Wavelet Analysis of Non-Stationary Financial Time-Series Data

Lakksan Kugananthavel
Supervisor : Euan McGonigle

May 29, 2024

Student ID : 30722624

This report is submitted in fulfilment of the requirements of the MMath Mathematics, Faculty of Social Sciences, University of Southampton.

Abstract

We introduce the concept of Wavelets and its applications providing a comparison of its advantages compared to other signal processing methods. We break down the types of Wavelet Transform and plot them for example functions. Discuss different models of time series data and introduce Locally Stationary Wavelet methods to analyse non-stationary time series data. We outline the theory of the Evolutionary Wavelet Spectrum and Localised Autocorrelation. We then adopt a case study format and investigate the S&P 500 and FTSE 100 datasets using these outlined methods. In closing, we discussed the limitations of our model and briefly discussed potential modes of continuation, referencing appropriate resources which provide extensive explanations for them.

Contents

1 Introduction/Background

Signal processing plays a crucial role in modern technology, enabling advancements in communication systems, medical diagnostics and many other fields. It provides the tools and techniques needed to extract meaningful information from raw data, making it a fundamental aspect of various scientific and engineering disciplines. The primary goal of signal processing is to extract useful information, enhance certain features and transform signals through filtering out noise, compressing data, extracting relevant features, and converting signals from one form to another. One method that stands out from the rest is the wavelet method.

Wavelets are short lived oscillations that are localised in time, informally described as ”little waves”. These ”little waves” are specially formulated to be used in representing, analysing and compressing data (such as signals, images or other functions). This idea is not new. Approximation using superposition of functions has existed since the early 1800’s, when Joseph Fourier discovered that he could superpose sines and cosines to represent other functions [Grattan-Guinness, 1990]. However, wavelets take this a step further and offer a more sophisticated frequency-analysis technique. In the history of mathematics, wavelet analysis shows many different origins. Much of the work was performed in the 1930s, and, at the time, the separate efforts did not appear to be parts of a coherent theory. For example, the first wavelet was proposed by Alfréd Haar in 1909 [Haar, 1910] but the study of wavelets, and even the term ”wavelet”, did not come until much later. In fact, the application of wavelets to signal and image processing is recent [Mallat, 1989], but it is without doubt that since its introduction wavelets have revolutionised our approach to signal processing.

2 Theory of wavelets

2.1 What are wavelets?

In order to best understand wavelets let us introduce the concept of a mother wavelet. A mother wavelet is a prototype or a basic waveform that, through dilation and translation, generates a set of ”daughter” wavelets covering a range of scales and positions. These wavelets form bases in which a signal can be decomposed into a wide range of scales. We can define a mother wavelet as any function that satisfies the following conditions:

1. Admissibility Condition:

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

   This condition states that the average value of the function must be zero.

2. Finite energy condition:

$$\int_{-\infty}^{\infty} |\psi|^2 dt < \infty$$

   This condition states that the total energy of the function must be finite

As all wavelets are generated from a mother wavelet these conditions formally describe all wavelets. There are many functions that satisfy these conditions and so there are many different types of mother wavelet.

2.2 Why do we use wavelets

In order to best understand the usefulness of wavelets let us take a look at another popular signal processing method Fourier analysis. Fourier analysis is based around the observation that a target function/data can be described as a sum of trigonometric functions, the data has an "alter ego" in the frequency domain if you will. We can find the specific series of sines and cosines that describe the target function using a mathematical operation known as a Fourier transform. We can use a Fourier transform to decompose the target function into a series of trigonometric functions, transforming our function into a frequency domain. We then observe the relative contributions of each frequency to the data (capturing frequency information) and reconstruct the function via an inverse Fourier transform. However, in transforming from the time domain into the frequency domain we lose all information about time. This makes it ill-suited for non-stationary signals for which the frequency changes over time.

Heisenbergs Principle of Uncertainty states that it is fundamentally impossible to perfectly capture both time and frequency resolution simultaneously [Vošvrda and Schurrer, 2015]. However, if we allow ourselves to sacrifice slightly on the preciseness of our resolution we can compromise and obtain a good approximation of both. This is key idea of wavelet analysis.

Advantages of Wavelet Analysis:

1. Fourier Analysis provides a global representation of a signal's frequency content. Doesn't offer localized information in the time domain, less effective for signals with non-stationary or time-varying characteristics. Wavelet transform excels in providing both time and frequency localisation. Allows for analysis of signal components at different scales, more suitable for capturing local features and transients.

2. Fourier Analysis is not well-suited for capturing transient events or abrupt changes in a signal. Tends to smear out information over the entire time domain. Wavelet Analysis efficiently represents transients, making it valu-

able in applications where the focus is on identifying specific events or anomalies in the data.

3. Sparsity of wavelets: A sparse representation means that the majority of coefficients in the wavelet transform are zero or close to zero. This means that a signal can be efficiently represented using only a small number of non-zero wavelet coefficients. The sparsity of wavelets makes them particularly suitable for tasks such as signal compression, denoising, and feature extraction, where an efficient representation of essential information is crucial. Providing reduced computational and storage complexity.

It should be noted however that wavelets are not always the superior method and that, depending on the function, a Fourier analysis may be more appropriate.

2.3 Types of wavelet

There are many different types of mother wavelet let us provide a basic introduction to a few of the most commonly used and important examples.

2.3.1 The Haar Wavelets

The simplest wavelet which serves as an excellent introduction to the topic. The Haar wavelet is a sequence of rescaled ”square-shaped” functions proposed in 1909 by Alfréd Haar [Haar, 1910]. The mother wavelet has form:

$$\psi(t) = \begin{cases} 1 & \text{if } 0 \leq t < \frac{1}{2}, \\ -1 & \text{if } \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

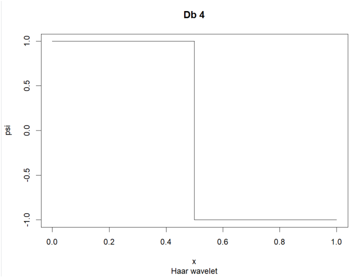


Figure 1: The Haar Mother Wavelet

Haar wavelet approximations widely used for obtaining numerical solutions of ordinary differential equations (ODEs) and fractional differential equations [Ülo Lepik and Hein, 2014]. Due to its simplicity it also has effective applications in signal and image compression as it provides a simple and computationally efficient approach for analysing the local aspects of a signal.

2.3.2 Morlet Wavelet

The Morlet wavelet is a complex valued wavelet that is Gaussian in both time and frequency. The Morlet wavelet is typically defined as the product of a complex sine wave and a Gaussian window [Cohen, 2018]

$$\psi(t) = e^{2i\pi ft} e^{\frac{-t^2}{2\sigma^2}}$$

where i is the imaginary operator ($i = \sqrt{-1}$), f is frequency in Hz, and t is time in seconds. σ is the width of the Gaussian, which is defined as

$$\sigma = \frac{n}{2\pi f}$$

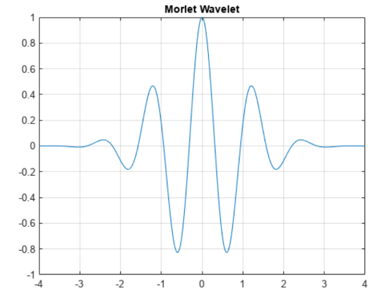


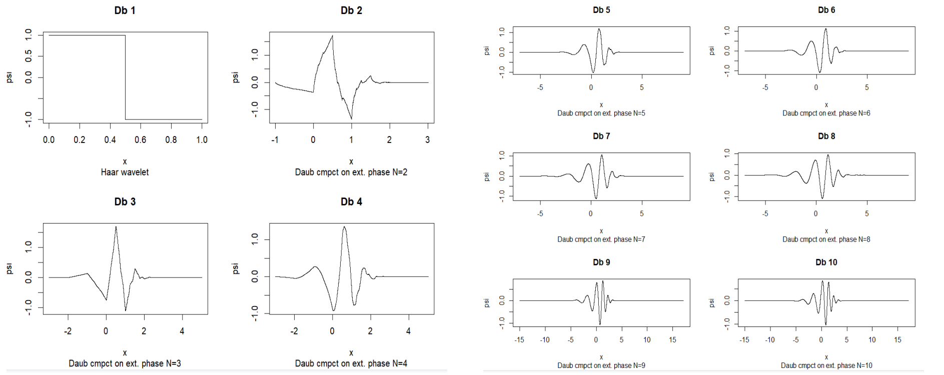
Figure 2: The Morlet Mother Wavelet

This wavelet has extensive applications in medicine specifically in the field of neuroscience for the analysis of brain signals, such as an Electroencephalogram (EEG) and a Magnetoencephalogram (MEG) [Bajaj, 2020].

2.3.3 Daubechies wavelet

Daubechies wavelets, named after the Belgian mathematician Ingrid Daubechies, are a family of orthogonal wavelets with compact support. The Daubechies wavelet family includes several members each characterized by a specific number of vanishing moments and approximation order. The choice of a particular Daubechies wavelet depends on the requirements of the specific application. This particular family of wavelets is very complex and a closed form representation of Daubechies wavelets does not exist so the following will provide a gentle overview of the topic but should by no means be considered comprehensive. Each Daubechies family wavelet is distinguished by an index number N. The index number refers to the number N of coefficients. Each wavelet has a number of zero moments or vanishing moments equal to half the number of coefficients $N/2$ [Walker, 2008].

We can use the R command



(a) Plot of Daubechies wavelets for $N = 1$ to $N = 4$

(b) Plot of Daubechies wavelets for $N = 5$ to $N = 10$

Figure 3: Family of Daubechies-N wavelets for $N = 1$ to $N = 10$

```
draw.default(filter.number=i, family="DaubExPhase", enhance=F, main=paste("Db", i))
```

to produce a plot of the family of Daubechies wavelets for $N = 1$ to $N = 10$ (Figure 3).

Notice how the Daubechies wavelet with $N = 1$ is the Haar wavelet. This is because the Daubechies wavelet transforms are defined in the same way as the Haar wavelet transform. They both work by computing running averages and differences via scalar products with scaling signals and wavelets. The only difference between them is in how these scaling signals and wavelets are defined. Also notice how he support of the functions increases with their index. This is because as you increase N, the Daubechies wavelet function becomes more oscillatory, allowing it to capture more complex features in the signal. As a consequence, the effective support of the wavelet function expands, encompassing a larger interval where it has significant values. This comes with a trade-off, as the increased frequency localization may lead to a broader time localisation.

Daubechies wavelets are a very popular family of wavelets that perform very well in many practical applications: in civil engineering ([Ma et al., 2003]), in power systems engineering ([Brito et al., 1998]), in biomedical engineering ([Mahmoodabadi et al., 2005]), and just generally in signal and image processing these wavelets are very powerful.

3 Wavelet Transform

3.1 Overview of Wavelet Transform

The wavelet transform uses a range of daughter wavelets generated from a mother wavelet to analyse a signal/function. A daughter wavelet can be generated from a mother wavelet by:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

$a, b \in R \quad a \neq 0$

Where b is a translation variable and a is a scaling variable. Varying the a and b variables will generate different daughter wavelets.

The father wavelet is associated with the scaling function in wavelet analysis and is generated from the mother wavelet. The scaling function represents the coarsest level of approximation in the multiresolution analysis. It spans a subspace that captures the low-frequency components of a signal. The father wavelet undergoes dilation and translation to generate a set of basis functions at different scales and positions. These basis functions are used to construct approximation spaces that represent the low-frequency content of the signal. The father and mother wavelets work in tandem to decompose a signal into different scales or resolutions. The father wavelet captures the coarse, low-frequency information of a signal, while the mother wavelet captures the fine-scale, high-frequency details, allowing for a multi-scale representation of the signal.

When we compute Fourier transforms we use trigonometric and exponential analysing functions to transform our signal from a 1 dimensional time domain $f(t)$ into a 1 dimensional frequency domain $f(\hat{w})$. In our function $f(\hat{w})$ we can only calculate the contribution of that frequency w to the signal, specifically we extract our frequency information from the signal using a Fourier transform.

The key difference between the wavelet transform that makes it unique compared to other methods is that it converts $f(t)$ into a 2 dimensional time-frequency surface $F(a, b)$.

The admissibility condition for the wavelet transform to hold is:

$$C = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(w)|^2}{|w|}\, dw < \infty \tag{2}$$

Where $\hat{\psi}(w)$ is the Fourier transform of $\psi(w)$ defined by

$$\hat{\psi}_{a,b} = \frac{1}{\sqrt{|a|}} e^{-ibw} \hat{\psi}(aw) \tag{3}$$

Changes in the time t will translate the mother wavelet along the time axis so varying b will generate a daughter wavelet $\psi_b = \psi(t-b)$. Similarly changing the frequency f will dilate (either shrink or stretch) the mother wavelet along the time axis, therefore varying a will generate a daughter wavelet $\psi_a = \psi(t/a)$. Putting this together we see that the value of our wavelet transform at a scale a and time b is equal to the contribution of our scaled and translated ”daughter” wavelet $\psi_{a,b} = \psi((t-b)/a)$ to our signal. The value of $F(a,b)$ will calculate the contribution of this daughter wavelet $\psi_{a,b}$ to our signal by the convolution of the two functions $< \psi_{a,b}, f >$. We vary our parameters a, b and repeat our convolution procedure to analyse our signal with daughter wavelets of different scales so that we can see what frequencies are prominent at a time point b. We refer to the generated $F(a,b)$ values as wavelet coefficients.

3.2 Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) calculates the wavelet coefficients at every possible scale.

$$CWT(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) \hat{\psi} \left(\frac{t-b}{a} \right) dt \tag{4}$$

The value $CWT(a,b)$ represents the wavelet coefficient at scale a and position b.

Let us compute the Continuous Wavelet Transform of a simple piece-wise polynomial (polynomial taken from [Nason, 2008])

$$y(x) = \begin{cases} 4x^2(3-4x) & 0 \leq x < 0.5 \\ \frac{4x(4x^2-10x+7)}{3} - 1.5 & 0.5 \leq x < 0.75 \\ \frac{16x(x-1)^2}{3} & 0.75 \leq x \leq 1 \end{cases}$$

We can compute the CWT of this function using the RWave package in R. We use the *cwt* command which computes the continuous wavelet transform of the polynomial with the (complex-valued) Morlet wavelet.

```
# Generate x values
x_values <- seq(0, 1, length.out = 512)

# Evaluate the function for each x value
y_values <- sapply(x_values, function_with_jump)
```

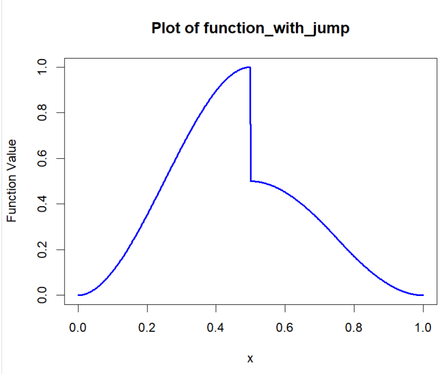
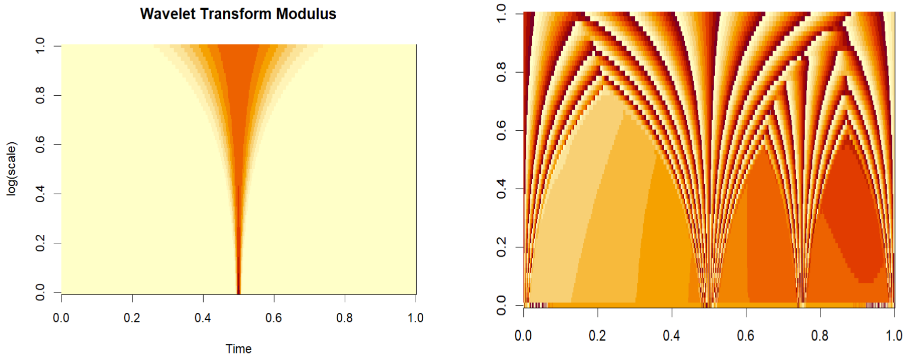


Figure 4: Plot of the piecewise polynomial $y(x)$ sampled at 512 equally spaced locations

```
cwt1 <- cwt(y_values, 5, 12)
image(Arg(cwt1))
```

Running the code generates the plots shown in figure 5. The x-axis represents positions along the signal (time), the y-axis represents scale (frequency), and the color at each x-y point represents the magnitude of the wavelet coefficient.

Clearly, these plots can be a struggle to make sense of even for a simple function like $y(x)$. This example demonstrates a major limitation of the CWT in that it generates a lot of data which makes it both difficult to analyse and possibly very computationally demanding for larger data sets. A much more efficient and commonly used alternative is the Discrete Wavelet Transform (DWT).



(a) Image plot of the modulus of the (complex valued) continuous wavelets coefficients

(b) Phase of the (complex valued) continuous wavelets coefficients

Figure 5: Time-scale view of $y(t)$ using a Continuous Wavelet Transform

3.3 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) is an alternative to the Continuous Wavelet Transform in which scales a and position b are chosen to be discrete. The most efficient framework for DWT is Multiresolution Analysis.

3.3.1 Multiresolution Analysis

Multiresolution Analysis (MRA) is a fundamental concept in wavelet analysis put into framework by [Mallat, 1989], providing a systematic way to decompose a signal into different scales or resolutions. The key idea is to represent a signal at various levels of detail, capturing both high and low-frequency components. This is particularly useful in understanding the behavior of a signal across different frequency bands.

Mathematically a Multiresolution Analysis on $\mathbb{R}$ is a sequence of subspaces $\{V_j\}_{j\in\mathbb{Z}} \subseteq L^2(\mathbb{R})$ satisfying:

(a) For all $j \in \mathbb{Z}$, $V_j \subseteq V_{j+1}$.

(b) $\text{span}\{V_j\}_{j\in\mathbb{Z}} = L^2(\mathbb{R})$. That is, given $f \in L^2(\mathbb{R})$ and $\epsilon > 0$, there is a $j \in \mathbb{Z}$ and a function $g(x) \in V_j$ such that $\|f - g\|_2 < \epsilon$.

(c) $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$.

(d) A function $f(x) \in V_0$ if and only if $D^{2j} f(x) \in V_j$.

(e) There exists a function $\phi(x)$, L^2 on $\mathbb{R}$, called the scaling function, such that the collection $\{T_n\phi(x)\}$ is an orthonormal system of translates and

$V_0 = \text{span}\{T_n\phi(x)\}.$

This is a lot to unpack so let us approach this intuitively. In MRA scales and positions are chosen based on powers of two, dyadic scales and positions. This is an efficient and accurate method of choosing a and b. In particular, if $a_0 = 2$ and $b_0 = 1$ then there exist $\psi(t)$ with good time-frequency localisation properties such that the set:

$$\{\psi_{k,n}(t) = 2^{-k/2}\psi(2^{-k}t - n) : k, n \in \mathbb{Z}\} \tag{5}$$

constitutes a countable complete orthonormal wavelet basis in $L^2(\mathbb{R})$. Now can define the equation for the Discrete Wavelet Transform of a signal $f(t)$ as the convolution $< f, \psi_{m,n} >$. From this point we begin a recursive decomposition of the signal which can be described as follows

1. We begin by choosing a scaling function/father wavelet $\phi(t)$ which spans a subspace capturing the low-frequency components of the signal. This function represents the coarsest approximation.

2. We choose an appropriate mother wavelet $\psi(t)$ to analyse the signal and extract the high frequency components over the signal.

3. The subspace is iteratively halved, effectively 'shrinking the window' for signal analysis, and we generate a family of wavelets from scaling and translating the father and mother wavelets to extract the frequency content over each subspace. This process is repeated until the desired level of detail is achieved.

The recursive decomposition results in a multiresolution pyramid, where each level corresponds to a different scale. The bottom level represents the coarsest approximation, and the higher levels capture finer details.

Figure (6) shows a visual representation of the essence of multiresolution analysis. A time series analysis shown in the leftmost image, the lines represent that this method has excellent time resolution as we know the value of the signal at every point in time t but no frequency information. The middle image shows an image of Fourier analysis which has excellent frequency resolution but no time information. The wavelet method is shown in the rightmost image which shows how the same signal is represented at different scales. The bottom level has good frequency resolution and as we go up through the levels we pull out frequency content and attain good localised time resolution.

3.3.2 Applying DWT on a signal

Practically applying the DWT on a signal involves the use of a series of filters which generate two sets of coefficients at each level. At the first level, half

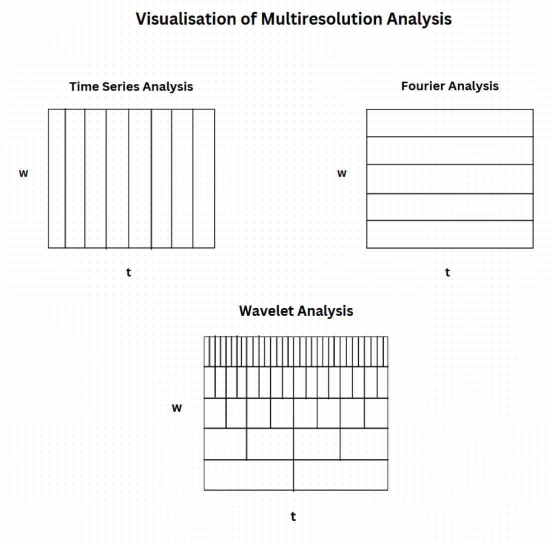


Figure 6: A visualisation of the method of multiresolution analysis produced in Canva [Can, 2024]

the frequencies of the signal are passed through a high pass filter $h[n]$ which give detailed coefficients that capture the high frequency data of the signal. Simultaneously, the other half of the frequencies of the signal are passed through a low pass filter $g[n]$ which generates approximation coefficients that characterize the low frequency components of the signal. The approximation coefficients provide a more generalized view of the signal's trends without capturing very rapid changes. The high pass $h[n]$ and low pass $g[n]$ filters are designed using the mother wavelet. We choose a mother wavelet based on how well it characterizes the signal, the specific choice of mother wavelet is crucial for the accuracy of our transform.

$$f_{low}[n] = (f * g)[n] = \sum_{k=-\infty}^{\infty} f[k]g[n-k]$$

$$f_{high}[n] = (f * h)[n] = \sum_{k=-\infty}^{\infty} f[k]h[n-k]$$

At the first level we have decomposed our signal by frequency into two halves captured by our detailed and approximation coefficients which we can reference as $d_{a,b}$ and $c_{a,b}$ respectively.

Since half the frequencies of the signal have now been removed, half the samples can be discarded according to Nyquist-Shannon sampling theorem (the

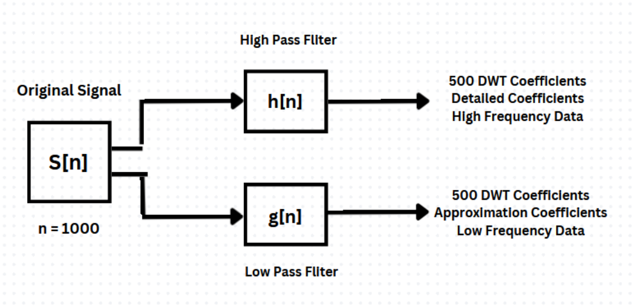


Figure 7: A visualisation of the first level of decomposition produced in Canva. Here S[n] represents a signal of input size n = 1000

sampling rate must be at least twice the highest frequency present in the signal [Shannon, 1948]).So, to further process the signal we subsample the filter output of the low-pass filter (i.e. the approximation coefficients $c_{a,b}$) by a factor of 2 and pass it through a new set of high-pass and low-pass filters with exactly half the cut of frequency of the previous ones.

$$f_{low}[n] = (f * g)[n] = \sum_{k=-\infty}^{\infty} x[k]g[n-2k]$$

$$f_{high}[n] = (f * h)[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-2k]$$

The decomposition halves the time resolution. The frequency resolution is doubled because the original frequency range is now divided into two distinct bands, allowing for a more detailed representation of the frequency content.

The low-pass filter provides a smoothed approximation of the signal (halving time resolution), the high-pass filter focuses on capturing high-frequency details (doubling frequency resolution). The combination of both filter outputs allows for a Multiresolution Analysis of the signal at different scales.

This decomposition is repeated, the approximation coefficients decomposed with high-pass and low-pass filters and then down-sampled, with the frequency resolution resolution improving at each level. We can continue this process until we are satisfied with the level of detail or until we reach the maximum number of levels $log_2 N$ where N is the size of our input signal. This process is shown visually through a filter bank below.

Using the *wavethresh* package we apply the Discrete Wavelet Transform on our simple polynomial from Figure 1 generating the wavelets coefficients which we plot onto a wavelet decomposition plot (Figure 9). We choose our mother wavelet to be the Daubechies Wavelet.

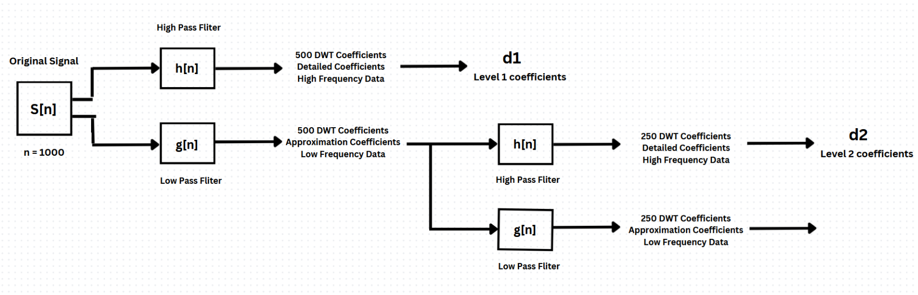
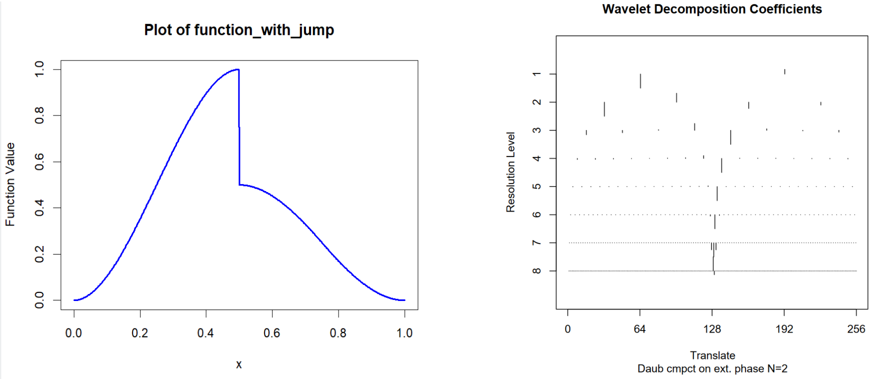


Figure 8: Filter bank showing the level by level decomposition of the signal



(a) Plot of the piece-wise polynomial $y(x)$ sampled at 512 equally spaced locations

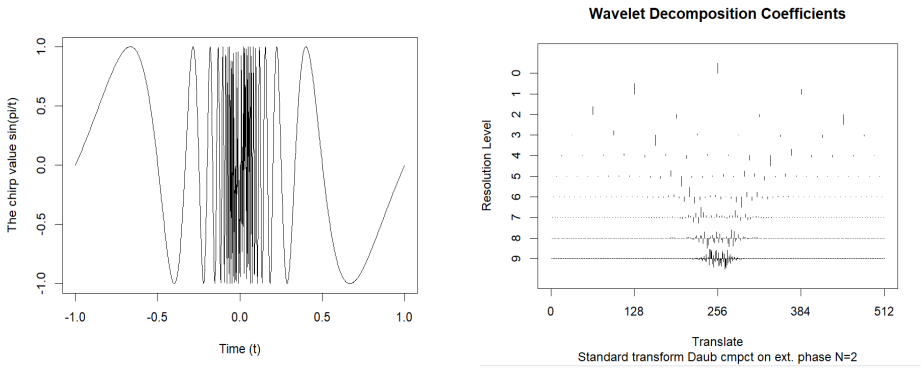
(b) The wavelet decomposition plot corresponding to the wavelet coefficients of the DWT of $y(t)$

Figure 9: Wavelet coefficients (right) of our simple piecewise polynomial (left)

```
wsd1 <- wd(y_values,family = "DaubExPhase")
plot(wsd1, scaling = "by.level")
```

Already it is clear that this is a much simpler representation of our signal than figure 5. The decomposition plot is an excellent illustration of the MRA framework we described previously. The higher resolution wavelet coefficients are shown in the lowest level of the decomposition plot and capture the high frequency components of $y(t)$. The lower resolution coefficients are shown in the lowest level capturing the low frequency components of $y(t)$. With this in mind we can easily interpret the decomposition plot and see that the "jump" at $x = 0.5$ is represented at the lowest level while the smooth increases and decreases in $[0, 1/2)$ and $[1/2, 1]$ of the function are represented in the highest level.

Let us inspect the power of the Discrete Wavelet Transform using another more substantial function (reproduced from [Nason, 2008]). Defining the chirp func-



(a) A plot of the chirp function against time t

(b) A wavelet decomposition plot of the chirp function (DWT)

Figure 10: The chirp function (left) and its Discrete Wavelet Transform (right)

tion by

$$y = \sin\left(\frac{\pi}{x}\right) \tag{6}$$

The nature of the signal can be clearly identified from the decomposition plot. The chirp function is characterized by violent oscillations around the midpoint of the signal. The decomposition plot shows the oscillations to be centred around the midpoint, showing that the Discrete Wavelet Transform extracts these high frequency oscillations at different scales while still maintaining a good image of their time localisation.

3.4 Non-decimated wavelets and the Stationary Wavelet Transform (SWT)

The Stationary Wavelet Transform, also referred to as the Undecimated Wavelet Transform, is an adapted form of the Discrete Wavelet Transform that maintains the entire signal length throughout the analysis. The Stationary Wavelet Transform avoids the traditional DWT approach of downsizing the signal at each level of decomposition, resulting in a shift-invariant wavelet transform that preserves the entire signal length throughout the analysis. This is shown clearly in figure 11.

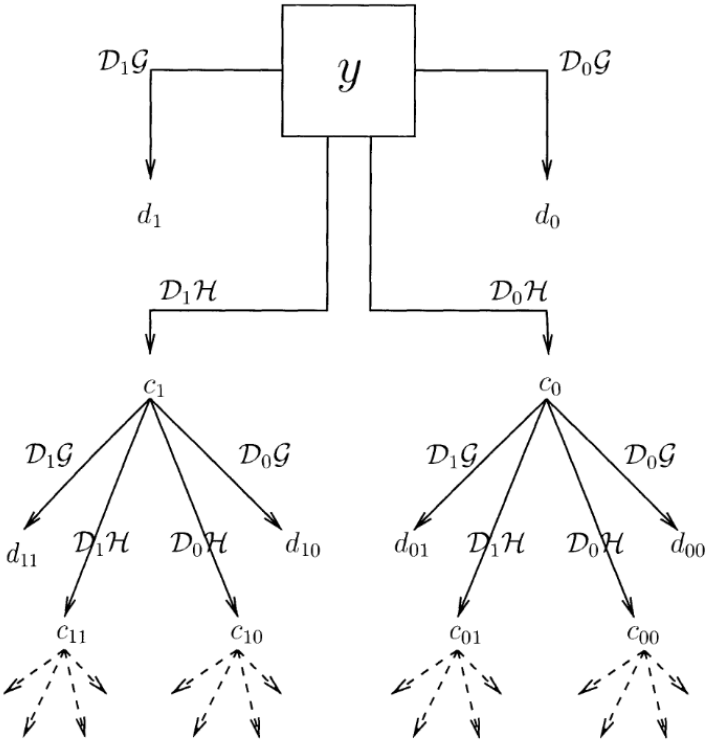
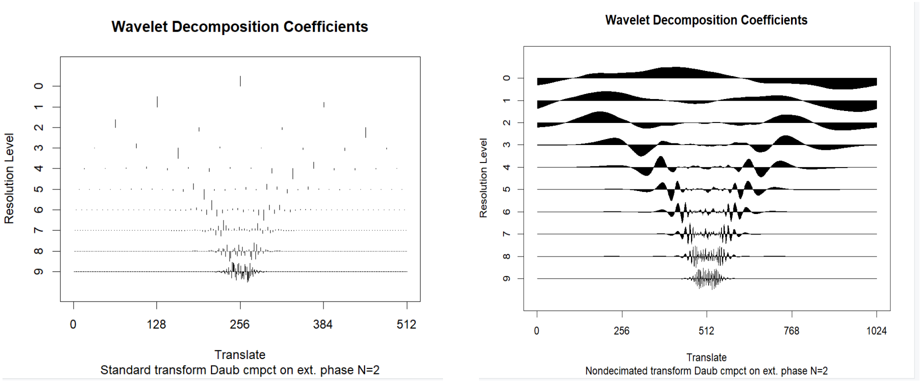

y
$\mathcal{D}_1\mathcal{G}$
$\mathcal{D}_0\mathcal{G}$
d_1
d_0
$\mathcal{D}_1\mathcal{H}$
$\mathcal{D}_0\mathcal{H}$
c_1
c_0
$\mathcal{D}_1\mathcal{G}$
$\mathcal{D}_0\mathcal{G}$
$\mathcal{D}_1\mathcal{G}$
$\mathcal{D}_0\mathcal{G}$
d_{11}
$\mathcal{D}_1\mathcal{H}$
$\mathcal{D}_0\mathcal{H}$
d_{10}
d_{01}
$\mathcal{D}_1\mathcal{H}$
$\mathcal{D}_0\mathcal{H}$
d_{00}
c_{11}
c_{10}
c_{01}
c_{00}

Figure 11: Flow diagram of the Stationary Wavelet Transform from [Nason, 2008]

The finest-scale wavelet coefficients are d_0 and d_1. The next finest scale are $d_{00}, d_{0,1}, d_{1,0}, d_{1,1}$. The coefficients that only have 0 in the subscript correspond to the usual wavelet coefficients while the coefficients with 1 in the subscript correspond to the wavelet coefficients generated solely by the SWT that aren't present in the DWT. We see from this flow diagram how double the coefficients are present in SWT than DWT and that downsizing is absent. It is worth noting that the labelling of filters and notation used in this figure are taken



(a) The wavelet decomposition plot of the DWT of the chirp function

(b) The wavelet decomposition plot of the SWT of the chirp function

Figure 12: Comparison of the Stationary Wavelet Transform with the Discrete Wavelet Transform of the chirp function

from [Nason, 2008] and hence different from the notation throughout this paper, specifically the low-pass filter is labelled H and the high pass filter is labelled G.

Returning to our chirp function we produce a wavelet decomposition plot of its Stationary Wavelet Transform with R (Figure 12).

```
chirpwdNonDecimated <- wd(chirp$y, filter.number = 2, family = "DaubExPhase", type = "station")
plot(chirpwdNonDecimated, scaling = "by.level")
```

While the Discrete Wavelet transform gives us good information about the nature of the signal, as the scales get coarser it is difficult to see any oscillations. These lower frequency oscillations are not redundant in the chirp function and without them our image of the function is limited. The missing oscillations are present in the Stationary Wavelet Transform of the signal and generally significant information about the oscillatory behaviour at medium and low frequencies (coarser scales) is retained, this is because the Stationary Wavelet Transform is not decimated at lower levels unlike the Discrete Wavelet Transform. It is clear to see why the SWT is more useful for time-frequency analysis than the DWT.

In general the Undecimated Wavelet Transform provides several advantages over the DWT:

1. Shift-Invariance: The SWT is shift-invariant, meaning that small shifts in the input signal result in small shifts in the transform coefficients. This property is well suited for applications such as signal denoising and time series analysis.

2. Redundancy: The SWT has a redundancy factor of 2, the number of coefficients in the transformed signal is doubled compared to the original signal. This redundancy can provide more accurate representation of signals with localized features.

3. Improved Frequency Localization: The SWT has improved frequency localization compared to the DWT, especially for high-frequency components. This is because the SWT avoids aliasing caused by downsampling.

All these reasons compound to explain why SWT is the preferred alternative to DWT. It is worth noting that, SWT can have higher computational complexity compared to the DWT due to the absence of downsampling. The production of each coefficient requires a fixed number of operations (which depends on the length of the wavelet filter in use), and so the computational effort required to compute the SWT is $\mathcal{O}(n \log_2 n)$ making it a slower algorithm than the discrete wavelet transform, which is $\mathcal{O}(n)$. Despite this the non-decimated algorithm is still considered a fast algorithm and this speed "drawback" is often mitigated by the benefits it offers, especially in applications that require shift-invariance or precise localization of high-frequency components. Another more significant trade off of the SWT in how by introducing redundancy, it leads to a loss of invertibility. This means that the original signal cannot always be perfectly reconstructed from its wavelet coefficients, unlike in DWT where exact reconstruction is guaranteed. However, this trade-off is generally acceptable in many applications where the advantages of SWT outweigh the need for exact signal reconstruction.

4 Modelling Non-stationary Time Series data

For the duration of this paper we will seek to apply wavelet methodologies on a set of time series data.

4.1 What is time series data

A time series is a sequence of data points recorded at equally spaced time intervals with definite ordering in time. Each data point in a time series is associated with a specific time stamp, indicating when the observation was made. We define $\{X_t, t \in D \subset \mathbb{Z}\}$ where each X_t is an observation of the time series data. For example we denote an observed discrete time series of length n by $x_1, \ldots, x_n$. In general, the key difference between time series data and 'ordinary' data is that time series observations X_t are not independent but typically possess a stochastic relationship between observations.

Time series data arise in many areas of science and technology and analysis of time series data is one of the most extensively researched and widely studied areas of statistics. This sort of analysis usually consists of designing/choosing an appropriate statistical model based on the data, fitting the model onto the data (i.e. a regression model), analysing the data using the model and reformulating the model based on our results. The goal of this form of analysis is to obtain a robust model for the series to obtain an efficient and accurate mathematical description of some key characteristics of our data and the relationship between the dependent variables. Another goal of this form of analysis is to forecast future values of the series based on collected data. Many methods and models exist to do this.

4.2 Formulating a Model

If a time series is a stationary stochastic process then it can be modelled with

$$X_t = \int_{-\infty}^{\infty} A(w) \exp(iwt) \, d\xi(w) \tag{7}$$

where $A(w)$ is the amplitude of the process and $d\xi(w)$ is a stochastic process with orthonormal increments. The parameter w can be interpreted as the frequency and thus we see that this formulation defines X_t as a weighted sum of a collection of sinusoids oscillating at various frequencies. The point with stationary processes is that the amplitude $A(w)$ does not depend on time. Hence, the frequency behaviour of the time series is independent of time. We see however that this formulation is not adequate to describe most time series models as often the frequency behaviour of the series actually changes with time. For such types of data it is necessary to introduce time dependency into our formulation; non-stationary time series.

4.2.1 Locally Stationary Wavelet Processes

One approach to model non-stationary time series data is to replace our amplitude function with a time varying version $A_t(w)$. This is the kind of idea promoted by [Priestley, 1965] and [Dahlhaus, 1997], which results in a time-frequency model. However a separate approach was introduced by [Nason et al., 2000] who created a rigorous time-scale model by replacing the set of harmonics $\{\exp(iwt),\ w \in (-\pi, \pi)\}$ with a set of discrete non-decimated wavelets. We define a time-series model based on this concept as a Locally Stationary Wavelet (LSW) process.

This approach developed by [Nason et al., 2000] provides a robust framework for modelling and understanding non-stationary time series. We prefer this approach as apposed to [Priestley, 1965] as the use of discrete non-decimated wavelets will provide us with a more detailed and comprehensive analysis of how the variance and other relevant statistical properties of the time series evolve over time and across different frequencies. Furthermore, the flexibility of the model, as apposed to Fourier-based methods in [Priestley, 1965] which rely on stationary assumptions, allow us to capture localized and transient features in the data. This is particularly relevant for modeling complex financial time series, where changes occur across multiple scales and transient localized features are regular. Another justification for this model lies in its simplicity as it circumvents the need for stochastic integration making it easier to implement and interpret practically. All these reasons make this an appropriate model for modelling non-stationary time series data such as stock data which exhibit locally stationary properties.

We can formally define a LSW process as follows:

Definition 4.1 (Locally Stationary Wavelet (LSW) processes)**.** The LSW processes are a sequence of doubly indexed stochastic processes $\{X_{t,T}\}_{t=0,\ldots,T-1}$ with $T = 2^J$, having the representation in the mean-square sense

$$X_{t,T} = \sum_{j=-J}^{-1} \sum_{n} w^0_{j,k;T} \psi_{jk}(t) \xi_{jk}$$

where ξ_{jk} is a random orthonormal increment sequence and where $\psi_{jk}(t)_{jk}$ is a discrete non-decimated family of wavelets for $j = -1, -2, \ldots, -J(T)$, $k = 0, \ldots, T-1$ based on a mother wavelet $\psi(t)$ of compact support [Nason, 2008]. These quantities have the following properties

1. $E(\xi_{jk}) = 0 \quad \forall j$ and k. Hence $E(X_{t,T}) = 0$ for all t and T.

2. $\text{Cov}(\varepsilon_{jk}, \xi_{im}) = \delta_{il}\delta_{km}$.

3. There exists for each $j \leq -1$ a Lipschitz continuous function $W_j(z)$ for $z \in (0,1)$ which fulfills the following properties:

   $$\sum_{j=-\infty}^{-1} |W_j(z)|^2 < \infty \quad \text{for } z \in (0,1),$$

   the Lipschitz constants L_j are uniformly bounded in j, and

   $$\sum_{j=-\infty}^{-1} 2^{-j} L_j < \infty,$$

   there exists a sequence of constants C_j such that for each T

   $$\text{Sup}_k \left| W^0_{j,u^j,T} - W_j(k/T) \right| \leqslant C_j/T,$$

   where for each $j = -1, \ldots, -J(T) = -\log_2(T)$, the supremum is over $k = 0, \ldots, T-1$, and where C_j fulfills

   $$\sum_{j=-\infty}^{-1} C_j < \infty.$$

This definition is taken from [Nason et al., 2000]

This definition specifies a time series model based on non-decimated discrete wavelets and enforces that we build a time series model $X_{t,T}$ out of a linear combination of oscillatory functions $\psi_{j,k}$ with random amplitudes $(w_{j,k;T}\xi_{j,k})$.

Here the function $W_j(z)$ is a rescaled Lipschitz continuous amplitude function, it is a time-varying function that describes the amplitude of the wavelet coefficients at a particular scale j and rescaled time $z = \frac{k}{T} \in (0, 1)$. A Lipschitz continuous function, simply put, is just a stable and smooth function with a bounded rate of change.

The smoothness constraints imposed on the function $W_j(z)$ in (3) of (4.1) are crucial to ensure that the model is feasible and provides an accurate model of non-stationary behaviour. The constraints forbid the random amplitudes

$$w_{j,k;T}\xi_{j,k}$$

from deviating too excessively from the smooth function $W_j(z)$ and prevent $W_j(z)$ from oscillating too wildly and exhibiting erratic behaviour, thereby allowing us to control the speed of evolution of the process. Controlling this speed of evolution is important to the model as it permits the pooling of more observations, from which we obtain good estimates for the process generator $W_j(z)$. A slower speed of evolution implies that more data points can be aggregated over time, resulting in more reliable and accurate estimates of $W_j(z)$. This ensures that the function remains stable and does not exhibit rapid, unpredictable changes, further enhancing the robustness of the model by allowing better estimation of the underlying processes.

4.2.2 The Evolutionary Wavelet Spectrum (EWS)

In order to gain deeper insights from the Locally Stationary wavelet (LSW) model defined in (4.1) we introduce the concept of the Evolutionary Wavelet Spectrum (EWS). The Evolutionary Wavelet Spectrum (EWS), $S_j(z)$ is defined by the equation:

$$S_j(z) = |W_j(z)|^2 \tag{8}$$

The EWS $S_j(z)$ measure how variance is distributed across scale j and location $z \in (0, 1)$. By decomposing the total variance into contributions from different scales, the EWS enables a multiscale analysis, capturing how the variance of the time series is distributed across different scales and over time. Computing the EWS allow us to perform a detailed analysis of the dynamic behavior of the time series.

4.2.3 Estimating the EWS

We can compute a direct estimate of the EWS via the wavelet Transform. A theoretical framework for the estimation of the EWS is provided by [Nason, 2008].

We begin by decomposing the time series into its wavelet coefficients across different scale and positions, capturing a good image of the time-frequency content. From [Nason et al., 2000] we define the empirical non-decimated wavelet coefficients of a time series x_t by

$$d_{j,k;T} = \sum_{t=1}^{T} x_t \psi_{j,k}(t) \tag{9}$$

Where $\psi_{j,k}$ are the scaled and translated wavelet basis functions derived from some appropriate mother wavelet. In practise we tend to take the mother wavelet to be a Daubechies wavelet particularly for simulations in R [Nason, 2008].

In the case of orthonormal wavelet coefficients obtained by the DWT a rescaling by some scale factor is necessary. This ensures that the energy of the time series is normalised across scales accommodating a valid and accurate comparison between different scales. In the case of standard orthonormal wavelet coefficients larger scales aggregate more of the signal components over time thus leading to higher energy hence the requirement for rescaling. However, in the case of the Non-Decimated Wavelet Transform the energy distribution across scales is naturally preserved and this rescaling is unnecessary. This is a result of the fact that, as previously described, the Non-Decimated Wavelet Transform does not downsample the signal and maintains the same number of coefficients at each scale. Thus we can obtain an estimate for the EWS in the form of a (raw) wavelet periodogram directly from the non-decimated wavelet coefficients $d_{j,k;T}$, another advantage of the non-decimated formulation. We find that the raw wavelet periodogram is given by:

$$I^j_{k,T} = |d_{j,k;T}|^2 \tag{10}$$

Here $I^j_{k,T}$ represents the raw wavelet periodogram at scale j and position k for a time series of length T. From [Nason et al., 2000] we have that if $I(z) = \left\{ I^j_{\lfloor zT \rfloor, T} \right\}_{j=-1,\dots,-J}$ represents the vector of raw periodograms at a rescaled time z then:

$$E\{I(z)\} = A\mathbf{S}(z) + O(T^{-1}) \tag{11}$$

Here we have a collection of the true EWS across different scales defined by

$\mathbf{S}(z) = S_j(z)_{j=-1\ldots-J}$ which is a collection of the spectral densities $S_j(z)$ at scale j and rescaled time z for $j \in -1, -J$. The matrix A essentially represents the bias of $I(z)$ and is defined by the inner products of the autocorrelation wavelets:

$$A_{j\ell} = \langle \Psi_j, \Psi_\ell \rangle = \sum_\tau \Psi_j(\tau)\Psi_\ell(\tau) \tag{12}$$

Where Ψ_j and Ψ_ℓ represents the autocorrelation wavelet for lag τ at scales j and ℓ respectively. Equation (12) effectively performs a convolution, measuring how the wavelets at different scales overlap and interact with each other. Thereby giving the bias in the raw periodograms of the vector $I(z)$.

In addition to the expectation of the raw wavelet periodogram $I(z)$ in (11) [Nason et al., 2000] also derives the variance of $I(z)$ as:

$$\operatorname{var}\left(I^j_{\lfloor zT \rfloor, T}\right) = 2\left\{\sum_\ell A_{j\ell} S_\ell(z)\right\}^2 + O(2^{-j}/T). \tag{13}$$

This implies that the wavelet periodogram is not a consistent estimator of the EWS as its variance does not vanish as the sample size $T \to \infty$. To remedy this issue we adopt the approach of smoothing the raw wavelet periodogram as a function of $\lfloor zT \rfloor$ (or k) for each scale j. Various techniques exists to perform this smoothing such as [Fryzlewicz and Nason, 2006]which advocates the use of Haar-Fisz transforms, this is however beyond the scope of this report.

Aside from special cases no closed form formula is known for A for Daubechies wavelets [Nason, 2008] thus numerical methods are required to obtain a good estimate for A. Upon computing A we can apply an obvious bias correction to our smoothed raw periodograms to obtain:

$$L(z) = A^{-1} I(z) \tag{14}$$

Where $L(z)$ represents our corrected wavelet periodogram and A^-1 represents the inverse of the matrix A. This equation corrects the bias in the raw periodograms to obtain a better estimate the true Evolutionary Wavelet Spectrum as shown by its mean in (15).

$$E\{L(z)\} = A\mathbb{S}(z) + O(T^{-1}) \tag{15}$$

Note also that in practical implementation, we interpolate the values of the wavelet coefficients $d_{j,k;T}$ to estimate the smooth continuous function $W_j(z)$

over the rescaled time interval $z = \frac{k}{T} \in (0, 1)$. This interpolation can be implemented using various numerical methods such as linear interpolation or, for more precision, spline interpolation.

4.2.4 Autocorrelation and Autocovariance

In addition to the variance across different time-scales we also want a way to identify any hidden linear relationships/trends within the data, specifically we want to identify and examine the level of correlation between lagged observations in the dataset. To do this we could employ a statistical measure known as autocovariance. The autocovariance describes the covariance between two points in the time series separated by a time lag τ. Where (generally) the covariance

$$\mathrm{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])] \tag{16}$$

is a measure of the relationship between two random variables X and Y and how they change together, specifically it indicates the direction of the linear relationship between two variables. A positive covariance indicates a positive correlation between two variables while a negative covariance indicates a negative correlation. The autocovariance is a specific type of covariance that measures the correlation of a variable to itself at different points in time. Traditionally for a time series B and some lag τ this is defined by the equation

$$\gamma(k) = \mathrm{Cov}(B_t, B_{t+k}) = E[(B_t - \mu)(B_{t+k} - \mu)] \tag{17}$$

In the case of Locally Stationary Wavelet Processes we build upon this definition to obtain a more specialised form which incorporates the notion of local stationary and wavelet decomposition.

From [Nason et al., 2000] the autocovariance of a Locally Stationary Wavelet (LSW) process $X_{t,T}$ for a finite time series of length T is given by:

$$c_T(z, \tau) = \mathrm{cov}\left(X_{\lfloor zT \rfloor, T}, X_{\lfloor zT \rfloor + \tau, T}\right) \tag{18}$$

Where $\lfloor zT \rfloor$ is the largest integer less than or equal to the localised time index zT. It is of note that zT converts the normalized time index z into an actual integer time index within the time series.

To approximate the behaviour of this process in large samples we seek to understand the asymptotic behaviour of the function. Taking the limit of (18) as $T \to \infty$ this autocovariance converges to $c(z, \tau)$ given by:

$$c(z,\tau) = \sum_{\jmath=-\infty}^{-1} S_\jmath(z)\Psi_\jmath(\tau) \tag{19}$$

The equation (19) establishes a relationship between the time varying autocovariance of $X_{t,T}$ and the Evolutionary Wavelet Spectrum $S_j(z)$ by an autocorrelation wavelet $\Psi_j(\tau)$. Breaking down the equation we see that it essentially splits the autocovariance into parts which are contributed by each scale j. For each wavelet scale, the contribution to the autocovariance is scaled by the EWS $S_j(z)$. This computes how much that particular scale j contributed to the autocovariance at time z. The contributions from all scales are summed over until we get the total autocovariance at time z for a lag τ. We see that this equation is analogous to a wavelet transform in and of itself.

Included in the equation (19) is the autocorrelation wavelet $\Psi_j(\tau)$ which measures how the wavelet coefficients at different scales and positions are correlated with each other across different time lags. From [Nason et al., 2000] we see that $\Psi_j(\tau)$ is defined by:

$$\Psi_j(\tau) = \sum_k \psi_{j.k}(0)\psi_{j.k\cdot}(\tau)$$

for all $j < 0$ and $\tau \in \mathbb{Z}$ This, as the name suggests, simply describes how wavelet functions at a specific scale j correlate with themselves over different time lags τ, i.e. it calculates the autocorrelation of the wavelet $\psi_{j,k}$ at lag τ. By summing the products of the wavelet function values at time 0 and time τ for all positions k.

While the autocovariance computed through (19) is informative of the extent of linear relationship within the data it is not the most robust measure. This is due to its lack of standardised units which make it difficult to interpret the strength of any such relationship as the autocovariance values are influenced by the scale of the data. The autocorrelation of a LSW process given in equation (20) overcomes this limitation by normalising the autocovariance by the variance at each scale and time point, thus removing the influence of the datas scale and providing a clear measure of the linear relationship between lagged time series values relative to their variance.

$$\rho(z,\tau) = \frac{c(z,\tau)}{c(z,0)} \tag{20}$$

The function (20) has range $\rho(z,\tau) \in [-1,1]$ where -1 indicates a perfect negative correlation and 1 indicates a perfect positive correlation. Due to this normalisation we will adopt the autocorrelation as the preferred method of trend analysis throughout the duration of our report.

4.3 Demonstrative Implementation in R

This is readily implemented in R making use of the powerful *locits* [Nason, 2023] and *wavethresh* packages [Nason, 2022]. Let us run through the non-stationary time series *BabyECG* as a demonstration. *BabyECG* is a medical dataset collected by the Institute of Child Health at the University of Bristol that records the heart rate of an infant during night time [Nason, 2008].

We can produce an estimate for the EWS for the BabyECG dataset with the function `ewspec3`:

```
spec <- ewspec3(dBabyECG, binwidth = 100)$S
```

Where *dBabyECG* is the detrended BabyECG dataset designed to have zero mean s.t it satisfies the conditions of a LSW. We have defined the smoothing parameter binwidth. The binwidth parameter is used to specify the width of the frequency bins when computing the wavelet spectrum. A larger binwidth will result in fewer frequency bands and broader spectral features, while a smaller binwidth will lead to more frequency bands and finer spectral resolution. We have guessed at a reasonable value of 100 for binwidth based upon the large number of observations in the dataset, experimentation with different binwidth values to find the most suitable parameters is welcome.

Plotting the EWS generated by (4.3) produces the figure (13)

We compute the localised autocovariance using the locits package in R. The function lacf computes the localized autocovariance and autocorrelation function for a nonstationary time series, performing all previously described steps of transforming, smoothing and bias correction. This function returns two matricies of the localised autocovariance and autocorrelation respectively at different lags and times. We can employ use of this function to analyse the data from the BabyECG time series.

```
L <- lacf(dBabyECG, binwidth = 100)
```

Retrieving the localised auto-correlation with `L$lacr` we obtain a matrix A with dimensions mxn. The rows m represent the various time points t covered by the dataset while the columns n represent different lags τ. For this specific scenario we see that the matrix A investigates 2048 time observations $m = 2048$ and 34 lags $n = 34$.

Due to the complexity of the resulting matrix we choose to perform our analysis of the localised autocorrelation through plots.

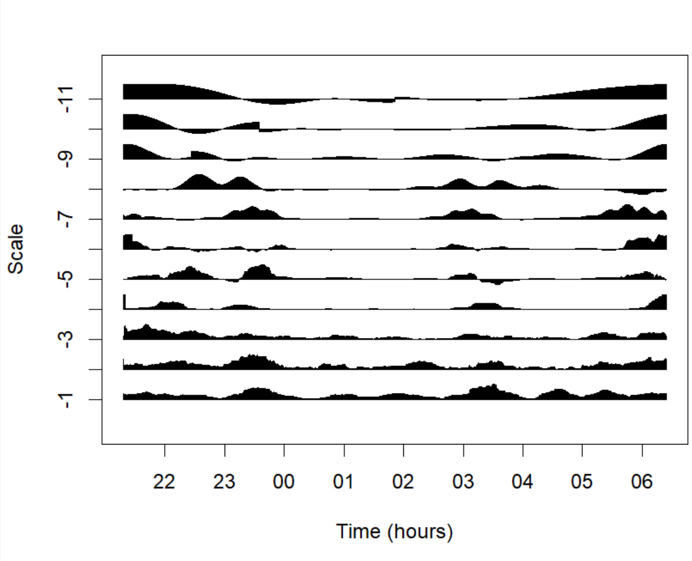


Figure 13: Evolutionary Wavelet Spectrum of the adjusted BabyECG time series with binwidth = 100

We begin by plotting the autocovariance at lag 0, the variance of x_t. Implementing this in r we produce the figure (14)

```
#This is the autocovariance at lag 0 -> just the variance of x_t
plot.ts(L$lacf[,1], main="Variance of BabyECG", xlab="Time (hours)", ylab="Variance", xaxt="n")
axis(1, at = tm2u, labels = tchar)
```

Now that we have an idea of how the variance evolves over time we can investigate the autocorrelation between lagged observations x_t and $x_t + \tau$.

To have an idea of which lags are significant we can plot the autocorrelation at different lags τ at some time t with

```
    plot.ts(L$lacr[t,])
```

With appropriate formatting we use this to plot the autocorrelation at lags τ for times $t = 5, t = 500, t = 1000, t = 2000$ as shown in figure (15).

From (15) we see that the relationship between the lags and the autocorrelation remains relatively consistent across time. Clearly, the lags with most impact on autocorrelation are lag 1, 3 and lag 5. Lag 1 has by far the strongest autocorrelation, lags 3 and 5 on the other hand have a relatively weaker but still somewhat significant autocorrelation. We see for all times t as the lag τ increases the autocorrelation stabilises to zero. From $\tau = 5$ onwards is when the convergence kicks in, after this point the autocorrelation between x_t and $x_{t+\tau}$

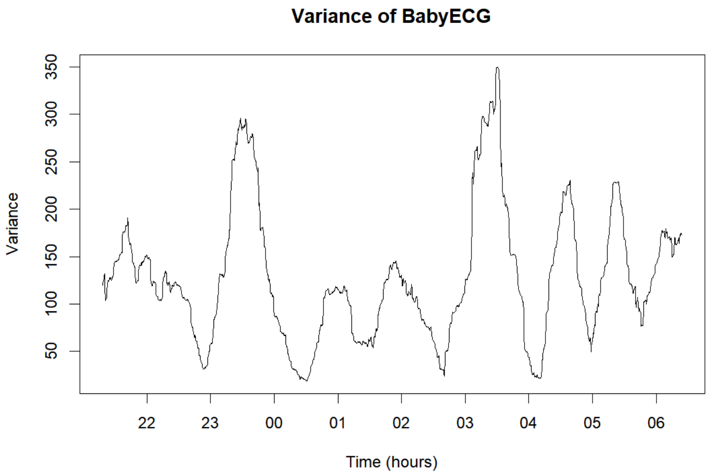
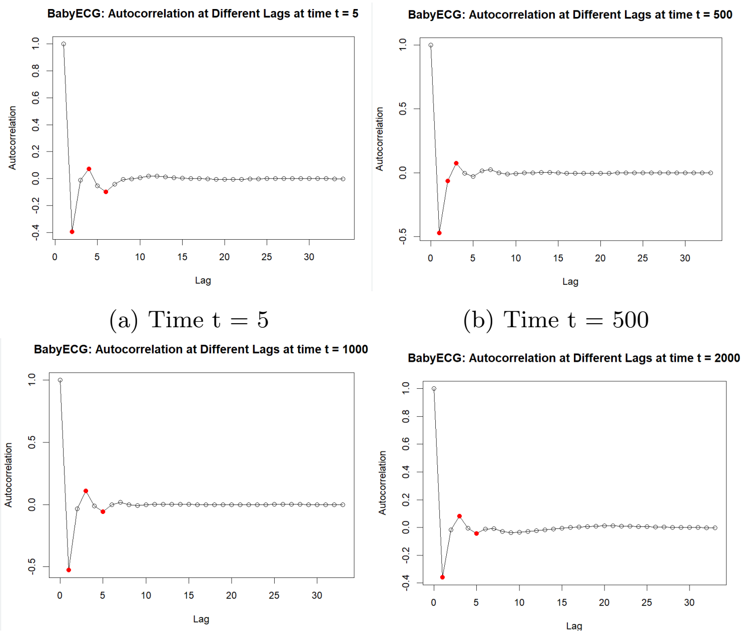


Figure 14: The variance of x_t at times t(hours)



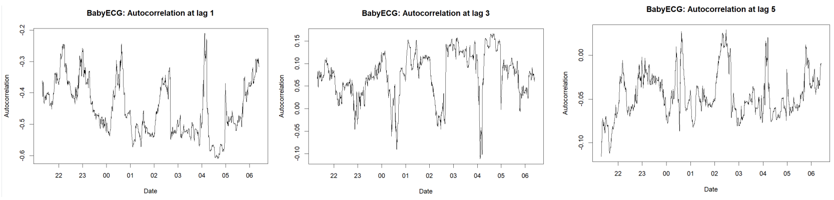
(a) Time t = 5

(b) Time t = 500

(c) Time t = 1000

(d) Time t = 2000

Figure 15: The Autocorrelation across different lags at time t



(a) Autocorrelation at Lag1 (b) Autocorrelation at Lag 3 (c) Autocorrelation at Lag 5

Figure 16: Autocorrelation at different lags of significance

becomes negligible. Indicating that there is very little linear relationship or correlation between the values of the BabyECG time series at time t and the values at time $t + \tau$ for $\tau > 5$, where τ is the time lag.

Extracting the most relevant lags we can now plot the autocorrelation across time at these lags (16)

5 A Case Study: Analysis of FTSE 100 and S&P 500 stock indexes

5.1 Introduction to the Datasets

For the remainder of this report we will focus on applying the methodology described and outlined to perform a qualitative analysis of financial time series datasets. Specifically, we will perform an analysis of closing stock data for the S&P 500 and FTSE 100 datasets. The S&P 500 is a stock index comprised of 500 of the largest companies listed on American stock exchanges. Similarly, the FTSE 100 is a stock index comprised of the largest 100 companies on the London Stock Exchange. Both stock indexes are composed of stocks from a variety of different sectors and generally tend to reflect the overall economic state of their respective countries.

Due to the nature of wavelet analysis we gain insights into how the frequency of our stock signal evolves over time. Importantly, we can identify long term trends and clearly distinguish them from short term fluctuations. Our ultimate goal is to perform an analysis of the variance at different scales to gain insights into the volatility of the market and how sensitive it is to short term fluctuations. We also aim to identify the strength of correlation between datapoints at different lags, this will inform us of how dependent future stock prices are on the current stock prices.

Figures (17) and (18) plot the closing prices of the UK FTSE 100 and US S&P 500 stock indexes respectively. These are daily closing prices over the trading

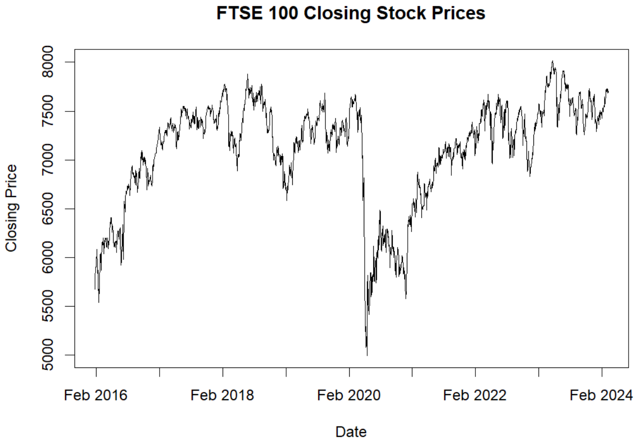


Figure 17: Plot of FTSE 100 closing prices from 2016 - 2024

days between (20/01/16 - 01/05/24) for the FTSE dataset and (20/01/16 - 08/03/24) S&P dataset. There is a 54 day discrepancy in trading days between the FTSE 100 and S&P 500 datasets. This is likely due to differences in stock closure dates (such as holidays) between American and UK markets. Ideally, we would like to able to analyse over same period but we are limited to 2048 observations, due to the nature of the DWT. This discrepancy wont have too significant of an impact on our analysis however and so we go forward with it.

Now that we have established our datasets we can begin our wavelet analysis. We compute the vector of differences with `diff(SnP$Close)` and `diff(FTSE)`, this calculates the change in closing prices from one day to the next and manipulates our data such that it satisfies the conditions of a Locally Stationary Wavelet (4.1) without too much loss of precision.

5.2 Analysis of the EWS

We begin our analysis with a wavelet transform of the corrected datasets, we produce the wavelet decomposition plots in (19) and (20). As expected we see a spike at all resolution levels around the midpoint this corresponds to the COVID-19 pandemic in 2020. The decomposition plot (19) seems to imply that the FTSE 100 stock index and the S&P plot (20) grew more volatile after the onset of the pandemic (about halfway through our time period). Particularly the S&P dataset which exhibts a lot more relative volatility in in finer levels past the second half of our downsampled time.

To gain more clarity and insights on the power of the variance we compute the Evolutionary wavelet spectrum of the S&P 500 and FTSE 100 datasets in (21) and (22) respectively. We can also plot the evolution of the total variance for

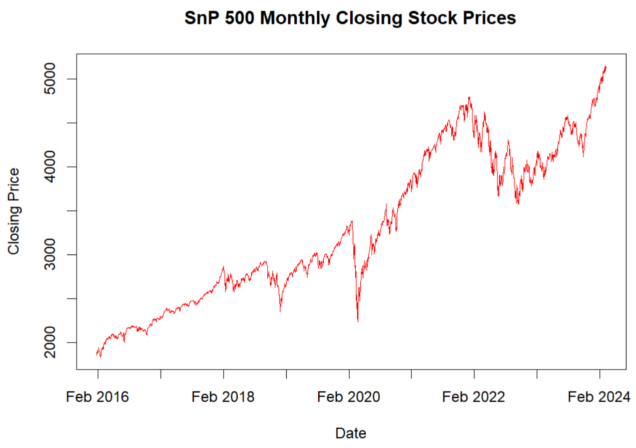


Figure 18: Plot of S&P 500 closing prices from 2016 - 2024

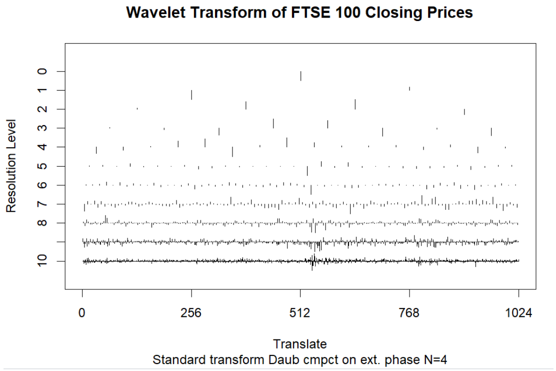


Figure 19: Wavelet coefficients of the corrected FTSE 100 dataset

both datasets as shown in figure (23).

We can break down our analysis into key phases.

For the S&P 500 dataset: 2016-2018: High variance at lower scales suggests strong influence of long term trends. Very low/almost negligible variance at higher scales indicates a generally stable market not as susceptible to short term fluctuations

2020: Following from wavelet transform see that the variance starts to noticeable increase specifically after the the onset of the pandemic around Feb 2020. Noticeable spike in volatility across higher scales and clear decrease in the variance at lower scales (0-2) indicating that the impact of long term/stable trends become less pronounced/weakened. The underlying long-term patterns and trends

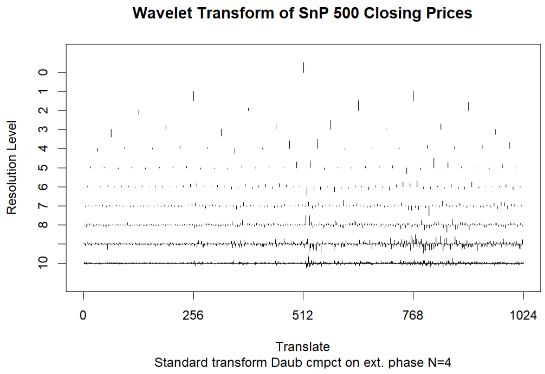


Figure 20: Wavelet coefficients of the corrected S&P 500 dataset

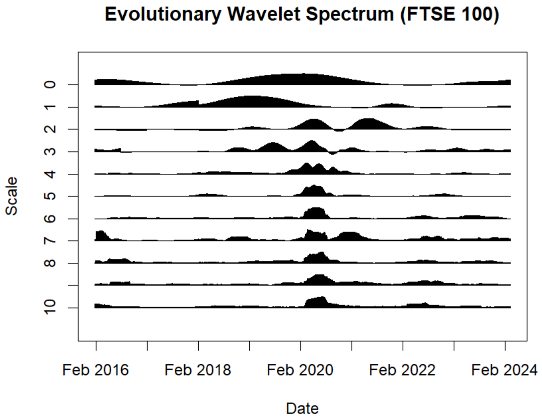


Figure 21: Evolutionary Wavelet Spectrum of the corrected FTSE 100 dataset

in the market data were disrupted or overshadowed by more volatile, shorter-term fluctuations. This corresponds to the onset of the COVID-19 pandemic which caused a major stock market crash, as is clearly visible from all the plots thus far, the pandemic was very abrupt and clearly shown to have a significant impact on the market. However, low power at low scales post 2020 suggests less prominent effects on the overall trend of the market. This indicates that the market stabilised quickly after the initial shock of the pandemic

2022:

The S&P figure (23a) also suggests another period of significant market volatility in 2022, as there is a spike in variance across all scales around the period of 2022-2023. This suggests an interesting period of the stock market as long term trends appear to have a weak but still present influence. Increased variance

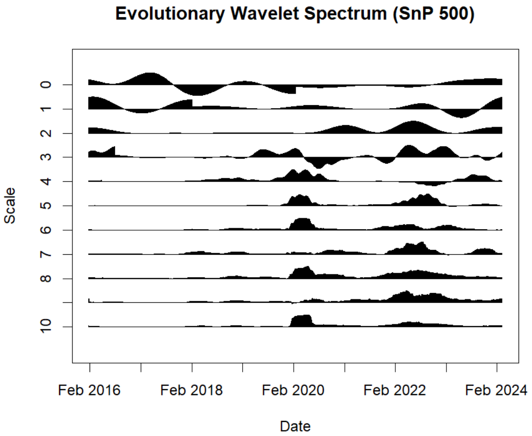


Figure 22: Evolutionary Wavelet Spectrum of the corrected S&P 500 dataset

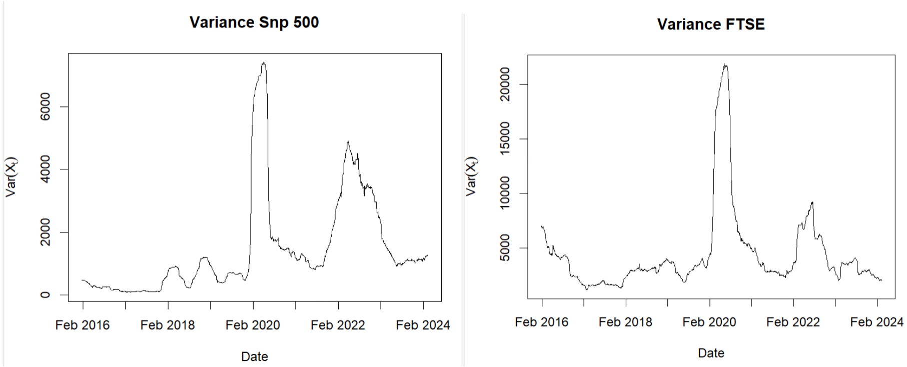
after a small recovery in higher scales (8-10) and most significantly the great spike in variance in scales (5-7) suggests that the market had high sensitivity to medium and short term fluctuations around this time.

This could be the result of the impact of many relevant events. In 2022, the long term effects of the pandemic were still relevant, inflation rates soared to very high levels, there were again interest rate hikes as well as considerable Geopolitical tension due to Russia's invasion of Ukraine. All of these factors could account for the spike in overall market volatility around this time.

2023-2024: We see that post 2022 the variance recovers across a range of scales and stock begins to return to following long term trends, although not quite to same extent as pre-covid, this suggests less uncertainty in the market and indicates a general recovery.

The FTSE 100 generally appears less volatile across different scales scales. However, similar to the S&P plot it also shows a significant spike across all scales in 2020, indicating that the UK market was similarly affected by the COVID-19 pandemic. This outcome is expected as the pandemic was a global issue that had impact on stocks and trading markets worldwide. The EWS (21) also shows generally consistent volatility across all scales post 2022, suggesting that market is quite resilient.

From the EWS it is clear that the FTSE market is more susceptible to long term trends than short term fluctuations as opposed to the to S&P 500. As evidence by the generally higher power at scale 0. Particularly, in 2020 the FTSE 100 shows high power at scale 0, indicating a significant long-term impact of the COVID-19 pandemic on the market. The long-term effects appear to be much more pronounced compared to the S&P 500, reflecting the substantial disruption



(a) Variance of the S&P 500 dataset over the period 2016-2024

(b) Variance of the FTSE 100 dataset over the period 2016-2024

Figure 23: Variance of the FTSE 100 and SnP 500 datasets

caused to the UK markets by the pandemic. Another thing revealed by the EWS (21) is that the various factors that influenced the S&P dataset did not have as prominent effect on the FTSE dataset. This is likely due to the differences in sector composition and market structure between the $S\&P$ 500 and FTSE 100 indicies.

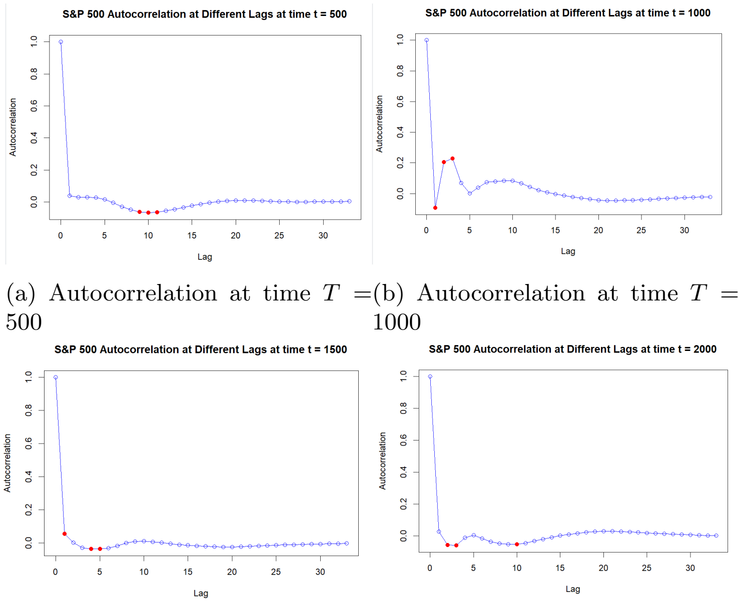
5.3 Analysis of Autocorrelation

Can employ am analysis of correlation using `locits` package and plot the localised autocorrelation between lagged times t and $t + \tau$. As mentioned, this is crucial as it may reveal any underlying trends/linear relationships in the data. An analysis can also reveal the strength of relationship between these lagged observations x_t and future observation $x_{t+\tau}$. This is especially useful when it comes to forecasting as understanding the nature of the relationship between different observations can help us predict how x_t may evolve with time

We highlight relevant lags to investigate by investigating the effect of different lags on the autocorrelation at some time t. To begin, we have chosen a vector of different times T to inspect. Here we will chose T to be biennial, $T = [500, 1000, 1500, 2000]$. Plotting this in R using the functions described in the previous section, produces figure (24) for the S&P dataset and figure (25) for the FTSE dataset.

Here we have highlighted the lags with the greatest absolute autocorrelation in red in each plot.

Our plots reveal that for the S&P 500 dataset lag 1 is significant as are lags 2, 3, 4, and 10. In the case of the FTSE 100 dataset we find the lags of significance to be 1,2,3 and 5. These particular lags generally exhibit larger absolute correlation



(a) Autocorrelation at time T = 500

(b) Autocorrelation at time T = 1000

(c) Autocorrelation at time T = 1500

(d) Autocorrelation at time T = 2000

Figure 24: Autocorrelation at different lags of significance for some constant time T: S&P 500 Dataset

compared to other lags, meaning that there exists a linear relationship between observations separated by this lag. The majority of the other lags highlighted in the plots have a negligible or very weak effect on the data, as their values are very close to zero, so they can be reasonably ignored. Therefore, we select these significant lags for further investigation.

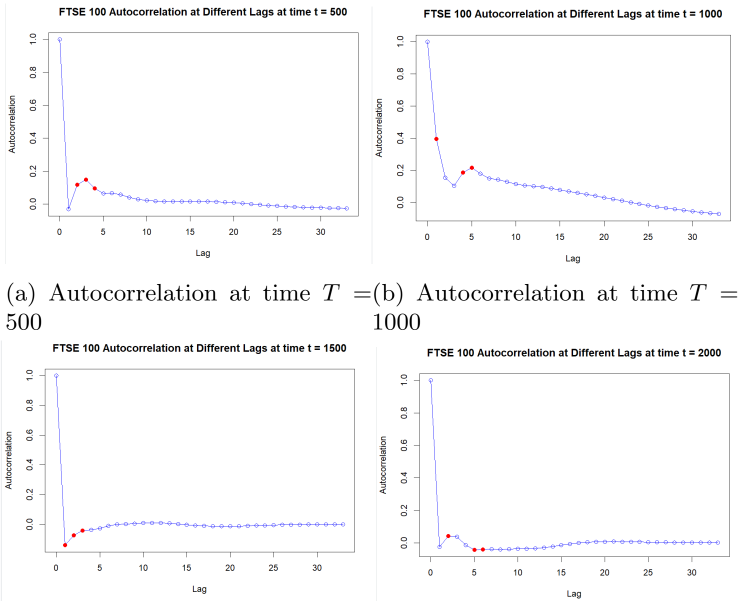
Proceeding further, we investigate the Autocorrelation of the dataset at these lags of significance by plotting the localised autocorrelation at these lags over our time frame t.

For the S&P 500 dataset this is shown in the figure (26). Also plotted in (27) is the autocorrelation of the S&P 500 dataset at lag 30, we have done this to see the extent of the impact of the market events:

We see that there are two periods of increased market volatility presented in the S&P plot:

2018:

In 2018 the $S\&P$ 500 plot reveals a period of increased market volatility. During this period, the autocorrelation function (ACF) shows relatively a significant



(a) Autocorrelation at time $T =$ 500

(b) Autocorrelation at time $T =$ 1000

(c) Autocorrelation at time $T =$ 1500

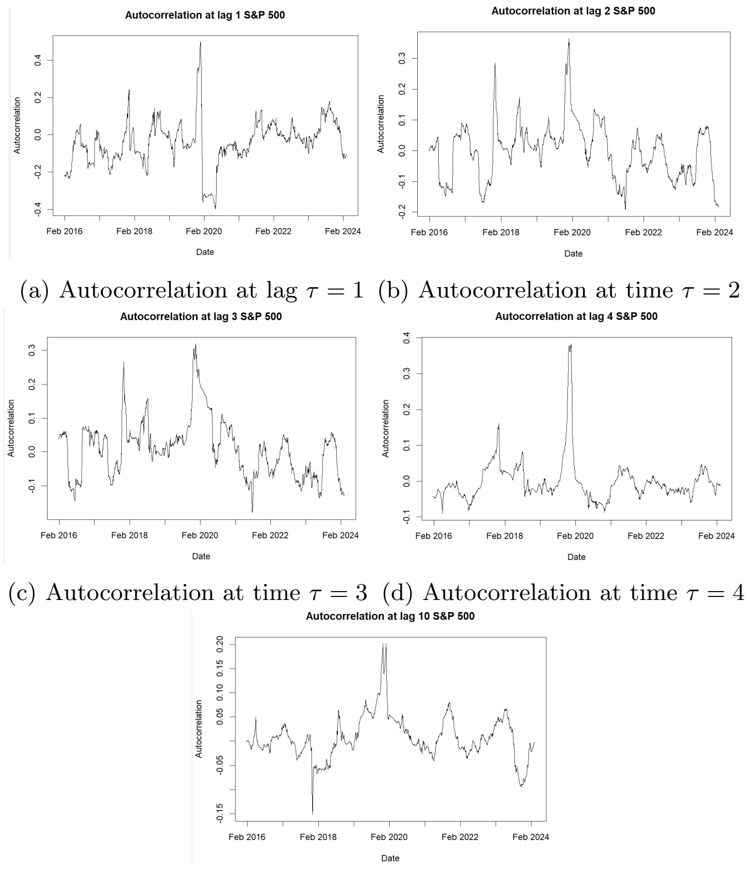
(d) Autocorrelation at time $T =$ 2000

Figure 25: Autocorrelation at different lags of significance for some constant time T: FTSE 100 Dataset

positive autocorrelation at lags 1, 2, and 3, taking values between 0.2 and 0.3. Additionally, there is a negative correlation of -0.15 at lag 10. This indicates a positive correlation between consecutive daily returns and a negative correlation between biweekly returns, suggesting a period of medium-term fluctuations and short-term volatility in market conditions. This observation can be corroborated to several market events. In 2018, the U.S. Federal Reserve raised interest rates several times, impacting borrowing costs and investor sentiment [Reserve, 2019]. Furthermore, the Tech sector experienced volatility due to increased scrutiny of major tech companies such as Facebook, Apple, Amazon, Netflix and Google which make up 11% of the S&P 500 index [Reserve, 2019]. However, despite this volatility we see that there is had a very weak/negligible long term impact/effect as seen by the low autocorrelation spike at lag 30.

2020:

In 2020, there is a strong negative autocorrelation at lag 1, indicating a strong inverse relationship between daily returns. This means that if the return was positive on one day, it would likely be negative the next. Such extreme market volatility is expected, given the massive economic shock caused by the COVID-19 pandemic. During the time of the pandemic, investors faced high uncertainty,



(a) Autocorrelation at lag $\tau = 1$ (b) Autocorrelation at time $\tau = 2$

(c) Autocorrelation at time $\tau = 3$ (d) Autocorrelation at time $\tau = 4$

(e) Autocorrelation at time $\tau = 10$

Figure 26: Autocorrelation at lag τ: S&P 500 Dataset

leading to rapid sell-offs and rebounds as new information about the pandemic and economic measures emerged. This high-frequency trading behavior only served to further exacerbate these rapid sell-offs. The ACF also shows strong positive autocorrelation at lags 2 and 3, and relatively strong autocorrelation at lag 10. This indicates that continued momentum from these rebounds and sell offs influenced trading over the next few days and still had effects up to two weeks after the initial movement. These findings suggest that a highly complex market dynamic during this time. Additionally, the long term effect on the stock prices is evident from the ACF at lag 30 (27) which shows a clear and comparatively large spike around this time.

2022:

In 2022, our LACF model for S&P appears to show a less pronounced downturn

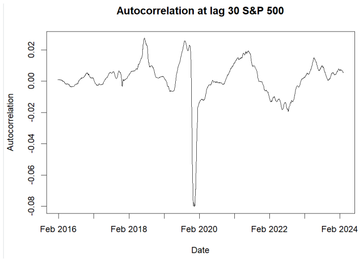
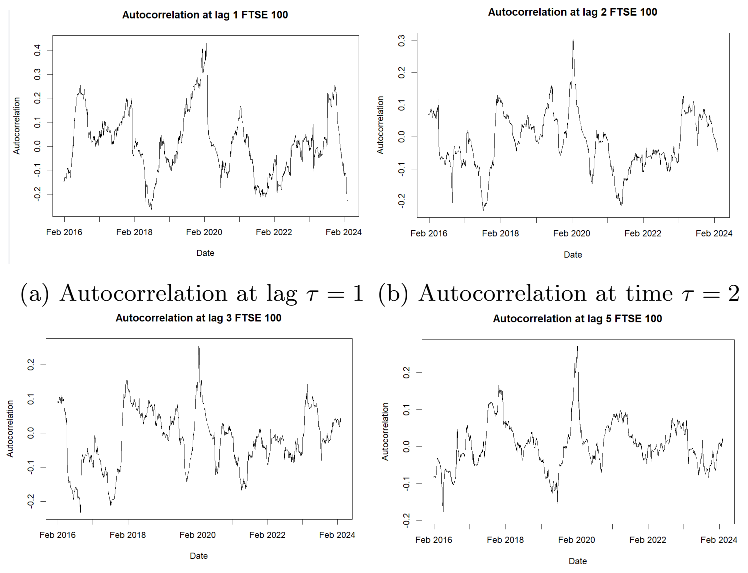


Figure 27: Autocorrelation at $\tau = 30$: S&P 500 Dataset

in Feb 2022 compared to our EWS and our stock market plots. This discrepancy is due to the dampening effect of the normalisation of the variance and informs us that, while there was significant market movement and variability around this time period it did not greatly impact the relative relationship between returns. This observation informs us that, despite notable volatility, the underlying autocorrelation structure of returns remained relatively stable.

We can produce similar plots for the FTSE 100 dataset the autocorrelation at its lags of significance are in figure (28) and of lag 30 in figure (29).



(a) Autocorrelation at lag $\tau = 1$ (b) Autocorrelation at time $\tau = 2$

(c) Autocorrelation at time $\tau = 3$ (d) Autocorrelation at time $\tau = 5$

Figure 28: Autocorrelation at lag τ: FTSE 100 Dataset

2018: An investigation of the autocorrelation of the FTSE also reveals that the FTSE had strong correlation both negative and positive in 2018 across the different lags Notably, lag 30 shows that this time period had a comparatively

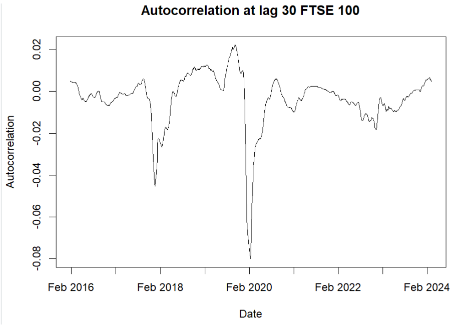


Figure 29: Autocorrelation at $\tau = 30$: FTSE 100 Dataset

large effect on the FTSE in the long term second only to the impact of the pandemic in 2020. During 2018, the FTSE fell by 12.5% the biggest annual decline since 2008. This notable decline was a consequence of stock market reactions to the Brexit referendum on 23 June 2016. The political and economic uncertainty surrounding Brexit negotiations in 2018 led to the weakening of the pound sterling and disruptions in global trade. This lack of stability undermined investor confidence, contributing greatly to the decline of the FTSE 100 in 2018 [Government, 2018]. This period also explains the spikes shown across some lags around 2016, which corresponds to when the Brexit vote first occurred. It is clear from analysis of the ACF function that the political and economic ramifications of Brexit continued to influence market behavior and investor sentiment during the subsequent years, culminating in significant volatility in 2018.

2020: In 2020, due to the severe impact of the Covid-19 pandemic the FTSE 100 showed a clear positive spike in autocorrelation across all short-term levels, indicating significant short-term volatility. During this period, the FTSE 100 crashed by over 34%. Furthermore, the comparatively large negative spike at lag 30, indicates (unsurprisingly) that the pandemic had a much stronger long-term effect than any other event. This further emphasises the global extent of the pandemic's impact, causing unprecedented disruptions in markets worldwide.

The conducted analysis of the variance and autocorrelation is generally consistent with our EWS analysis and highlighting key periods of interest and allowing clear analysis and comprehension of both the strength and period of effect of key events around this period. Specifically our analysis suggests that the S&P market is more resilient than the FTSE, a widely accepted fact. This indicates that our model aligns well with reality. The reason for its comparative strength lies in the fact that the S&P includes a more diverse range of sectors, allowing downturns in one sector to be offset by stability in others. Additionally, its stronger global presence helps buffer against domestic economic losses or downturns. Our analysis also highlighted key phases of market volatility and underlying trends in both markets in 2018 and 2020. We also discovered through our autocorrelation analysis that the extreme market volatility in 2022 (in the S&P dataset) revealed no significant underlying trends in the returns this year.

5.4 Possible Continuations

In this section we briefly discuss the limitations of the model, how it can be improved and touch on possible continuations to it.

5.4.1 Limitations of the model and possible improvements

One significant limitation of the model is the necessity to transform data into a zero-mean form to satisfy the conditions of the Locally Stationary Wavelet (LSW) model (4.1). This transformation, while essential, introduces noise into the data, and may potentially create correlations where none existed. This noise causes a loss in the integrity of our data thereby affecting the accuracy of our results. This is however a necessary trade off due to the nature of our methodology.

Another limitation is the absence of a specific measure to identify whether data points were anomalous, leading us to rely on visual inspection for periods of fluctuation. This lack of standardisation can lead to an unreliable model. An improvement that could overcome this limitation would be to construct hypothesis tests to calculate confidence intervals and identify anomalous data points. Such tests would standardize the detection of anomalies thereby enhancing the model's reliability by providing a statistical measure of uncertainty and robustness against outliers.

Another limitation is the condition on the input size as, due to the dyadic nature of the DWT, we must limit our dataset to observations of 2^n. This requirement can be restrictive as it forces us to either truncate or pad our data to fit this format, potentially leading to the loss of valuable information or the introduction of artificial data points. This constraint limits the flexibility of the model in handling datasets of arbitrary lengths and can affect the robustness of the analysis. There are however alternatives to our chosen method of analysis that we can adopt which overcome this issue, such adopting methods of non-dyadic wavelet analysis [Pollock and Cascio, 2006]

5.4.2 Wavelet coherence

Wavelet coherence is a powerful tool for analyzing the relationship between two time series in both the time and frequency domains. It allows us to identify localized correlations at different scales, providing insights into the dynamics of the interactions between the FTSE 100 and S&P 500 and allow us to quantify the strength of their cross correlation.

The wavelet coherence is computed using the Continuous Wavelet Transform (CWT) instead of the DWT, a potential drawback as it can be computationally costly. For two time series $X(t)$ and $Y(t)$ and their respective CWTs $CWT_Y(a,b)$ and $CWT_X(a,b)$ (computed from (4)) we mathematically define the wavelet coherence as:

$$R^2_{XY}(a,b) = \frac{\left|S\left(CWT_X(a,b)CWT^*_Y(a,b)\right)\right|^2}{S\left(|CWT_X(a,b)|^2\right)\cdot S\left(|CWT_Y(a,b)|^2\right)} \tag{21}$$

Here S denotes a smoothing operator in both the time domain and scale domain, in practise this smoothing operator is often implemented as a gaussian or another suitable kernel. This definition is taken from [Yang et al., 2016]

The equation (21) computes the wavelet coherence between $X(t)$ and $Y(t)$ and has range $R^2_{XY}(a,b) \in [0,1]$. The value taken by $R^2_{XY}(a,b)$ informs us of the degree of linear relationship between the two time series at a scale a and time b. Thus the wavelet coherence function is analogous to a correlation function, but builds upon it by analyzing the correlation between two time series not just globally, but locally in both time and frequency domains. By examining the interdependence and connection between two major stock indices, we can understand how the FTSE 100 and S&P 500 influence each other over different time scales. This analysis can reveal periods of strong interdependence and phases where the indices move independently, doing this will help us to identify specific events and factors that have the most significant impact on the global market. A thorough wavelet coherence analysis can be used to report valuable insights and recommendations to potential investors. For example, understanding the correlation structure between major indices can inform portfolio diversification strategies and inform risk management strategies.

5.4.3 Forecasting the movement of stock prices

A natural continuation would be to build a forecasting model using our results to predict future returns over some reasonable time frame. This would involve using wavelet methods to denoise the data and applying appropriate methodologies to aid in such a formulation. One such methodology may be the integration of econometric tools such as the Autoregressive Integrated Moving Average (ARIMA) model as adopted in [Andriy, 2018]. ARIMA models can handle trends and seasonality effectively, which can improve the model's ability to capture underlying patterns in non-stationary time series data. Applying wavelet denoising on the data is extremely advantageous as it helps to remove random fluctuations and noise that may obscure the underlying patterns in the datasets. This process can significantly improve the quality of the data, leading to more accurate and reliable forecasts. For example, wavelet-based denoising can filter out high-frequency noise while preserving important low-frequency trends, informing a more accurate predictive model.

After identifying a range of potential forecasting models we would split the relevant time series dataset into training and test sets. Using machine learning principles we would then apply our forecasting model on training set to predict the values of the test set. The accuracy of these models can then be assessed through statistical measures such as the Mean Squared Error (MSE). From such an evaluation we aim to choose the optimal models to perform the forecasting. This could be repeated a number of times until a refined model is arrived at. A combination of this outlined method with wavelet-based denoising can offer a robust approach to handling non-stationary data and significantly aid in the precision of stock price forecasts.

There does in fact exist a potential forecasting model that can be constructed directly using LSW processes, that follows quite naturally from our model. This forecasting model is introduced and extensively described in [Bellegem et al., 2003].

6 Conclusion

This project introduced the concept of wavelets, detailing the theory and key concepts behind it, as well as outlining its general applications. We applied wavelet methodologies to analyze non-stationary financial time series, specifically addressing the problem of non-stationarity with Locally Stationary Wavelet (LSW) processes. Through a time-scale analysis of two major stock indexes, the $S\&P$ 500 and the FTSE 100, we gained insights into the variance over time at different scales using the Evolutionary Wavelet Spectrum (EWS) and examined underlying linear relationships between lagged observations through the localized autocorrelation. Our investigation identified key periods of stock market volatility. We gained insights into the long term impact of different historical events such as the COVID-19 pandemic in 2020 and the influence of various factors such as the raising of interest rates/big tech scrutiny on the market in 2018. We were able to corroborate our findings with real historical events and the influence of substantial economic factors, demonstrating that our results are generally consistent with reality. The lack of overwhelmingly anomalous or overtly incorrect results further validated the sensibility of our model. Our analysis also revealed the non-stationary of the datasets. All of which strongly imply that we have constructed a sensible model. While our analysis uncovered key features of the time series, there are potential improvements to be made. These include adopting non-dyadic wavelet analysis techniques or constructing a hypothesis test to identify anomalous data. Although complex, these improvements would formalize our analysis and enhance the accuracy of anomaly detection.

In conclusion, our investigation confirms that wavelet methods are an appropriate and effective tool for analyzing non-stationary financial time series, through which we can identify key underlying trends and hidden correlations. Coupled with machine learning principles, insights from properly conducted analyses could significantly inform forecasting models, providing valuable guidance for future financial predictions and strategies.

7 References

References

[Can, 2024] (2024). Canva. https://www.canva.com.

[Akaike et al., 1980] Akaike, H., Granger, C. W. J., Joyeux, R., Newbold, P., Nicholls, D. F., Quinn, B. G., Priestley, M. B., Zhao-Guo, C., and Hannan, J. (1980). Journal of time series analysis volume 1 issue 1. *Journal of Time Series Analysis*, 1(1).

[Andriy, 2018] Andriy, S. (2018). *Wavelet Transform in Financial Time Series Analysis: Denoising and Forecast*. PhD thesis, Kent State University.

[Bajaj, 2020] Bajaj, N. (2020). *Wavelets for EEG Analysis*.

[Bellegem et al., 2003] Bellegem, S. V., Fryzlewicz, P., and Sachs, R. (2003). A wavelet-based model for forecasting non-stationary processes.

[Brito et al., 1998] Brito, N., Souza, B., and Pires, F. (1998). Daubechies wavelets in quality of electrical power. In *8th International Conference on Harmonics and Quality of Power. Proceedings (Cat. No.98EX227)*, volume 1, pages 511–515.

[Cohen, 2018] Cohen, M. X. (2018). A better way to define and describe morlet wavelets for time-frequency analysis. *bioRxiv*.

[Dahlhaus, 1997] Dahlhaus, R. (1997). Fitting time series models to nonstationary processes. *Annals of Statistics*, 25:1–37.

[Fryzlewicz and Nason, 2006] Fryzlewicz, P. and Nason, G. P. (2006). Haar–Fisz estimation of evolutionary wavelet spectra. *Journal of the Royal Statistical Society Series B*, 68(4):611–634.

[Government, 2018] Government, U. (2018). Eu exit: Long-term economic analysis technical reference paper.

[Grattan-Guinness, 1990] Grattan-Guinness, I. (1990). *The entry of Fourier: heat theory and Fourier analysis, 1800–1816*, pages 583–632. Birkhäuser Basel, Basel.

[Haar, 1910] Haar, A. (1910). Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69:331–371.

[Hartmann, 2016] Hartmann, D. L. (2016). Section 9: Wavelets. https:

//atmos.uw.edu/~dennis/552_Notes_9.pdf. [Online; last accessed 05-January-2024].

[Ma et al., 2003] Ma, J., Xue, J., Yang, S., and He, Z. (2003). A study of the construction and application of a daubechies wavelet-based beam element. *Finite Elements in Analysis and Design*, 39(10):965–975.

[Mahmoodabadi et al., 2005] Mahmoodabadi, S., Ahmadian, A., and Abolhasani, M. (2005). Ecg feature extraction using daubechies wavelets.

[Mallat, 1989] Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693.

[Nason, 2008] Nason, G. (2008). *Wavelet Methods in Statistics with R.* Springer.

[Nason, 2022] Nason, G. (2022). *wavethresh: Wavelets Statistics and Transforms.* R package version 4.7.2.

[Nason, 2023] Nason, G. (2023). *locits: Tests of stationarity and localized autocovariance.* R package version 1.7.7.

[Nason et al., 2000] Nason, G. P., Sachs, R. V., and Kroisandt, G. (2000). Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(2):271–292.

[Pollock and Cascio, 2006] Pollock, D. and Cascio, I. L. (2006). Non-dyadic wavelet analysis. Department of Economics, Queen Mary College, University of London.

[Priestley, 1965] Priestley, M. B. (1965). Evolutionary spectra and non-stationary processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 27(2):204–237.

[Reserve, 2019] Reserve, F. (2019). Monetary policy report – february 2019, part 2: Monetary policy.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(4):623–656.

[Vošvrda and Schurrer, 2015] Vošvrda, M. and Schurrer, J. (2015). Wavelet coefficients energy redistribution and heisenberg principle of uncertainty. https://library.utia.cas.cz/separaty/2015/E/vosvrda-0449775.pdf. [Online; last accessed 05-January-2024].

[Walker, 2008] Walker, J. (2008). *A Primer on Wavelets and Their Scientific Applications.*

[Yang et al., 2016] Yang, L., Cai, X. J., Zhang, H., and Hamori, S. (2016). Interdependence of foreign exchange markets: A wavelet coherence analysis. *Economic Modelling*, 55:6–14.

[Ülo Lepik and Hein, 2014] Ülo Lepik and Hein, H. (2014). *Haar Wavelets: With Applications.*