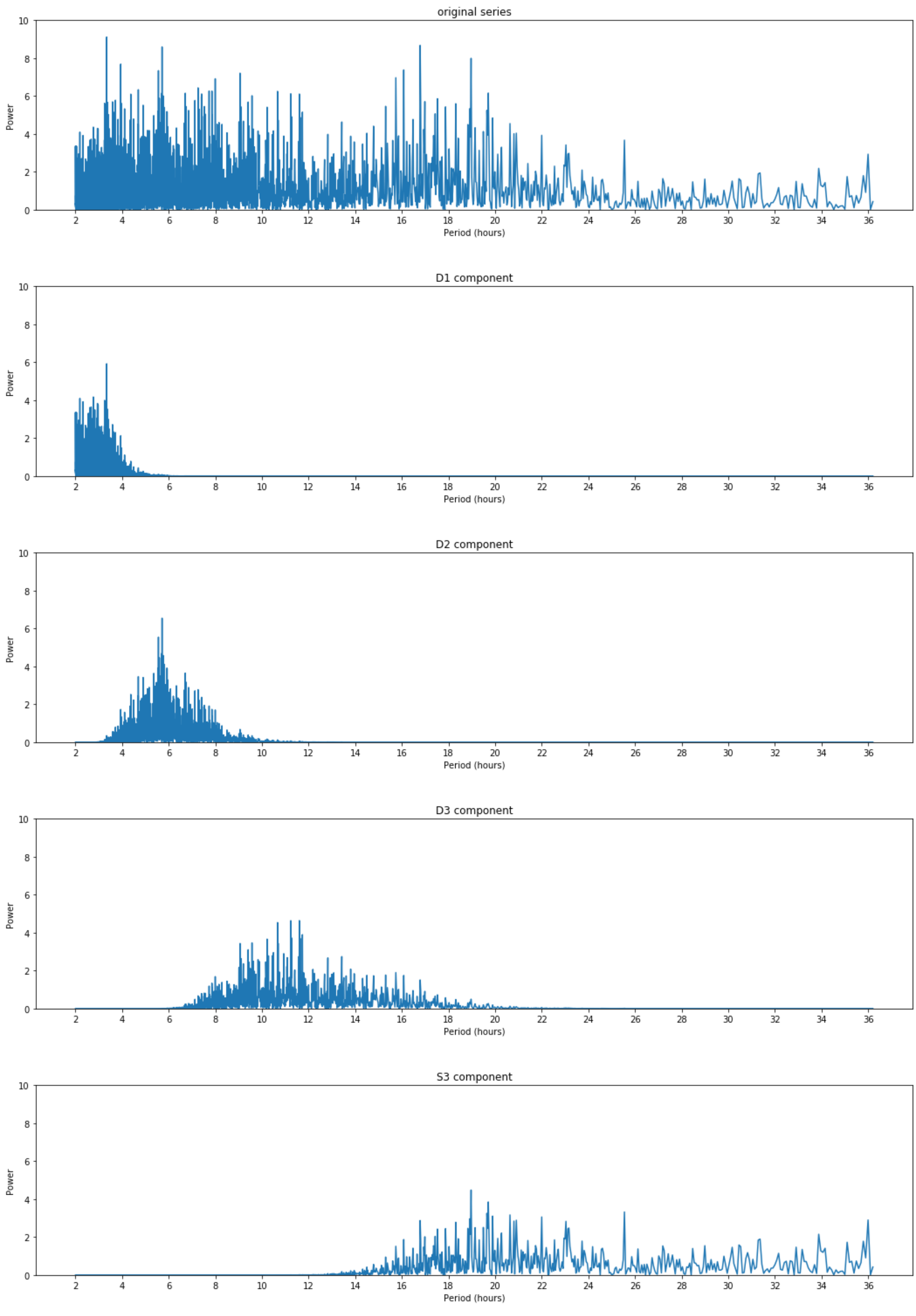


Figure 3.2.11 The power spectrums for the original series and 3 decompositions. The series used in this case was bitcoin returns sampled hourly from 1/6/2017 - 11/11/2018. The graphs labelled $D1, D2, D3$ *are the power spectrum of the details are decomposition level 1, 2 and 3 respectively while the graph labelled* $S3$ *is the power spectrum of the smooth at decomposition level 3. It is a demonstration of how*

the details accounts for an approximate band of frequencies, with the upper and lower bound of this band increasing exponentially a function of J, the decomposition number. Note, the plots only account for frequencies with periods up to 36.2 hours to make the graphs easier to read.

This dissertation will later refer to the details accounting for a certain frequency band at decomposition j; in this case, the frequency band specified is that band contained by the bounds specified in equations 3.2.11a and 3.2.11b.

This dissertation also later talks of details accounting for a single frequency. This simplification is made in order to make it easier to interpret the results intuitively and in this case, that single value of frequency refers to the mean of the lower and upper bounds mentioned in equations 3.2.11a and 3.2.11b respectively. Note, the labelling of these series with these values is not precise and the results should be considered with this in mind.

3.2.12. Wavelet Variance

The wavelet variance as it is described here is as described in (Percival and Walden 2006). For the DWT, it can be shown that energy is preserved in the multiresolution analysis, and as such can be reproduced after the decomposition:

$$||x||^2 = \sum_{j=1}^{J} ||d_j||^2 + ||s_J||^2$$

where $||x||^2$ is the energy of signal x.

As the energy is proportional to variance (variance is the energy divided by the length of the series), this reasoning is also true for the variance. However, this relationship does not hold for the MODWT and as such, the detail coefficients must be used rather than the details themselves to calculate the wavelet variance. Although energy is not conserved in the process of calculating the details, it is conserved in the calculation of the detail coefficients:

$$||x||^2 = \sum_{j=1}^{J} ||\tilde{W}_j||^2 + ||\tilde{V}_J||^2$$

where,

- $||x||^2$ is the energy of series x.

In order to calculate the variance at these different scales, the following method is used:

$$\sigma^2(\tilde{W}_j) = \frac{1}{N_j}\sum_{L_j-1}^{N-1}\tilde{W}_j^2$$

$$L_j = (2^j - 1)(L - 1) + 1$$

where,

- L is the length of the wavelet filter used,
- $\tilde{W}_j$ is the detail coefficients at scale j
- j is the index of the decomposition, j=1 being the decomposition at the lowest scale,
- N is the length of the series,
- N_j is calculated as follows:

$$N_j = N - L_j + 1.$$

3.2.13. Implementation

All results of the multiresolution analysis were obtained using functions written in Python. Numpy [6], Pandas [7] and Matplotlib [8] were libraries used to aid the development of the software but the wavelet logic was written without using third party libraries except to verify the results. The functions are attached in the Appendix. The results of these functions were cross checked using three well known wavelet libraries mentioned in section 5.4.

3.3. Autoregressive Model

An autoregressive model (AR) is a model that seeks to explain the value of a time series x_t as the summation of k previous values of that series from x_{t-k} to x_{t-1}, where k is the number of lags taken into account. A different weight is assigned to each of these lags from 1 to k to form a linear model from these lagged values. The notation for an autoregressive model is defined by the number of lags used; AR(k) would be an autoregressive model using k lags. The formula for the model is as follows:

$$x_t = \sum_{i=1}^{k} \beta_i x_{t-i} + \varepsilon_t$$

where,

- x_t is the values of the series at time t,
- β is the weight for a given lag, these weights are called the autoregression coefficients,
- ε_t is the residual at time t.

It is built upon the assumption that the signal x holds memory of its previous values. The residuals ε are assumed to be Gaussian white noise. The signal is assumed to have zero mean, however if that assumption is not made, which is the case for the model used in this dissertation, a constant value is added into the model also.

The problem to be solved is finding the values of β that best fit the series x. One of the ways of estimating these β values is to use unconditional maximum likelihood. This was the method used in this dissertation and is explained well for an ARMA (Autoregressive moving average) model in (Conejo et al. 2005). The basic idea of the method is to transform the model into a probabilistic model and select the coefficients that maximise the likelihood of the data subject to the constraints of the model.

3.3.1. Applying an Autoregressive Model

A vector autoregressive model (VAR) is an autoregressive model which is composed using lags of multiple variables instead of just one. The VAR model used by Tetlock was calibrated for a sampling frequency of 1-day and that makes it slightly different to the model dealt with here as the data in this dissertation is sampled at a 5-minute sampling rate. His model was defined as follows:

$$Dow_t = \alpha_1 + \beta_1 \cdot L5(Dow_t) + \gamma_1 \cdot L5(BdNws_t) + \delta_1 \cdot L5(Vlm_t) + \lambda_1 \cdot Exog_{t-1} + \varepsilon_t$$

where,

- Dow_t is the returns of the Dow Jones index at time t,
- $L5(x)$ is the summation of 5 lags of variable x,
- $BdNws_t$ is the sentiment proxy used, generated from the column "Abreast of the Market" from the Wall Street Journal,
- Vlm_t is the detrended log of the daily volume on the New York Stock exchange,
- $Exog$ includes five lags of the detrended Dow residuals to proxy for past volatility, dummy variables for day-of-the-week and January to control for other potential return anomalies, and a dummy variable for the October 19, 1987 stock market crash to ensure the results were not driven by this single observation.

Given that all of these variables assume daily sampling frequency, the model had to be trimmed down completely in order to get a model that dealt with high frequency data. The model used instead was just a simple 5 lags of returns:

$$r_t = c + \beta_1 r_{t-1} + \beta_2 r_{t-2} + \beta_3 r_{t-3} + \beta_4 r_{t-4} + \beta_5 r_{t-5} + \varepsilon_t$$

where,

- r_t is the returns at time t,
- c is a constant,
- β_n is the weight on lag n,
- ε_t is the residual at time t.

3.3.2. Implementation

The model was implemented using the statsmodels library in Python[9].

3.4. Ordinary Least Squares

Ordinary Least Squares (OLS) is a method of fitting a model to data that aims to minimise the squared difference between the model's approximation of the data and the data itself. It is used by Tetlock to fit the VAR model and also in this dissertation to fit lines of best fit that will be mentioned later.

Suppose a model is defined as follows with N observation and p variables:

$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \varepsilon_i$

This can be represented in vector form as

$y_i = x_i^T \beta + \varepsilon_i$

where,

- y_i is the value of the response variable at observation i,
- x_{ik} is the value of variable k at point observation i,
- β_k is the coefficient of variable k,
- x_i^T is a column vector of length p of all variables in the equation at observation i,
- β is a vector of length p containing the coefficients of all variables.
- ε_i is the residual at observation i.

Using matrix notation, the entire model can be written as:

$y = X\beta + \varepsilon$

where,

- y is a vector of length n containing the value of the response variable for every observation,
- X is a nxp matrix containing the n observations of all p variables,
- β is a vector of length p containing the coefficients of all variables.
- ε is a vector length n containing the residuals for all N observations.

The goal of OLS it to choose the parameters β such that the value of the squared errors $\varepsilon^T\varepsilon$ is minimised. It can be shown that this value of β that minimises the squared errors is obtained with the following equation [1]:

$\beta = (X^T X)^{-1} X^T y$

3.4.1. Implementation

OLS was used to fit the trend lines used in section 4.5. The Numpy [6] Python library was used to calculate the line of best fit via OLS.

3.5. Applications

3.5.1. Investigating the Relationship Between the Wavelet Model and the Residuals of the AR Model

Tetlock showed that using a sentiment proxy he could account for a large portion of the noise in the vector autoregressive model for modelling the returns of the DOW Jones index (Tetlock 2007). This was a very effective way of showing the impact a sentiment proxy can have on the model, but it also shows the importance of noise in the model. There may be many more features of interest contained in the residuals that are unexplored and this makes the residuals series alone one worth investigating. In this section, a model is constructed to compare the AR approach of isolating the residuals to the details produced in a multiresolution analysis.

The multiresolution analysis splits the input series up into its details at various different scales, each one representing the activity of a certain band of frequencies. This model looks to examine which detail represents the residuals most closely. Intuitively, one would think that the highest frequency component contains most of the noise just by looking at the time series.

The details of the S&P 500 return series were extracted by carrying out a multiresolution analysis using the MODWT for 6 decompositions. The least asymmetric Daubechies wavelet of length 8, often referred to as “la8”, was used as the basis function for this analysis due to the fact that it is the basis most widely used in Gençay, Selçuk and Whitcher’s work. The coefficients for the wavelet and scaling filter of that wavelet can be seen in table 3.5.1

The residuals of the AR model of the S&P 500 returns were extracted using the model described in section 3.3. Five lags of returns were used as features in the model and the model was fitted using unconditional maximum likelihood.

In order to compare the details with the residuals, a histogram was plotted comparing the distribution of the residuals with the distribution of the details for each decomposition from one through six. A correlation plot was also plotted to examine the correlation between the residuals and the details. A correlation plot in this case is a plot such that a point on the plot represents a time t. The x-coordinate of that point is given by the z score of the residuals of the AR model at that time t and the y-coordinate of that point is given by the z score of the details, D_j at that time t. Each value of t is represented on the plot.

The flow of this analysis is seen in figure 3.5.1

Table 3.5.1 The coefficients of h and g, the wavelet and scaling filters respectively of the LA8 wavelet.

Index	**h (Wavelet Filter)**	**g (Scaling Filter)**
0	0.032223	-0.075766
1	0.012604	-0.029636
2	-0.09922	0.497619
3	-0.297858	0.803739
4	0.803739	0.297858
5	-0.497619	-0.09922
6	-0.029636	-0.012604
7	0.075766	0.032223

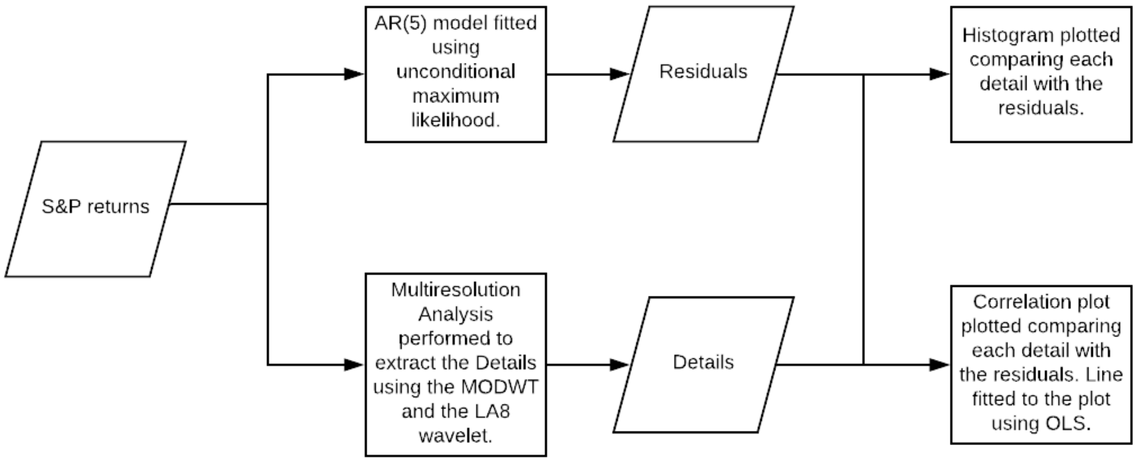


Figure 3.5.1. The flow of the comparison between the residuals and the details

3.5.2. Comparing Risk Across Scales

One of the ways in which multiresolution analysis is used in this dissertation is in analysing the variance of the component series at all of these scales. The variance of the volatility is thought of as a good proxy for risk, this makes it a useful metric to use to compare the behaviour of assets. Using the wavelet variance, the variance of the volatility can be examined across multiple scales which can yield interesting results: see (Gallegati and Gallegati 2007) and (Gençay, Selçuk, and Whitcher 2001a). Bitcoin is considered to be a very risky asset and S&P 500, being a proxy for the market, is considered a safer asset to invest in. This analysis seeks to discover if those considerations are represented in the data.

The wavelet variance was calculated at twelve scales for all three datasets for both returns and volatility. The returns were plotted as well as volatility for comparison, a linear relationship was expected for returns in order to verify that it obeys a linear scaling law.

3.5.2.1. Scaling Laws

The motivation for this work was based on a curiosity in how risky Bitcoin would appear over the long run relative to the more well known, safer assets in Ryanair and

S&P 500. However, it was also influenced by a curiosity about the scaling laws of risk in finance. (Gençay, Selçuk, and Whitcher 2001b) and (Vuorenmaa, T.A., 2005) have shown a break in the scaling law for foreign exchanges and the stock market. As per the CAPM of (Sharpe 1964) and Lintner (Lintner 1965), risk is expected to scale linearly with time. However, Gençay and Vuorenmaa observed that the relationship was not linear and that there was a break in the linear relationship after scales greater than one day. The analysis in this part of the dissertation was carried out in order to find out whether or not the assets analysed display this same property.

In order to test whether or not this variance break occurs, the same methodology that produced figure 7.10 in (Gençay, Selçuk, and Whitcher 2001c) was used. This involves calculating the wavelet variance for multiple scales and fitting a line to the scales of one day or less and fitting a separate line to scales greater than one day on a log scale. This was done using the ordinary least squares technique described in section 3.4 of this dissertation.

4. Results

4.1. Data Used

Table 4.1 Descriptive Statistics for S&P 500, Ryanair and Bitcoin returns and volatility. The darker the shade of green the higher the value relative to other values of that metric in the dataset.

	Mean	Median	Standard Deviation	Min	Max	Kurtosis	Skew
S&P 500 Returns	0.000005	0	0.000838	-0.01759	0.017291	42.28	0.42
Ryanair Returns	-0.000007	0	0.002139	-0.132016	0.059857	641.77	-9.497
Bitcoin Returns	0.000007	0.000031	0.002675	-0.040147	0.055213	19.72	0.049
S&P 500 Volatility	0.000476	0.000274	0.00069	0	0.01759	75.73	6.07
Ryanair Volatility	0.001033	0.000659	0.001873	0	0.132016	1044.55	21.61
Bitcoin Volatility	0.001589	0.000897	0.002152	0	0.055213	35.087	4.26

Shown in table 4.1 is the descriptive statistics of the dataset. Note Bitcoin has the highest standard deviation for both returns and volatility while S&P 500 has the lowest in both returns and volatility. The kurtosis of Ryanair returns and volatility are very high with respect to the other two assets while the absolute skew of Ryanair returns and volatility are high with respect to the other two assets. Ryanair also has the highest max and the lowest min.

4.2. AR Model

The results of the autoregressive model are shown in table 4.2. The 1st, 2nd and third lag are statistically significant at a 95% confidence level. The first and third lags have a negative coefficient.

Table 4.2 Model information of a 5 lagged returns autoregressive model fitted to S&P 500 returns. It can be seen that 3 of the lags proved to be statistically significant.

Variable	**Coefficient Value**	**P value**	**Significant at 95% confidence level**
Constant	0.000	0.310	No
1 Lag Returns	-0.033	0.000	Yes
2 Lag Returns	0.015	0.010	Yes
3 Lag Returns	-0.019	0.001	Yes
4 Lag Returns	0.002	0.766	No
5 Lag Returns	-0.001	0.841	No

The autocorrelation plot is shown in figure 4.2, the autocorrelation plot shows the correlation between the series and its k lags where k is equal to 5 in this case. As can be seen in the figure 4.2, the 1st, 2nd and 3rd lags are statistically significant, similarly to the 1st, 2nd and 3rd lags being statistically significant to the AR model in table 4.2.

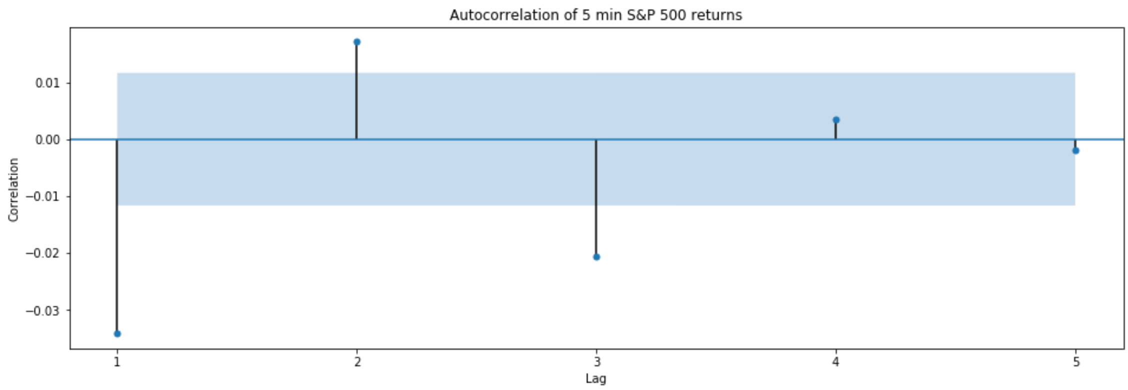
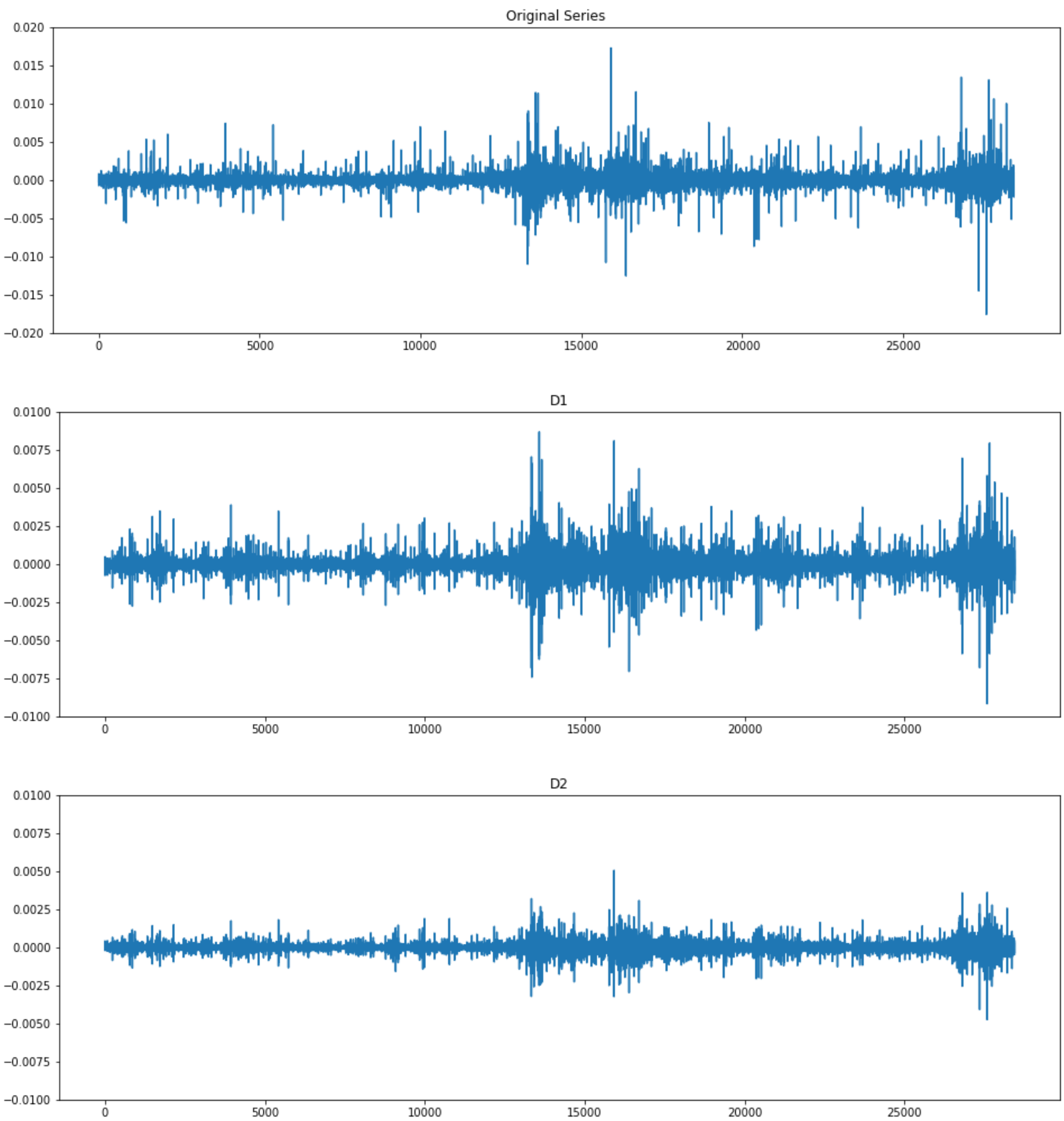


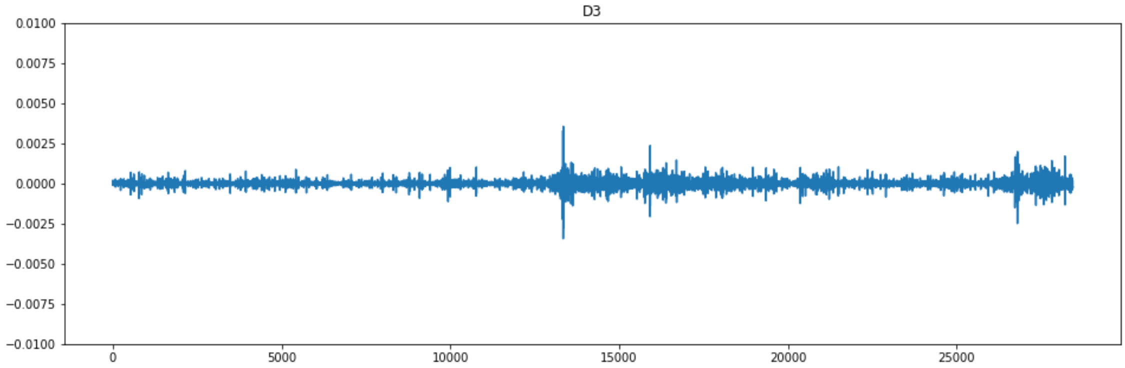
Figure 4.2 Autocorrelation plot of S&P 500 returns. The lags on the x-axis are 5 minutes long corresponding to the 5-minute sampling rate. The dark blue zone shows the bounds outside which the

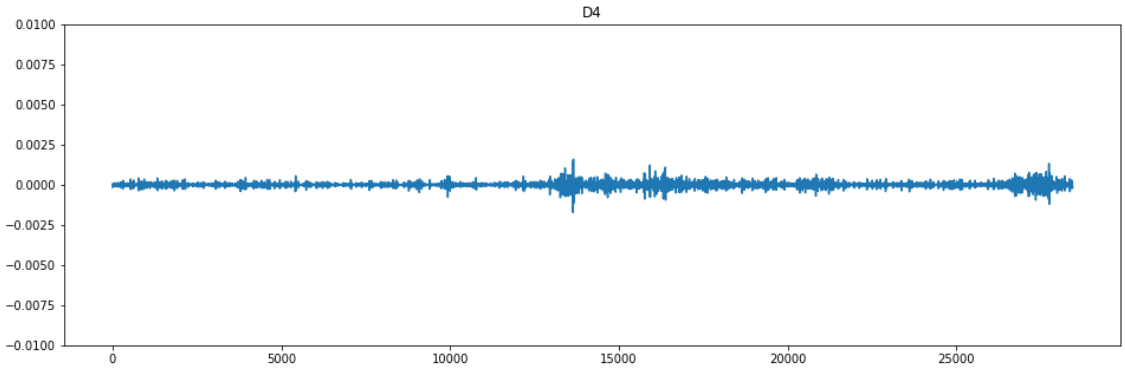
correlation is significant at a 95% confidence level. Similarly, to the results in table 4.2., lags 1, 2 and 3 prove significant.

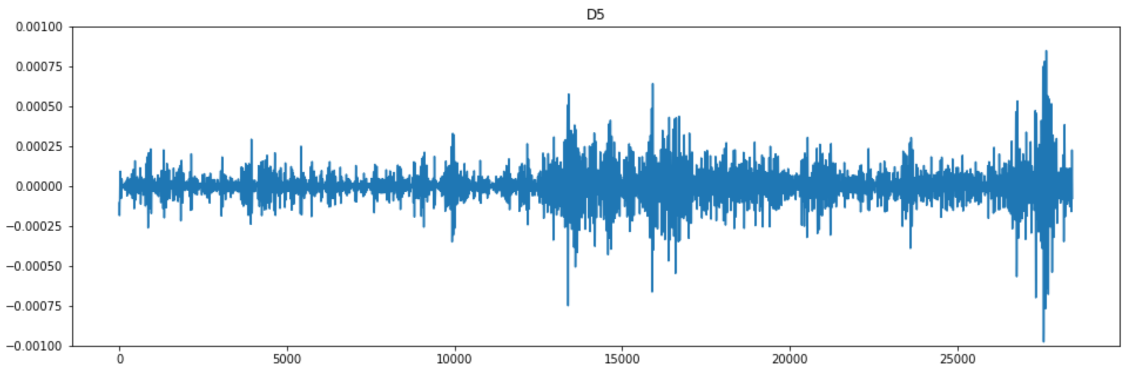
4.3. Wavelet Model

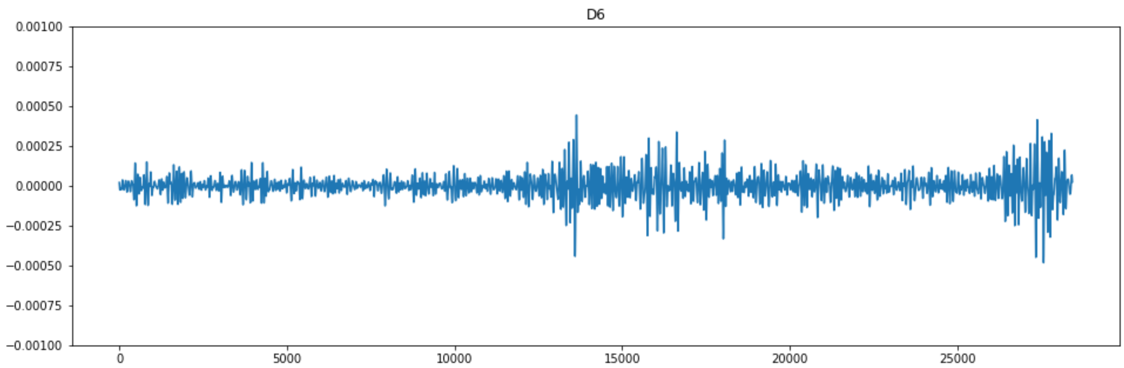
The results of the wavelet decomposition are shown in figure 4.3. Due to the noisy nature of the signal, it is difficult to deduce any information from these graphs alone regarding these components' relationship to the residuals of the AR model.




D3
0.0100
0.0075
0.0050
0.0025
0.0000
−0.0025
−0.0050
−0.0075
−0.0100
0
5000
10000
15000
20000
25000


D4
0.0100
0.0075
0.0050
0.0025
0.0000
−0.0025
−0.0050
−0.0075
−0.0100
0
5000
10000
15000
20000
25000


D5
0.00100
0.00075
0.00050
0.00025
0.00000
−0.00025
−0.00050
−0.00075
−0.00100
0
5000
10000
15000
20000
25000


D6
0.00100
0.00075
0.00050
0.00025
0.00000
−0.00025
−0.00050
−0.00075
−0.00100
0
5000
10000
15000
20000
25000

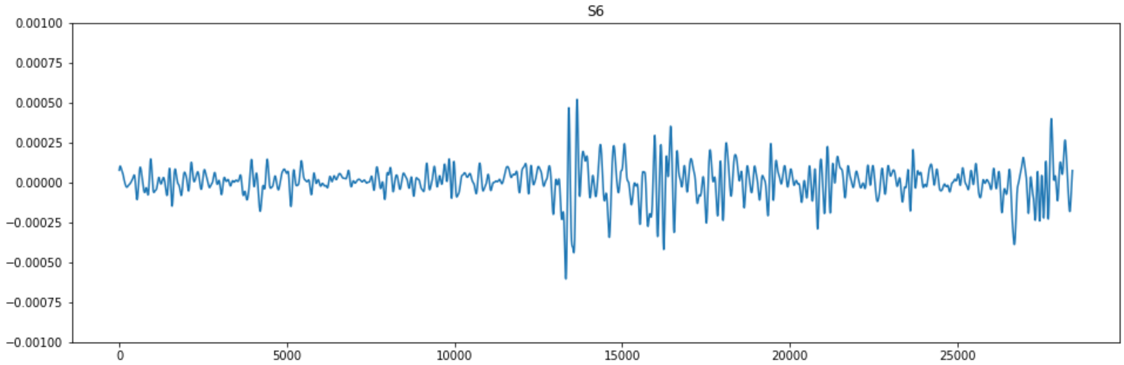


Figure 4.3. The wavelet decomposition of S&P 500 5-minute returns. D1-D6 show the wavelet details and S6 shows the wavelet smooth. On the x-axis is the sample number; this was chosen as the x-axis as opposed to datetimes as there were no values on the weekends and as such there would be jumps in the graph that might be misleading. Note the larger scale on the y-axis for the original series plot relative to the rest of the plots and the larger scale on the y-axis of D1-D4 relative to D5-S6. The decomposition was carried out using the MODWT and the least asymmetric Daubechies wavelet of length 8.

Table 4.3 shows the meaning of each of the decomposition levels. Rough bounds of the periods of the frequencies represented in the details are provided however they are not strict bounds as per section 3.2.11. As can be seen in the table, all of these details contain frequencies of period less than 1 day.

Table 4.3 The corresponding frequency bands for each decomposition number. The days are taken to be 6.5 hours long as that is the length of time that the stock market is most commonly open for (9:30 am until 4:00 pm EST).

Decomposition number (J)	**Lower bound (mins)**	**Upper bound (mins)**	**Lower bound (hours)**	**Upper bound (hours)**	**Lower bound (days)**	**Upper bound (days)**
1	10	20	0.2	0.3	0.0	0.1
2	20	40	0.3	0.7	0.1	0.1
3	40	80	0.7	1.3	0.1	0.2
4	80	160	1.3	2.7	0.2	0.4
5	160	320	2.7	5.3	0.4	0.8
6	320	640	5.3	10.7	0.8	1.6

4.4. Investigating the Relationship Between Wavelet Decomposition and Residuals of the AR Model

In this section, results are presented from the attempted comparison of the details of the multiresolution analysis and the residuals of the AR model. Figures 4.4a, 4.4b and 4.4c show histogram comparisons of the details and the residuals. Based on these histograms and the histograms of the remaining details, it seems that the higher the decomposition, the sharper the peak of the distribution. As is apparent from these visualisations, the histogram of D1 appears to be the most comparable to that of the residuals.

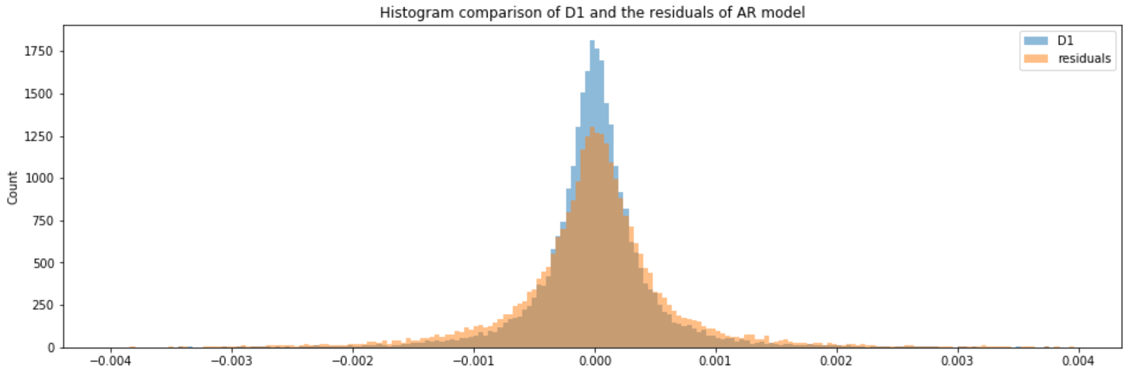


Figure 4.4a A histogram plot comparing the distributions of the first detail and the residuals of the AR model, both extracted from the S&P 500 returns.

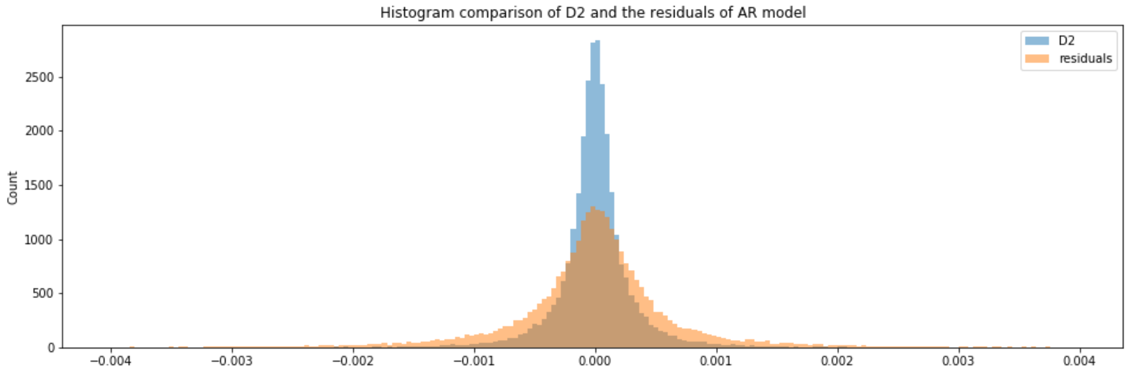


Figure 4.4b. A histogram plot comparing the distributions of the second detail and the residuals of the AR model, both extracted from the S&P 500 returns.

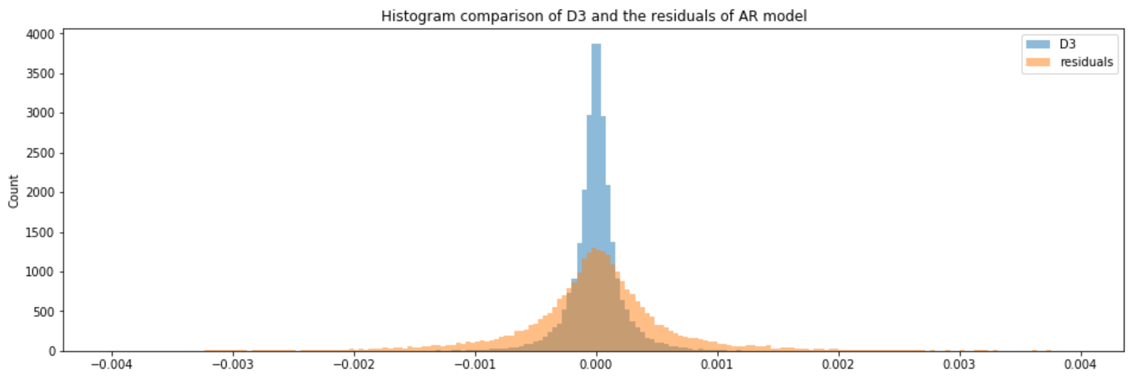


Figure 4.4c A histogram plot comparing the distributions of the third detail and the residuals of the AR model, both extracted from the S&P 500 returns.

The correlation plots of D1, D2, D3 and the residuals are shown in figures 4.4d, 4.4e and 4.4f. The correlation between all three details and the residuals is a positive one with D1 having the highest Pearson's correlation coefficient at 0.75.

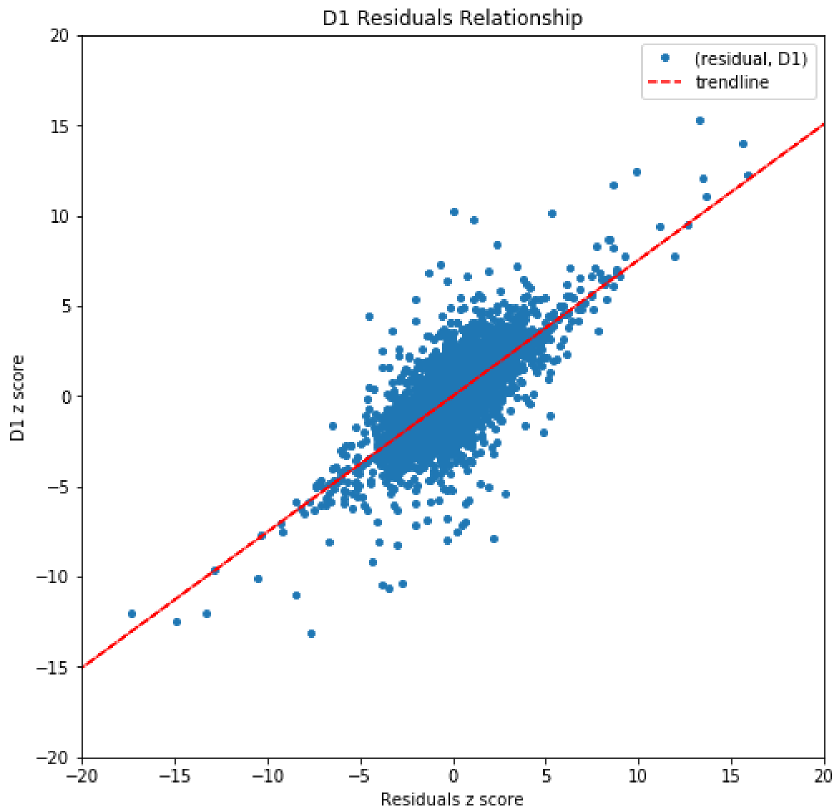


Figure 4.4d. Plot of the z score of the first detail (D1) vs the z score of the residuals for the AR model. A point on this graph is formed by getting the value of the z score for D1 for some time t and mapping it to the value of the z score for the residuals at that same time t. Pearson's correlation coefficient for these two series is 0.75.

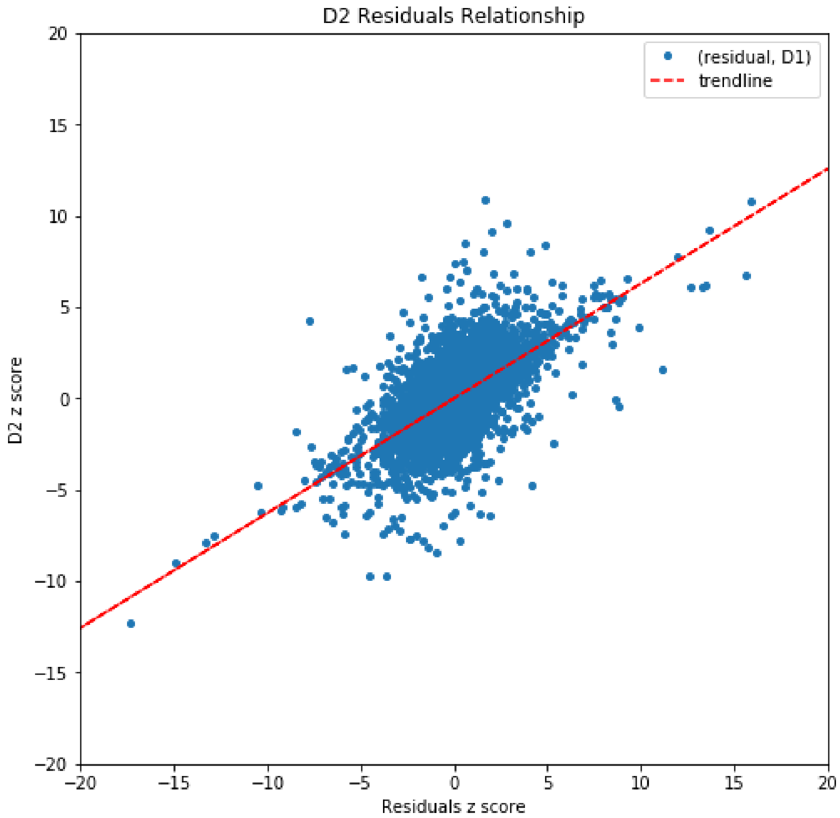


Figure 4.4e. Plot of the z score of the second detail (D2) vs the z score of the residuals for the AR model. A point on this graph is formed by getting the value of the z score for D2 for some time t and mapping it to the value of the z score for the residuals at that same time t. Pearson's correlation coefficient for these two series is 0.63.

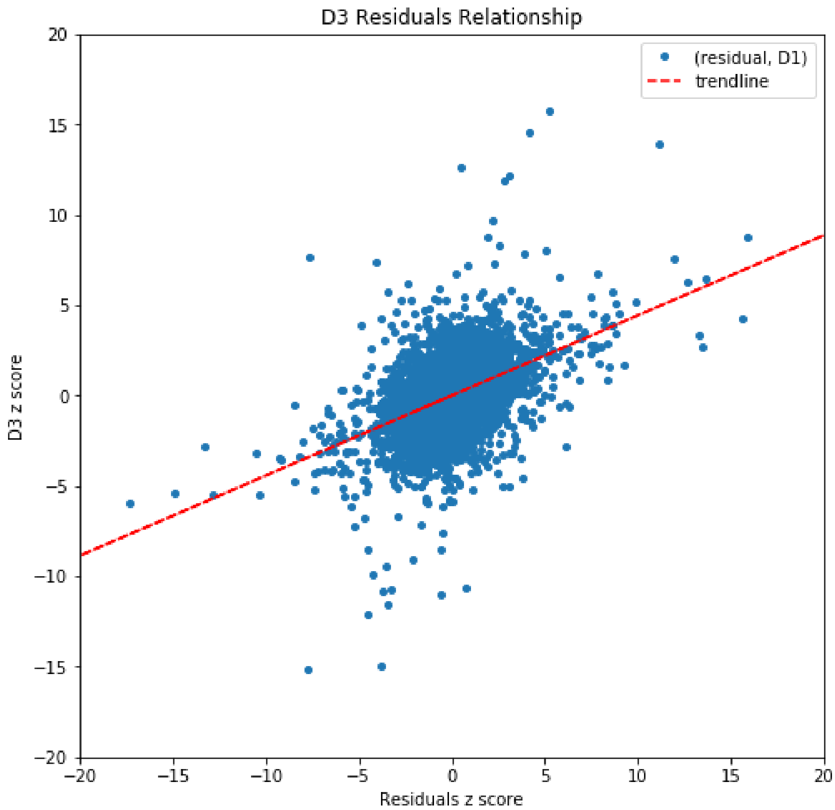


Figure 4.4f. Plot of the z score of the third detail (D3) vs the z score of the residuals for the AR model. A point on this graph is formed by getting the value of the z score for D3 for some time t and mapping it to the value of the z score for the residuals at that same time t. Pearson's correlation coefficient for these two series is 0.44.

4.5. Comparing Risk Across Scales

Figures 4.5a, 4.5b and 4.5c show the plot of wavelet variance of returns against scale. The results are plotted on a log scale. In all three plots a linear relationship is observed.

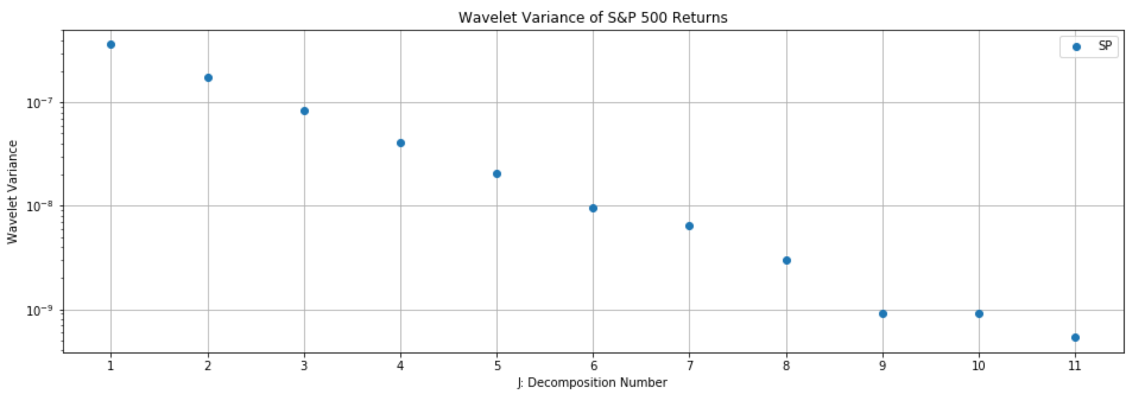


Figure 4.5a Wavelet variance of S&P 500 returns across 11 decompositions.

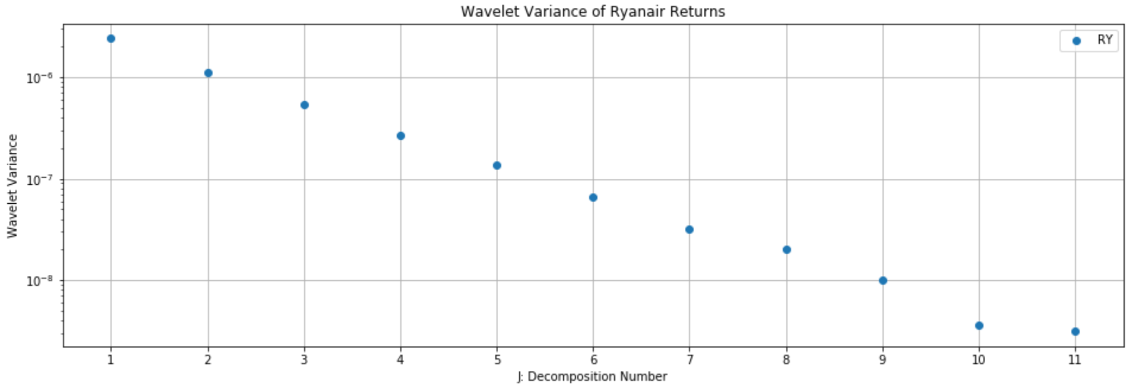


Figure 4.5b Wavelet variance of Ryanair returns across 11 decompositions

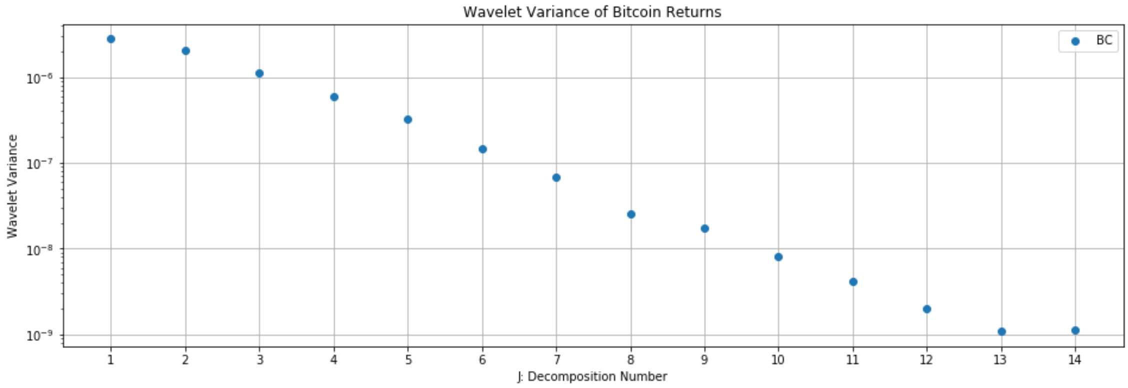


Figure 4.5c Wavelet variance of Bitcoin returns across 11 decompositions

Figures 4.5d, 4.5e and 4.5f show the plots of wavelet variance of volatility against scale. There is a trend line fitted to the intraday frequencies and separately to the frequencies associated with larger periods. All 3 plots show a break in the variance trend after one day. Interestingly, the S&P 500 variance break appears to be the most severe. The variance of the bitcoin volatility series appears to break twice, once at decomposition 8 and then again at decomposition 12.

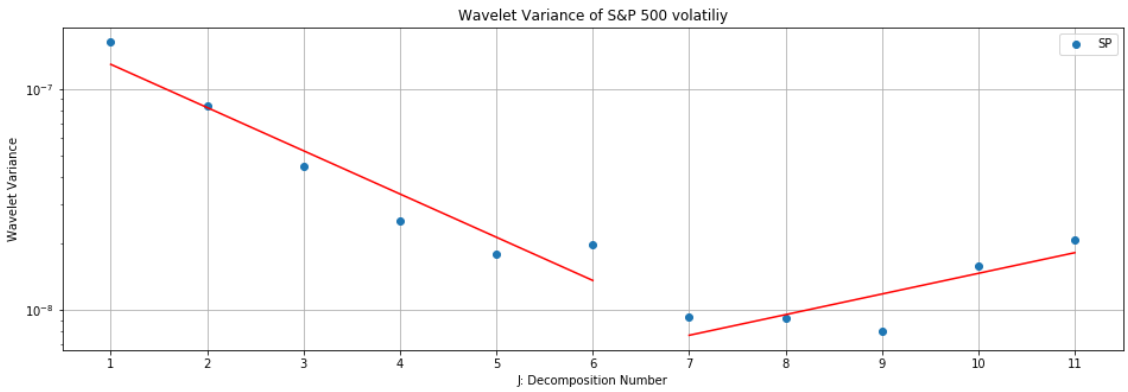


Figure 4.5d Wavelet variance of S&P 500 volatility across 11 decompositions. A trend line was fitted to the first 6 levels and separately to the last 5 levels using Ordinary Least Squares. The

first 6 levels represent activity of frequencies with a period equal to or less than one day, the last 5 represent activity of frequencies with a period greater than one day.

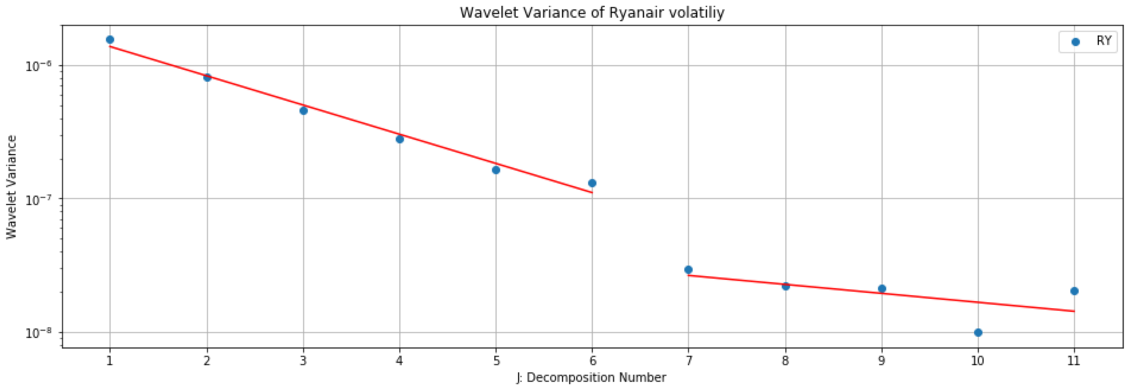


Figure 4.5e Wavelet variance of Ryanair volatility across 11 decompositions. A trend line was fitted to the first 6 levels and separately to the last 5 levels using Ordinary Least Squares. The first 6 levels represent activity of frequencies with a period equal to or less than one day, the last 5 represent activity of frequencies with a period greater than one day.

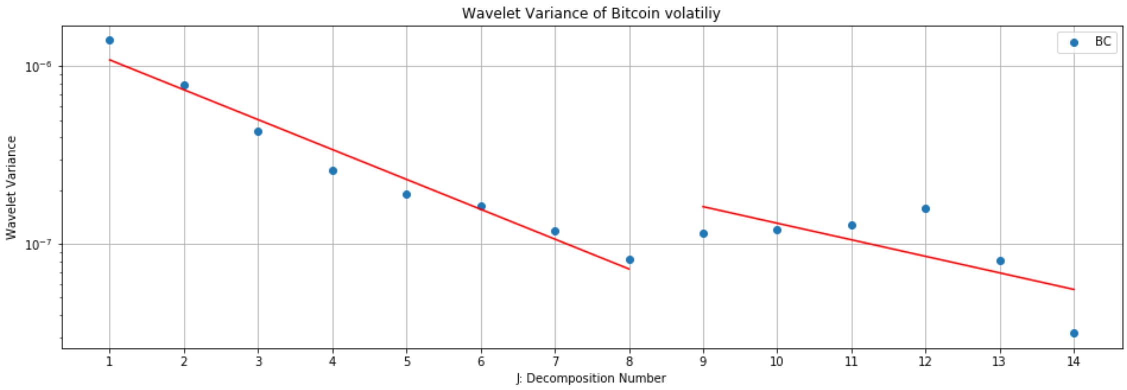


Figure 4.5f Wavelet variance of Bitcoin volatility across 14 decompositions. A trend line was fitted to the first 8 levels and separately to the last 6 levels using Ordinary Least Squares. The first 8 levels represent activity of frequencies with a period equal to or less than one day, the last 6 represent activity of frequencies with a period greater than one day.

Figures 4.5g, 4.5h and 4.5i show a comparison between the wavelet variance of the three series. It is interesting to note that bitcoin has a much higher variance over long time horizons however a smaller variance than Ryanair over horizons less than one day. Although, S&P 500 has the lowest variance across the majority of scales, Ryanair climbs above it for the last two scales. Based on Figure 4.5i, it appears that the second variance breaks of Bitcoin and Ryanair line up with the one-month mark.

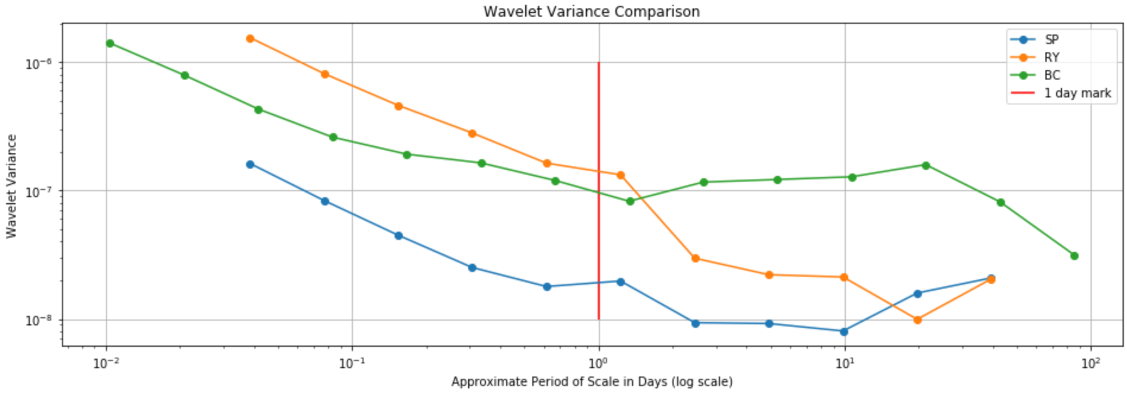


Figure 4.5g Wavelet variance of the volatility of S&P 500, Ryanair and Bitcoin with the x-axis transformed to represent days rather than decompositions. This is to account for the fact that Bitcoin has a 24-hour day as opposed to the other datasets which have 6.5-hour days. A red line shows the one-day mark. The periods given here are approximate periods of the frequencies contained in the details, taken as the average of the lower bound and the upper bound discussed in table 3.2.2.11.

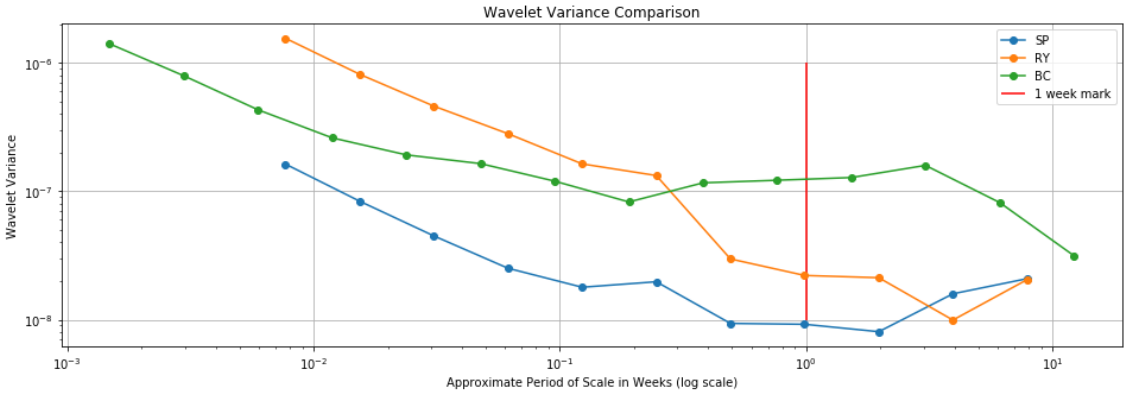


Figure 4.5h Wavelet variance of the volatility of S&P 500, Ryanair and Bitcoin with the x-axis transformed to represent weeks rather than decompositions. This is to account for the fact that Bitcoin has a 7-day week as opposed to the other datasets which have 5-day weeks. A red line shows the one-week mark. The periods given here are approximate periods of the frequencies contained in the details, taken as the average of the lower bound and the upper bound discussed in table 3.2.11.

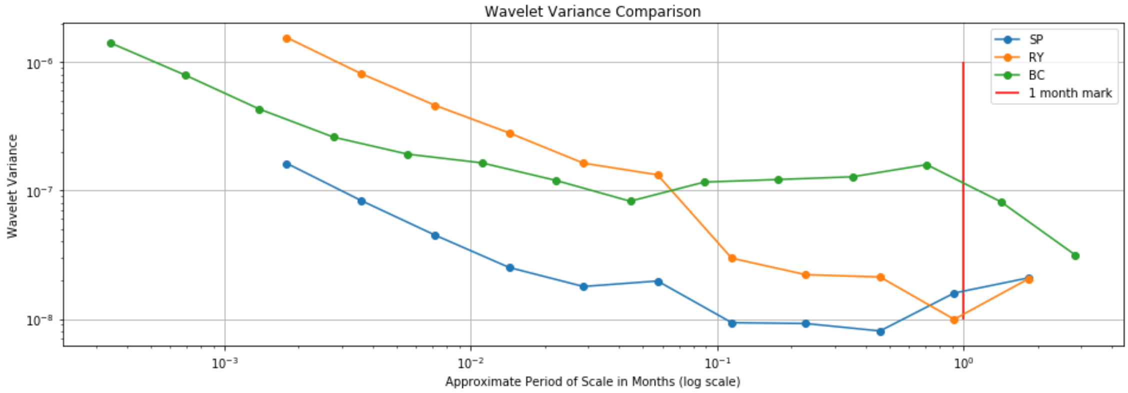


Figure 4.5i Wavelet variance of the volatility of S&P 500, Ryanair and Bitcoin with the x-axis transformed to represent months rather than decompositions. A red line marks the one-month mark. This was done in order to see if the apparent second breaks line up with monthly frequencies. The periods given here are approximate periods of the frequencies contained in the details, taken as the average of the lower bound and the upper bound discussed in section 3.2.11.

5. Discussion

5.1. Data Used

The results of the descriptive statistics fit with what was expected. Bitcoin was expected to vary the most and S&P 500 was expected to vary the least and the standard deviation scores back that up. What is most interesting about these statistics is that Ryanair has by far the highest kurtosis for both returns and volatility, it also has the highest max and the lowest min. The kurtosis in particular suggests that Ryanair has a very high number of outliers. So, although Bitcoin varies more in general, Ryanair produces more extreme values relative to its distribution.

5.2. Investigating the Relationship between Wavelet Decomposition and Residuals of AR model

The generation of the noise in an autoregressive model requires a parametric approach; it requires the user to generate a model which they feel models the input series well. This will generally require domain knowledge and will require the creator of the model to make several judgement calls such as how many lags to take into account and which other variables to include in the equation. For this reason, the results of this method can be affected quite significantly by biases of the creator of the model.

The wavelet approach is a non-parametric approach. It requires no domain knowledge as it is simply a signal transform. It also requires minimal data pre-processing as outliers in the data don't have a huge impact on the model. It is a more general way of breaking down a signal into its component parts.

The residuals of the AR model correlate positively with all details examined. Based on the correlation coefficient and the similarity of the histograms, it seems D1 represents the residuals best. However, the degree to which it represents the residuals is unclear. The fact that there is a relatively strong correlation for all three correlation plots shown suggests that although D1 has the highest correlation, it certainly doesn't provide a complete picture of the noise as a lot of the noise is captured in other details. Due to this uncertainty over the similarity, the results of the investigation are inconclusive.

5.3. Comparing Risk Across Scales

The results of this section provide interesting findings regarding the comparison of the three assets examined. It would make sense after a visual inspection of the trend of Bitcoin since its inception to suspect that Bitcoin is a very volatile series over long periods. The results of this analysis certainly back this up. The figures in the results of section 4.5 show it having a much higher variance of volatility at higher scales than the other two assets. This makes sense as the value of Bitcoin is almost impossible to gauge while Ryanair being a company with many assets will always have its price revert over the long run due to the fact that it is tied to valuable assets.

Having said that, values of the wavelet variance of the volatility of Bitcoin and Ryanair are quite similar over scales of one day or less, much higher than the values for S&P 500. From this we can conclude that although Bitcoin is riskier if you are trading over long horizons, Ryanair and Bitcoin are both very risky at short horizons relative to S&P 500.

It was interesting that the analysis was able to replicate the results of (Gençay, Selçuk, and Whitcher 2001b) in showing that a variance break occurs for all three assets at the one-day mark. This variance break was expected but the fact that the break occurs in the Bitcoin series is of interest as it was hard to expect what its behaviour might look like given how short a lifespan it has so far. In contrast, Figures 4.5a, 4.5b and 4.5c show the wavelet variance of the returns scaling linearly for all three assets.

It was also interesting to see that a second variance break seems to occur at the one-month mark for Bitcoin and Ryanair. However, given this is an engineering dissertation rather than an economics dissertation, the hypothesising as to why that might be will be left to the better educated in this area.

5.4. The Python Implementation

In order to ensure that the Python implementation of the wavelet theory used in this dissertation was consistent with the literature, the results produced were compared with four packages:

1. Waveslim: a wavelet package in R written by Brandon Whitcher,[2]
2. PyWavelets: a wavelet package in Python,[3]
3. The Wavelet Toolbox package provided in the default packages in MATLAB, [4]
4. Gretl was also used to verify the results of the AR model. [5]

These packages allowed for cross examination and verification of the results.

6. Conclusion

It is unclear whether or not there is a strong relationship between the residuals of the AR model and the wavelet model. The fact that AR models of returns series in finance only account for a small percentage of the variance (In Tetlock's VAR model, negative sentiment only explained 1.52% of the residual variation and was the highest of any variable of in the model) means that although they can provide information on what features help contribute to a better model and thus give a better understanding of what effects the market, the residuals largely reflect the data that is used to produce the model. I suspect that is the reason that the residuals correlate with all of the details however I cannot be sure about this suspicion.

The results of the variance analysis are a little more interesting, this method of using multiresolution analysis to investigate the risk of assets across different timescales is an effective one and produced interesting results that confirmed suspicions regarding how risky Bitcoin is over long time periods. It also demonstrated how stable S&P 500 is over short timescales which makes sense given it is an aggregation of multiple different assets.

Both of these applications show the worth of multiresolution analysis in finance. Although the results of the multiresolution analysis can be difficult to interpret, the nonparametric approach means that the model does not get influenced by biases that are inherently incorporated into traditional financial models. This detachment from biases is a feature that is very desirable in a world filled with influences that make it hard to stay objective.

6.1. Future Work

One of the ways this research could be taken forward is to compare more companies using the method proposed in the section 3.5.2 of this dissertation. Given only three assets were examined in this analysis, the comparison was not as comprehensive as it could have been. It would be interesting to see how other cryptocurrencies compare to Bitcoin; are they as risky over long time periods? The answer to that question could provide some interesting insights into the dynamics of these new age assets.

There is certainly room for the wavelet method and the autoregressive method to be compared using a different autoregressive model. The model used in this dissertation was very simple and as such, it perhaps didn't capture the data as well as it might have done. It would be interesting to see if more complex models have a similar relationship to the details of the multiresolution analysis.

7. Bibliography

Black, F. 1986. 'Noise', *Journal of Finance*, 41: 529-43.

Calderón, Alberto. 1964. 'Intermediate spaces and interpolation, the complex method', *Studia Mathematica*, 24: 113-90.

Conejo, A. J., M. A. Plazas, R. Espinola, and A. B. Molina. 2005. 'Day-ahead electricity price forecasting using the wavelet transform and ARIMA models', *IEEE transactions on power systems*, 20: 1035-42.

Cooley, James W., and John W. Tukey. 1965. 'An algorithm for the machine calculation of complex Fourier series', *Mathematics of computation*, 19: 297-301.

Daubechies, I. 1988. 'Orthonormal Bases of Compactly Supported Wavelets', *Communications on Pure and Applied Mathematics*, 41: 909-96.

Gabor, Dennis. 1946. 'Theory of communication. Part 1: The analysis of information', *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93: 429-41.

Gallegati, M. 2008. 'Wavelet analysis of stock returns and aggregate economic activity', *Computational Statistics & Data Analysis*, 52: 3061-74.

Gallegati, M., and M. Gallegati. 2007. 'Wavelet variance analysis of output in G-7 countries', *Studies in Nonlinear Dynamics and Econometrics*, 11.

Gençay, Ramazan, Faruk Selçuk, and Brandon Whitcher. 2005. 'Multiscale systematic risk', *Journal of International Money and Finance*, 24: 55-70.

Gençay, Ramazan, Faruk Selçuk, and Brandon Whitcher. 2001a. 'Differentiating intraday seasonalities through wavelet multi-scaling', *Physica A: Statistical Mechanics and its Applications*, 289: 543-56.

Gençay, Ramazan, Faruk Selçuk, and Brandon Whitcher.. 2001b. 'Scaling properties of foreign exchange volatility', *Physica A: Statistical Mechanics and its Applications*, 289: 249-66.

Gençay, Ramazan, Faruk Selçuk, and Brandon Whitcher.. 2003. 'Systematic risk and timescales', *Quantitative Finance*, 3: 108-16.

Gençay, Ramazan, Faruk Selçuk, and Brandon J Whitcher. 2001c. *An introduction to wavelets and other filtering methods in finance and economics* (Elsevier).

Godfrey, Acrmdog, Richard Conway, David Meagher, and Gearoid ÓLaighin. 2008. 'Direct measurement of human movement by accelerometry', *Medical engineering & physics*, 30: 1364-86.

Grossmann, A., and J. Morlet. 1984. 'Decomposition of Hardy Functions into Square Integrable Wavelets of Constant Shape', *SIAM journal on mathematical analysis*, 15: 723-36.

Gultekin, M. N., and N. B. Gultekin. 1983. 'Stock-Market Seasonality - International Evidence', *Journal of financial economics*, 12: 469-81.

In, F., and S. Kim. 2006. 'The hedge ratio and the empirical relationship between the stock and futures markets: A new approach using wavelet analysis', *Journal of Business*, 79: 799-820.

Jones, R. H. 1980. 'Maximum-Likelihood Fitting of Arma Models to Time-Series with Missing Observations', *Technometrics*, 22: 389-95.

Kim, S., and F. In. 2003. 'The relationship between financial variables and real economic activity: Evidence from spectral and wavelet analyses', *Studies in Nonlinear Dynamics and Econometrics*, 7.

Kim, Sangbae, and Francis In. 2007. 'On the relationship between changes in stock prices and bond yields in the G7 countries: Wavelet analysis', *Journal of International Financial Markets, Institutions and Money*, 17: 167-79.
Lee G, Gommers R, Wasilewski F, Wohlfahrt K, O'Leary A, Nahrstaedt H, and Contributors, "PyWavelets - Wavelet Transforms in Python", 2006-, https://github.com/PyWavelets/pywt [Online; accessed 2019-04-09].
Lintner, J. 1965. 'Security Prices, Risk, and Maximal Gains from Diversification', *Journal of Finance*, 20: 587-616.
Mallat, S. G. 1989. 'A Theory for Multiresolution Signal Decomposition - the Wavelet Representation', *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 11: 674-93.
Percival, D. P. 1995. 'On Estimation of the Wavelet Variance', *Biometrika*, 82: 619-31.
Percival, Donald B, and Andrew T Walden. 2006. *Wavelet methods for time series analysis* (Cambridge university press).
Percival, Donald B., and Harold O. Mofjeld. 1997. 'Analysis of subtidal coastal sea level fluctuations using wavelets', *Journal of the American Statistical Association*, 92: 868-80.
Pinello, Luca. 2011. 'Multi Layer Analysis', *arXiv preprint arXiv:1107.5349*.
Polikar, Robi. 1999. 'The story of wavelets', *Physics and modern topics in mechanical and electrical engineering*: 192-97.
Ramsey, J. 2002. 'Wavelets in economics and finance: Past and future', *Studies in Nonlinear Dynamics and Econometrics*, 6.
Ramsey, J. B. 1999. 'The contribution of wavelets to the analysis of economic and financial data', *Philosophical Transactions of the Royal Society a-Mathematical Physical and Engineering Sciences*, 357: 2593-606.
Ramsey, J. B., and C. Lampart. 1998. 'Decomposition of economic relationships by timescale using wavelets - Money and income', *Macroeconomic dynamics*, 2: 49-71.
Ramsey, J. B., D. Usikov, and G. M. Zaslavsky. 1995. 'An Analysis of Us Stock-Price Behavior Using Wavelets', *Fractals-an Interdisciplinary Journal on the Complex Geometry of Nature*, 3: 377-89.
Ranta, Mikko. 2010. *Wavelet multiresolution analysis of financial time series* (Vaasan yliopisto).
Renaud, Olivier, Jean-Luc Starck, and Fionn Murtagh. 2003. 'Prediction based on a multiscale decomposition', *International Journal of Wavelets, Multiresolution and Information Processing*, 1: 217-32.
Rowland, Todd and Weisstein, Eric W. "Compact Support." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/CompactSupport.html, Date: 9/4/2019.
Selçuk, Faruk, and Ramazan Gençay. 2006. 'Intraday dynamics of stock market returns and volatility', *Physica A: Statistical Mechanics and its Applications*, 367: 375-87.
Sharpe, W. F. 1964. 'Capital-Asset Prices - a Theory of Market Equilibrium under Conditions of Risk', *Journal of Finance*, 19: 425-42.
Tetlock, P. C. 2007. 'Giving content to investor sentiment: The role of media in the stock market', *Journal of Finance*, 62: 1139-68.

Vuorenmaa, T.A., 2005. A wavelet analysis of scaling laws and long-memory in stock market volatility. *Noise and Fluctuations in Econophysics and Finance*, *5848*, pp.39-55.

Website References

[1] https://web.stanford.edu/~mrosenfe/soc_meth_proj3/matrix_OLS_NYU_notes.pdf,

Date accessed: 09/04/2019

[2]https://cran.r-project.org/web/packages/waveslim/index.html

Date accessed: 10/04/2019

[3]https://pywavelets.readthedocs.io/en/latest/

Date accessed: 10/04/2019

[4]https://www.mathworks.com/products/wavelet.html

Date accessed: 10/04/2019

[5]http://gretl.sourceforge.net/mac-intel-quartz.html

Date accessed: 10/04/2019

[6]http://www.numpy.org/

Date accessed: 10/04/2019

[7]https://pandas.pydata.org/

Date accessed: 10/04/2019

[8]https://matplotlib.org/

Date accessed: 10/04/2019

[9]https://www.statsmodels.org/stable/index.html

Date accessed: 10/04/2019

8. Appendix

Imports

pandas as pd

numpy as np

MODWT

```
def modwt(input_series, filter_length=2, j=1,
       g=[1/np.sqrt(2), 1/np.sqrt(2)],
       h=[1/(np.sqrt(2)), -1/(np.sqrt(2))]):
    """
    Function that computes the modwt of the input series
    Default wavelet is the haar wavelet
    Args:
        Input_series: Series modwt will be performed on
        Filter_length: length of wavelet filter basis
        J: decomposition number
        g: scaling coefficients
        h: wavelet coefficients
    returns:
    (cD, cA)
    """
    gtilda = np.array(g)/np.sqrt(2)
    htilda = np.array(h)/np.sqrt(2)

    vj = np.zeros(input_series.shape)
    wj = np.zeros(input_series.shape)

    N = input_series.shape[0]

    for t in range(N):
        k = t
        wj[t] = htilda[0]*input_series[k]
        vj[t] = gtilda[0]*input_series[k]
        for n in range(1, filter_length):
            k = k - 2**(j-1)
            if k < 0:
```

```
                k = k%N
            wj[t] += htilda[n]*input_series[k]
            vj[t] += gtilda[n]*input_series[k]
    return wj, vj
```

IMODWT

```
def imodwt(W, V, filter_length=2, j=1,
        g=[1/np.sqrt(2), 1/np.sqrt(2)],
        h=[1/(np.sqrt(2)), -1/(np.sqrt(2))]):
    '''
    Function that takes in the detail and smooth coefficients and returns the
parent series
    using an imodwt
    Default wavelet is the haar wavelet
    Args:
        W: detail coefficients
        V: Smooth coefficients
        Filter_length: length of wavelet filter basis
        J: decomposition number
        g: scaling coefficients
        h: wavelet coefficients
    Returns:
        Smooth Coefficients of previous decomposition level
    '''
    if len(W) == 0:
        W = np.zeros(V.shape)
    if len(V) == 0:
        V = np.zeros(W.shape)
    v_output = np.zeros(W.shape)
    N = W.shape[0]
    htilda = np.array(h)/np.sqrt(2)
    gtilda = np.array(g)/np.sqrt(2)

    for t in range(N):
        k=t
        v_output[t] = W[k]*htilda[0] + V[k]*gtilda[0]
        for n in range(1, filter_length):
            k += 2**(j-1)
            if k >= N:
                k = k%N
```

```
            v_output[t] += W[k]*htilda[n] + V[k]*gtilda[n]
    return v_output
```

Decomposition stopping at obtaining the detail/smooth coefficients

```
def cmra(input_series, num_decompositions, h, g, L, verbose=False):
    """
    Perform decomposition for coefficients using modwt
    Args:
        input_series: the series to be decomposed
        num_decompositions: The number of times the modwt will be performed
        h: The coeffs of the high pass filter of the wavelet
        g: The coeffs of the low pass filter of the wavelet
        L: The length of the wavelet

    Returns:
        List of the series in order cd1,cd2,...,cdj,caj
    """
    results = []
    cA = input_series
    for i in range(num_decompositions):
        cD, cA = modwt(cA, j=i+1, filter_length=L, h=h, g=g)
        results.append(cD)

    results.append(cA)
    return results
```

Multiresolution Analysis

```
def mra(input_series, num_decompositions, h=None, g=None, L=None):
    """
    Perform mra using modwt
    Args:
        input_series: the series to be decomposed
        num_decompositions: The number of times the modwt will be performed
        h: The coeffs of the high pass filter of the wavelet
        g: The coeffs of the low pass filter of the wavelet
```

```
        L: The length of the wavelet

    Returns:
        Dictionary of "D1, D2,. ..., SN"
    """
    results = {}
    if h == None:
        la8 = pd.read_csv("../datasets/la8.csv", index_col=0)
        h = la8.h
        g = la8.g
        L = 8


    cA = input_series
    for i in range(num_decompositions):
        cD, cA = modwt(cA, j=i+1, filter_length=L, h=h, g=g)
        cD = imodwt(cD, [], j = i+1, filter_length=L, h=h, g=g)
        for j in range(i, 0, -1):
            cD = imodwt([], cD, j = j, filter_length=L, h=h, g=g)
        results["D"+str(i+1)] = cD

    for j in range(num_decompositions, 0, -1):
        cA = imodwt([], cA, j=j, g=g, h=h, filter_length = L)
    results["S"+str(num_decompositions)] = cA

    return results
```

Wavelet Variance

```
def wavelet_variance(series, J, L):
    """
    Computes the wavelet variance for a given series, scale and wavelet length

    Args:
        series: the series of detail/smooth coefficients the variance is being
computed for
        J: the level to which that series has been decomposed by the wavelet
transform
        L: the length of the wavelet used to decompose the series
```

```
    Returns:
        A floating point value; the wavelet variance

    """
    L_j = (2**J - 1)*(L-1) + 1
    s = 0
    N = series.shape[0]
    N_j = N-L_j+1
    for t in range(L_j - 1, N):
        s += series[t]**2
    s = s/N_j
    return s
```

Brick wall function

```
def bw(series, j, L):
    """
    Sets the first n values of the series to nan. This is due to problems
associated with computing the variance using boundary values.

    Args:
        series: The input series
        j: the level to which that series has been decomposed by the MODWT
        L: The length of the wavelet filter used

    Returns:
        The series with the first n values set to nan

    """
    up_to_value = (2**j -1)*(L-1)
    series[:up_to_value] = np.nan
    return series
```

Wavelet Variance for multiple decompositions

```
def complete_wavelet_variance(input_series, num_decompositions, fil="la8"):
    """
    Calculates the wavelet variance for the input series across multiple
scales. Assumes the wavelet coefficients are stored in ../datasets
```

```
Args:
    input_series: series to be decomposed
    Num_decomposition: number of times the series will be decomposed
    fil: Wavelet filter used
Returns:
    Dictionary of D1_var, D2_var....
"""
la8 = pd.read_csv("../datasets/la8.csv", index_col=0)

if fil == "haar":
    haar = pd.read_csv("../datasets/haar.csv", index_col=0)
    h = haar.h
    g = haar.g
    L = 2
elif fil=="la8":
    h = la8.h
    g = la8.g
    L = 8
coeffs = cmra(input_series, num_decompositions, h, g, L)
results = {}
for i in range(num_decompositions):
    bw_i = bw(coeffs[i], i+1, L)

    var = wavelet_variance(bw_i, i+1, L)
    results["D"+str(i+1)+"_var"] = var
return results
```