Do Day-Trading Gurus Obey the Laws of Entropy?

Tobias Rodriguez del Pozo

4th Year Project Report
Computer Science
School of Informatics
University of Edinburgh

2022

Abstract

The study of using past prices to predict future ones is a problem as old as financial markets themselves. In an age of stock "savants" selling courses in the hundreds of dollars, zero-commission apps allowing millions of people to day trade, and an explosion of those who claim to have found price patterns that all but guarantee profit, a methodical approach to the study of price-action trading is needed.

To that end, this project aims to apply entropic methods to discern the effectiveness of different price patterns, and apply them to trading strategies. These methods imply that pure candlestick pattern based trading is unlikely to be a learnable system, and consequently that pure price-action trading on single financial instruments are unlikely to be profitable.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Tobias Rodriguez del Pozo*)

Acknowledgements

Thank you to Dr. Valerio Restocchi, for his help throughout the project, and continual advice.

Thank you to everyone who has supported me for the last four years: my parents, Adam, Fabian, Gregor, Graham, Fred, Dr. Sean Brocklebank, and Dr. Julian Bradfield.

Contents

Chapter 1

Introduction

1.1 Outline

Over the last 20 years, 95% of all professional investment funds in the US investing in large-cap stocks have failed to outperform the S&P 500 [3]. For investors in smaller and mid-sized companies the story is much the same; 91% and 93% failed to outperform their respective indices (S&P MidCap and SmallCap) [3]. In fact, across all of the indices, S&P Global tracked as part of their latest SPIVA (Standard and Poor's Indices Versus Active) report, the best performing active investors are in real estate, where 84% *underperformed* relative to their benchmark [3].

The problem is clear: virtually every investor loses money relative to the market. This problem is as old as financial markets themselves. In his 1688 book, *Confusión de Confusiones*, Joseph de la Vega describes financial markets as:

> *"Piedra de toque de los sagaces y piedra de túmulo de los atrevidos"* [24]

Translating to "a touchstone for the intelligent, and a tombstone for the audacious".

In the modern-day, where trading is cheaper and easier than ever before, there has been a surge in *technical analysis* [19], which is trading based on using past prices to predict future ones. This has ushered in a new wave of "Wall Street Gurus", day-traders such as Rayner Teo, Ryan Scribner, and The Lifestyle Trader, who claim to have solved the market.

Using price patterns, or in some cases star alignments[1], they propose that by using their methods, a profit is guaranteed. Though such coveted information is typically locked away under a $500 price tag.

On the other hand, the *Efficient Market Hypothesis* (EMH) suggests that any form of technical analysis will not consistently beat the market, as the price of an asset is reflective of all information known about it [9]. The literature regarding technical analysis remains divided and there is no clear consensus on the profitability of it [20],

[1] No, we're *not* kidding: https://en.wikipedia.org/wiki/Planetary_Stock_Trading

meaning that further investigation into the information that price changes contain is needed.

Consequently, with 300 years of hindsight, modern mathematics, and computers on our side, we seek to answer the following question:

> What can entropy tell us about the informational content of prices?

The central finding of this report is that through the lens of entropic methods, irrespective of asset class or time horizon, a time series of prices is highly unlikely to be informative. Specifically, by sampling a given series of prices, these methods indicate that we are unable to learn from them and that we can distinguish very little (if anything) about the underlying market.

The body of this report is composed of 7 chapters.

- Background (Chapter 2) outlines the existing literature both in technical analysis and complex systems and introduces the mathematics behind both methods used.
- Methodology (Chapter 3) gives the methodology used to analyze financial time series and subsequently apply the techniques outlined in Chapter 2.
- Findings (Chapter 4) discusses the results of experiments run on different markets, assets, and time ranges.
- Applications (Chapter 5) explains a handful of simple trading strategies developed using the findings.
- Discussion (Chapter 6) lists some possible improvements for trading strategies and discusses the findings.
- Conclusion (Chapter 7) evaluates the experiments run in Chapters 4 and 5 and comments on future directions for the body of work.

1.2 Definitions

- *Financial Instrument* — any asset that is traded in a financial market. These include (but are not limited to) stocks, bonds, cryptocurrencies, currencies, and commodities.
- *Technical Analysis* — using a time series of past prices to predict future ones.
- *Fundamental Analysis* — looking at the business model, cash flows, balance sheets, etc. of a particular company to determine whether or not to buy/sell it.
- *Speculation* — buying an asset in hopes that it will appreciate, or selling (shorting) an asset in hopes that it will depreciate.
- *Investing* — buying an asset whilst having a *direct* say in how it is controlled, eg. purchasing a majority stake in a company to change its business model.
- *Signal* — information that one uses as the “trigger” to buy or sell a financial instrument, eg. if the price is above $10, buy, and if it is above $15, sell.
- *Candlestick* — characterizing price changes in a given time period by a *quadruple*, [Open Price, Close Price, High Price, Low Price]. If the Open Price is greater than the Close Price, this is a *bearish* (downwards) candle, colored red or black. If the Close Price is greater than the Open Price, this is a *bullish* (upwards) candle, colored green or white.
- *Body* — the difference between the Open Price and the Close Price.
- *Shadows* — the *upwards* shadow is the difference between the High Price and the Body. The *downwards* shadow is the difference between the Low Price and the Body.
- *Candlestick Pattern* — a collection of one or more candlesticks, these are normally given names, such are *engulfing*, *three crows*, etc.
- *Alpha* (α) — how much a trading strategy outperforms the market.
- *Position* — what trades you currently are engaging in, ie. what assets you have bought (or sold).
- *Long Position* — buying an asset, expecting the price to increase.
- *Short Position* — selling an asset *borrowed* from another party, hoping to buy it back when the price of the asset falls.
- *Closing a Position* — selling an asset you had previously bought, or buying *back* an asset you had previously shorted.

Chapter 2

Background

This chapter begins by providing an overview on the existing literature on technical analysis, noise, and trading. Then, we introduce the mathematics, notation, and set-up for using entropy to find maximally informative samples of a system. Finally, we explain where this report fits in to the broader body of work in finance and statistical mechanics.

2.1 Technical Analysis

2.1.1 Candlestick Patterns

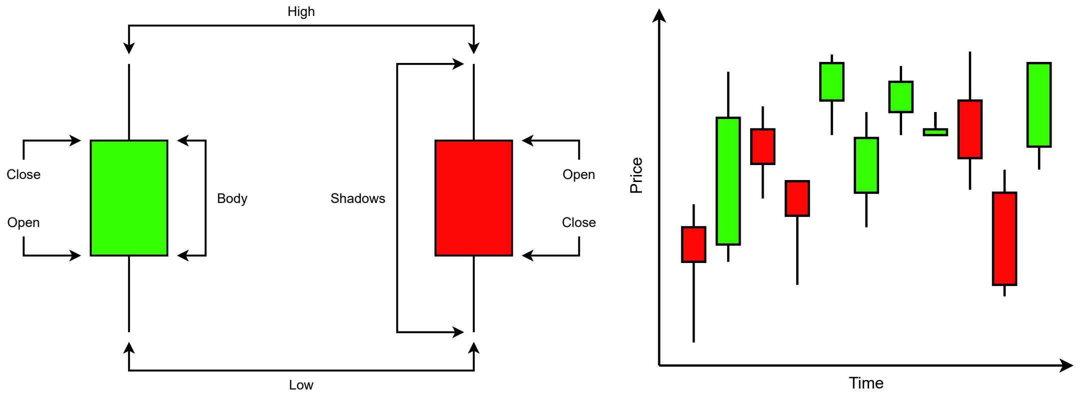


Figure 2.1: Anatomy and plot of candlesticks.

One of the most common ways to represent a time series of prices is through candlestick patterns or candlestick charts. The way to construct these is by taking a series of prices, dividing them into categories of a particular time period, and then calculating prices at the beginning (Open Price) and end (Close Price) of that period, in addition to the maximum and minimum prices throughout the time period. These four numbers form a *candle*, and a series of candles can then be plotted on a graph. Throughout this report, we take candlestick patterns to mean daily intervals, as they are considered the most common and reliable time interval to use [20].

Each candle is composed of a “body”, which is the difference between the open and close prices, and a “shadow” or “wick”, which are the high and low prices for the time interval. A candle can then be colored green (“bullish”) if the close price is greater than the open indicating that the price moved up during the time period, or red (“bearish”) if the opposite is true. Modern candle patterns were first popularized in Europe and North America by Steve Nison in 1990 [18] and have become virtually ubiquitous in financial time series analysis.

To classify different types and patterns of candles, many of them are given names depending on the relationship between the open, high, low, and close prices. For instance, a “Hammer” is when the high price is equal to either the open or close, but the low price is significantly lower than both the open and the close. This would represent the price falling greatly and then rising back up close to where it opened. Some more named examples can be found in Figure 2.2.

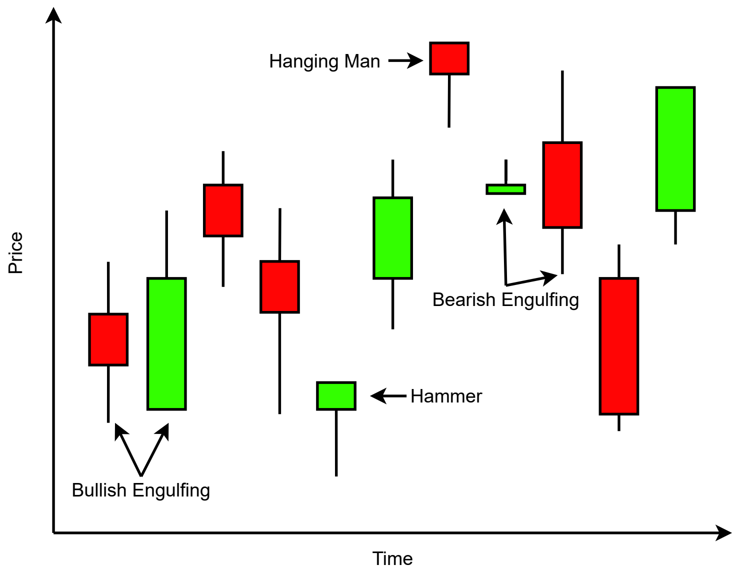


Figure 2.2: Hammer, Hanging Man, and Engulfing Patterns.

2.1.2 Trading Theory

The core theory behind technical analysis is that of using past prices to predict future ones. Using candlestick patterns, a common way to do this is to first identify a candlestick pattern, check if it is associated with either an upwards or downwards change in price and then act on that information by either buying or selling.

So, as an example, suppose that the “Bearish Engulfing” pattern from Figure 2.2 historically corresponded to a decrease in price during a given time period. Then, if at the end of a given day, we saw that a “Bearish Engulfing” had occurred over the last two days, we would look to sell that asset the following day, expecting a further decrease in price in the future.

To that end, one of the most common goals of technical analysis is finding reversals [20]. The idea of a reversal is that the price has hit a temporary “peak” or “trough” in the market and that it will reverse in direction in the following days. The way to profit off of finding a reversal, therefore, is simply by buying at a trough and selling at the peak, and vice versa if one is short selling. By doing this, a speculator aims to profit from the difference between the high point and low point of the price.

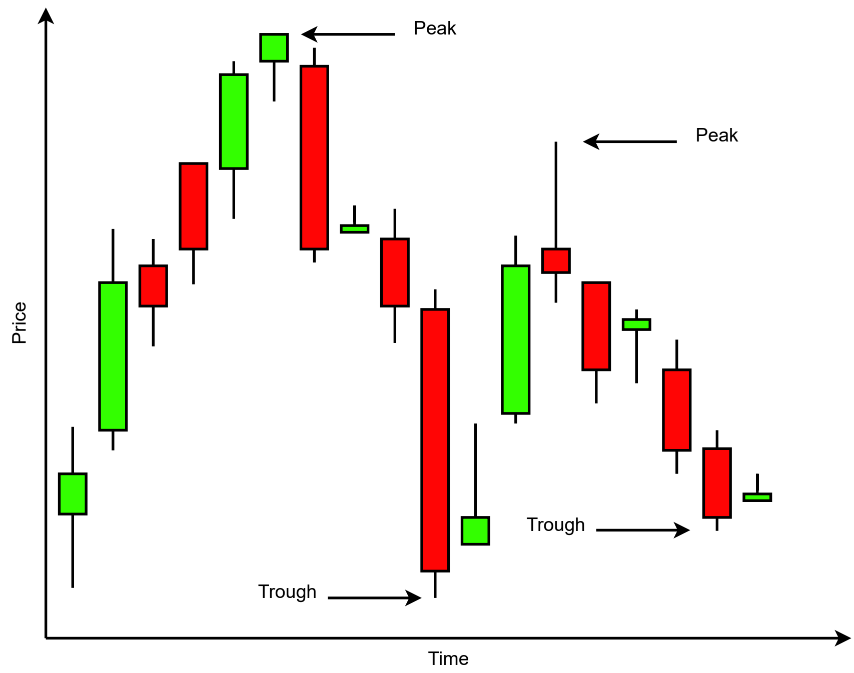


Figure 2.3: Graph showing peaks/troughs on a candlestick chart.

In Figure 2.3, there is a labeled graph of this where if one were able to know the future would time reversal trades to perfectly coincide with the peaks and troughs in the market.

2.1.3 Evidence

The literature on the feasibility of any form of technical analysis is highly divided [19]. Eugene Fama, Nobel Prize winner and widely considered the "father of modern finance" [7], first formalized the "Efficient Market Hypothesis" (EMH) [9] in 1970. For preciseness, we take the EMH to mean the *semi-strong* EMH, which Fama posits occurs when the price of a financial instrument accounts for all public information about the stock [9]. That is, the only way to profit from speculation is through insider information.

The EMH goes further than simply refuting technical analysis, theorizing that any investment that doesn't rely on insider trading will not outperform the market in the long run.

Technical analysis is both increasing in popularity, and there is a growth of literature with empirical findings of successful trading strategies using technical analysis [19]. However, studies in technical analysis are normally riddled with methodological mistakes, chief among which is data snooping, ie. using hindsight [20]. Although blatant mistakes are rare (such as failing to correctly partition training and backtesting datasets), more subtle mistakes are commonplace.

The most common forms of data snooping include tailoring a trading strategy to a fixed asset for a fixed period of time and as a consequence experiments are not run on large enough datasets [20]. Additionally, studies frequently construct trading strategies *ex-post*, whereby upon examination of a particular asset, a trading strategy is developed around it and then tested. Given that any trading strategy requires some level of subjective judgment, this inadvertent data snooping often means that results are not generalizable [19].

The consequence of these two things is that when a research paper on technical analysis is written, it is for the most part ignored and quickly falls into obscurity. Therefore, it's unsurprising that the most recent meta-study on technical analysis was published in 2007 (with an addendum in 2011) [19]. As such, the literature on technical analysis remains fragmented and controversial.

This is particularly problematic in candlestick-based technical analysis, where the evidence is particularly contested. On the one hand, proponents of it claim that candlestick patterns are highly generalizable, whereby the same patterns can be applied across many different assets to achieve profits. Yet, on the other hand, virtually all papers with empirically successful findings only found them for very specific patterns in very specific assets [11] [14].

2.1.4 Noise Reduction

The double-edged sword of noise in financial markets is best explained by Fischer Black in his 1986 paper, "Noise.":

> *Noise creates the opportunity to trade profitably, but at the same time makes it difficult to trade profitably* [5].

Noise in markets increases liquidity but distorts prices.

2.1.4.1 Noise & Candlesticks

The amount of data produced for a single financial instrument is enormous. For instance, the Bitcoin-US Tether (BTC-USDT) market on the Binance exchange regularly sees over 100,000 trades per day [4], and order book information is significantly larger than this. Moreover, this is only a single asset on a single exchange; markets for Bitcoin also include Bitcoin-USD Coin (BTC-USDC), Bitcoin-TrueUSD (BTC-TUSD), and Bitcoin-Binance USD (BTC-BUSD) to name a few. These markets also exist on multiple exchanges meaning that the daily trades are much higher than this.

However, virtually all of these trades do not give meaningful information about the future price of Bitcoin. This is where candlesticks are useful, as they can turn hundreds of thousands of prices, each representing a trade, into four prices for a single day.

2.1.4.2 Noise & Technical Analysis

Noise also plays a role in technical analysis in terms of one of the theoretical explanations to support the profitability of technical analysis: noisy efficient markets.

In a noisy efficient market, the current price of an instrument does *not* contain all available information regarding it because we have some variable level of noise added to the current price [20]. Therefore, it would follow that as the amount of noise changes, so too does the price.

This hypothesis does have some empirical findings namely that if there was no noise trading in financial markets, very little trading would occur, as the only thing that could change prices was new information being revealed about the asset [20]. However, given the amount of trading that does occur, then it follows that there is a high level of noise trading in all markets.

This means that assuming that the EMH holds in the long run, ie. that the market will "correct" from the noisy price to the true price and if you can separate noisy price changes from true price changes, then there exists an opportunity to trade profitably when the disparity between the two is big enough.

Therefore, it follows that if market participants hope to use technical analysis to trade profitably, they need some way to reduce this noise.

To that end, we introduce the entropic methods in the following section, with the aim of discerning *informative* price movements from noise.

2.2 Entropy & Complex Systems

2.2.1 Overview

We begin by giving an outline of complex systems: systems made up of many components that interact with each other and whose behavior is intrinsically near-impossible to model. The seminal paper concerning this report is Matteo Marsili, Iacopo Mastromatteo, and Yasser Roudi's 2013 paper "On Sampling and Modeling Complex Systems" [16].

Consider some complex system, S. This system takes in variables $s = s_1, \ldots, s_n$, and seeks to optimize some unknown objective function $U(s)$. We can observe some of the variables that go into the system, as well as the states of the system, ie. how the system behaves to optimize $U(s)$.

Let $\underline{s} = s_1, \ldots, s_i$ be the variables that are known to us and $\bar{s} = s_{i+1}, \ldots, s_n$ be the variables that are unknown to us [16]. Therefore, $s = (\underline{s}, \bar{s}) = (s_1, \ldots, s_i, s_{i+1}, \ldots, s_n)$. Note that we don't know what we don't know: n is unknown.

Moreover, considering that the system seeks to optimize $U(s)$ it follows that there exists an optimal solution for the variables, s^*, such that:

$$s^* = \arg\max_s U(s)$$

For example, consider a piece of text. The author, through writing, is trying to achieve *something*. The precise nature of this *something* is unknown to us and therefore forms the unknown objective function, U. We assume that the author is trying to optimize U and that the words were chosen by the author play a role in this.

In this case, our known variables are the words in the text, which therefore constitute $\underline{s}$. However, there are many other variables that we can't observe and likely don't even know of, such as editorial constraints, cultural influences, or how the author was feeling that morning [2]. These constitute $\bar{s}$.

If we assume that every word is chosen optimally in regards to U, then it follows that each word in the text is an observation of $\underline{s}^* \in s^*$ [16]. From now on, we take $\underline{s}$ to mean $\underline{s}^*$, as every observed state is assumed to be maximizing U.

2.2.2 Sampling the System

Only using $\underline{s}$, what can we learn about the system?

We begin by sampling the system M times. We now have $\underline{S} = \{\underline{s}^1, ..., \underline{s}^M\}$ observations, each corresponding to an optimal solution for $U(s)$. Next, we count the frequency of each unique $\underline{s}^i \in \underline{S}$, denoted by $K_{\underline{s}^i}$. The discrete probability distribution of each $\underline{s}^i$ is given by:

$$P(\underline{s}^i) = \frac{K_{\underline{s}^i}}{M} \text{ [16]} \tag{2.1}$$

We can then derive the discrete probability distribution for $K_{\underline{s}^i}$:

$$P(K_{\underline{s}^i}) = \frac{km_k}{M} \text{ [16]} \tag{2.2}$$

Where m_k is the number of states observed k times.

In our example of a text, Equation (2.1) would involve generating the relative frequencies of each word. Equation (2.2) would calculate the relative frequencies of the number of unique words appearing exactly k times in the text.

2.2.3 Entropy

In 1948, Claude Shannon introduced the concept of information entropy, in his paper "A Mathematical Theory of Communication" [21]. In it, he defines the informational entropy of a discrete random variable X, with $x_1, ..., x_n$ outcomes, and probabilities $P(x_1), ..., P(x_n)$ to be

$$H(X) = -\sum P(x_i) \ln(P(x_i) \quad \text{[21]}$$

The key idea behind entropy is that we are measuring the average information gained based on the outcome of a trial.[21] For instance, the informational entropy of flipping a fair coin tells us that we are unable to gain any information about the next flip, and

[2] We note that these unknown unknowns also means that the exact nature of U is also unknown to the author.

therefore entropy is maximal. Equally, flipping a biased coin would give us a lower entropy, as due to the bias we can gain information about the outcome of the next flip.

Therefore, considering our previous set-up, where we defined two discrete probability distributions, $P(\underline{s}^i)$ and $P(K_{\underline{s}^i})$, it follows that we can calculate their entropies to determine how informative they are. To that end, we introduce the two defining equations of this report:

$$H[\underline{s}] = -\sum_{\underline{s}} \frac{K_s}{M} \ln \frac{K_s}{M} = -\sum_k \frac{km_k}{M} \log \frac{k}{m} \text{ [16]} \tag{2.3}$$

$$H[K] = -\sum_k \frac{km_k}{M} \log \frac{km_k}{m} = H[\underline{s}] - \sum_k \frac{km_k}{M} \log m_k \text{ [16]} \tag{2.4}$$

These are the key findings made in Marsili's paper.

This method is well-suited for complex systems as neither $H[\underline{s}]$ nor $H[K]$ rely on $\underline{s}$, meaning that we are able to derive information about the system only through the distribution of $\underline{s}$ throughout the samples, not $\underline{s}$ itself [16]. This is important as it allows us to determine which $\underline{s}^i$ are the most informative independently of our understanding of them.

Moreover, as seen in Equation (2.4) $H[K]$ is a function of $H[\underline{s}]$, meaning that as we vary the size of M by grouping sequences of observed states together we would expect a relationship between the two which we can then interpret. Marsili argues that by doing so we should expect to see the states with the highest $H[K]$ to be the most informative [16].

2.2.4 Breaking Things Down

The core concept behind this method is that by taking our M samples of the system as a single state, calculating $H[\underline{s}]$ and $H[K]$, then splitting up the samples into two states of size $M/2$, and repeating the process until we terminate with M states of length one, the states which contain the highest $H[K]$ are the most informative [16].

We define the following procedure to calculate $H[\underline{s}]$ and $H[K]$.

1. Get every unique sample in $\underline{S}$.
2. For every sample:
    - 2.1. Calculate $H[\underline{s}]$ and $H[K]$ with 1 state of length M.
    - 2.2. Find the next smallest divisor of M, d.
    - 2.3. Calculate $H[\underline{s}]$ and $H[K]$ with d states of length M/d.
    - 2.4. Normalize $H[\underline{s}]$ and $H[K]$ by dividing them both by $\log(M/d)$.
    - 2.5. Repeat 2.2-2.4 until no divisors of M remain.
3. Output every sample and their sequence of associated $H[\underline{s}]$ and $H[K]$.

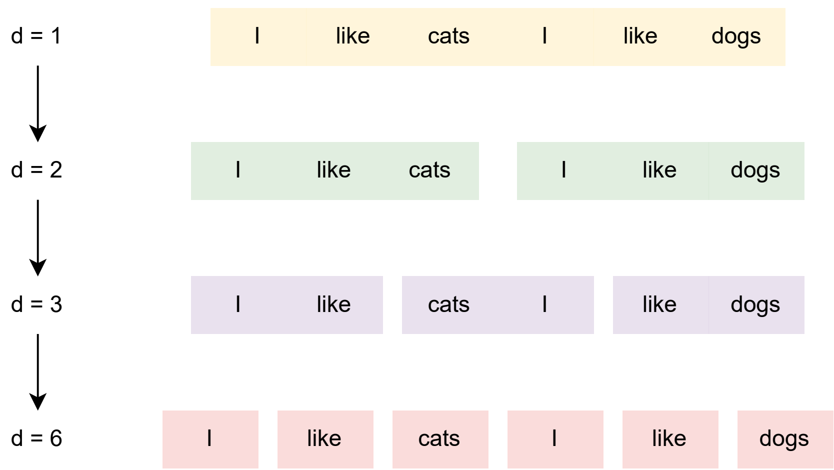


Figure 2.4: Division procedure for generating $H[K]$ and $H[\underline{s}]$ for the sentences "I like cats. I like dogs."

Then, we can look at the sequence of values $H[K]$ and $H[\underline{s}]$, and determine the maximally informative states of the system; those that give us the most insight as to the underlying optimization problem, $U(s)$.

Figure 2.4 shows an example of the division for the simple sentence "I like cats. I like dogs.". We point out that in the cases of $d = 2$ and $d = 3$, we build on the notion in Section 2.2.2 and treat the word and state frequencies as the probability distributions for the frequency of the word *in* that group instead of across all observed samples.

2.2.5 Visual Representation of Entropies

One of the best ways to interpret the results of $H[K]$ and $H[\underline{s}]$ calculations from Section 2.2.4 is to plot them on a graph, given that $H[K]$ is a function of $H[\underline{s}]$. In systems that we can learn from, those in which $\underline{s}$ provides us with some information regarding U, we would expect to see a clear separation between the $H[K]$ values for informative versus uninformative states. For systems whose observations of $\underline{s}$ provide us with no information, we would expect no clear separation in terms of the informational content $H[K]$ that each state gives us.

To illustrate this point, we ran the procedure in Section 2.2.4 on two different datasets.

First, on a list of strings chosen uniformly at random from 8 potential options.

Second, on Mary Shelley's novel "Frankenstein". We plotted the normalized $H[K]/H[\underline{s}]$ graphs (by dividing both $H[K]$ and $H[\underline{s}]$ by $\log M$), for selected states (words), shown in Figure 2.5.

For the random words, we sampled them uniformly 75,000 times (the number of words in Frankenstein). As seen in Figure 2.5, for the text made up of entirely random words, there is virtually no difference between them. Their peak is also relatively low with none of them exceeding $H[K] = 0.4$.

For Frankenstein, we see that "creature" (what Victor Frankenstein's creation is referred

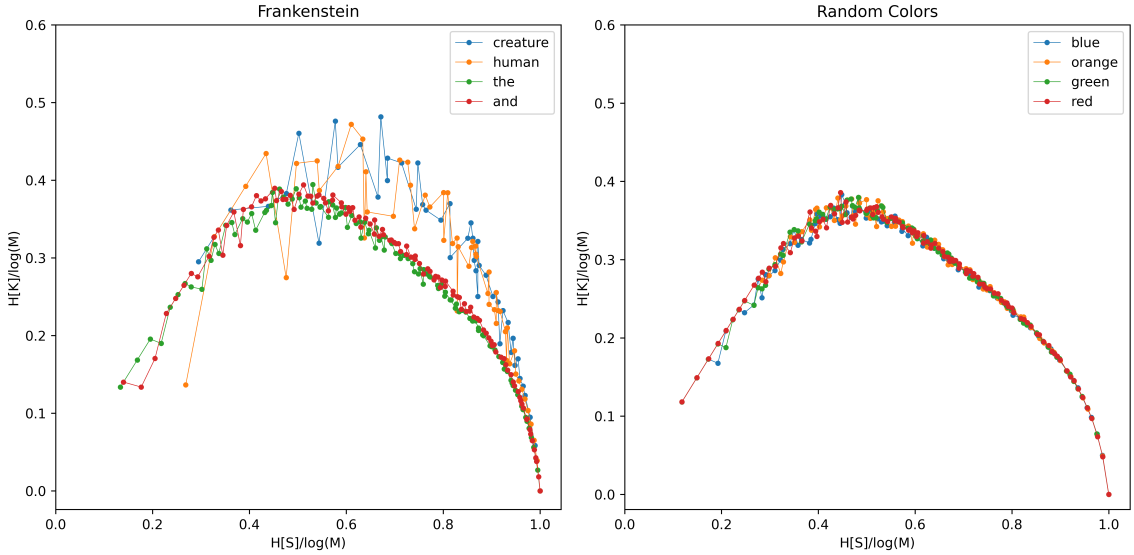


Figure 2.5: $H[K]$ and $H[\underline{s}]$ for Frankenstein and randomly chosen words.

to as throughout the novel) and "human" have higher $H[K]$ values for virtually every value that $H[\underline{s}]$ takes, compared to "filler" words such as "the" and "and". This indicates that they contain more information regarding Shelley's unknown optimization function U.

This method is not perfect, with words such as "day", "other", and "came" all ranking higher than some words which we would *subjectively* consider to be important, such as "Elizabeth" (the fiancé of Victor Frankenstein), "death", and even "Frankenstein", despite all of them appearing frequently throughout the book.

Therefore, we argue that states which contain the highest implied informational content must satisfy two conditions.

First, there needs to be a clear difference in both the average and maximal values of $H[K]$. That is, when plotting $H[K]$ as a function of $H[\underline{s}]$, informative states should take higher $H[K]$ values than those that are uninformative for virtually all $H[\underline{s}]$.

Second, these states should have higher absolute informational content. Looking at our random colors example above, we point out that even a random state has a peak $H[K]$ of around 0.4. Therefore, it would follow that any informative states ought to have a higher $H[K]$ than this, irrespective of the $H[K]$ values of the other states.

2.2.6 Order Matters

We also point out that this method is distribution agnostic: even if the underlying distribution of samples is not uniform we can still differentiate those $\underline{s}$ which are uninformative. The reason for this is that because we consider sequences of $\underline{s}$, then the placement of each $\underline{s}$ matters. That is, if a state is uncommon but roughly uniformly distributed throughout the samples as M/d decreases in size then it would still be considered noise.

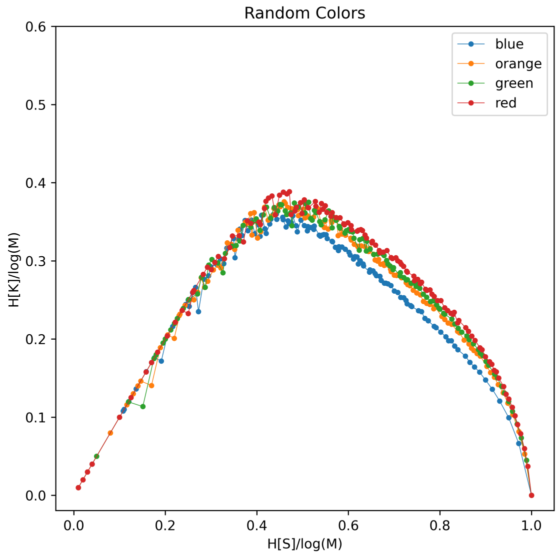


Figure 2.6: Random Colors that Follow a Zipf Distribution

Figure 2.6 shows this, whereby a sample of 75,000 states were selected from a Zipf Distribution[3], however, we can see that there is still very little difference between them.

This is because this method differentiates the *placement* of the states in the sequence of samples $\underline{s}^1, \ldots, \underline{s}^M$ from merely the frequency of the states in the samples.

2.2.7 Importance of Many Samples

In this analysis, arguably the most important thing is a large sample size: M should be as large as possible. As an illustrative example, we plotted $H[K]$ and $H[\underline{s}]$ for random words, only with $M = 200$ instead of $M = 75,000$.

As we can see in Figure 2.7, not knowing that the data is uninformative *a priori*, we might conclude that "blue" is highly informative relative to "orange". However, this is a consequence of the sample size, rather than the information we can gain from the underlying system.

Of course, we don't need the true probability distribution and sequences of $\underline{s}^i$ of the system to model it, however, we ought to get as close as possible to the distribution of the underlying system, and to do so we need as many samples as possible.

2.2.8 Power Law Considerations

Marsili argues that the systems which are *most* suited for these techniques are those whose $\underline{s}$ follow power-law distributions, and specifically those that most closely follow a power-law distribution with an exponent of $\mu = 2$, Zipf's Law [16].

[3] A power law distribution with an exponent of 2, whereby the second most common element is half as common as the first, the fourth element is half as common as the second, and so on. A more detailed discussion can be found in Section 2.2.8

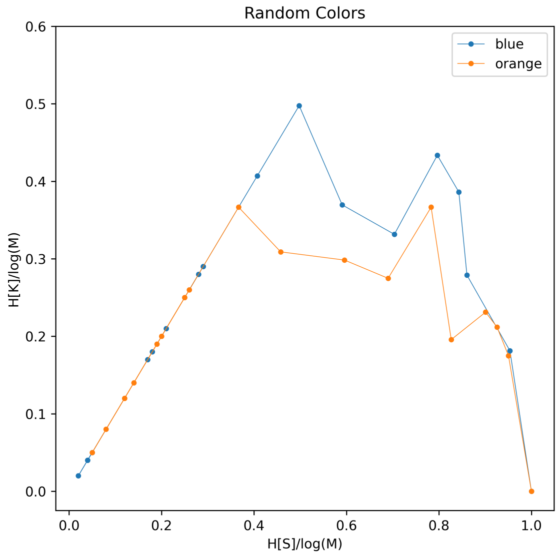


Figure 2.7: Random Colors with a Small Sample Size

That is, if we have M samples of $\underline{s}$, and rank them by frequency $K_{\underline{s}^i}$, we would expect the systems for which we can derive the *most* information from to have the frequency inversely proportional to one over the rank squared:

$$K_{\underline{s}^i} \propto \textsc{Rank}(\underline{s}^i)^{-2}$$

This may imply that these methods are *particularly* well suited to most distributions that follow a power-law distribution, especially when $\mu \approx 2$.

Additionally, Marsili indicates that maximal $H[K]$ tends to occur when, following the procedure from Section 2.2.4, the d states of size M/d most closely follow a $\mu = 2$ power law distribution [16].

2.3 Applications

2.3.1 Existing Applications

There have been a variety of papers on applications of Marsili's methods in complex systems. To give some concrete examples:

The process of selecting the number and size of hidden layers in neural networks has been observed to follow this pattern [8]. Treating the process of selecting the optimal neural network configuration from a large hypothesis class as a complex system, with some adjustments, can be used to find maximally informative configurations of the network [8].

Protein structuring to determine the most relevant amino acids in a given cell can be found with this method [16]. Treating the observed amino acids as $\underline{s}$, and then sampling how they behave in a cell, suggests that we can find more relevant amino acids in the cell-function optimization function [16].

Most relevant for this report, however, is the application to find keywords in a text. By treating each word as an observed state of the system, a text is made up of M words. Therefore, given a word we can calculate $H[\underline{s}]$ and $H[K]$, for varying lengths of text (states). The words with the maximal $H[K]$ are deemed to be more informative as to the underlying optimization function of the author, and can therefore be considered keywords [16].

Montemurro and Zanette introduced a similar technique in 2001 [17], purely by looking at $H[\underline{s}]$. They argue that the keywords in a text are those that have the maximal difference in $H[\underline{s}]$, and the entropy of a random reshuffling of the text [17]. The concept behind this is that words with higher $H[\underline{s}]$ give us more information about the underlying system, and because word position is important, the *most* important words no longer provide as much information when randomly shuffled.

2.3.2 New Applications

From our perspective, we can treat financial markets as a complex system. We assume that the markets are trying to optimize some U. Knowing that the price of an instrument is a variable in the system (arguably the most important one), it follows that each price seen in the market is therefore trying to optimize U. We can sample the system by looking at its historical prices. Thus, each price of the market can be treated as an $\underline{s}^i$.

To do so, we view the series of daily classified candlestick prices as “words” in a text, with the full text being composed of a list of the classified candles, each one representing a day. Then, by calculating $H[K]$ and $H[\underline{s}]$ for the candles observed, Marsili’s method would provide us with the most informative ones: those which are the least noisy and most closely related to the underlying U.

To empirically evaluate the predictive success of the most informative candles, we treat them as price signals for a trading model and evaluate their performance across a wide range of assets.

With all of this in mind, we believe that this report serves two purposes.

First, a novel application of existing methods in noise reduction in markets. Most of the academic literature focuses on non-entropic methods, such as jump detection, filters, kernels, etc. Moreover, existing papers on entropy either primarily discuss information flow between news-driven trading activities [13], or representations of data [12].

Second, a discussion about the empirical observations made across markets, and an extensive amount of tests for a trading strategy built using these techniques.

Chapter 3

Methodology

3.1 Setup

To calculate $H[K]$ and $H[\underline{s}]$, we make the following changes to the data.

As discussed in Chapter 2, we treat the prices as daily. Then, we take the [OPEN, HIGH, LOW, CLOSE] prices for the day, and encode them as a list of candlesticks using the technical analysis library, TA-LIB [22]. Then, we apply Marsili's and Montemurro's techniques to this list to find the key candles.

These transformations still preserve the information of the underlying system, as we are simply providing an abstraction away from the raw data, but the behavior of the time series is preserved.

3.2 Library

To run the experiments, we implemented a Python [10] library, made of three parts. Code for both the ENTROPY and CANDLEPATTERNS classes can be found in Appendix A.1 and Appendix A.2, respectively.

First, there is the ENTROPY class, which, given *any* list of strings will run both Marsili's and Montemurro's methods and subsequently rank the most informative words. This can be run on any system whose sampled states $\underline{s}$ can be represented as a list of strings, and can therefore be used for non-financial applications, such as to find keywords in a book (Figure 2.5, Figure 2.7, Figure 2.6).

Second, to apply it to financial markets, we implemented a CANDLEPATTERNS class, which takes any time series of [OPEN, HIGH, LOW, CLOSE] prices, and returns a list of named candles.

Third, we used BACKTESTING.PY [15] to build and test trading strategies using entropy calculations.

3.3 Data Collection

The data used for these experiments are single-day candlestick patterns on a variety of different financial instruments. All data was gathered using the YFINANCE Python package, which is a wrapper for the Yahoo Finance API [2].

We ran the experiments both on the raw data, and the log-normalized data. The reason for log-normalization is because it only looks at relative prices by calculating:

$$p_{norm} = \ln\left(\frac{p_t}{p_{t+1}}\right)$$

Where p is the quadruple of [OPEN, CLOSE, High, LOW], t is the given trading day, and $t+1$ is the following trading day. This means that we can correct for longer-term trends in prices, and instead focus on daily changes. Additionally, log-normalizing has the benefit of taking into account the asymmetric nature of percentage changes, namely that a +1% change in price, followed by a -1% change in price is greater than a -1% price change, following by a +1% change in price.

However, the trading strategies were run only on the raw data to maintain strictly positive numbers. There was also not a significant difference in results between the two data types, as the candle calculations were daily and therefore maintain their characteristics irrespective of log-normalization.

Another consideration is that the fact that there are on average 250 trading days in a year means that many instruments have limited data. Therefore, we picked instruments with more trading days. This does introduce some survivor bias, as stocks that haven't been de-listed from the S&P 500 over many years are more likely to be better investments. However, this is unlikely to make a large difference, as the experiments were run across multiple assets and on 444/500 stocks currently listed on the S&P.

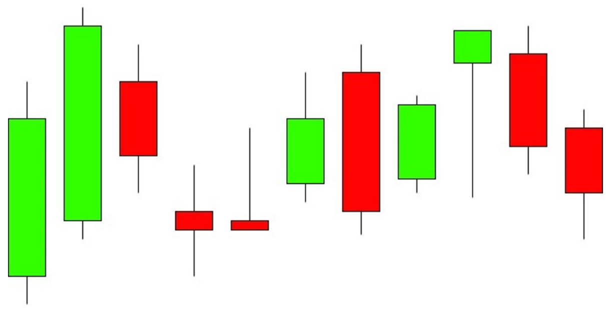
Finally, as Marsili's method requires dividing the data into as many integer-length pieces as possible, we removed elements from the beginning of the dataset to maximize the number of divisors present in the series with the constraint of not removing more the 25% of the original data. This means that we can obtain more data points of $H[\underline{s}]$ and $H[K]$ and therefore better evaluate which candles are the most informative.

3.4 Prices to Candlesticks

To classify the candles, we used TA-LIB in Python which has a classifier for candlestick patterns. Given a series of prices, TA-LIB returns whether or not they correspond to named candles. It can classify 60 different candle patterns, for sequences of one, two, and three candles [22].

Each candle was further classified into "bull" or "bear", depending on whether the open price was higher or lower than the close price. Candles that TA-LIB didn't classify were marked as BULLNONE or BEARNONE, accordingly. In the case of multiple candles being assigned to the same trading day, we gave priority to rarer candle patterns. The reason for this is that from our perspective, less common candlesticks may be more informative.

For instance, this pattern of price-candles:



Would be transformed into the following array

[BULLNONE, BULLMARUBOZU, BEARHARAMI, BEARSPINNINGTOP, BEARGRAVESTONEDOJI, BULLNONE, BEARENGULFING, BULLNONE, BULLHANGINGMAN, BEARNONE, BEARNONE]

3.5 Entropy Calculations

3.5.1 Marsili

The first analysis of the data involved using the calculations outlined previously. We ran the algorithm to find $H[\underline{s}]$ and $H[K]$ for each unique candle observed in the time series.

However, there emerged the challenge of ranking the candles. This is because even though we have several thousand data points per instrument, the scale of the data produced was still relatively small. In Marsili's analysis, he used Charles Darwin's "The Origin of Species" (~120,000 words). Montemurro used "The Complete Works of William Shakespeare" (~900,000 words) and in Figure 2.5 we used "Frankenstein" (~75,000 words).

These datasets are much larger than the data for daily financial markets (~3,000-10,000 data points). Moreover, both of the aforementioned texts have thousands of unique words, whereas TA-LIB is only able to classify a maximum of 122 different candlesticks (60 patterns + NONE, each bull or bear).

This meant that in many cases, there simply were not enough samples of the system to see the distinct differences seen in other work. Candles frequently had equal or similar values for $H[K]$ and there was high variance in the values. When differences in $H[K]$ were large (Figure 4.3), this may have been an outcome akin to that in Figure 2.7: randomness with a small sample size.

However, we are constrained in terms of sample size. The reason we didn't decrease the time period in question for candles (eg. 1-minute intervals) is precisely because we empirically know that they represent noise trading. Therefore, if we were to decrease the time period, we risk only adding noise to the model and not achieving more informative results.

There may also be completely different objective functions for different time periods. Intraday trading may be considered an entirely different complex system and thus trying to infer information about multi-day trends with intraday sampling is unlikely to yield generalizable results.

To overcome these issues we used two ranking methods to evaluate the most informative candles.

3.5.2 Ranking

The first method, used by Marsili, simply takes the maximal $H[K]$ for each unique word (candle) [16]. This works well in most cases, however, due to only having 122 candles, and most commonly 40-60, it meant that multiple candlesticks could have equal maximal $H[K]$ scores.

The second method used is the sum of the $H[K]$ scores. The reasoning for this is that we are interested in candles that not only have high *maximal* information but that on average have more information. Therefore, summing their $H[K]$ scores enables us to distinguish candles that have a high peak, but are low everywhere else, from those that consistently have higher $H[K]$ values.

3.5.3 Montemurro & Zanette

We use the same application in candle patterns, taking the series of candles and shuffling them, and consequently finding the maximal difference in the average $H[\underline{s}]$, and a random shuffling $\hat{H}[\underline{s}]$ [17]. In the case of a tie, the candle with the higher *maximal* difference was given priority.

Chapter 4

Findings

The findings across the board indicated relatively small differences between the most and least informative candles, supporting the claim that prices alone are unlikely to be an informative system. We ran experiments on a variety of different assets, and have included some sample results below.

4.1 Stocks

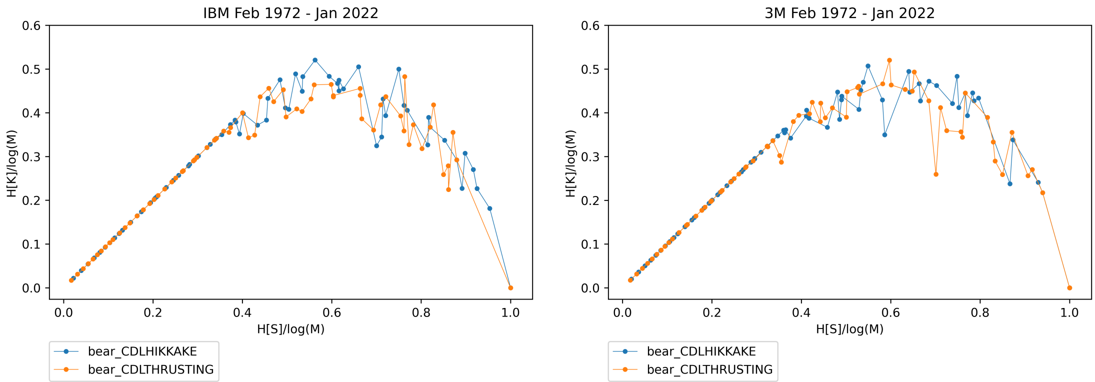


Figure 4.1: IBM Stock (left), 3M Stock (right)

Figure 4.1 and Figure 4.2 show the results of the experiments for two candles: BEARTHRUSTING and BEARHIKKAKE. These two candles were chosen as they ranked the highest for IBM stock. The reason these stocks were chosen was because IBM, 3M, and Lockheed Martin had among the most datapoints available, which we know from [16] implies that we are better able to model the system.

As seen, there is some variation for the same candle across different stocks, however, the differences between the candles are relatively small, even though their ranks vary significantly in terms of the theoretical information value they provide. Both of these factors point towards randomness, after all, if all candles were equally uninformative then we would expect the difference between their entropies to be similar.

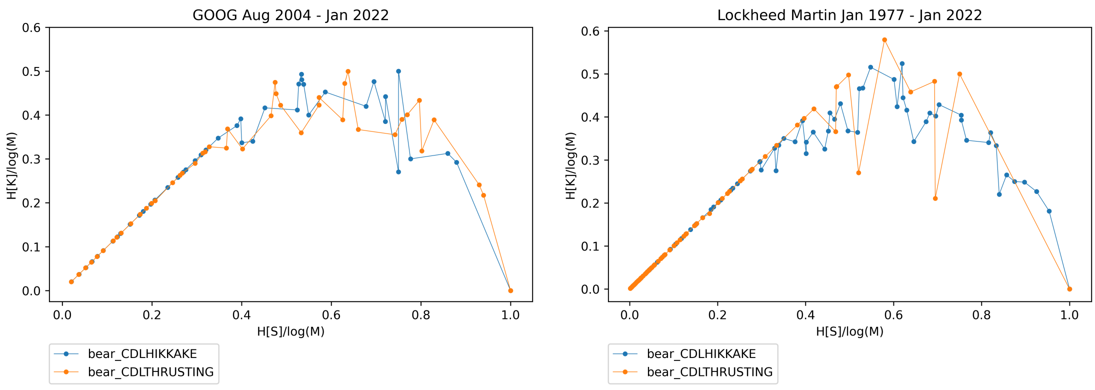


Figure 4.2: Google Stock (left), Lockheed Martin Stock (right)

Taking a look at Lockheed Martin, we see what we believe to be one of the limitations of the data available to us, in the form of the behavior of the BEARTHRUSTING candle. The majority of the data points are clustered in the bottom left portion of the $H[\underline{s}]/H[K]$ graph, but the candle has a high maximal $H[K]$. Increasing the number of samples (and therefore the number of times we would expect a given candle to be sampled), would enable the distribution of $H[K]$ to take on *more* unique values as the size of each state increases, shifting the distribution towards one with more spread out $H[\underline{s}]$ values for that candle.

One possible insight, which again may be due to sheer randomness, is the fact that both candles are lower in absolute terms for Google than for the other stocks. This may indicate that the time series of prices for Google is less informative as a whole than for the others — though with 30 years less of data that may be unsurprising.

4.2 Commodities

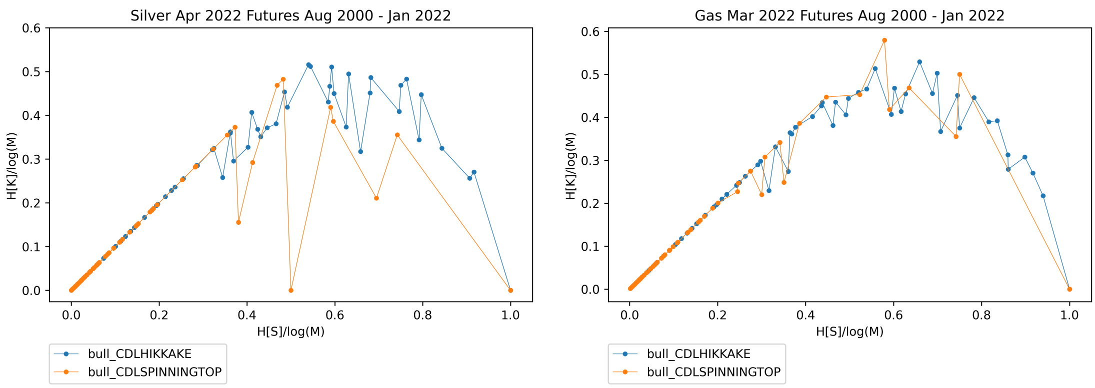


Figure 4.3: Silver Futures April 2022 (left), Natural Gas Futures March 2022 (right)

The April '22 Silver Futures market had the largest difference of any of the experiments (Figure 4.3), both in terms of peak $H[K]$, but also in terms of the area under the curve of $H[K]$. There are two possible explanations for this.

The first is that in running experiments across many different assets, we were bound to "get lucky" in terms of finding a significant result, even if the general distributions of $H[K]$ are random, due to all candles containing the same information (as in Figure 2.7).

The second is that in this particular market, BULLHIKKAKE candles contain much more information as to the underlying objective function. Therefore, should we speculate about price changes, it would be reasonable to begin by examining instances of the BULLHIKKAKE as these methods suggest that it is likely to contain less noise.

The chart on the right shows the same two candles for March '22 Natural Gas Futures. This is where the ranking methodology outlined in Section 3.5.2 becomes relevant, as we observe that the BULLSPINNINGTOP candle has a *higher* maximal $H[K]$, but the BULLHIKKAKE has a higher average $H[K]$.

An important thing to note is that the relative importance of maximal versus average $H[K]$ in this report is primarily a subjective decision. Further investigation into this topic is left for future work.

4.3 Bonds

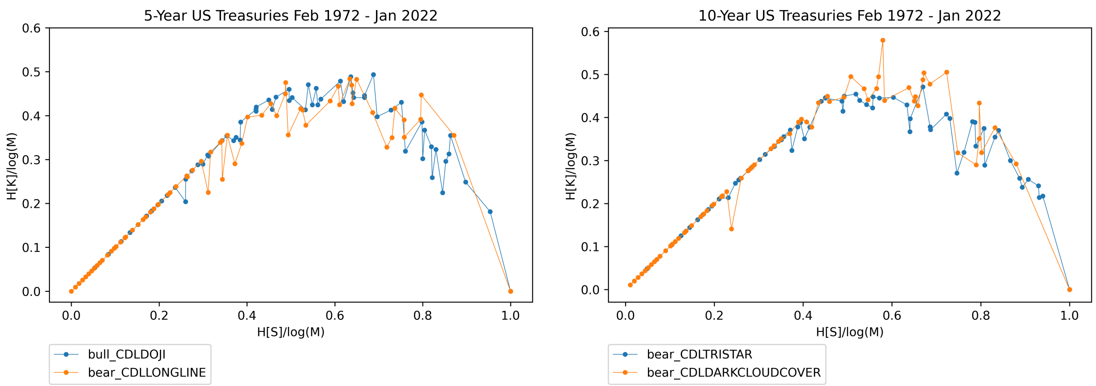


Figure 4.4: 5-Year US Dollar Treasuries (left), 10-Year US Dollar Treasuries (right)

Next, we ran Marsili's technique on 5 and 10-year treasure notes (Figure 4.4). For 5 year treasuries, BULLLONGLINE candles had the maximal $H[K]$. For BULLDOJI, we see that they *tend* to be slightly higher on average, however, the distribution seems to indicate that both candles represent noise. Additionally, their peak $H[K]$ are similar, again suggesting that we are unlikely to be able to learn from the system.

For 10 year notes, we see the most informative candle pattern is a three-candle pattern, the BEARTRISTAR. It has both a noticeably higher maximal $H[K]$ and a higher average $H[K]$.

When comparing the two markets, there are no clear similarities in their maximally informative candles, and this trend holds for 30-year treasuries, too. This supports the hypothesis that we can consider the market for each instrument (especially those the size of the US-treasury market) to be its own complex system.

One possible explanation for treasuries, in particular, might be as follows: a speculator may purchase treasuries of different time periods with different goals, that is, shorter-term treasuries likely serve an entirely different purpose than long-term ones. For instance, an insurance company may purchase 5-year treasuries in anticipation of the expected payouts in that year, but may purchase 30-year treasuries primarily for their cash flows. Therefore, the optimization function of each market participant, and subsequently the market as a whole, is different.

4.4 Foreign Exchange (FX)

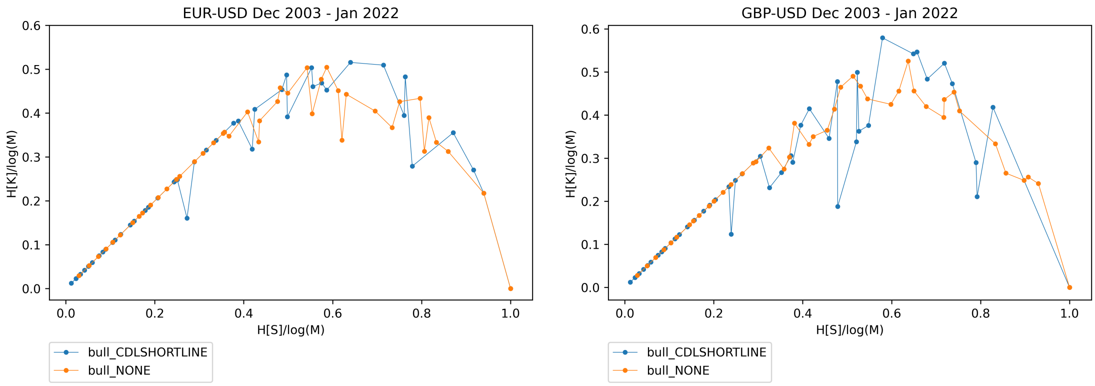


Figure 4.5: Euro-US Dollar FX (left), British Pound-US Dollar (right)

Of interest in this experiment (Figure 4.5) is the difference in maximally $H[K]$ between the two assets. We see that the GBP-USD market has a higher maximal $H[K]$ (0.579) than the EUR-USD market (0.531). This, much like in the Google case explained in Figure 4.2, may imply that we can derive more information from GBP-USD, indicating that the market may be less random.

4.5 Cryptocurrencies

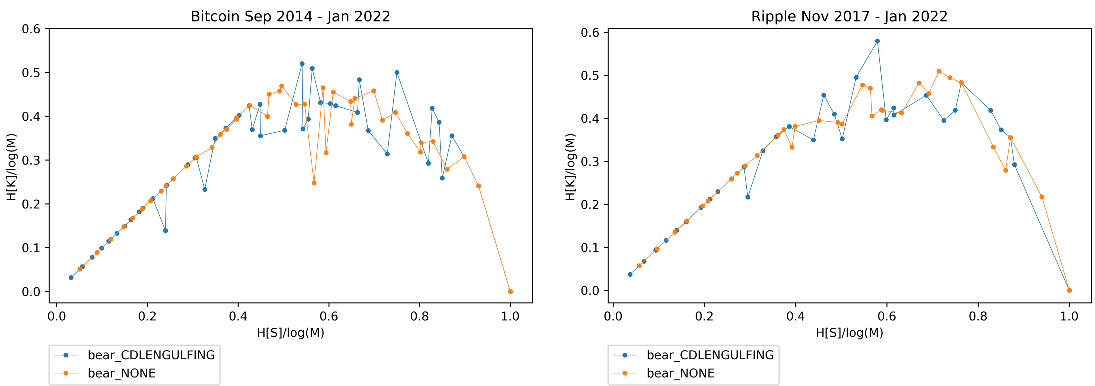


Figure 4.6: Bitcoin-US Dollar Spot Market (left), Ripple-US Dollar Spot Market (right)

Perhaps the most well-known example of both irrational market participants and an extraordinarily high signal-to-noise ratio, we see the following (Figure 4.6).

First, for Bitcoin, there is very little difference in the average between BEARENGULFING and BEARNONE (where TA-LIB is unable to classify the candle). This could imply that given a small difference in information, the total amount of meaning we can extract from the system is significantly less. Moreover, BEARNONE being the least informative candle does (fortunately) suggest that named candles can provide more insight.

Second, if we compare the two candles used for Bitcoin to Ripple, we see that the BEARENGULFING is also more informative than BEARNONE. Perhaps the complex systems of cryptocurrency markets have some similarities.

We see that across markets, with perhaps the exception of Silver Futures, the difference between the implied informational content of the candles tends to be relatively small. This indicates that the underlying prices are random, and therefore not informative. However, to put this hypothesis to the test, we applied these techniques in the form of two trading strategies.

Chapter 5

Applications

Now that we have learned something about the relative importance of each candlestick, we designed some trading strategies utilizing that information. The first strategy outlined is very simple, only using two candlesticks as signals, "common knowledge" about what the patterns might mean in terms of price, and the role that luck plays in trading.

The second strategy is a step up from this, fully automating candlestick selection according to their $H[K]$ values, utilizing stop-losses, and backtesting on several different asset classes.

5.1 A Simple Trading Strategy

Coming back to the discussion of reversals outlined in Section 2.1.2, we define the following goals for our trading strategy.

1. Find the most informative candles.
2. Within those, isolate ones that are *most associated* with a reversal in price.
3. Split the candles into "buy" signals, ie. an upwards reversal and "sell" signals, ie. a downwards reversal.
4. When we see one of those candles, we either buy or sell the instrument.

To implement the trading strategy, we use BACKTESTING.PY, a lightweight Python module built for backtesting trading strategies [15]. In BACKTESTING.PY, we define our strategy as a class, extending the methods in the STRATEGY class in the module. Next, before running the backtest we define INDICATOR variables, which are arrays of data corresponding to buy/sell signals.

Finally, we pass testing data containing [OPEN, HIGH, LOW, CLOSE] prices (candlesticks) and run the backtest. The BACKTEST class steps through each row of the data and indicators, revealing them to the strategy. Upon completion, we receive a summary of the strategy's performance.

In testing these strategies we make the following assumptions:

1. We are always able to execute buy and sell orders at the close price for the day.
2. Transaction costs are fixed at 0.2%. This is realistic in the short term, but probably not in the long term.
3. Our position size never changes, that is, we always buy or sell our entire position size and not portions of our portfolio.
4. If we don't currently hold any of the stock, a sell order acts as a short position.

Let us consider the Lockheed Martin (LMT) stock time series in Figure 5.1.

We first limit our date range to the last 10 years. Then, we split the data into training and testing data, with 2000 days in the training, and 516 in the testing data ($\sim$ 80/20 split).

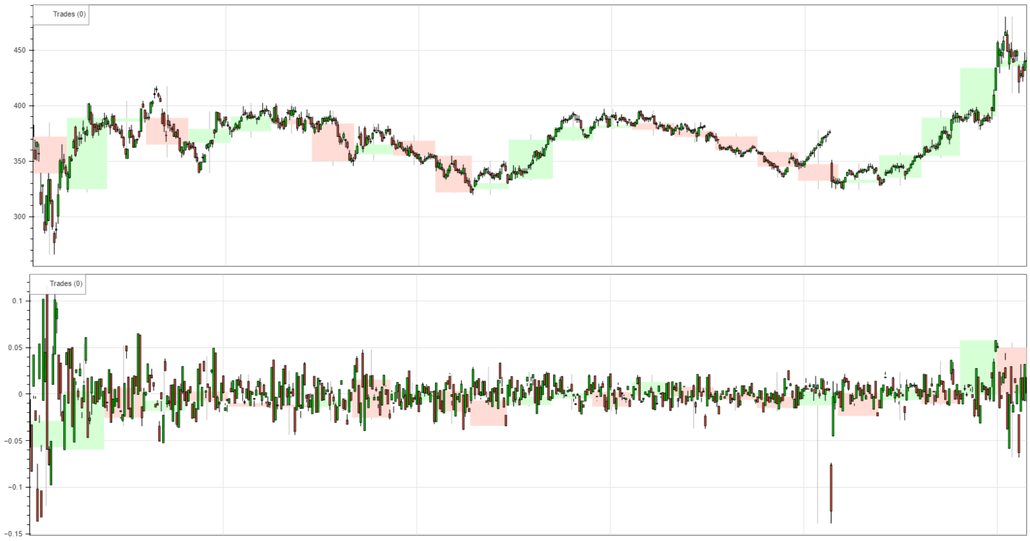


Figure 5.1: LMT Stock Price (top) and LMT Log-Normalized Stock (bottom)

Next, we run both Marsili's and Montemurro's analyses to find the most informative candles on the training data. We only consider candles that appear at least 10 times in the dataset, leaving us with 34 candles (out of the 48 classified).

Of those, we find the 5 most informative candles according to the two Marsili ranking systems outlined in Section 3.5.2 and the Montemurro ranking.

1. Marsili Max:
   1.1. BEARKICKING: 0.579
   1.2. BULLMORNINGSTAR: 0.579
   1.3. BEARNONE: 0.579
   1.4. BULLKICKING: 0.579
   1.5. BULLHIKKAKE: 0.579
2. Marsili Sum

   2.1. BEARNONE: 11.9
   2.2. BULLNONE: 11.5
   2.3. BULLLONGLINE: 10.9
   2.4. BEARENGULFING: 10.8
   2.5. BULLHIKKAKE: 10.6

3. Montemurro
   3.1. BULLNONE: 22.9
   3.2. BEARNONE: 22.2
   3.3. BULLENGULFING: 18.5
   3.4. BULLHIKKAKE: 18.2
   3.5. BEARENGULFING: 18.1

We consider the candlesticks most commonly associated with reversals. To do so, we use the "The Pattern Site" [6], which contains information on the candles that traders believe to be most important. We use the following criteria when evaluating potential signals:

1. Does the candle rank highly on all 3 measurements?
2. Is the theoretical performance of the candle a reversal?
3. Is the reversal downwards or upwards?

For a buy signal, the clear choice seems to be the BULLHIKKAKE candle, which is in the top 5 in the Montemurro and Marsili Sum rankings and also has a maximum $H[K]$ of 0.579. It is normally associated with an increase in price [6].

For the sell signal, the BEARENGULFING candle could be a good choice. It ranks well in the Sum and Montemurro rankings, but only has a maximum $H[K]$ of 0.554, trailing behind several candles, and is seen as an indication of a future decrease in price [6].

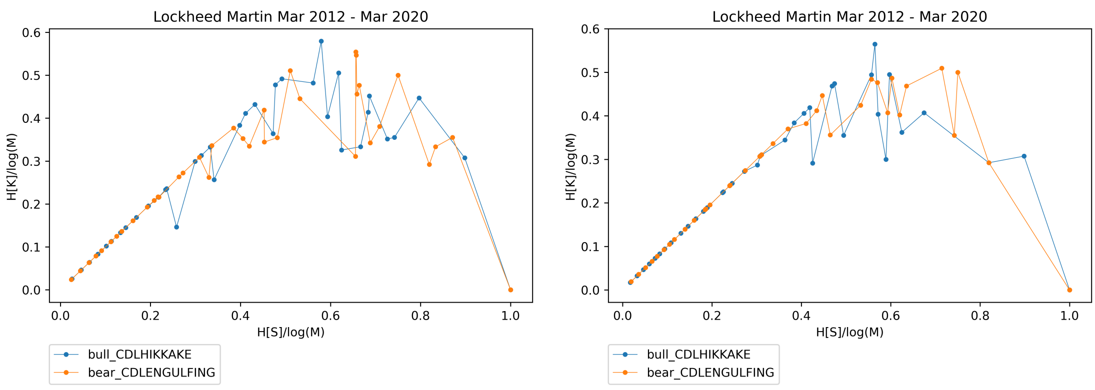


Figure 5.2: Entropy for our chosen candles. Log-Normalized Prices (left), Prices (right)

Therefore, we define our Indicator as being equal to 1 (buy) if a BullHikkake is seen, -1 (sell) if a BearEngulfing is seen, and 0 (hold) otherwise.

We run the strategy.

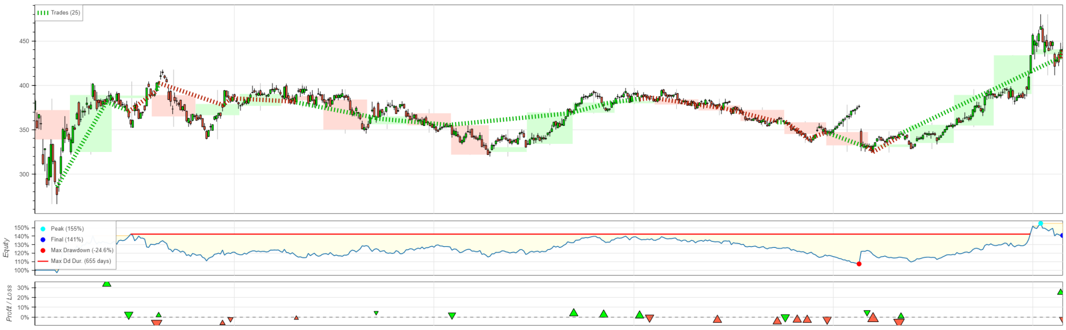


Figure 5.3: In descending order: LMT Stock Price, Equity (Portfolio Value), Trades Made Profit/Loss

```
Equity Final [$]                 14083.114803
Equity Peak [$]                  15508.822262
Return [%]                          40.831148
Buy & Hold Return [%]               15.195962
Return (Ann.) [%]                    18.20077
Volatility (Ann.) [%]               32.463016
Max. Drawdown [%]                  -24.620957
Avg. Drawdown [%]                   -5.587871
# Trades                                   25
Win Rate [%]                             48.0
Best Trade [%]                      33.718523
Worst Trade [%]                     -6.237576
Avg. Trade [%]                       1.389471
```

Figure 5.4: Returns of Trading Strategy

As we can see in Figure 5.4 and Figure 5.3, the strategy performs well for Lockheed Martin stock. Buying and holding would have seen a 15% return, whereas this strategy achieved returns of around 40%.

However, there are a few considerations to be had. Looking at the “Win Rate”, the percentage of profitable trades, this is only 48% or 12 out of 25 trades made. Moreover, the return on our average trade was on average 1.4%, with our best trade giving us a 33.7% return.

Specifically, our best trade on the 28th of April 2020 and our second-best trade on the 21st of March 2022 (+24%) account for virtually all of the returns. All other trades had less than $\pm 5\%$ profit/loss.

This indicates that we got lucky. Moreover, these candles were selected partially for their implied informational content, but also used “common knowledge” in regards to interpreting them. This is clearly not generalizable and uses highly subjective judgment.

Therefore, we move on to build a more generalizable trading strategy, with no human involvement in determining “buy” and “sell” decisions, using these principles.

5.2 A Less Simple Trading Strategy

5.2.1 Candle Ranking

To evaluate the candles, we take a weighted average of the maximal $H[K]$ and the sum of $H[K]$. Next, we take the 5 most informative candles and calculate their average returns for 2 weeks in the future (10 trading days), as well as at 2, 3, 5, and 7 trading-day intervals. We also calculate the *Sharpe Ratio* for each candle, given by the expected value of the returns divided by the standard deviation of the returns. This is a measure of risk-adjusted returns. This information is stored in the form of a MODEL, treating each candle as the index and the various statistics as columns.

We treat highly informative candles with positive Sharpe Ratios (positive returns) as buy signals and negative ones as sell signals. For any other candle, we hold.

5.2.2 Stop-Loss

One important consideration is that when markets are moving upwards, as they have for the last century and certainly in the last decade, we would expect virtually every candle to have positive expected returns. This means that we are much less likely to encounter indications to sell, compared with indications to buy.

Moreover, it means that if the price does fall and we don’t see a sell candle, then we have no way to close unprofitable positions. As such, we introduce a stop-loss, which is the maximum decrease in price that can occur before we close our position.

We used a 14-day ($n = 14$) Average True Range (ATR) stop loss on orders. What this is is an average of the maximum daily range in prices, over a given lookback period as a proxy for volatility, given by the formula:

$$\text{ATR} = \left(\frac{1}{n}\right)\sum_{i=1}^{n} \max(\textsc{High}_i - \textsc{Low}_i, |\textsc{High}_i - \textsc{Close}_i|, |\textsc{Low}_i - \textsc{Close}_i|)$$

On a long position, the ATR can only increase. Consider entering a position at an entry price of \$10 and an ATR of \$0.05. We would set a stop-loss order for 2x-ATR, meaning that if the price falls below \$9.90, we would close the trade. Therefore, if the price increases to \$10.50 with an ATR of \$0.05, we would update our stop-loss to be \$10.40.

Similarly, if the price had instead risen to \$10.50 and the ATR had also risen to \$0.1, we would place a stop-loss at \$10.30. The opposite is true for a short position, where we want the price to fall and therefore our 2x-ATR stop-loss would be above the close price.

Therefore, we exit a position in one of two ways: either we see an informative candle that is associated with a price drop or the price changes to $\pm$2x-ATR.

5.2.3 Results

The results from the backtests are unanimous: it's almost impossible to meaningfully and consistently beat the market with this strategy.

This finding is unsurprising given that, as outlined in Section 1.1, between 85% and 95% of professional investment funds fail to beat their benchmark [3].

We ran this strategy on ∼2000 backtests in total; 444 individual stocks, the S&P 500 ETF, 37 futures (primarily commodities), 4 US treasuries, 24 currencies, and 8 cryptocurrencies. These were run, where applicable, for 5, 10, and 20 year lookback periods followed by the maximum range of the available data.

All datasets were split into an 80/20 training/test split, to train and test the entropy model. Then the α for the strategy, win rate, and other metrics were calculated. We used the return from buying and holding the asset as a benchmark for the strategy.

5.2.3.1 α Distributions

Across these tests, we begin by plotting our α, which we take to be the difference between the return from buying and holding (%) and the return (%) from the trading strategy. This is perhaps the most important metric for any investment, as it measures how the trading strategy performed relative to the market.

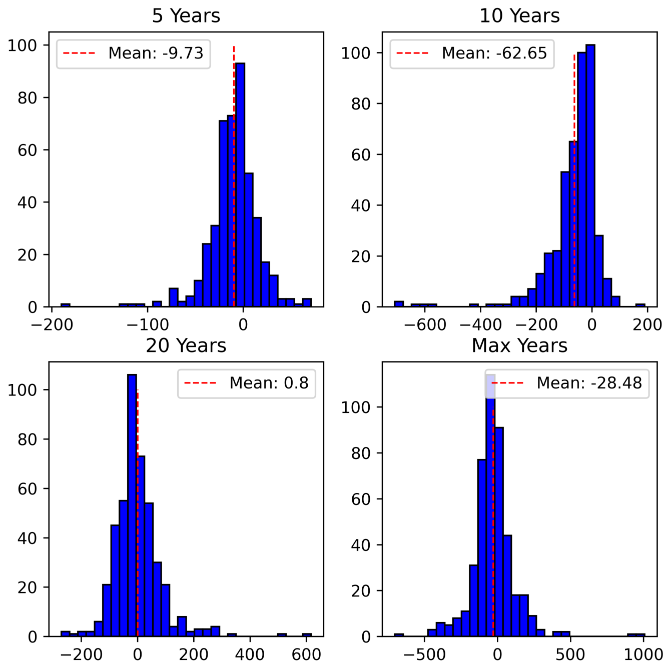


Figure 5.5: α for stock results over time periods (higher α is better). As we can see, the average performance of the trading strategy for stocks was primarily grouped slightly below zero though the tails of the distributions varied significantly. Of note is the particularly poor performance over a 10 year time period.

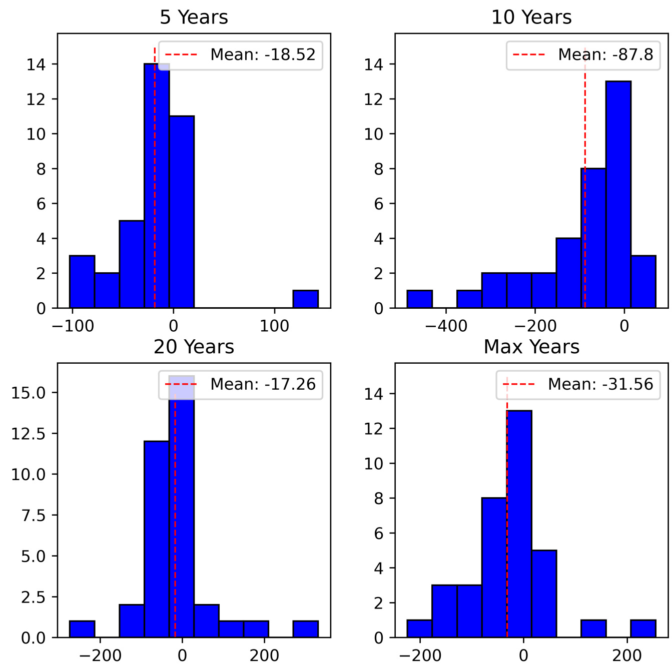


Figure 5.6: α for futures results over time periods. The performance was worse than that of stocks, which may be due to the fact that prices for futures are more volatile and haven't seen the large upwards trends that stocks have. Returning to our discussion of Silver Futures from Chapter 4, they had an α of 23 over the maximum time period and a win rate of 54%, possibly supporting the hypothesis that they are a more informative market.

Regarding cryptocurrencies, whose graphs are omitted due to having 8 of them and only having between 3 and 7 years worth of data, we saw an α of -200% for Bitcoin with the others ranging from $\alpha = 0.6$ to 43%.

Bonds were the *most* bleak, with a return of -100% across all 4 of them.

5.2.3.2 Power-Law Exponents

A calculation was made using the POWERLAW package for Python [1]. We calculated μ, the exponent for the power-law that most closely fit the data, and σ, the standard error. This was calculated using the frequency and rank for the candles of each asset in the training data. The following graphs show the α of the strategy with the power-law exponent, with a line of best fit overlaid.

As explained in Section 2.2.8, if the series of candlesticks (and therefore prices) was non-random, we would expect to see *higher* returns associated with exponents closer to $\mu = 2$, indicating that the candlestick distribution more closely follows Zipf's Law.

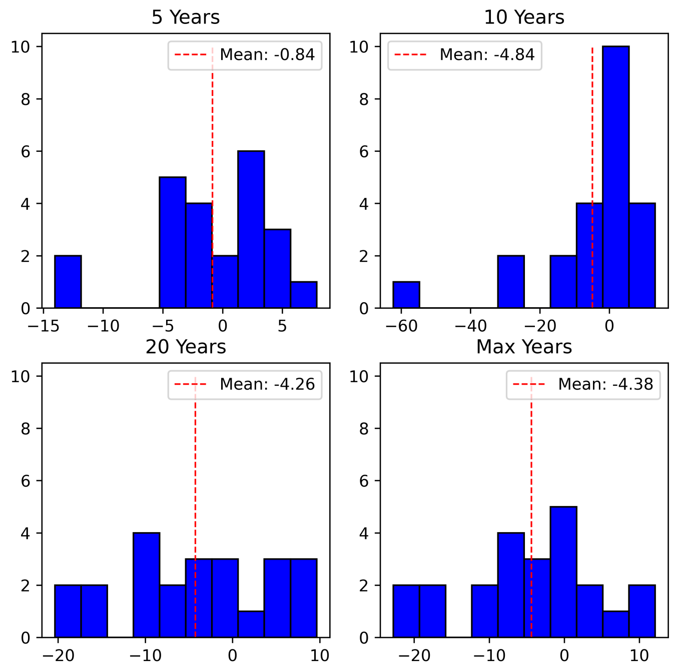


Figure 5.7: α for currency results over time periods. We saw a slight, yet consistent underperformance. This could be due to the fact that currencies, because they are traded in "pairs" (meaning that to buy a currency you must sell another currency) and tend to have more direct government intervention regarding their value, are less volatile than other assets. Moreover, many currency pairs haven't seen the large increases in value seen in stocks and therefore both returns and losses are more contained (Figure 5.8).

5.2.3.3 Win Rate

Next, as a proxy for measuring the "luck" seen in the backtesting we plot graphs showing the win rate, which is the % of profitable trades out of all trades made, relative to α. In theory, the returns of the trading strategy should be closely correlated with their win rate, whereas the closer the trading strategy approximates random chance, we would expect to see a handful of large successful trades constituting the majority of the returns and a handful of large unsuccessful trades constituting the majority of the loses.

There is a problem with this measure, though, which is that we have no way to benchmark it. Due to our benchmark being buying and holding, we only have one trade to compare it to meaning that the win rate as a measure in and of itself may not be a particularly good indicator. Moreover, if the price trends upwards we would expect an artificially increased win rate as increasing prices mean that the probability of a profitable trade is much higher.

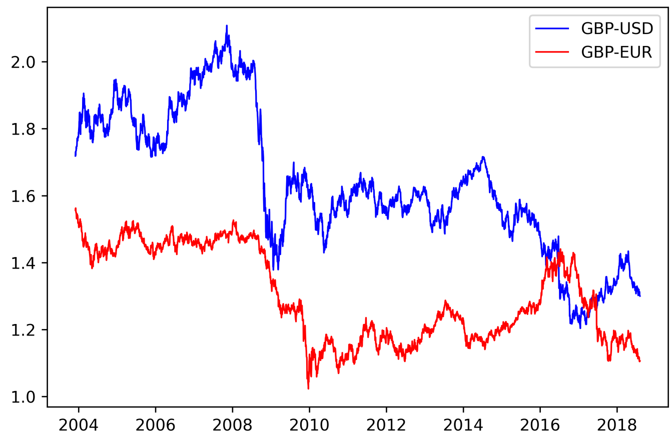


Figure 5.8: Exchange rates often don't have large swings and therefore both losses and returns are more contained relative to other assets.

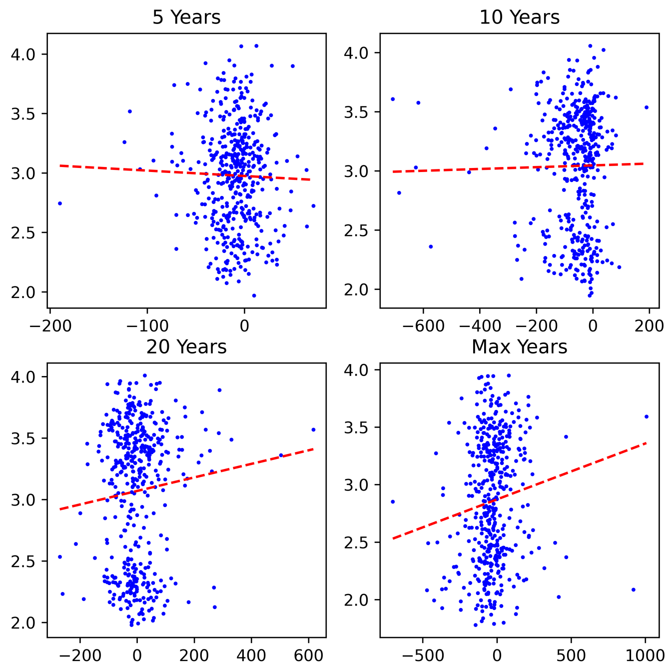


Figure 5.9: α versus μ power-law exponent for stocks. There is virtually no association of returns with a power-law fit of $\mu = 2$. This once again points toward financial markets not being learnable, and the distribution of candle patterns being uninformative of the underlying system. We point out that the lines of best fit for all time periods other than 5 Years slope *upwards*, suggesting that markets with exponents further away from 2 actually performed better.

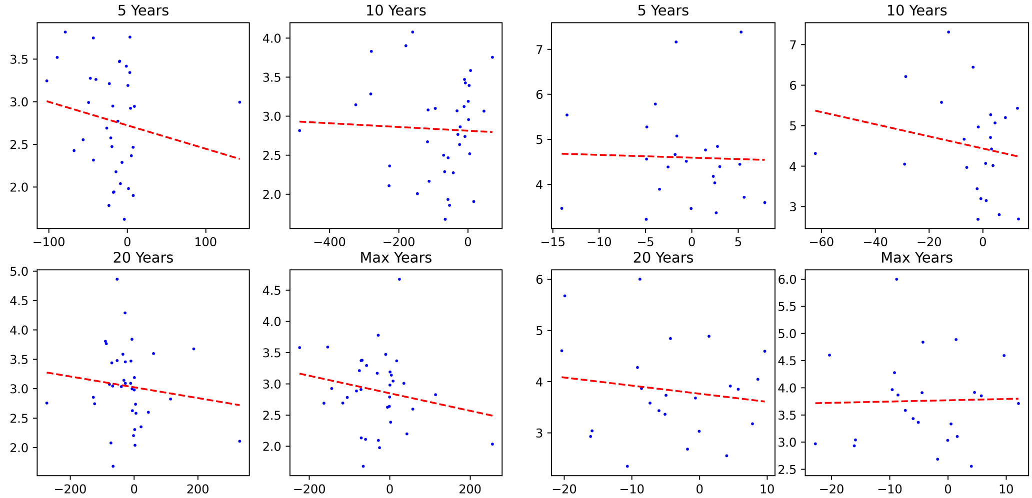


Figure 5.10: α versus μ power-law exponent for futures (left) and currencies (right). The trend seen for stocks in Figure 5.9 continues across asset classes, with futures and currencies also exhibiting no clear relationship between μ and α. However, unlike stocks, the lines of best fit have a slight downwards slope, indicating that better returns may be very loosely related to an exponent closer to 2. This may also be a consequence of having fewer assets (and therefore more randomness), as there are around 10 times more stocks on the S&P than all others combined. Cryptocurrencies and bonds also exhibited similar behaviors.

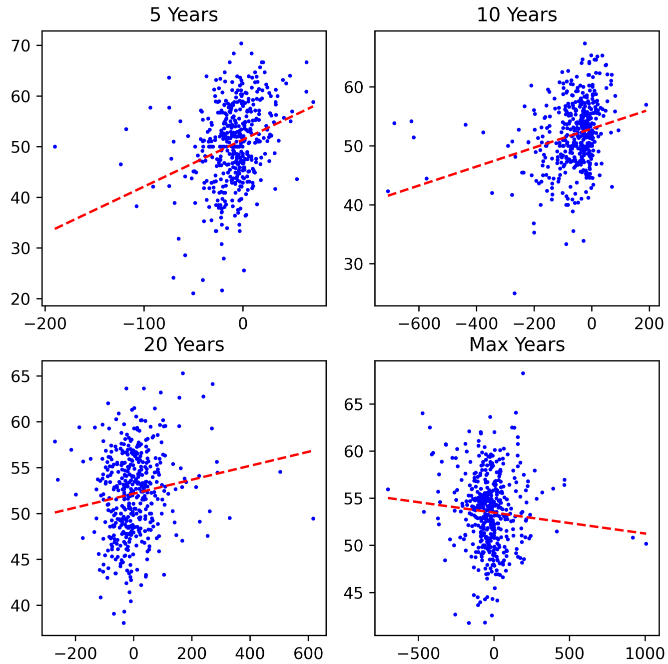


Figure 5.11: α versus win rate for stocks. Further indications of randomness constituting the majority of the returns can be seen in stocks, with a weakly positive relationship between win rate and α.

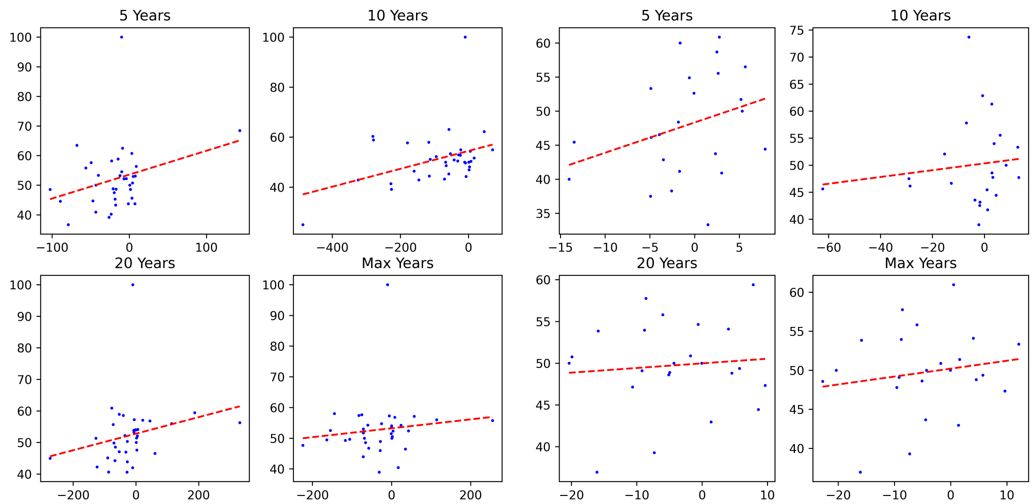


Figure 5.12: α versus win rate for futures (left) and currencies (right). We see much of the same story for futures and currencies. Note that the 100% win rate outlier for futures came from Orange Juice contracts, where a single trade was made on the second day of backtesting. This is an illustrative example of win rate not necessarily being a good measure, due to having no benchmark.

Chapter 6

Discussion

6.1 Improvements

We believe that the findings in Chapter 4 and Chapter 5 constitute a relatively extensive analysis of the implied informational content in a variety of different financial markets. However, we note that the primary trading strategy used to test the predictive value of informative candles may be improved in several ways.

First, the most informative candles may change, and could therefore be continually re-calculated. This may result in the buy/sell signals changing with the passage of time, which may lead to better returns. However, we point to the findings in Chapter 4, which indicate that even when looking at the entire dataset without partitioning it into training/testing data, the differences between the candles are still small, and indicate an uninformative system.

Second, in the applications of these methods so far, the objective function of the complex system is not changing. That is, Mary Shelley's U is assumed to be the same for the two years it took her to write Frankenstein. However, the objective function of financial markets may have changed. The advent of electronic trading, the post-2008 financial crisis regulations, and the current ultra-low interest rate environment may have all shifted the U we are trying to model. Markets are drastically different from what they were 15 years ago and as such our methods fall short in accounting for that.

However, a restriction in the date range to account for this problem leads to a decrease in sample size, which may cause inaccurate results. As such, when evaluating the 5-Year performances from Chapter 5 perhaps the objective function they were trying to model was more consistent but fell short due to a lack of sufficient samples.

Third, we believe that improvement of the "sell" signal beyond that of $\pm$2x-ATR may lead to better returns. Making adaptations to design the trading strategy to be entirely based on candlestick signals may see improvements as perhaps the shortcoming of the strategy outlined in Chapter 5 is entirely from failing to close unprofitable positions in time, or not keeping positions open for long enough.

Despite these areas for improvement, we believe that given the relatively small differences observed between candles classified as highly informative and uninformative, further strategies built on this technique may see a modest change in findings, but the general outcome is unlikely to change.

6.2 Future Work

In addition to the improvements outlined in Section 6.1, we think that the potential for applications of these techniques in financial markets remains large.

Given that this report suggests that candlestick price patterns alone are unlikely to be informative, adding additional information in each $\underline{s}$ may be extremely valuable. This may include factors such as interest rates, inflation, growth indicators, and trading volume. This may give us more insight as to the importance of price patterns under specific conditions and incorporating a wider range of inputs to the entropy models may cause greater differences in the informational content.

Moreover, these methods have applications in other financial domains other than technical analysis. For instance, analyzing orders being sent to an exchange may enable both brokers and exchanges to better understand liquidity conditions. Or, by applying these methods to macroeconomic trends, central banks can better understand key indicators for the business cycle.

Chapter 7

Conclusion

The project set out to give a proof-of-concept in financial applications of entropy methods as a basis for technical analysis. By encoding a series of price movements into candlestick patterns, labeling, and concatenating the candlesticks into a list, and then ranking them according to their $H[K]$ and $H[\underline{s}]$ values, we were able to analyze the differences in the information different candles provide.

To do this, we implemented a library, which first turns numerical candlesticks into a labeled series, and then runs the entropy methods to find the most informative ones. We observe that the differences between the most and least informative candles tend to be quite small, indicating that the total information to be gained from the system is therefore small. Moreover, random variations in the candles may explain away some of the differences found.

Following on from this, we built and tested a trading strategy using these methods as inputs to build a model, and found that these strategies universally underperformed relative to buying and holding. This further points to the fact that a series of prices alone is unlikely to provide meaningful information in regards to the underlying system of the financial market. This has two potential consequences.

The first is that Eugene Fama is right and the EMH holds. If it is the case that there is little to no difference between candles deemed to be maximally informative and those that are minimally informative, then it follows that one cannot learn about the system. Meaning that, if all price movements are equally uninformative, then predicting future prices based on past ones is impossible.

Second, if because of noise, herd behavior, or other factors [20] lead to noisy efficient markets then there is the potential to predict future prices. By adapting the methods in this report, we may still be able to distinguish noise trading from informative price changes. However, a time series of prices alone does not contain enough information to do so. This means that by only looking at the prices, the system may appear uninformative to us, even though it need not be.

Consequently, Marsili's methods will remain applicable to financial markets and technical analysis may still be profitable, but prices alone are unlikely to provide a sufficient basis for it.

So, yes, Day-Trading Gurus obey the laws of entropy, but not in a way that implies anything more than getting lucky.

Bibliography

[1] Jeff Alstott. *powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions*. 2014. URL: https://pypi.org/project/powerlaw/.

[2] Ran Aroussi. *yfinance*. 2022. URL: https://pypi.org/project/yfinance/.

[3] Guarav Sinha Berlinda Liu. "SPIVA U.S. Scorecard". In: ().

[4] Binance. URL: https://data.binance.vision/.

[5] FISCHER BLACK. "Noise". In: *The Journal of Finance* 41.3 (1986), pp. 528–543. DOI: https://doi.org/10.1111/j.1540-6261.1986.tb04513.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1986.tb04513.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1986.tb04513.x.

[6] Thomas Bulkowski. *The Pattern Site*. 2022. URL: https://thepatternsite.com/CandlePerformers.html.

[7] University of Chicago. "Eugene Fama - Biography". In: (2022). URL: https://www.chicagobooth.edu/faculty/directory/f/eugene-f-fama#biography.

[8] O Duranthon, M Marsili, and R Xie. "Maximal relevance and optimal learning machines". In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.3 (Mar. 2021), p. 033409. DOI: 10.1088/1742-5468/abe6ff. URL: https://doi.org/10.1088/1742-5468/abe6ff.

[9] Eugene F. Fama. "Efficient Capital Markets: A Review of Theory and Empirical Work". In: *The Journal of Finance* 25.2 (1970), pp. 383–417. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2325486.

[10] Python Software Foundation. *Python Programming Language*. 1995. URL: https://python.org.

[11] Adrian Heinz et al. "Bullish and Bearish Engulfing Japanese Candlestick patterns: A statistical analysis on the S&P 500 index". In: *The Quarterly Review of Economics and Finance* 79 (2021), pp. 221–244. ISSN: 1062-9769. DOI: https://doi.org/10.1016/j.qref.2020.06.006. URL: https://www.sciencedirect.com/science/article/pii/S1062976920300806.

[12] Jan Kozak, Krzysztof Kania, and Przemysław Juszczuk. "Permutation Entropy as a Measure of Information Gain/Loss in the Different Symbolic Descriptions of Financial Data". In: *Entropy* 22.3 (2020). ISSN: 1099-4300. DOI: 10.3390/e22030330. URL: https://www.mdpi.com/1099-4300/22/3/330.

[13] Anqi Liu et al. "The Flow of Information in Trading: An Entropy Approach to Market Regimes". In: *Entropy* 22.9 (2020). ISSN: 1099-4300. DOI: 10.3390/e22091064. URL: https://www.mdpi.com/1099-4300/22/9/1064.

[14] Tsung-Hsun Lu, Yung-Ming Shiu, and Tsung-Chi Liu. "Profitable candlestick trading strategies—The evidence from a new perspective". In: *Review of Financial Economics* 21.2 (2012), pp. 63–68. ISSN: 1058-3300. DOI: https://doi.org/10.1016/j.rfe.2012.02.001. URL: https://www.sciencedirect.com/science/article/pii/S1058330012000092.

[15] Zach Lûster. *Backtesting.py*. Version 0.3.3. 2021. URL: https://kernc.github.io/backtesting.py/.

[16] Matteo Marsili, Iacopo Mastromatteo, and Yasser Roudi. "On sampling and modeling complex systems". In: *Journal of Statistical Mechanics: Theory and Experiment* 2013.09 (Sept. 2013), P09003. ISSN: 1742-5468. DOI: 10.1088/1742-5468/2013/09/p09003. URL: http://dx.doi.org/10.1088/1742-5468/2013/09/P09003.

[17] MARCELO A. MONTEMURRO and DAMIÁN H. ZANETTE. "ENTROPIC ANALYSIS OF THE ROLE OF WORDS IN LITERARY TEXTS". In: *Advances in Complex Systems* 05.01 (2002), pp. 7–17. DOI: 10.1142/S0219525902000493. eprint: https://doi.org/10.1142/S0219525902000493. URL: https://doi.org/10.1142/S0219525902000493.

[18] S. Nison. *Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East, Second Edition*. Penguin Publishing Group, 2001. ISBN: 9780735201811. URL: https://books.google.co.uk/books?id=rbn8NeXOYV4C.

[19] Cheol-Ho Park and Scott H. Irwin. "A reality check on technical trading rule profits in the U.S. futures markets". In: *Journal of Futures Markets* 30.7 (2010), pp. 633–659. DOI: https://doi.org/10.1002/fut.20435. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fut.20435. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fut.20435.

[20] Cheol-Ho Park and Scott H. Irwin. "WHAT DO WE KNOW ABOUT THE PROFITABILITY OF TECHNICAL ANALYSIS?" In: *Journal of Economic Surveys* 21.4 (2007), pp. 786–826. DOI: https://doi.org/10.1111/j.1467-6419.2007.00519.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-6419.2007.00519.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-6419.2007.00519.x.

[21] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x.

[22] TicTacTec LLC. *TA-Lib: Technical Analysis Library*. Version 0.4. 2007. URL: https://ta-lib.org/about.html.

[23] Algo Trading and Investment. *Peak and Trough Graph*. 2020. URL: https://algotrading-investment.com/2020/04/06/turning-point-peak-trough-swing-high-swing-low-and-zigzag/.

[24] J. de la Vega and H. Kellenbenz. *Confusion de Confusiones [1688]: Portions Descriptive of the Amsterdam Stock Exchange*. Publication. Baker Library, Har-

vard Graduate School of Business Administration, 1957. URL: https://books.google.co.uk/books?id=16%5C_aDwAAQBAJ.

Appendix A

Relevant Code

A.1 Entropy Class

```
class Entropy:
  def __init__(self, data):
      data = data[: self.calc_best_len(len(data))]
      unique_values = np.unique(data)
      iterables = [unique_values, ["hs", "hk"]]
      multi_index = pd.MultiIndex.from_product(
          iterables, names=["value", "calculation"]
      )
      self.entropies = pd.DataFrame(columns=multi_index)

      for i in unique_values:
          hs_frac, hk_frac = self.calc_entropies(data, i)
          self.entropies[i, "hs"] = hs_frac
          self.entropies[i, "hk"] = hk_frac

      self.maxes, self.sums, self.aucs = self.get_most_informative
  ()
      self.montemurro = self.montemurro_shuffling(data)
      self.frequencies = data.value_counts()

  def calc_best_len(self, n):
      best_fact = {0}
      j = n
      best_len = 0
      while j > 0.75 * n:
          all_fact = set(
              ft.reduce(
                  list.__add__,
                  ([i, j // i] for i in range(1, int(j ** 0.5) +
  1) if j % i == 0),
              )
          )
          if len(all_fact) > len(best_fact):
              best_fact = all_fact
              best_len = j
          j -= 1
```

```
        return best_len

    def h_of_s(self, states):
        _, counts = np.unique(states, return_counts=True, axis=0)
        unique, counts = np.unique(counts, return_counts=True)
        M = states.shape[0]
        to_return = -np.sum((unique * counts) / M * np.log(unique / M)) / np.log(M)
        if np.isnan(to_return):
            return 1
        return to_return

    def h_of_k(self, states):
        _, counts = np.unique(states, return_counts=True, axis=0)
        unique, counts = np.unique(counts, return_counts=True)
        M = states.shape[0]
        to_return = -np.sum(
            (unique * counts) / M * np.log(unique * counts / M)
        ) / np.log(M)
        if np.isnan(to_return):
            return 0
        return to_return

    def word_probabilities(self, states, word):
        prob = np.count_nonzero(states == word, axis=0) / states.shape[0]
        return prob

    def calc_entropies(self, samples, word):
        length = len(samples)
        hs = []
        hk = []
        for i in range(length):
            if length % (i + 1) == 0:
                new_shape = [(i + 1), math.floor(length / (i + 1))]
                hs_app = self.h_of_s(
                    self.word_probabilities(
                        np.reshape(np.array(samples, copy=True), new_shape), word
                    )
                )
                hk_app = self.h_of_k(
                    self.word_probabilities(
                        np.reshape(np.array(samples, copy=True), new_shape), word
                    )
                )
                hs.append(hs_app)
                hk.append(hk_app)
        hs = np.array(hs)
        hk = np.array(hk)
        return zip(*sorted(zip(hs, hk), key=lambda x: x[0]))

    def get_most_informative(self):
        max_max = {}
        max_sum = {}
```

```python
        max_auc = {}
        for i in list(self.entropies.columns.levels[0]):
            max_max[i] = max(self.entropies[i, "hk"])
            max_sum[i] = sum(self.entropies[i, "hk"])
            max_auc[i] = np.trapz(self.entropies[i, "hk"], self.entropies[i, "hs"])
        return dict(sorted(max_max.items(), key=lambda item: item[1])), \
               dict(sorted(max_sum.items(), key=lambda item: item[1])), \
               dict(sorted(max_auc.items(), key=lambda item: item[1]))

    def montemurro_shuffling(self, data):
        maxes = {}
        unique_values = np.unique(data)
        for i in unique_values:
            maxes[i] = 0

        for i in range(10):
            shuffled = data.sample(frac=1)
            for j in unique_values:
                hs, _ = self.calc_entropies(shuffled, j)
                maxes[j] += sum(np.absolute(hs + self.entropies[j, "hs"])) / 10

        return dict(sorted(maxes.items(), key=lambda item: item[1]))
```

A.2 CandlePatterns Class

```python
class CandlePatterns:
    def __init__(self, raw_data):
        self.data = raw_data[["Open", "Close", "High", "Low"]]
        self.data = self.data.dropna(how="any")
        self.data = self.__classify_candles()
        self.data = self.__backfill_candles()

    def __classify_candles(self):

        for candle in CS_PATTERNS:
            self.data[candle] = getattr(ta, candle)(
                self.data["Open"],
                self.data["High"],
                self.data["Low"],
                self.data["Close"],
            )
        return self.data

    def __candle_overlap(self):
        counts = dict(self.data[CS_PATTERNS].count())
        overlaps = self.data[CS_PATTERNS].apply(
            lambda x: list(x.dropna()), axis=1
        )
        for i, v in overlaps.items():
            if not v:
```

```
                overlaps[i] = "NONE"
            elif len(v) == 1:
                overlaps[i] = v[0]
            else:
                rarest = v[0]
                for j in v:
                    if counts[j[5:]] < counts[rarest[5:]]:
                        rarest = j
                overlaps[i] = rarest
        self.data["Pattern"] = overlaps
        return self.data

    def __backfill_candles(self):
        self.data = self.data.dropna(how="any")
        self.data[CS_PATTERNS] = self.data[CS_PATTERNS].replace(0, np.NaN)
        self.data.replace(
            -100,
            pd.Series(["bear_" + i for i in self.data.columns], self.data.columns),
            inplace=True,
        )
        self.data.replace(
            -200,
            pd.Series(["bear_" + i for i in self.data.columns], self.data.columns),
            inplace=True,
        )
        self.data.replace(
            100,
            pd.Series(["bull_" + i for i in self.data.columns], self.data.columns),
            inplace=True,
        )
        self.data.replace(
            200,
            pd.Series(["bull_" + i for i in self.data.columns], self.data.columns),
            inplace=True,
        )
        self.data = self.__candle_overlap()
        self.data.loc[
            (self.data["Pattern"] == "NONE") & (self.data["Open"] < self.data["Close"]),
            "Pattern",
        ] = "bull_NONE"
        self.data.loc[
            (self.data["Pattern"] == "NONE") & (self.data["Open"] > self.data["Close"]),
            "Pattern",
        ] = "bear_NONE"
        return self.data[["Close", "Open", "High", "Low", "Pattern"]]
```