Analysis of Historical Profitability:

Moving Average Strategies on ES Futures

Reese McDonald

Graduate Student of University of Montana, Missoula

BMKT 699: MSBA Capstone

Professor Andrew Connor

03/24/2023

Executive Summary

Purpose

This work and analysis were undertaken for several reasons:

1. To discover if any configuration of trading rule of a basic moving average (MA) cross-over trading strategy would have performed better than a buy and hold strategy over the span of the data.
2. To visualize the back-testing of any trading rule configuration in action.
3. To create an analytical tool for measuring MA strategy profitability on any ticker, any time frame, and any range of MA period lengths.
4. To serve as necessary first steps in the exploration of price data that are foundational to the beginning algorithmic trader.

Methodology

- Simple, weighted, and exponential signal and control MAs, along with directional trading signal and price movement columns for each MA style, were created from raw ES hourly price data with Python as the programming language.
- Profit columns were calculated by multiplying the signal value (-1 for a short position or 1 for a long position) by the associated movement in price.
- Profit columns were then summed to give the total amount each moving average strategy would have made over the span of the data for comparison against a buy and hold strategy (price of last hour – price of first hour).
- This process was repeated for every signal and control period length combination within the range of (1, 50) and profit totals were visualized in heat maps for each MA style.

Key Findings

- The four visualizations that are featured in this paper constitute the primary work product that resulted from this undertaking. With them, it is possible to see how each strategy performs over the length of a set of price data, and where and why a given strategy wins or loses on an hourly basis.
- A buy and hold strategy (purchasing an asset to hold long-term to capitalize from long-term appreciation) would have made $143,325. The SMA had 217/2500 trading rule pairs that beat the market (AKA the buy and hold strategy), the WMA had 135/2500 pairs that beat the market, and the EMA had only 4/2500 that beat the market. The most profitable strategy at $327,437 had an SMA signal of period 27 and an SMA control of period 24 which is an inverse trading rule given that signal > control which contrasts from the conventional scenario where signal < control.

Limitations

- This system does not make predictions based on sentiment or fundamental indicators, but rather on an assumption that price trends persist more often than they change direction at any timeframe. There is a whole world of causation to price behavior that has not been considered in this work.
- Running price data that is more granular than a `1min` timeframe through the Python script may result in extremely high demand on memory storage and processing.

Recommendations

- The data spans over seventeen years and four months and includes 104,439 hours of open, high, low, and close (OHLC) prices. Over this amount of time, it is reasonable to assume that this algorithm would encounter, and account for, a comprehensive variation of technical situations which would be expressed in the performances of these MA strategies.

- As such, it may also be reasonable to assume that the strategies that did well within this span of time might continue to do so over any span of time that is similar in length.
- Some of the most profitable strategies were the inverses of failing strategies. A willingness to utilize a counter-intuitive application of MA cross-over based on these findings may well result in an algorithm that outperforms the market in the long-run.

Introduction/Background

This section contains all the background concepts that are needed to understand the engineering and analytical portions of this research. A seasoned trader will likely find this whole section and the section on market theory to be review. Those readers should consider skipping ahead to the section of the study that discusses prior related research and read on.

This research has been conducted on futures price data to the end of helping the audience understand more about price action and the moving average cross-over strategy, specifically those of the ES contract. All the concepts just mentioned will be explained in detail later in the text. To understand futures trading itself, it is necessary to explain some related concepts. A stock index is comprised of a selection of stocks based on some criteria and every index has different criteria for populating its basket of stocks. [1] The Standard & Poor 500 index (S&P 500) takes the top 500 companies in the United States by market capitalization. Market capitalization is defined as the number of shares that a publicly traded company currently has issued multiplied by the market value of each share. The top five publicly traded companies by market cap as this sentence is being written are Apple, Microsoft, Alphabet (Google), Amazon, and Berkshire Hathaway.

An index futures contract establishes an agreement between a buyer and seller of a contract. [2] To buy a contract (long position) and hold it to expiration means that the buyer agrees to buy the index at the purchase price (point value in dollars times number of points) on the date of expiration. To sell a contract (short position) and hold until expiration obligates the seller to sell the index. There are many kinds of futures, some of which involve the physical buying and selling of real commodities such as barrels of oil or lean hogs, but the focus of this research is on the ES contract, which is the futures instrument that corresponds with the S&P 500 market index. Futures are not to be confused with exchange-traded funds (ETFs), which are traded much like stocks. The scope of this research excludes the pricing mechanics of ETFs, so it is enough to know that they are generally priced based on the values of the underlying stocks within the index.

Futures move in points and each point is worth a certain dollar amount. [3] They are interesting financial instruments for many reasons. Firstly, because they are priced with a point-based system, they influence the trader to think in points and dollar amounts rather than percentages. Every futures ticker has its own point system and point value. The ticker “ES” is the e-mini futures contract for the S&P 500 and is worth $50 per point. The entire point value of the ES contract (the “price”) is currently 3947 points as this sentence is being written. That means that the whole contract is worth $50 * 3947 pts = $197,350. A margin requirement is the minimum amount of money the trader/hedger/investor needs to have in their futures trading account in order to control a futures contract. Tradovate is a popular futures trading platform which only requires $500 to control an ES contract. [4] If the total value of the asset/contract is divided by the margin requirement, the leverage multiple is the result. Leverage is essentially borrowing money to control a more valuable asset than one could have with only the liquid capital they possess. A leverage multiple of 2X implies that the trader borrowed the same amount of total capital at his disposal so he can control twice the amount of financial instrument, and profit or lose twice as quickly. The leverage multiple of the ES contract, as this sentence is being written, is $197,350/$500 = 394.7X. When an ES futures contract is purchased or sold short, the trader then has control of a ~ $200,000 asset depending on the point value of the contract. The price fluctuations of such an asset are very steep, so it is possible to lose some or all of the margin requirement very quickly. Generally, a little more than the margin requirement is needed in a futures account to control a futures contract because of the steep price fluctuations. To take control of an equivalent value of SPY (S&P 500 ETF) without any broker-issued leverage would require the full $197,350. Thus, futures contracts can be extremely financially hazardous to the inexperienced user and should be practiced upon extensively before trading or investing in them.

To profit from the price fluctuations of a contract, a trader needs to take a position. Betting that the asset will appreciate in value by buying the asset to sell at a later date is called "going long" or "taking a long position". Betting that the asset will depreciate in value by borrowing the asset, selling it now, and buying it back later is called "short selling" or "taking a short position". The time duration for which the asset is held long or short can vary from a fraction of a second to many years depending on the trading style of the user. Scalping occurs when a trader completes many trades in a single day and profits from small fluctuations in price with trades that last seconds. Day trading is a bit less aggressive, but the positions a day trader takes on will typically be closed by the end of the trading session that day. Swing trading occurs when a trader holds a position for days, weeks, or months before closing it. Investing involves holding an asset for years or even decades in hopes of attaining large long-term gains. [5]

The nature of futures lends itself to short term trading quite well, but it can be awkward to swing trade or invest in them due to the contract expiration feature of futures. Some futures contracts expire every month while some expire every three months. When a futures contract is spoken of, sometimes the overall contract is being referred to and other times it is the sub-contract that has an expiration date. It is possible to swing trade or invest in futures so long as the next contract is consistently being rolled over to, which simply involves closing the position of the contract that is set to expire soon and opening the same position on the newer contract.

A few words on technical analysis will be in line to understand why anyone would use a moving average for trading in the first place. [6] At its heart, technical analysis (TA) is about identifying quantitatively discernable patterns in price movements, volume, and open interest. The definition of volume will be discussed briefly later, and the concept of open interest lies outside the scope of this research, so the focus here will be on the price action component of TA. There are innumerable ways to quantify and transform price action data and the outcome of doing so is usually a technical indicator of some kind. Most people think of chart patterns, such as the triangle, double-bottom, or head-and-shoulders formations, and candle stick theory when they think of technical analysis. Figures 1, 2, and 3 illustrate these given examples of price action structures. It is notable that these are real examples of price action and that they were easy to find in hourly ES data since they occur with such regularity in all price action. There is a vast litany of patterns to learn, and each has a set of implications for position sizing, risk control, and direction. This short mention/discussion of TA should be enough to at least conceptualize the craft of charting in order to understand the subject matter of this text and study.

Figure 1: An example of a double-bottom formation in price action. This pattern occurs when price finds support at or near the same level twice and indicates that there may be a short-term market bottom at this level. The opposite of this formation is the double-top.

Figure 2: An example of a triangle formation in price action. This formation requires that upper and lower trend lines converge and usually indicates that the market will continue in the direction that was occurring prior to its formation. Triangles also occur during positive trends.

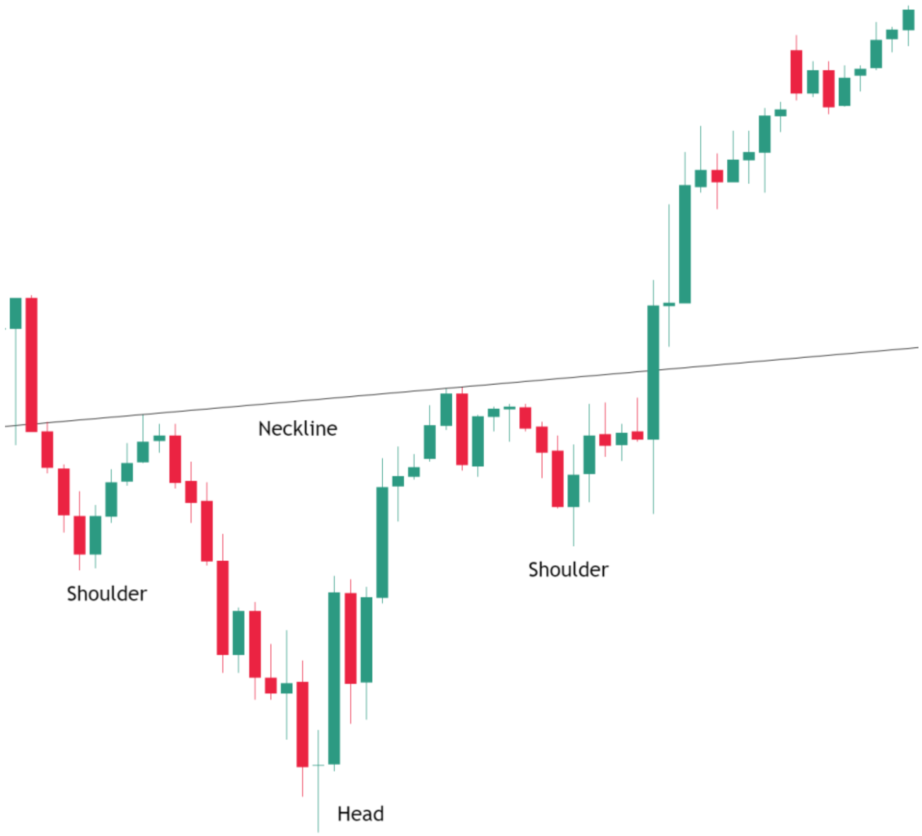

Figure 3: An example of a reverse head and shoulders formation in price action. The opposite of this formation would be a normal head-and-shoulders formation where the head and shoulders components of the structure would be on the other side of the neckline and would predict falling price as opposed to rising price once price breaks the neckline as is seen here.

Though these patterns of price action are indeed programmatically quantifiable, many traders who use these techniques rely on eyeballing a price chart along with the gut instinct that they derive from personal anecdote to determine if they have a trade set up with good a probability of profiting. This is one way of seeing the market, but algorithmic trading demands a more quantitative approach to describe and make predictions on price action phenomena. [7] There are many different indicators that are all calculated in different ways and are also used in very different ways. Just a few of these indicators include the relative strength index (RSI), volume-weighted average price (VWAP), on-balance volume (OBV), Bollinger bands, pivot points, and Fibonacci-retracement ratios. All are outside the scope of this research. The class of technical indicator that will be extensively explored here is the moving average.

A moving average is a common indicator that is used on time series data and helps to smooth out the noise of the data. [8] It also helps to identify trends in the data. Generally, if the moving average has a positive slope and the time series data is above the moving average, an upward trend is indicated. If the moving average has a negative slope and the time series data is below the moving average, then a negative trend is indicated. A moving average with a slope that stays close to zero (flat) with time series data that oscillates above and below the moving average with some recent regularity indicates a sideways trend. The way that this indicator expresses itself through price data is no different. See Figures 5 and 6 below for an illustration of these concepts.

The acronym OHLC stands for open, high, low, close and is an important aspect of how price data is typically stored and visualized as "candle sticks". [9] These are four measures of price that can be used for any given period of time. The open price occurs at the very beginning of a given period, say one hour, and the close price occurs at the very end of the same period. The high price is the largest numerical value that occurred in the span between the open and close prices, while the low price is the smallest numerical value in the same span of time. The opening price of a candle is very often the same price, or very close, as the closing price of the preceding candle. Figure 4 below shows real ES candlesticks. The “candle stick” moniker comes from the resemblance of the data visualization style to candles. As such, the portion of the candle from its open to its close is called the body and the thin lines that jut from the top and bottom of the body represent the high and low of the period and are called upper and lower wicks.

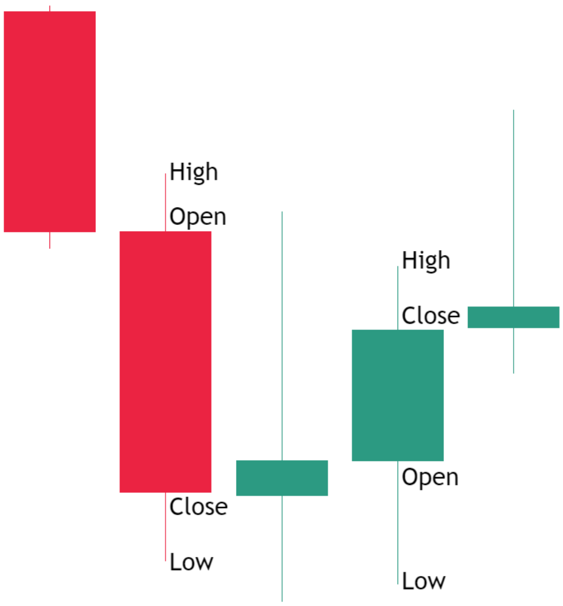

Figure 4: A close view of candle sticks and their labeled OHLC components.

Moving averages of price data can be calculated with any one of these price metrics or some derivation of them, though this isn’t as typical a practice as simply using O, H, L, or C. To calculate a simple moving average, one needs to determine how many periods of time they would like to average. This is simply known as the “period” of the moving average. To calculate a single moving average value with a period of ten and using the open price, one would sum the current period’s open price with the nine that came before it. The resultant value is stored in the same record as the most recent point of open price data. Then the next open price is summed with the nine before it and this process continues for the rest of the data until every record contains a moving average value, except for the records where there weren’t enough preceding records to calculate a moving average of the given period. For visualization, this means that a ten-period moving average indicator line spans the entire chart alongside the price data itself except for those first nine points in time. Note that the first nine records of any price data cannot contain a moving average value with a period of ten because there aren’t and will never be enough time periods available to calculate this specific metric until the tenth period. It follows that it is impossible to calculate moving average values for the first 24 records in price data if the moving average has a period of 25, and so on. The equation of the simple moving average is as follows [10]:

$$SMA = (A_1 + A_2 + \dots + A_n) / n$$

Where:

A_n = the price at period n
n = number of total periods or the “period length”

A moving average cross-over trading strategy is a simple configuration of trading rules wherein a moving average style is selected, and a directional trading position (either betting that the price will go up or down) is initiated based on one moving average's position relative to the other. [11] The two most important components of this strategy are the fast-moving average (the signal) and the slow-moving average (the control). The signal always has a smaller period length than the control in a traditional MA cross-over context. The effect of the differing period lengths allows for the identification of trend reversals. A moving average with a shorter period stays closer to price action, smooths less, and lags less than a moving average with a longer period. So, when the trend is clearly to the upside, the signal (shorter period) will always be between the actual price action and the control (longer period) with price action on top. The converse is also true. If the trend is clearly to the downside, the control will have larger values than the signal, and the signal with have larger values than the price action. So, when a peak or trough in price manifests, there must always be some point in time where the signal crosses the control and goes from above it to below and vice versa. This cross-over of the signal of the control is the directional trading signal of this strategy. When the signal crosses the control from the bottom, the trading signal is to the upside and a trader or algorithm must take a long position to adhere to the rules of the strategy. When the signal crosses from the top of the control and is now below it, the trading signal is to the downside and the trader or algorithm must take a short position to adhere to the rules of the strategy.

Figure 5 is an example of such a trade to the downside, or a "short sell" using simple moving averages. The blue curve is the signal, and the red curve is the control. The red circle highlights the moment where the signal crosses the control from above and to the downside. It also indicates the moment when the algorithm would instruct the trading platform to sell short some specified number of contracts, opening a short position. The green circle highlights the moment where the signal crosses the control from the bottom to the upside and when the algorithm would instruct the platform to close the short position by buying the borrowed contract back. Price fell during this snapshot of price action, and because the short trade was a bet that price would fall, the trade was profitable roughly by the vertical price distance between the red circle and the green circle.

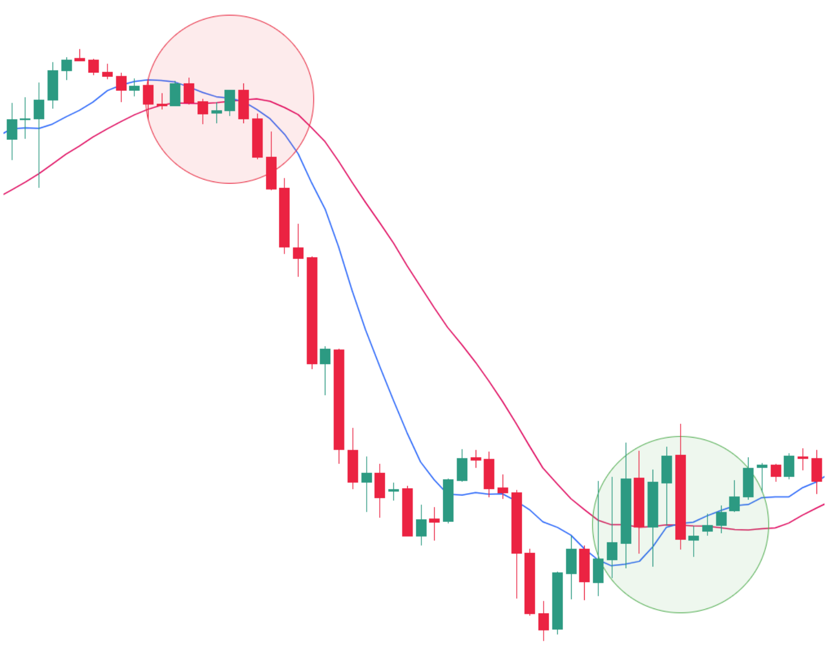

Figure 5: An example of a short trade based on an MA cross-over strategy.

Figure 6 illustrates a contrasting scenario where the blue signal MA crosses the red control MA from below and to the upside, signaling a long trade that would profit from an increase in price. The green circle highlights the moment in time where a specified number of contracts would be purchased by order of the algorithm and a long position opened, while the red circle indicates where those same contracts would be sold by the algorithm and the long position closed because the blue signal curve crossed the red control curve from above and to the downside. Price rose during this snapshot of price action, and because the long trade was a bet that price would rise, the trade was profitable roughly by the vertical price distance between the green circle and the red circle.

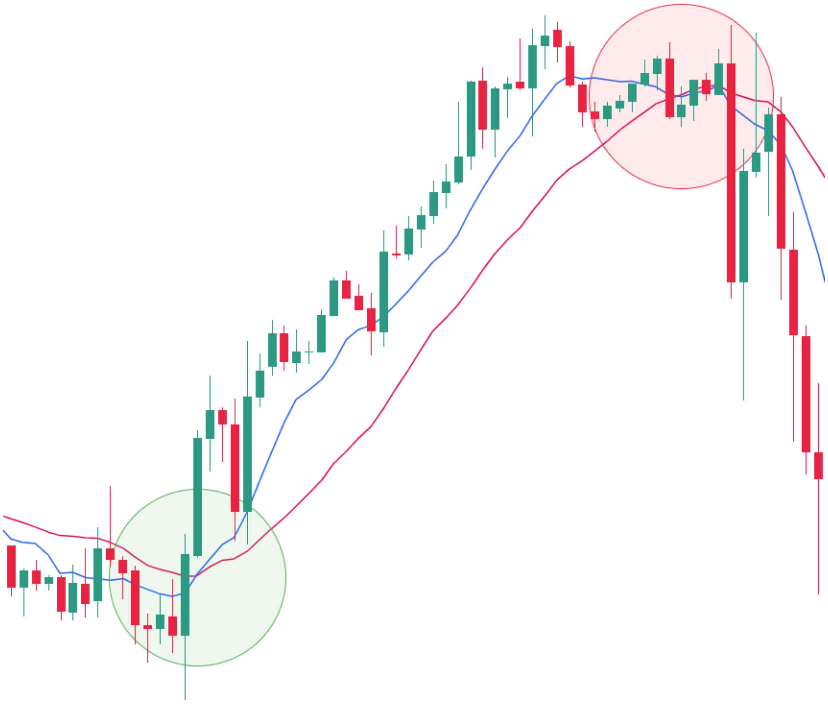

Figure 6: An example of a long trade based on an MA cross-over strategy.

Market Theory

A crucial touch point for this analysis must be at least a brief discussion on whether there is sufficient opportunity for profit in financial markets at all, whether using technical, fundamental, or sentiment analysis as vehicles for the discovery of those opportunities. Fundamental analysis (FA) "… measures a security's intrinsic value by examining related economic and financial factors. Intrinsic value is the value of an investment based on the issuing company's financial situation and current market and economic conditions." [12] There are two opposing schools of thought pertaining to market theory to which every investor or trader should adhere if they are to follow a stable protocol in transactional decision-making.

These dichotomous categories are labeled as Efficient vs Inefficient Market Hypotheses. The following definition of market efficiency [13] serves to accurately express the concept: "Market efficiency is a relatively broad term and can refer to any metric that measures information dispersion in a market. An efficient market is one where all information is transmitted perfectly (everyone receives the information), completely (everyone receives the entire information), instantly (everyone receives the information at once), and for no cost (everyone receives the information for free)." The Efficient Market Hypothesis (EMH) implies that no reliable or recurring opportunities for profit exist in financial markets because the whole of market participants for a given asset or financial instrument immediately act in a way that renders the current price of said asset or financial instrument true to its *humanly rational* value given every *known* underlying fundamental element. This is to say that speculation of any form in financial markets would be a moot, if not self-destructive, endeavor under the assumption, since it would be impossible to consistently predict the directional price movement of a given asset. It is important to note that EMH does not assert that every asset is always priced at its true value, but the most accurate value given all underlying information known to market participants.

Martin Sewell stated beautifully in the abstract of his 2011 research note [14], “The efficient market hypothesis (EMH) asserts that financial markets are efficient. On the one hand, the definitional ‘fully’ is an exacting requirement, suggesting that no real market could ever be efficient, implying that the EMH is almost certainly false. On the other hand, economics is a social science, and a hypothesis that is asymptotically true puts the EMH in contention for one of the strongest hypotheses in the whole of the social sciences. Strictly speaking the EMH is false, but in spirit is profoundly true. Besides, science concerns seeking the best hypothesis, and until a flawed hypothesis is replaced by a better hypothesis, criticism is of limited value.” The bold assertion that the EMH is asymptotically true may be accurate to one degree or another, but the fact remains that traders and institutions do outperform market indices with regularity, hence the existence of hedge funds and proprietary trading firms on Wall Street and elsewhere.

The three forms of EMH are weak, semi-strong, and strong [15]. The weak form hypothesis assumes that all historical asset prices are reflected in current asset prices, and thus market technicians (chartists) should not be able to consistently earn abnormal returns using technical analysis (TA). The semi-strong form of EMH postulates that all public information is so quickly factored into an asset’s pricing by market actors that no form of TA or FA can be used to predict price direction in ways that can lead to abnormal returns. The strong form of EMH postulates that all information, whether public or private, is factored into an asset’s price and no trader or investor can consistently achieve abnormal returns with respect to a buy and hold strategy. This research report rejects EMH altogether as an absolute assumption but recognizes the extreme difficulty in predicting directional price movement of any asset, and the scarcity of such predictive accuracy even in inefficient markets where information may be ubiquitously asymmetric and many opportunities for profit exist in the fluctuation of asset prices.

Prior Related Research

There have been several bodies of research that attempted to measure MA strategy historical profitability. The articles discussed here were found by searching for research on MA trading strategies using Google Scholar. One of such works by Eero Pätäri and Mika Vilska [16] examined the performance of “dual moving average crossover” (DMAC) trading strategies in the Finnish stock market from 1996 to 2012. They applied the DMAC to the 25 individual components of a Finnish stock index that has ticker OMXH25, and to the index itself. This index is comprised of the top 25 Finnish stocks by market capitalization. They used daily closing price data and allowed their “short-term” MAs (signals) to take period lengths between 1-20 days and allowed their “long-term” MAs (controls) to take period lengths between 50-200 days, which gave them, “… 3020 trading rules by testing all these combinations”. The research that Pätäri and Vilska conducted is different in many ways from the research that is detailed in this report. Firstly, they used different ranges for their short and long-term MA period lengths, and they applied their strategies to data of a daily timeframe instead of an hourly timeframe. They conducted their research under the assumption that a trader can either take a long position (betting that the market will go up), or no position (immune to price decline) which means that their strategy cannot profit directly from price decline holding other economic forces constant. They also applied their trading rules to stocks instead of futures and drew a comparison between how the strategies performed on the index itself vs a performance aggregation of its component stocks. An important difference in trading these two classes of financial instrument is that stocks are costly to sell short, because one must borrow the asset at an interest premium to sell immediately and buy back later, and futures are not costly to sell short because there is no associated interest premium involved in shorting futures. They may have left out the possibility of short positions to simplify their research and resulting model, and their analysis does account for reinvested stock dividend payouts, which is not a factor when trading futures. Their “… results show that most of the active DMAC strategies examined have outperformed the corresponding passive B and H (buy and hold) strategies...”

This outcome should be expected since extended periods of price decline would be avoided by exiting the market when negative trends were confirmed by MA crossovers.

Another noteworthy body of work that Massoud Metghalchi, Juri Marcucci, and Yung-Ho Chang conducted in 2011 [17] "… examines the profitability of several simple technical trading rules for 16 European stock markets over the 1990 to 2006 period." They stated in the report abstract that their "… empirical results support the hypothesis that technical trading rules can outperform the buy and hold strategy after accounting for transaction costs." Their study also used price data of daily granularity and they tested their selection of trading rules on 16 European stock market indices from January 1990 to May 2006. They use only one "short moving average" of "1-day" and only the 20, 50, 100, and 200-day period MAs as "long MAs". They test three moving average styles that include the Standard Moving Average (SMA), the Increasing Moving Average (IMA), and the Arnold and Rahfeldt's Autoregressive Moving Average (ARMA). The SMA in their research is synonymous with the SMA in this research and the definitions of the other MA styles that they use are not consequential to the findings of this report. The purpose of their mention is to state that they are different than what were chosen for this analysis.

Ben R. Marshall, Nhut H. Nguyen, and Nuttawat Visaltanachoti [18] published a body of research in 2016 that compared and contrasted MA trading rules with time series momentum (TSMOM) trading rules. Their MA trading rule only requires one MA where the position of price relative to the MA determines the trade signal. So, price itself, in this case, behaves as the signal and the one MA plays the role of control. The TSMOM strategy rule requires, to generate a long signal, that a price rise above a historical price of some number of days toward the past that would be the same as the MA period length of the MA strategy. A sell signal is generated in a converse way where the current price falls below the price of 200 time units ago. They found that both strategies are "… less effective on large stock-dominated market indices…" than on individual stocks, and that the returns of the two trading rules are highly correlated (0.78), but their MA trading rule tends to provide earlier entries and exits and larger profits than that of the TSMOM.

These examples of prior works that analyze MA rule strategy performance being applied to various financial instruments represent only a fraction of the body of research that members of finance, investment, and trading communities have conducted and published. The goals and findings of this research are distinct from these examples of existing research and literature and will enrich the analytical perspective of MA rule strategies and their specific employment to index futures contracts.

Exploration of ES Price Data

The data analyzed in this study is of ES futures and comes with only six columns: Date, Open, High, Low, Close, and Volume. This set was purchased from Download Historical Intraday E-Mini S&P 500 Futures (CME) (ES) Data (15 Years Data) (firstratedata.com) for $59.95 and includes separate data sets that vary in time granularity. Time frames included in the package were 1 hour, 30 minutes, 5 minutes, and 1 minute. The data span from 2005-09-06, 16:00:00 to 2023-01-13, 16:00:00, which is a couple years more than 15 as is asserted in the link above. The data does not span the entire life of the contract, which dates back to 1997, but was the best set available at a practical price for the purposes of this project. The analyses of this data were performed with Python as the programming language in Jupyter Notebook. After the data was read in as "ES", the 'Date' column was converted to date time format, sorted in ascending order, and set as the index. An additional index column that simply numbers each record was stored as 'Time_Unit'. Table 1 below shows the head and tail of the data.

Table 1: Looking at the head and tail of the data after formatting and sorting the 'Date' column, setting it as the index, and creating a supplementary informal index called 'Time_Unit'.

Date	Open	High	Low	Close	Volume	Time_Unit
2005-09-06 16:00:00	1146.250	1146.750	1146.000	1146.500	1711	1
2005-09-06 17:00:00	1146.250	1146.750	1146.250	1146.500	744	2
2005-09-06 18:00:00	1146.500	1146.750	1146.000	1146.250	395	3
2005-09-06 19:00:00	1146.000	1146.250	1146.000	1146.000	280	4
2005-09-06 20:00:00	1146.250	1146.500	1146.000	1146.000	370	5
...	...	...	...	...	...	...
2023-01-13 12:00:00	3990.750	4001.250	3989.750	3999.750	95284	104435
2023-01-13 13:00:00	4000.000	4009.750	3993.750	4008.000	99130	104436
2023-01-13 14:00:00	4008.000	4011.500	4004.500	4009.500	93394	104437
2023-01-13 15:00:00	4009.750	4024.250	4009.000	4017.750	214852	104438
2023-01-13 16:00:00	4017.750	4020.000	4014.250	4019.000	58017	104439

It was necessary to create some additional columns for use in the exploratory analysis portion of this project. The 'Mean' column is simply an average of the open, high, low, and close prices for each period. The column 'span' is a calculation of the 'High' price of each candle minus the 'Low' price and the column 'body' is a calculation of the 'Open' price minus the 'Close' price. They measure the entire vertical distance that price traveled and the distance of the price movement from beginning to end within the period respectively. Another interesting metric is a function of 'Volume' which is the number of contracts that were traded within a period. This metric is called 'vol-vma' and is volume minus a ten-period moving average of volume, which gives us an unrefined measure of relative volume. There are yet many other important columns to mention, but they will be covered in later sections. Table 2 shows a slice of the data after the new columns have been added. Note that some of these metrics were created only to demonstrate different ways that this simple data can be manipulated. Not all of them will serve a further role in this research.

Table 2: Looking at a slice of the data after creating the columns necessary for some light exploratory analysis.

Date	Open	High	Low	Close	Volume	Time_Unit	Mean	span	body	vma	vol-vma
2005-09-06 21:00:00	1146.250	1146.250	1145.750	1146.000	329	6	1146.062	0.250	-0.250	NaN	NaN
2005-09-06 22:00:00	1145.750	1146.250	1145.750	1146.000	312	7	1145.938	0.250	0.250	NaN	NaN
2005-09-06 23:00:00	1146.000	1146.250	1145.500	1145.750	637	8	1145.875	0.500	-0.250	NaN	NaN
2005-09-07 00:00:00	1145.500	1146.000	1145.500	1146.000	148	9	1145.750	0.000	0.500	NaN	NaN
2005-09-07 01:00:00	1146.000	1146.250	1146.000	1146.000	178	10	1146.062	0.250	0.000	510.400	-332.400
2005-09-07 02:00:00	1146.000	1146.750	1145.750	1146.000	1020	11	1146.125	0.750	0.000	441.300	578.700
2005-09-07 03:00:00	1146.250	1147.000	1145.500	1146.000	4265	12	1146.188	1.000	-0.250	793.400	3471.600
2005-09-07 04:00:00	1146.000	1146.500	1145.500	1145.750	5183	13	1145.938	0.750	-0.250	1272.200	3910.800

Exploratory Analysis of ES price candles

Python’s matplotlib.pyplot package is used in this section to visualize some of the metrics from the last section. The first histogram is of volume per hour and can be seen in Figure 7 and Table 3 displays some descriptive statistics of ‘Volume’. The volume per hour at third quartile is only 98,715 while the largest volume over the span of this data is 1,600,843. The distribution of volume per hour is right skewed to say the least, and the number of contracts traded in an hour rarely reaches the range of 300,000+.

Table 3: The descriptive statistics of volume for every hour within the data set.

```
ES['Volume'].describe()
```

```
count     104418.000
mean       69263.764
std       104343.398
min            1.000
25%         4995.000
50%        17114.000
75%        98715.750
max      1600843.000
Name: Volume, dtype: float64
```

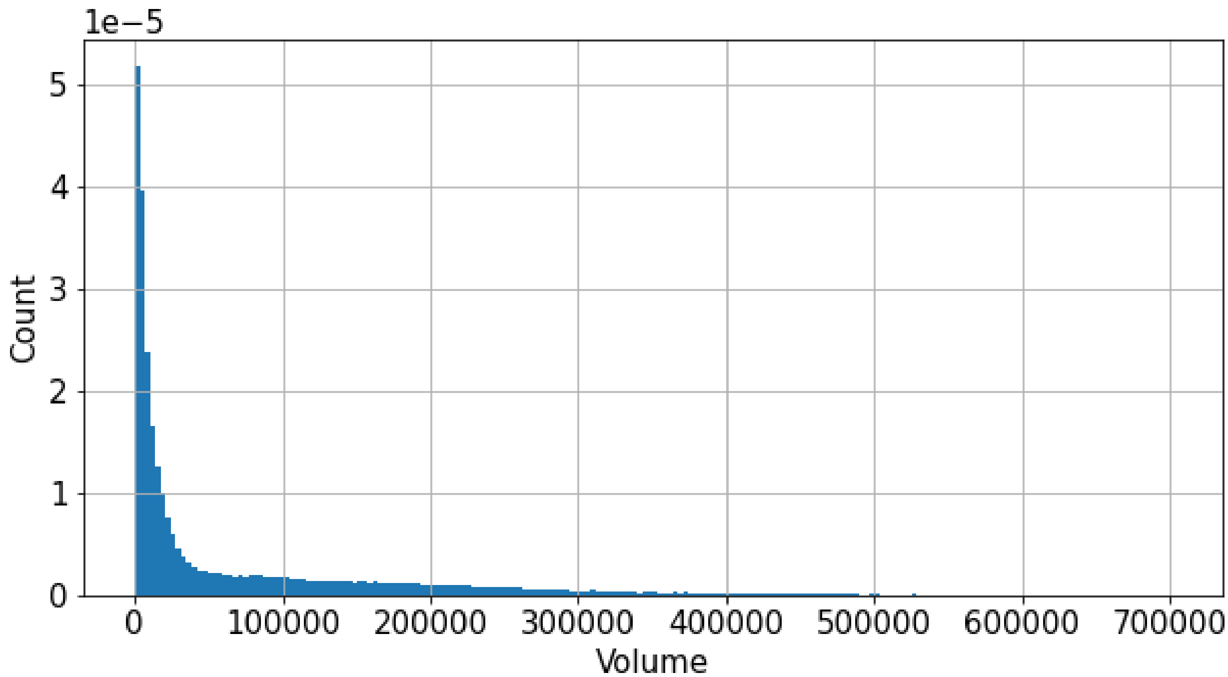

Figure 7: A histogram of hourly volume that includes all records in the data set.

The ‘movement’ variable is calculated later in the research for the purpose of profit calculation, but it has a rightful place in the data exploration section of this report. Table 4 and Figure 8 provide a summary of ‘movement’. The mean is very close to zero, being only slightly positive. The largest open-to-open movements were 142.75 points to the upside and -120.25 point to the downside. The range of the chart was reduced to show only the bins that caught enough counts to be visible. Figure 8 and Table 4 should illustrate just how rare large movements are. The vast majority of open-to-open hourly movements are under ten points either to the upside or the downside.

Table 4: Descriptive statistics of price movement from open-to-open across the entire data set.

```
1 ES['movement'].describe()
```

```
count    104418.000
mean          0.028
std           5.537
min        -120.250
25%          -1.250
50%           0.000
75%           1.500
max         142.750
Name: movement, dtype: float64
```

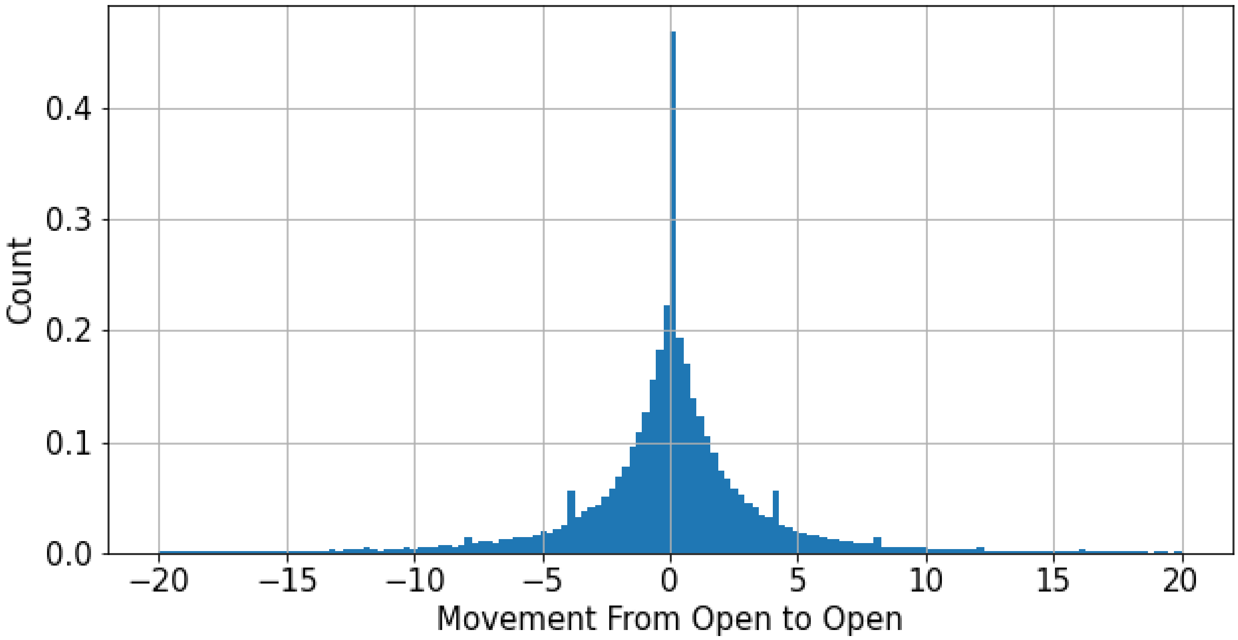

Figure 8: A histogram of hourly price movement.

A summary of candle 'span' from high to low can be read in Table 5 and Figure 9. The largest candle size in the data set was worth 205.5 points and the smallest, not surprisingly, was 0. As above, the range of the histogram chart in Figure 9 was reduced to show mostly visible bins. The average candle size is 5.8 points.

Table 5: Descriptive statistics of candle span from high to low for every period in the data set.

```
1 ES['span'].describe()
```

```
count    104439.000
mean          5.799
std           7.112
min           0.000
25%           2.000
50%           3.750
75%           6.750
max         205.500
Name: span, dtype: float64
```

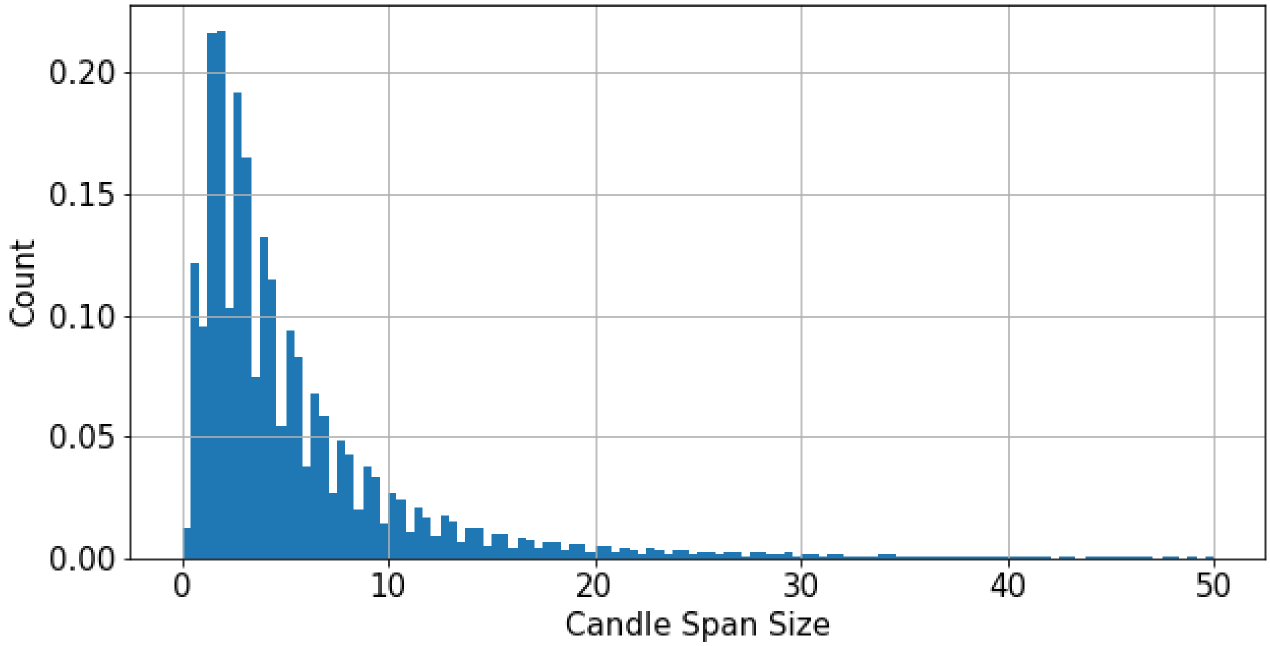

Figure 9: A histogram of candle span.

Table 6 and Figure 10 summarize the candle 'body' variable. As expected, the candle span descriptive statistics eclipse those of candle body. The implications of these measures lie outside the scope of this research, but there are insights hidden here that may arm the experienced trader with better judgement upon further study. For example, knowing how rare or common certain values of these metrics can help one to adhere to rational expectations on an hourly basis when managing trades.

Table 6: Descriptive statistics of candle body size, from open to close, including every record in the ES data.

```
ES['body'].describe()
```

```
count    104439.000
mean          2.820
std           4.687
min           0.000
25%           0.500
50%           1.500
75%           3.250
max         142.750
Name: body, dtype: float64
```

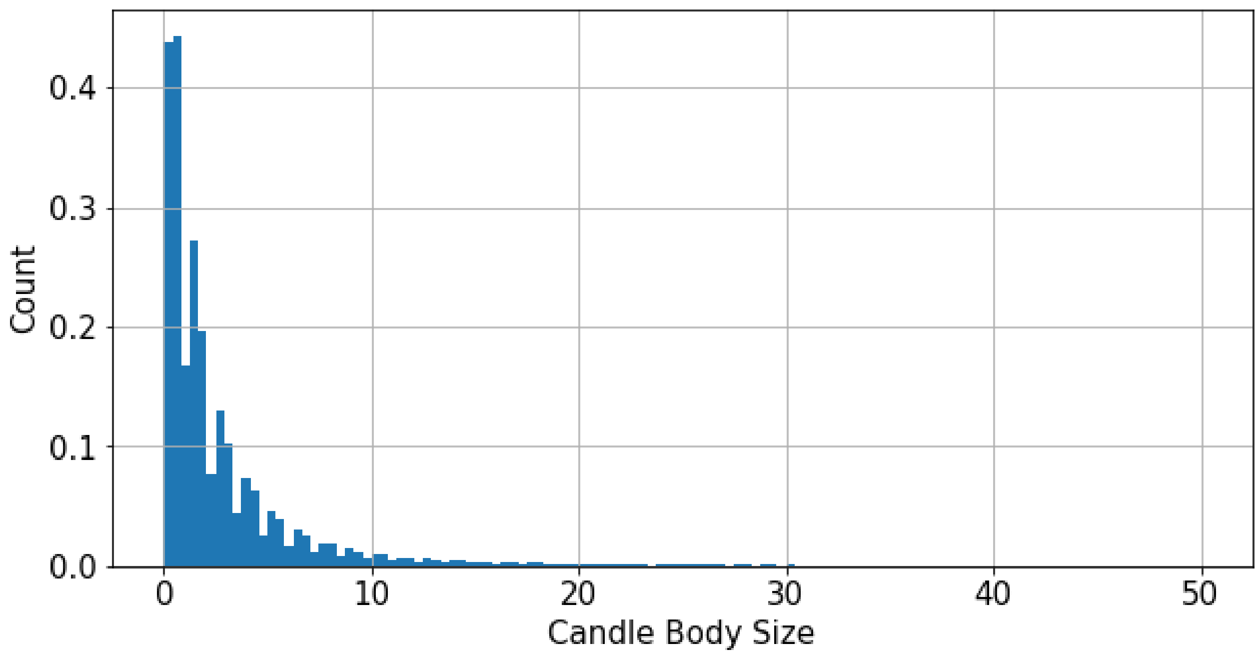

Figure 10: A histogram of candle body size.

Basic Back-Testing of Each Moving Average Style

The first part of this research attempts to identify, through back-testing, which moving average (MA) style, between the simple, weighted, and exponential styles, performed the best over the data span of the ES contract. These MA styles will be described in detail in later sections. Back-testing involves programmatically applying the trading rules of a given strategy to historical price data in an attempt to measure the strategy's effectiveness over time. [19] For this part of the study, a signal period of 9 and a control period of 20 on an hourly time frame are used and the goal is to discover whether any of these strategies would have outperformed a simple buy and hold strategy and which among them would have performed the best. A buy and hold strategy simply involves holding an asset without the intention of selling it over an extended period of time under the assumption that it will appreciate in value. The questions of this research arise from the common saying that one cannot consistently "beat the market", here meaning to earn more than the index appreciates over many consecutive years. It should be noted that the selection of time frame and signal/control lengths in this portion of the work is arbitrary and that the words 'period' and 'length' are interchangeable in the context of the signal and control throughout this writing.

The most important elements to this section of the work are the visualizations and the overall profit calculations. First, the code calculates profit as a product of price movement from open to open and the trading entry signals produced by the moving average crossovers. How exactly this is done with code and the logic behind the calculations will be thoroughly explained shortly. Then a section of code visualizes the price action, moving averages, buy/sell signals, and performance in the form of cumulative profit of each moving average. This whole process will be broken down step by step in the code explanations to follow. To accomplish this goal, signal and control moving averages for each of the MA styles must be calculated for a given price metric. The 'Open' metric suffices and provides the convenience of being available to use at the beginning of a period, as is not the case with the 'Close' metric. The 'High' and 'Low' metrics also can't be known until the close of the candle period. Any metric can be used in back testing a strategy, but the 'Open' metric is slightly more convenient. If another metric is used, some alterations will need to be made to the data to make sure that the signal of each cross-over strategy is matched properly with the appropriate measure of price movement.

The coding and calculation process for the simple moving average (SMA) is straightforward. First, objects are defined that contain the numerical period lengths for the signal and control SMA's. The code and output in Figure 11 show exactly how these tasks are accomplished programmatically using Python. The rolling().mean() function takes the selected price metric 'Open' from the data frame and creates two new columns 'sma_s' and 'sma_c' that are its signal and control MAs with respective period lengths. Notice from the 'Time_Unit' column that this is a display of records eight through twenty-two and note the missing observations due to MA calculation lag. The signal MA values start at record nine and the control values start on record 20 which are the same as their period lengths. Here again is the mathematical definition for the SMA for quick reference.

$SMA = (A_1 + A_2 + \ldots + A_n) / n$

Where:

A_1 = the price at position 1
A_n = the price at period n
n = number of total periods or the "period length"

```python
# The signal variable is used for calculations of fast (signal) MA's
# The control variable is used for calculations of slow (control) MA's
# 'prix_met' stands for price metric and is set to 'Open' here

signal = 9
control = 20
prix_met = 'Open'

ES.drop(columns=['sma_s', 'sma_c'], inplace=True, errors='ignore')

ES['sma_s'] = ES[prix_met].rolling(signal).mean()
ES['sma_c'] = ES[prix_met].rolling(control).mean()

ES[7:22] # Printing this range shows how the calculation omits records for our SMA's
```

Date	Open	High	Low	Close	Volume	Time_Unit	sma_s	sma_c
2005-09-06 23:00:00	1146.000	1146.250	1145.500	1145.750	637	8	NaN	NaN
2005-09-07 00:00:00	1145.500	1146.000	1145.500	1146.000	148	9	1146.083	NaN
2005-09-07 01:00:00	1146.000	1146.250	1146.000	1146.000	178	10	1146.056	NaN
2005-09-07 02:00:00	1146.000	1146.750	1145.750	1146.000	1020	11	1146.028	NaN
2005-09-07 03:00:00	1146.250	1147.000	1145.500	1146.000	4265	12	1146.000	NaN
2005-09-07 04:00:00	1146.000	1146.500	1145.500	1145.750	5183	13	1146.000	NaN
2005-09-07 05:00:00	1145.750	1146.000	1145.000	1145.250	1225	14	1145.944	NaN
2005-09-07 06:00:00	1145.250	1145.750	1145.000	1145.000	1239	15	1145.833	NaN
2005-09-07 07:00:00	1145.000	1146.000	1144.750	1146.000	2476	16	1145.750	NaN
2005-09-07 08:00:00	1145.750	1146.000	1144.250	1145.000	7409	17	1145.722	NaN
2005-09-07 09:00:00	1144.750	1146.000	1143.500	1144.500	69921	18	1145.639	NaN
2005-09-07 10:00:00	1144.500	1146.750	1143.250	1146.750	118638	19	1145.472	NaN
2005-09-07 11:00:00	1146.750	1148.250	1144.750	1145.750	88173	20	1145.556	1145.838
2005-09-07 12:00:00	1145.750	1147.500	1145.500	1147.000	24728	21	1145.500	1145.812
2005-09-07 13:00:00	1147.000	1148.500	1145.500	1147.500	62232	22	1145.611	1145.850

Figure 11: Calculating SMA signal and control columns from ES['Open'].

Calculating the weighted moving average (WMA) is not quite as simple, but only requires a small bit of mathematical understanding to conceptualize how it can be done. There are different ways to go about this, but here, each component of the rolling period is assigned a value based on which position it takes in the series window. The earliest period's price is assigned a weight value of one, the next gets a value of two, and so on until all periods within the length have a value. If a WMA has a period of 20, then the most recent period will have a weight value of 20. The weight value of each period is divided by the sum of all the weight values, so each period is now bound to some weighted fraction of one. All these fractions should add up to one. Then, the 'Open' price of each period is multiplied by its weighted fraction. To calculate the WMA from here simply requires summing the weighted product of each period and dividing it by the period length. Because the period components were weighted in ascending order, the most recent price value influences the WMA the most, which results in a MA that follows price action more closely and responsively. There will be an illustration later in the document of how these MAs differ via visualization once they have all been added to the data frame. The following is the mathematical definition of WMA used in this study and Table 7 shows the resulting columns of the calculations of 'wma_s' and 'wma_c' which are the signal and control WMAs respectively. [8]

$$WMA = (A_1*1 / s + A_2*2 / s + \ldots + A_n*n / s) / n$$

Where:

A_n = the price at period n
n = number of total periods or the “period length”
s = (1 + 2 + … + n)

Table 7: Displaying the new WMA signal and control columns.

```
ES[7:22]
```

Date	Open	High	Low	Close	Volume	Time_Unit	sma_s	sma_c	wma	wma_s	wma_c
2005-09-06 23:00:00	1146.000	1146.250	1145.500	1145.750	637	8	NaN	NaN	NaN	NaN	NaN
2005-09-07 00:00:00	1145.500	1146.000	1145.500	1146.000	148	9	1146.083	NaN	NaN	1145.972	NaN
2005-09-07 01:00:00	1146.000	1146.250	1146.000	1146.000	178	10	1146.056	NaN	NaN	1145.956	NaN
2005-09-07 02:00:00	1146.000	1146.750	1145.750	1146.000	1020	11	1146.028	NaN	NaN	1145.944	NaN
2005-09-07 03:00:00	1146.250	1147.000	1145.500	1146.000	4265	12	1146.000	NaN	NaN	1145.989	NaN
2005-09-07 04:00:00	1146.000	1146.500	1145.500	1145.750	5183	13	1146.000	NaN	NaN	1145.989	NaN
2005-09-07 05:00:00	1145.750	1146.000	1145.000	1145.250	1225	14	1145.944	NaN	NaN	1145.939	NaN
2005-09-07 06:00:00	1145.250	1145.750	1145.000	1145.000	1239	15	1145.833	NaN	NaN	1145.800	NaN
2005-09-07 07:00:00	1145.000	1146.000	1144.750	1146.000	2476	16	1145.750	NaN	NaN	1145.633	NaN
2005-09-07 08:00:00	1145.750	1146.000	1144.250	1145.000	7409	17	1145.722	NaN	NaN	1145.633	NaN
2005-09-07 09:00:00	1144.750	1146.000	1143.500	1144.500	69921	18	1145.639	NaN	NaN	1145.439	NaN
2005-09-07 10:00:00	1144.500	1146.750	1143.250	1146.750	118638	19	1145.472	NaN	NaN	1145.211	NaN
2005-09-07 11:00:00	1146.750	1148.250	1144.750	1145.750	88173	20	1145.556	1145.838	1145.670	1145.467	1145.670
2005-09-07 12:00:00	1145.750	1147.500	1145.500	1147.000	24728	21	1145.500	1145.812	1145.662	1145.506	1145.662
2005-09-07 13:00:00	1147.000	1148.500	1145.500	1147.500	62232	22	1145.611	1145.850	1145.775	1145.806	1145.775

The final indicator to be calculated and tested is the exponential moving average (EMA), for which the equation is surprisingly unlike the other two. Generally, the EMA of the current period is calculated using the price value of the current period and the EMA of the period directly prior. The first order of operation is to calculate a value for the simple moving average at the period length. The first EMA value is calculated by adding two terms together. The first term is the price directly after the initial SMA calculation multiplied by the smoothing factor, and the second is the prior period’s SMA multiplied by one minus the smoothing factor, or the multiplicative inverse of the smoothing factor. By convention, the smoothing factor or “alpha multiplier” is defined as 2/(n+1), so a longer period EMA gives less weight to the current period’s price than a shorter period EMA. As subsequent EMA values are calculated, the prior period’s EMA value is substituted in the place of the initial SMA value in the second term of the first EMA calculation. This process continues to create an EMA curve of the open price in this case. Table 8 displays a slice of the data after the signal and control EMA columns have been calculated and added to the frame. [8]

$EMA_c = P_c * a + EMA_p * (1-a)$

Where:

EMA_c = current EMA value
P_c = current price
$a = 2 / (1+n)$
n = length of period
EMA_p = EMA value of the prior period

Table 8: Displaying the new EMA signal and control columns.

```
ES[7:22]
```

Date	Open	High	Low	Close	Volume	Time_Unit	sma_s	sma_c	wma_s	wma_c	ema_s	ema_c
2005-09-06 23:00:00	1146.000	1146.250	1145.500	1145.750	637	8	NaN	NaN	NaN	NaN	NaN	NaN
2005-09-07 00:00:00	1145.500	1146.000	1145.500	1146.000	148	9	1146.083	NaN	1145.972	NaN	NaN	NaN
2005-09-07 01:00:00	1146.000	1146.250	1146.000	1146.000	178	10	1146.056	NaN	1145.956	NaN	1146.067	NaN
2005-09-07 02:00:00	1146.000	1146.750	1145.750	1146.000	1020	11	1146.028	NaN	1145.944	NaN	1146.053	NaN
2005-09-07 03:00:00	1146.250	1147.000	1145.500	1146.000	4265	12	1146.000	NaN	1145.989	NaN	1146.093	NaN
2005-09-07 04:00:00	1146.000	1146.500	1145.500	1145.750	5183	13	1146.000	NaN	1145.989	NaN	1146.074	NaN
2005-09-07 05:00:00	1145.750	1146.000	1145.000	1145.250	1225	14	1145.944	NaN	1145.939	NaN	1146.009	NaN
2005-09-07 06:00:00	1145.250	1145.750	1145.000	1145.000	1239	15	1145.833	NaN	1145.800	NaN	1145.857	NaN
2005-09-07 07:00:00	1145.000	1146.000	1144.750	1146.000	2476	16	1145.750	NaN	1145.633	NaN	1145.686	NaN
2005-09-07 08:00:00	1145.750	1146.000	1144.250	1145.000	7409	17	1145.722	NaN	1145.633	NaN	1145.699	NaN
2005-09-07 09:00:00	1144.750	1146.000	1143.500	1144.500	69921	18	1145.639	NaN	1145.439	NaN	1145.509	NaN
2005-09-07 10:00:00	1144.500	1146.750	1143.250	1146.750	118638	19	1145.472	NaN	1145.211	NaN	1145.307	NaN
2005-09-07 11:00:00	1146.750	1148.250	1144.750	1145.750	88173	20	1145.556	1145.838	1145.467	1145.670	1145.596	NaN
2005-09-07 12:00:00	1145.750	1147.500	1145.500	1147.000	24728	21	1145.500	1145.812	1145.506	1145.662	1145.627	1145.574
2005-09-07 13:00:00	1147.000	1148.500	1145.500	1147.500	62232	22	1145.611	1145.850	1145.806	1145.775	1145.901	1145.710

Notice how the values of the EMA columns at the end begin one record after the other two MA styles. This occurs because the first SMA value was needed, in both the case of the signal and the control, to calculate the first EMA value.

Figure 12 visually compares how these MAs behave differently from one another. They remain close to one another and behave similarly, but their calculations cause them to frequently diverge. The pink curve is the WMA, the orange is the EMA, and the dark purple curve is the SMA. All three in the visualization were calculated with a period length of 20. The most notable difference occurs between the SMA and the WMA. The WMA, it being the case that more recent observations are weighted more heavily in its calculation, stays closer to price action and changes direction earlier than the SMA when price action changes direction. This can lead to better entries and exits, but also to more fake outs during volatile conditions where the trend direction is strong with sudden large movements against it. This causes the algorithm to change direction momentarily and misguidedly, resulting in papercut losses and messy trading behavior. The SMA is generally a smoother curve and causes an algorithm to stay in positions for longer. This can be a good thing in emotional markets, but entries and exits won't be executed as promptly or aggressively. Due to the erratic nature of price action and the magnitude of all the possible scenarios that can occur, an exhaustive analysis of their behavioral differences won’t be included here. It suffices to understand that the different MAs cause different trading activity that results in different profitability in different technical scenarios.

Figure 12: Visualizing the differences between the behaviors of the three MA styles.

Trading signals can be derived now that the signal and control MA columns have been generated for the simple, weighted, and exponential MA styles. Recall that if the signal is above the control, the strategy mandates a long position and if the signal is below the control, the strategy requires a short position. The following Python function in Figures 13 and 14 shows how to do this programmatically. Table 9 shows the signal and control MAs and their signals isolated from the rest of the columns in the data set.

```python
def signal_gen(signal, ma_signal, ma_control):
    ES[signal] = np.where(ES[ma_signal] > ES[ma_control], 1, 0) # Long signal
    ES[signal] = np.where(ES[ma_signal] < ES[ma_control], -1, ES[signal]) # Short signal
```

Figure 13: Defining a function that generates trading signals based on the signal and control MA’s positions relative to one another.

```python
signal_gen('signal','sma_s','sma_c')
signal_gen('signal2','wma_s','wma_c')
signal_gen('signal3','ema_s','ema_c')
```

Figure 14: Applying the function to each MA style pair to generate trading signals for each.

Table 9: Printing the directional trading signals next to their respective MA pairs to confirm that the signals were properly generated. A value of 1 represents a long position and a value of -1 represents a short position.

```
ES[['sma_s', 'sma_c', 'signal', 'wma_s', 'wma_c', 'signal2', 'ema_s', 'ema_c', 'signal3']][80:90]
```

Date	sma_s	sma_c	signal	wma_s	wma_c	signal2	ema_s	ema_c	signal3
2005-09-12 01:00:00	1154.917	1151.800	1	1155.017	1153.667	1	1154.382	1152.187	1
2005-09-12 02:00:00	1155.083	1152.188	1	1154.983	1153.948	1	1154.456	1152.431	1
2005-09-12 03:00:00	1155.194	1152.650	1	1155.017	1154.239	1	1154.615	1152.699	1
2005-09-12 04:00:00	1155.083	1153.050	1	1154.828	1154.392	1	1154.542	1152.847	1
2005-09-12 05:00:00	1154.694	1153.388	1	1154.411	1154.387	1	1154.233	1152.862	1
2005-09-12 06:00:00	1154.389	1153.713	1	1154.072	1154.350	-1	1153.987	1152.875	1
2005-09-12 07:00:00	1154.139	1153.987	1	1153.694	1154.235	-1	1153.689	1152.839	1
2005-09-12 08:00:00	1153.833	1154.225	-1	1153.367	1154.093	-1	1153.451	1152.807	1
2005-09-12 09:00:00	1153.611	1154.237	-1	1153.200	1153.976	-1	1153.361	1152.825	1
2005-09-12 10:00:00	1153.583	1154.287	-1	1153.278	1153.954	-1	1153.489	1152.937	1

The next step, to the end goal of determining how each moving average style would have compared to the others in terms of profitability, is to multiply the trading signal values by the movement of the open price from period to period. There are four possible outcomes for each period, not considering the magnitude of price movement:

1. The signal is long, the algorithm instructs the trading platform to buy contracts, and the price movement is positive as the model predicts (up): resulting in + profit for that period.
2. The signal is short, the algorithm instructs the trading platform to sell contracts short, and the price movement is negative as the model predicts (down): resulting in + profit for that period.
3. The signal is long, the algorithm instructs the trading platform to buy contracts, and the price movement is negative against the model's prediction (down): resulting in – loss for that period.
4. The signal is short, the algorithm instructs the trading platform to sell contracts short, and the price movement is positive against the model's prediction (up): resulting in – loss for that period.

The model predictions are far from perfect. There will be many instances where a user decides on a trading direction based on what the algorithm predicts, but price moves in the opposite direction of what the strategy predicts. The user will enjoy a capital gain when scenarios 1 or 2 play out and suffer a capital loss when scenarios 3 or 4 play out.

Because the signal for each MA style pair is generated at the beginning of the period, the price movement needs to be matched with the signal, which requires that the 'movement' column be shifted up one row. This gives an accurate matching of the directional trading signal of each period with the associated movement of price. The code and output below in Figure 15 show how the point movement now corresponds with the starting point value and not the ending point value. After this step, the point profit value for each period can be calculated by multiplying the directional trading signal by the open price movement. Table 10 previews the result of the latter step.

```python
# This line calculates movement from Open to Open and shifts it up one row
ES['movement'] = ES['Open'].diff().shift(-1)
```

```python
ES[['Open', 'movement']][1000:1010]
```

Date	Open	movement
2005-11-06 20:00:00	1127.000	0.000
2005-11-06 21:00:00	1127.000	-0.500
2005-11-06 22:00:00	1126.500	-0.750
2005-11-06 23:00:00	1125.750	0.000
2005-11-07 00:00:00	1125.750	0.000
2005-11-07 01:00:00	1125.750	0.250
2005-11-07 02:00:00	1126.000	1.250
2005-11-07 03:00:00	1127.250	0.250
2005-11-07 04:00:00	1127.500	-0.250
2005-11-07 05:00:00	1127.250	-0.750

Figure 15: Generating the open price 'movement' column.

Table 10: Showing the point profit columns for each MA style pair after their calculation.

Date	sma_point_profit	wma_point_profit	ema_point_profit
2005-09-07 12:00:00	-1.250	-1.250	1.250
2005-09-07 13:00:00	-0.500	0.500	0.500
2005-09-07 14:00:00	-0.000	0.000	0.000
2005-09-07 15:00:00	1.250	1.250	1.250
2005-09-07 16:00:00	-1.500	-1.500	-1.500
...	...	...	...
2023-01-13 11:00:00	6.000	6.000	6.000
2023-01-13 12:00:00	-9.250	-9.250	-9.250
2023-01-13 13:00:00	-8.000	-8.000	-8.000
2023-01-13 14:00:00	-1.750	-1.750	1.750
2023-01-13 15:00:00	-8.000	8.000	8.000

Now the point profit columns for each pair can be totaled to reveal the hypothetical past performance of each strategy using the different MA styles. Recall that the signal length is 9 and the control length is 20. It would be possible to exhaustively run this script manually with every conceivable combination of signal and control periods to get the performance of each set of trading rules for each MA style. It makes much more sense to do this programmatically, though, which will be the next large part of this work to be discussed.

The subject of this study is the e-mini S&P 500 futures contract which is worth $50 per point. To have bought an ES contract at the date where this data starts (minus the rows that contained N/As along the way due to MA calculations) and have consistently rolled over to each newest contract before the prior contract's expiry until the date where the data ends would have resulted in $143,200 in profit. This would be the simple buy and hold strategy in the context of futures. Table 11 displays each strategy's aggregated outcome with each trend-following strategy (moving average cross-over) having a 9-period signal and a 20-period control. The clear winner in this case is the buy and hold strategy with the SMA coming in 2nd and the WMA coming in last place in terms of performance. It may be tempting to draw the conclusion that the SMA style is superior to the others in terms of profitability, but this would be a premature assumption. It may well be the case, but further analysis is warranted to better inform a decision about this.

Table 11: Total profit for each moving average on a one-hour time frame with a 20-period control and 9-period signal spanning the entire data set.

Strategy	Returns
Buy and Hold	$143200.00
Simple Trend	$59175.00
Weighted Trend	$-74375.00
Exponential Trend	$20625.00

A visualization of the work that has been done thus far will wrap up this section. The chart below in Figure 15 shows the price action of the last 500 hours of the data set range of ES 'Open' (blue) with SMA9 (yellow) and SMA20 (green) following along. The arrows indicate the change in trading position where SMA9 crosses SMA20. Green arrows indicate where the algorithm would have chosen long positions and the red arrows indicate where it would have sold short. As it is built, the algorithm is never idle and is always either in a long or short position. The other lines on the chart that are hovering above the real price action and SMAs represent the cumulative profit of each moving average style holding constant the lengths for signal and control. The SMA cumulative profit curve is in red, the profit curve for the WMA is in purple, and the curve for the EMA is brown. These cumulative profit sums at every hour have been altered to begin at the same price point as the contract to better illustrate the difference in growth patterns between the buy/hold strategy (which is the 'Open' price action of the contract, "blue line") and the MA strategies. One element of this chart that should stand out is the growth that the MA trading strategies enjoy while price declines. This divergence occurs because of the inability of a buy and hold strategy to return profit from a declining technical environment. [20] Take note where the cumulative profit curves decline. During this period, the trend of the blue price action line is horizontal, and the action stays volatile around its moving averages. Periods like these are bad for trend-following MA systems because the algorithm produces false signals with higher frequency. An MA system performs best when price action is stronger and more consistently moving in one direction with little noise. Unfortunately, these conditions are somewhat rare in time-based price data.

Figure 15: A visualization of the 9/20 SMA trading strategy in action along with the cumulative profit of each MA style for comparison for the last 500 hours of the data set.

Now would be a valid moment to mention the difference between time-based orientation and volume/tick-based orientation in working with price data. When a time frame is spoken of regarding price charts, one refers to the record keeping of price per unit of time, but there are different ways to keep record of price. Price can also be recorded and visualized in terms of volume (the number of contracts traded) or ticks (the number of times that price moves). These ways of conceptualizing the movement of price may significantly impact the performance of moving averages in signaling trade opportunities because of the way that they smooth price action into clearly identifiable crests and ridges with waves that tend to exhibit very defined slopes. As stated before, the moving average strategy depends on a steady wave slope to work its best. More erratic price fluctuation (noise) and flat price trend can lead to profit-eating numbers of false signals. Tick and volume-based price data lies beyond the scope of this writing and won't be considered here, though their eventual inclusion will serve as the next milestone in the continued progress of this project, long-term. It is also worth mentioning that this entire analysis can also be conducted on the one-minute, five-minute, and thirty-minute time frames that were included in the purchased data file that contains the one-hour data set. All that would require is a small change to the code that reads in the text file, and the rest of the script will run through. One could use this same script to analyze any futures ticker data, but minor changes to the code in several places would be in order for it to make sense, such as the futures ticker name and contract point value.

Iterating Through Trading Rule Configurations

Now that the ranking for one trading rule pair (9-period signal and 20-period control) of each strategy has been established, as well as how they compare to a buy and hold strategy, the questions become: which signal and control combination, with period length values between one and fifty, would have generated the most profit for each MA style? Which MA is associated with the most profitability across all trading rule pairs? The first order in the process of iterating through all possibilities given a range of period lengths is to create a list of said range which can be seen in Figure 16. Once that is obtained, the task becomes to iterate through the list with a for loop to calculate signal and control SMAs with each length. This process adds 100 columns to the data frame. Their calculation and naming can be seen in the code chunk below in Figure 18. It was necessary to create separate copies of the original data frame for each MA style this time around to keep their sizes manageable which is the code in Figure 17.

```
MAs = list(range(1, 51))
prix_met = 'Open'
```

Figure 16: Creating a list of period lengths for use in iteration.

```
# Creating copies of the data set for the other two MA styles
ESw = ES.copy()
ESe = ES.copy()
```

Figure 17: Making separate copies of ES for WMA and EMA calculations.

```
sma_s_names = []

for ma in MAs :

    ES[f"sma_s{ma}"] = ES[prix_met].rolling(ma).mean()
    sma_s_names.append(f"sma_s{ma}")

sma_c_names = []

for ma in MAs :

    ES[f"sma_c{ma}"] = ES[prix_met].rolling(ma).mean()
    sma_c_names.append(f"sma_c{ma}")
```

Figure 18: Generating columns and lists of column names for signal SMAs with periods 1-50 and columns for control SMA's with periods 1-50.

This next chunk of code in Figure 19 generates directional trading signals for every possible combination of SMA signal and control lengths within the specified range of (1, 50), or every possible trading rule configuration within the range of possible period lengths, in other terms. This chunk, and the two that are similar to it, by far required the most computational power and time to run. Below it in Figure 20 is a header display of how these columns were named. This code and calculation add 2500 columns to the data frame and this process will be repeated twice more for the other two MA styles. In all, this part of the MA iteration process creates 7,900 new columns * 104,439 rows = 825,068,100 observations.

```
# The SMA signal generator
signal_cols_s = []

for sname in sma_s_names :
    for cname in sma_c_names :
        # Long signal
        ES[f"{sname} {cname} signal"] = np.where(ES[sname] > ES[cname], 1, 0)
        # Short signal
        ES[f"{sname} {cname} signal"] = np.where(ES[sname] < ES[cname], -1, ES[f"{sname} {cname} signal"])
        ES.dropna(inplace=True)
        signal_cols_s.append(f"{sname} {cname} signal")
```

Figure 19: The code chunk that generates pairwise signals for all possible combinations of the SMA signal and control columns.

```
signal_cols_s
```

```
['sma_s1 sma_c1 signal',
 'sma_s1 sma_c2 signal',
 'sma_s1 sma_c3 signal',
 'sma_s1 sma_c4 signal',
 'sma_s1 sma_c5 signal',
 'sma_s1 sma_c6 signal',
```

Figure 20: Showing how the signal columns are named.

The function for creating WMAs by iterating through the period list is largely the same as what was built for the back-testing section but was not shown there. The function is displayed below in Figure 21. The only difference is the extra 'rule' variable that is needed for naming the columns in an orderly fashion. To avoid too much redundancy, the rest of the iteration code is not displayed for the WMA since it is very similar to the code for the SMA. These columns were calculated and stored in a nearly identical way to the SMA columns. Note that this data frame is called 'ESw' and is a copy of 'ES' that was made specifically for the WMA columns.

```
data = ESw['Open']

def weighted_moving_average(data, period, rule):

    weights = []
    weights_calc = []

    # Calculating weights
    for i in range(period + 1) :
        pd.to_numeric(weights_calc.append(i))
    for i in weights_calc :
        weights.append(i/sum(weights_calc))
    weights.pop(0)

    # Using weights to calculate the WMAs
    wma = []
    for i in range(len(data) - period + 1):
        wma.append(sum(data[i:i+period] * weights) / sum(weights))

    # Adding the WMAs to ES data frame
    wma = [float('nan')] * (period - 1) + wma
    ESw[f"wma{rule}{period}"] = wma
```

Figure 21: The function for creating many WMAs at once using an iterative for loop.

Again, to avoid too much redundancy, only the function for the EMA is shown below in Figure 22, as the process of creating all the EMA columns is identical to how the other two were done apart from their calculations. This function was modified in the same way as the one for the WMA to make it suitable for iterating through the period range and naming the columns appropriately as they were calculated at the same time.

```python
def exponential_moving_average(data, column_name, period, rule):

    # Calculate the weighting multiplier
    alpha = 2 / (period + 1)

    # Calculate the initial EMA value using a simple moving average
    sma = data[column_name].rolling(window=period, min_periods=period).mean()
    ema_init = sma[period-1]

    # Calculate the EMA values for the remaining data
    ema_values = []
    for i in range(period, len(data)):
        ema = alpha * data[column_name][i] + (1 - alpha) * ema_init
        ema_init = ema
        ema_values.append(ema)

    # Adding the EMAs to the ES data frame
    ema_values = [float('nan')] * (period) + ema_values
    data[f"ema{rule}{period}"] = ema_values
```

Figure 22: Defining the function for creating many EMAs at once using an iterative for loop.

Recall from the prior section that after the signals have been generated, the next step in the process is to calculate a profit column and then sum the column and multiply it by point value in dollars to show the overall performance of a strategy's trading rule configuration. This is done for all 7500 trading rule configurations in Figures 24 and 25 which will then be visualized for each MA style, resulting in three separate profit heat maps, each with 50 rows and columns representing the signal/control MA period length permutations, and each map representing the performances of 2500 trading rule configurations. Before any of that can be accomplished, a price movement column must be built by which to multiply the signal columns to get point profit columns as seen in Figure 23. To reiterate, the movement column is shifted up one row because the movement value must be paired with the signal that determines the trading direction, and thus, profit or loss for its period. Making a trade decision at the very beginning of a period and sticking with it until the beginning of the next means that the movement from the beginning of the decision period to the beginning of the new period is the movement of the decision period on which profit or loss is based. Figure 26 displays how the point profit columns are named.

```python
# These lines create 'movement' columns in each MA's data frame
ES['movement'] = ES['Open'].diff().shift(-1)
ESw['movement'] = ESw['Open'].diff().shift(-1)
ESe['movement'] = ESe['Open'].diff().shift(-1)
```

Figure 23: Creating columns that contain price movement from period to period using the 'Open' metric.

```python
profit_cols_s = []
profit_cols_w = []
profit_cols_e = []

def point_profit_gen(df, sig_cols, prof_cols):
    for sig_col in sig_cols :
        df[f"{sig_col} point profit"] = df['movement']*df[sig_col]
        prof_cols.append(f"{sig_col} point profit")

    df.dropna(inplace=True)
```

Figure 24: Defining a function that calculates point profit columns to add each data frame and creates lists of those column names.

```
point_profit_gen(ES, signal_cols_s, profit_cols_s)
point_profit_gen(ESw, signal_cols_w, profit_cols_w)
point_profit_gen(ESe, signal_cols_e, profit_cols_e)
```

Figure 25: Applying the point profit generation function to each data frame.

```
profit_cols_s = profit_cols_s.copy()
profit_cols_s
```

```
['sma_s1 sma_c1 signal point profit',
 'sma_s1 sma_c2 signal point profit',
 'sma_s1 sma_c3 signal point profit',
 'sma_s1 sma_c4 signal point profit',
 'sma_s1 sma_c5 signal point profit',
 'sma_s1 sma_c6 signal point profit',
 'sma_s1 sma_c7 signal point profit',
 'sma_s1 sma_c8 signal point profit',
 'sma_s1 sma_c9 signal point profit',
```

Figure 26: Viewing how the profit columns were named using SMA profit columns as an example.

For quick reference, the first code chunk and output in Figure 27 are for the total profit for a buy and hold strategy. The code chunk after in Figure 28 defines a function that totals the point profit columns of each trading rule configuration of each MA style and multiplies those figures by the number of contracts and the point value of each contract. The function then prints each trading pair's total profit. The number of contracts was set to one and the point value of each contract was set to 50 since this analysis was conducted on the ES contract which has a point value of $50. The lines of code following the definition of 'calc_profit' in Figure 29 show the application of this total profit calculation function to the point profit columns for each trading rule pair for each MA style. Only the output head of the EMA data frame is shown for orientation to what transformations have occurred. The 'profit_totals' lists contain only the profit totals of each trading rule pair and were appended to new tables that only contain the trading rule pairs and associated total profit over the span of the data set.

```
# Number of e-mini contracts traded
# (one micro is one tenth of an e-mini, a full contract is five e-minis)
# Full value: $250
# e-mini value: $50
# Micro value: $5

contracts = 1
point_value = 50

#buy/hold futures dollars
dollars_bh = ((ES[prix_met][-1])-(ES[prix_met][0]))*point_value*contracts
dollars_bh = "${:.2f}".format(dollars_bh)

print('Buy and Hold Strategy Returns')
print(dollars_bh)
```

```
Buy and Hold Strategy Returns
$143337.50
```

Figure 27: Recalling the return for a buy and hold strategy from the beginning to the end of the modified data.

```python
profit_totals_s = []
profit_totals_w = []
profit_totals_e = []

def calc_profit(data, prof_cols, prof_tots):
    for prof_col in prof_cols :
        prof_tots.append(data[prof_col].sum()*point_value*contracts)

        print('Returns for '+prof_col.replace(' signal point profit', '')+ ' Strategy')
        print("${:.2f}".format(data[prof_col].sum()*point_value*contracts))
        print()
```

Figure 28: Defining a function that totals each point profit column and multiplies it by the point value and number of contracts traded.

```python
calc_profit(ES, profit_cols_s, profit_totals_s)
```

```python
calc_profit(ESw, profit_cols_w, profit_totals_w)
```

```python
calc_profit(ESe, profit_cols_e, profit_totals_e)
```

```
Returns for ema_s1 ema_c1 Strategy
$0.00

Returns for ema_s1 ema_c2 Strategy
$-170012.50

Returns for ema_s1 ema_c3 Strategy
$-149312.50

Returns for ema_s1 ema_c4 Strategy
$-142112.50

Returns for ema_s1 ema_c5 Strategy
$-75412.50
```

Figure 29: Running the prior function on each data frame and viewing the output head of its application to the EMA data frame.

The next step towards visualizing the findings of total profit for all MA styles and their trading rule pairs requires some new and simplified tables. The first code chunk below in Figure 30 builds empty tables for each MA style to then be populated with their respective trading rule pair and associated total profit data with the second code chunk in Figure 31. After, each strategy performance table is sorted by profitability and the most and least profitable signal and control pairs are displayed for the SMA, WMA, and EMA in tables 12, 13, and 14 respectively.

```python
# Create column name list.
col_names =  ['strategy', 'profit']

# Create an empty dataframe with columns.
strategy_performances_s  = pd.DataFrame(columns = col_names)
strategy_performances_w = strategy_performances_s.copy()
strategy_performances_e = strategy_performances_s.copy()
```

Figure 30: Creating separate data frames for trading rule pair and performance of each MA.

```python
strategy_performances_s.strategy = signal_cols_s
strategy_performances_s.profit = profit_totals_s

strategy_performances_w.strategy = signal_cols_w
strategy_performances_w.profit = profit_totals_w

strategy_performances_e.strategy = signal_cols_e
strategy_performances_e.profit = profit_totals_e
```

```python
strat_perfs_s = strategy_performances_s.sort_values(by=['profit'], ascending = False)
strat_perfs_w = strategy_performances_w.sort_values(by=['profit'], ascending = False)
strat_perfs_e = strategy_performances_e.sort_values(by=['profit'], ascending = False)
```

Figure 31: Assigning trading rule pair and respective performance to new data frames and sorting by performance.

Table 12: Viewing the top and bottom performers of the SMA.

```python
strat_perfs_s
```

	strategy	profit
1323	sma_s27 sma_c24 signal	327450.0
1373	sma_s28 sma_c24 signal	285512.5
1677	sma_s34 sma_c28 signal	282437.5
1578	sma_s32 sma_c29 signal	282137.5
1678	sma_s34 sma_c29 signal	277337.5
...	...	...
1433	sma_s29 sma_c34 signal	-277337.5
1431	sma_s29 sma_c32 signal	-282137.5
1383	sma_s28 sma_c34 signal	-282437.5
1177	sma_s24 sma_c28 signal	-285512.5
1176	sma_s24 sma_c27 signal	-327450.0

2500 rows × 2 columns

Table 13: Viewing the top and bottom performers of the WMA.

```python
strat_perfs_w
```

	strategy	profit
1630	wma_s33 wma_c31 signal	195950.0
1679	wma_s34 wma_c30 signal	193500.0
1981	wma_s40 wma_c32 signal	190350.0
1834	wma_s37 wma_c35 signal	189100.0
1631	wma_s33 wma_c32 signal	186587.5
...	...	...
1582	wma_s32 wma_c33 signal	-186587.5
1736	wma_s35 wma_c37 signal	-189100.0
1589	wma_s32 wma_c40 signal	-190350.0
1483	wma_s30 wma_c34 signal	-193500.0
1532	wma_s31 wma_c33 signal	-195950.0

2500 rows × 2 columns

Table 14: Viewing the top and bottom performers of the EMA.

```
1 strat_perfs_e
```

	strategy	profit
50	ema_s2 ema_c1 signal	170012.5
551	ema_s12 ema_c2 signal	156212.5
501	ema_s11 ema_c2 signal	151862.5
100	ema_s3 ema_c1 signal	149312.5
150	ema_s4 ema_c1 signal	142112.5
...	...	...
3	ema_s1 ema_c4 signal	-142112.5
2	ema_s1 ema_c3 signal	-149312.5
60	ema_s2 ema_c11 signal	-151862.5
61	ema_s2 ema_c12 signal	-156212.5
1	ema_s1 ema_c2 signal	-170012.5

2500 rows × 2 columns

The following data frame in Table 15 is a result of splitting the strategy column of each profit table into signal and control period length columns using some light text manipulation. The signal and control columns are then converted to numerical data so that these figures can be plotted in order on x and y axes.

Table 15: An example of a 'strat_perfs' data frame using the SMA version after parsing the strategy column into new rule component specific columns.

```
1 strat_perfs_s
```

	strategy	profit	sma_s	sma_c
1323	sma_s27 sma_c24 signal	327450.0	27	24
1373	sma_s28 sma_c24 signal	285512.5	28	24
1677	sma_s34 sma_c28 signal	282437.5	34	28
1578	sma_s32 sma_c29 signal	282137.5	32	29
1678	sma_s34 sma_c29 signal	277337.5	34	29
...	...	...	...	...
1433	sma_s29 sma_c34 signal	-277337.5	29	34
1431	sma_s29 sma_c32 signal	-282137.5	29	32
1383	sma_s28 sma_c34 signal	-282437.5	28	34
1177	sma_s24 sma_c28 signal	-285512.5	24	28
1176	sma_s24 sma_c27 signal	-327450.0	24	27

2500 rows × 4 columns

Now that some neat profit-to-rule pair tables have been built for each MA style, heat maps that visualize total profit for each MA and each rule pair over the span of the data set can finally be built. The following code chunk in Figure 32 defines 'heat_map_viz' which does precisely that.

```python
def heat_map_viz(title, strat_perf, _ma_s, _ma_c):
    plt.title(title)
    plt.rcParams['figure.figsize'] = 20,15
    X = strat_perf[_ma_s]
    Y = strat_perf[_ma_c]
    Z = strat_perf['profit']
    data = pd.DataFrame({'X': X, 'Y': Y, 'Z': Z})
    data_pivoted = data.pivot("X", "Y", "Z")
    ax = sns.heatmap(data_pivoted)
    plt.xlabel('Control Period')
    plt.ylabel('Signal Period')
    ax.invert_yaxis()
    plt.show()
```

Figure 32: Defining a function that visualizes the total point profit of each trading rule configuration.

The first heat map shown below in Figure 33 uses SMA data. It’s important to note that the plot function above plots variable ‘X’ on the y-axis and the variable ‘Y’ on the x-axis. The signal length for each MA is defined as ‘X’ and is plotted on the y-axis. The control length for each MA is defined as ‘Y’ and is plotted on the x-axis. The ‘Z’ variable is total profit and is expressed through “heat” or color variation. Darker purple colors represent that the rule pair lost money and lighter orange-beige colors represent positive profit figures. Shades of fuchsia represent pairs that broke even or came close to it. The ranges are different for each map, so a brighter color on one map doesn’t necessarily mean that pair was more profitable than a darker colored pair on another map.

There is a noticeable diagonal line of symmetry that stems from the origin and rises to the upper right with slope = 1 in each heat map. This occurs because the code does not specify that the signal MA must have a smaller period length value than the control MA, which is typically the convention with MA trend-following strategies. As such, the visualization includes the inverse of each strategy where signal < control to include each strategy where signal > control. In effect, this allows one to see clearly that making the opposite decisions of a severely losing MA strategy can result in a market-beating strategy. All observations (rule pairs) below or to the right of this line of symmetry have a signal length that is shorter than its corresponding control length, which is the conventional way of thinking about MA cross-over strategies. All others that are to the left or above the diagonal are inverse cross-over strategies. An inverse cross-over strategy, where signal length > control length, simply implies that the opposite directional trading decision is made for any rule pair where signal length < control length. For example, there is a bright node at (24, 27) which is an inverse cross-over strategy with a control of 24 and a signal of 27, of the conventional strategy at (27, 24) with a control of 27 and a signal of 24. There is a saying that no technical system can consistently beat the market year over year. The (24, 27) strategy’s performance seems to suggest otherwise with a total profit of $327,450 over the span of the data, which is more than double of what a buy and hold strategy would have earned. In fact, there are several visible clusters of trading rules that would have outperformed the market in Figure 33.

```
1  heat_map_viz('SMA Strategy Profits', strat_perfs_s, 'sma_s', 'sma_c')
```

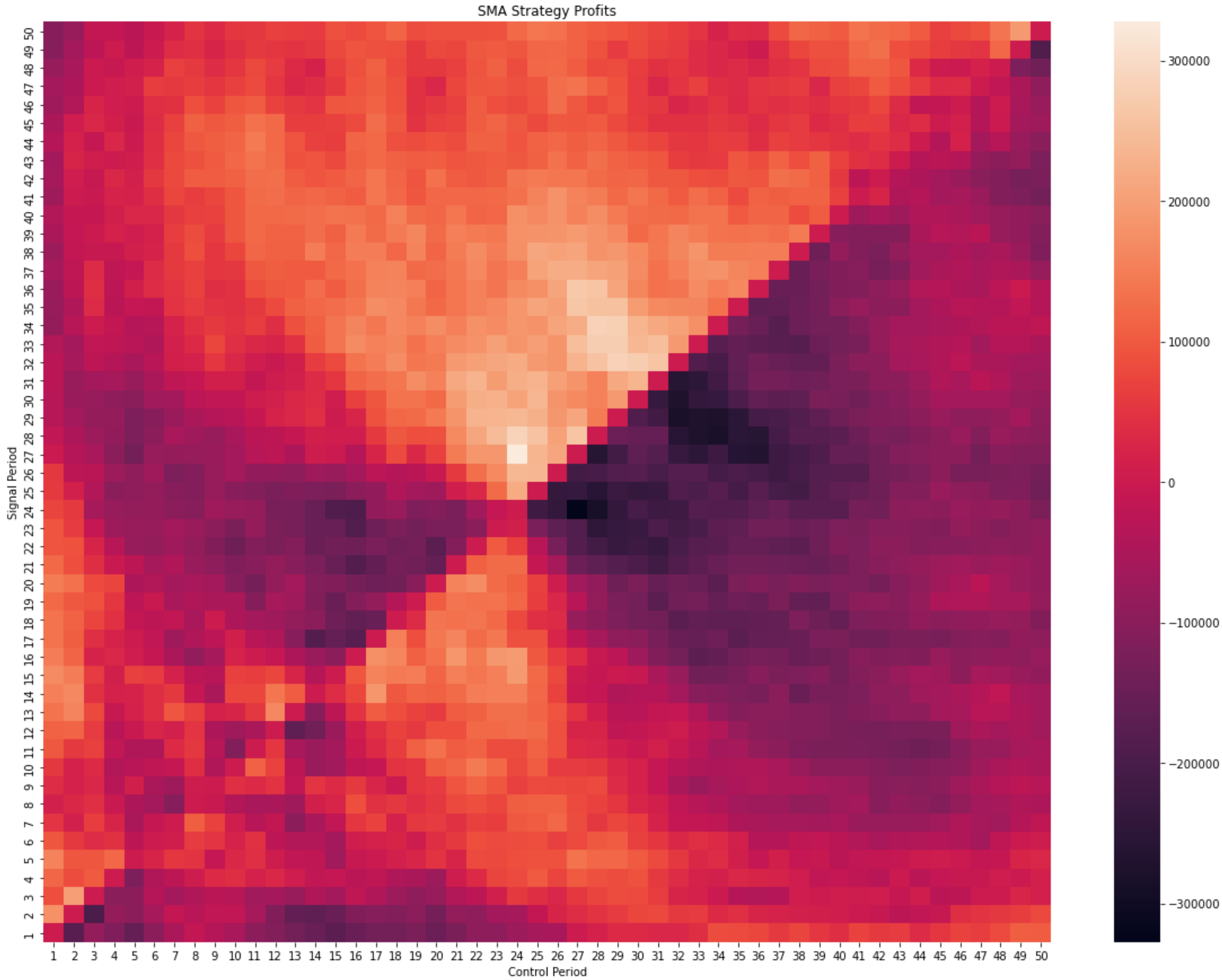

Figure 33: Applying 'heat_map_viz' function to SMA signal and control periods and their profit totals.

The following chart from Figure 34 displays the profit heat map for the WMA over the span of the data set. Its structure is similar in ways to that of the SMA. Again, there are multiple visible clusters of trading rule pairs that would have beat the market, though less decisively than the clusters seen in the SMA map. The insights provided by these heat maps led to the question of, which MA style performs better overall within the period length ranges given that the conventional signal < control rule be followed. One could ask the same of the inverse trading pairs of all MA styles. Lastly, one could apply simple mean difference to discover whether the conventional rule performed better than the inverse rule for each MA style.

```
1 heat_map_viz('WMA Strategy Profits', strat_perfs_w, 'wma_s', 'wma_c')
```

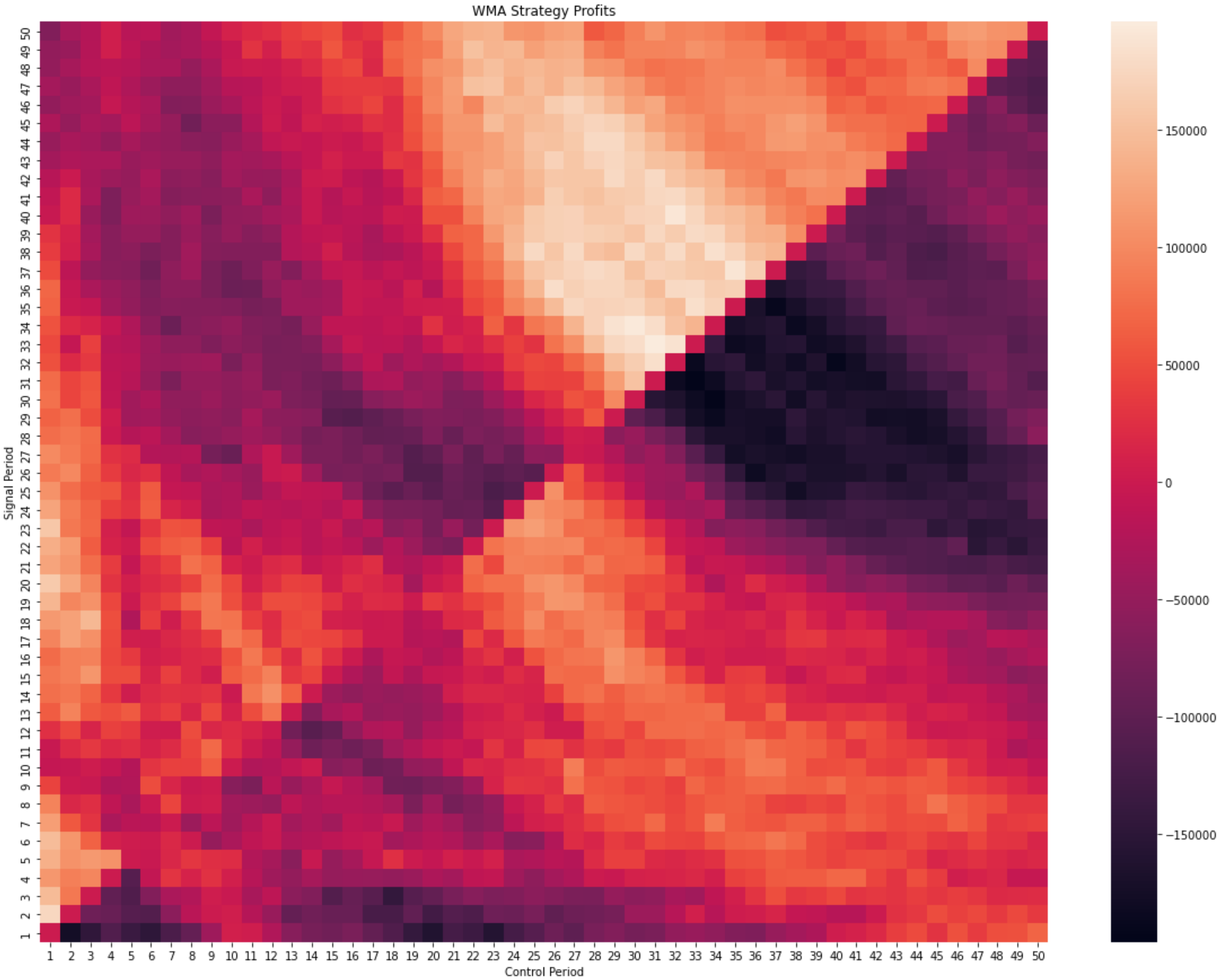

Figure 34: Applying 'heat_map_viz' function to WMA signal and control periods and their profit totals.

The EMA heat map in Figure 35 does exhibit some profitable clusters on the conventional-rule and inverse-rule sides of the diagonal line, but the color patterns also appear flatter and the range of profit smaller. This would suggest that both gains and losses while using the EMA seem to be more toned down compared to the other two styles. An implication of these visualizations is that it is possible to outperform the market with an automated technical trading system. Another would be that one could choose a strategy from one of these heat maps and perhaps expect to beat the market over a long enough length of time. This data spans over seventeen years and was tested on an hourly time frame. The sheer number of observations at disposal for this research may suggest that the strategies that did beat the market could continue to do so over comparable lengths of time. The types and variety of technical situations encountered in this data could be like those of any slice of price data of the same contract into the future.

```
heat_map_viz('EMA Strategy Profits', strat_perfs_e, 'ema_s', 'ema_c')
```

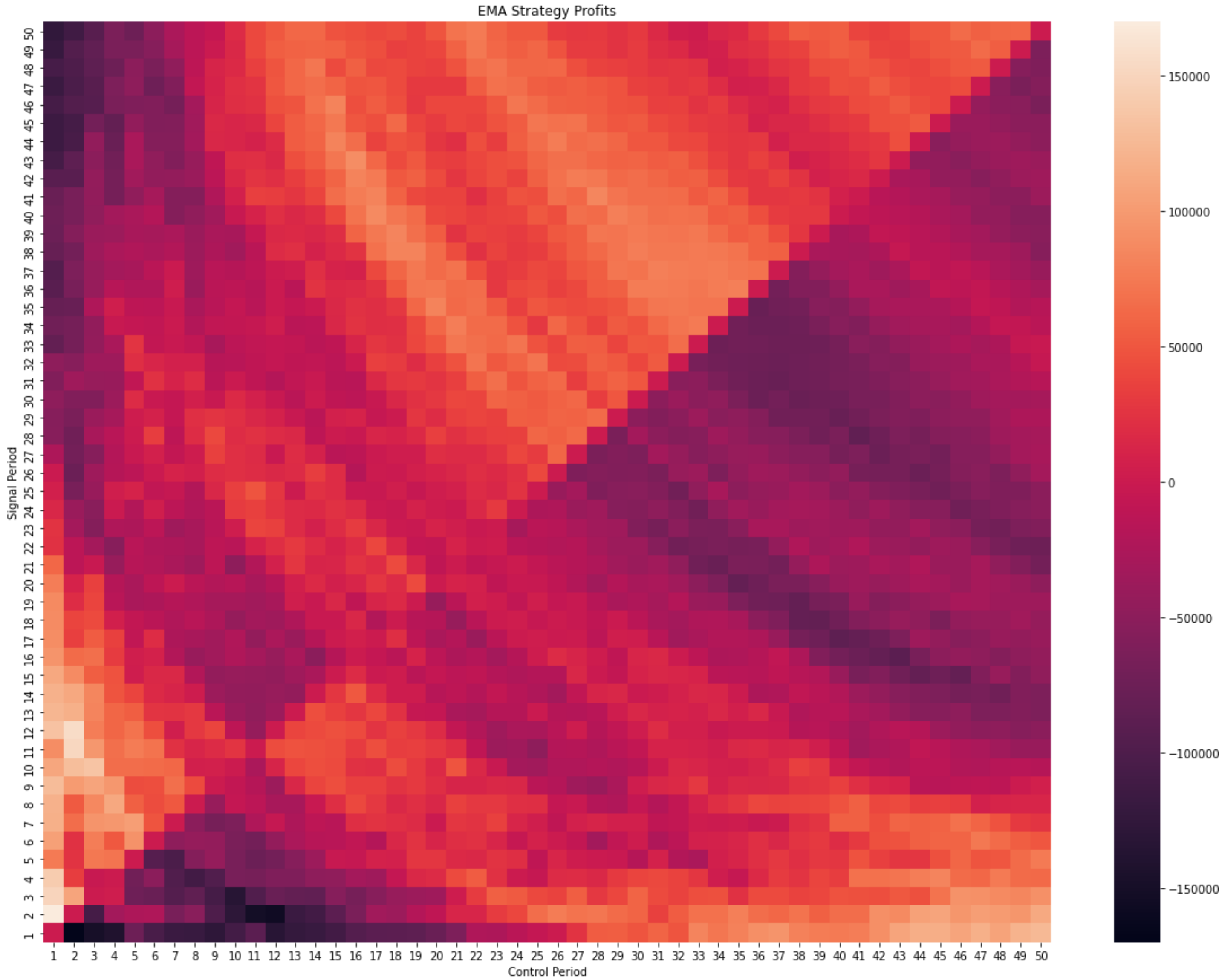

Figure 35: Applying 'heat_map_viz' function to EMA signal and control periods and their profit totals.

ANOVA/Tukey's HSD

The profit data had to change form a bit in order to conduct an appropriate statistical test to determine the significance of mean differences between each MA and whether the rule was conventional or inverse. Recall that the conventional rule mandates that signal < control and the inverse rule requires that control < signal. All this means is that the conventional rule for a signal/control pair will produce the opposite of the signals from the inverse rule, holding the MA style constant. Reformatting the data to prepare it for an ANOVA analysis was relatively simple. A new data frame, called 'rule_split_hsd', was created to include the profit of each MA/rule combination. The data frame only includes 'profit' and 'rule' where 'rule' is a categorical identifier. Table 16 shows the top and bottom of this data frame. There are six possible categorical identifiers for each profit and include 'SMA_conv', 'WMA_conv', 'EMA_conv', 'SMA_inv', 'WMA_inv', and 'EMA_inv'. The bit before the '_' indicates the MA style and the part after indicates whether the profit figure came from the conventional-rule side of the heat map or the inverse-rule side. The signal and control lengths have been decoupled from the profit figures and discarded since they are not relevant to the statistical analysis. Note that there are 7,350 profit figures included in the table while there were 7,500 total MA/rule pairs. This was intentional because all trading rule pairs where signal = control generated no signals and therefore generated no profit or loss. That, and using either ‘<’ or ‘>’ to define whether a pair was conventional or inverse while restructuring the data left them out.

Table 16: Head and tail of ‘rule_split_hsd’.

```
1 rule_split_hsd
```

	profit	rule
0	191900.0	SMA_conv
1	190487.5	SMA_conv
2	188800.0	SMA_conv
3	186237.5	SMA_conv
4	185600.0	SMA_conv
...	...	...
7345	-116537.5	EMA_inv
7346	-116862.5	EMA_inv
7347	-118112.5	EMA_inv
7348	-120062.5	EMA_inv
7349	-125512.5	EMA_inv

7350 rows × 2 columns

The new data frame was exported to Rstudio to conduct the ANOVA and Tukey’s HSD tests. Figure 36 directly below shows this code. An important disclaimer to make here is that the perfectly inverse relationship between the conventional rule and inverse rule of each MA introduces dramatic collinearity to the model. Another possible source of collinearity would be the fact that all three MA styles are calculated using the same price metric. This collinearity could easily work its way through to the profit figures. The Tukey’s HSD is mostly meant to create pair-wise mean differences and confidence intervals for the comparison of overall MA performances and determine whether it might be better to actually do the opposite of what the strategies predict.

````
```{r}
# Fit ANOVA model
model <- aov(rule_split_hsd$profit ~ rule_split_hsd$rule)

# Perform Tukey's HSD test
tukey <- TukeyHSD(model)

# Print the results
summary(model)
print(tukey)
```
````

Figure 36: R code that performs a Tukey’s Honestly Significant Difference test on ‘rule_split_hsd’

The results of the ANOVA in Table 17 demonstrate decisively that the categorical variable ‘rule’ is highly significant ($F(5, 7344) = 394.9$, $p < 2e-16$), indicating strong evidence of significant differences in means among the groups. This only shows that there is a statistically significant difference between at least two of the categories. A Tukey’s HSD was necessary to see the mean differences and their significance between each MA/rule combination.

Table 17: Output of the ANOVA model on 'profit' and 'rule'.

```
                    Df    Sum Sq   Mean Sq F value Pr(>F)    
rule_split_hsd$rule    5 1.088e+13 2.176e+12   394.9 <2e-16 ***
Residuals           7344 4.046e+13 5.509e+09                   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Table 18 shows the mean differences between every possible combination of the 'rule' categories. The questions of this research were specific to the differences over conventional and inverse rule, and among the MA styles. Table 19 is a reduction of the output to only include these mean differences for quicker visibility.

Table 18: Output of the Tukey's HSD test on 'profit' and 'rule'.

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = rule_split_hsd$profit ~ rule_split_hsd$rule)

$`rule_split_hsd$rule`
                      diff        lwr        upr     p adj
EMA_inv-EMA_conv   38873.94  30325.281  47422.597 0.0000000
SMA_conv-EMA_conv -35167.00 -43715.658 -26618.342 0.0000000
SMA_inv-EMA_conv   74040.94  65492.281  82589.597 0.0000000
WMA_conv-EMA_conv -13435.09 -21983.750  -4886.434 0.0001107
WMA_inv-EMA_conv   52309.03  43760.373  60857.689 0.0000000
SMA_conv-EMA_inv  -74040.94 -82589.597 -65492.281 0.0000000
SMA_inv-EMA_inv    35167.00  26618.342  43715.658 0.0000000
WMA_conv-EMA_inv  -52309.03 -60857.689 -43760.373 0.0000000
WMA_inv-EMA_inv    13435.09   4886.434  21983.750 0.0001107
SMA_inv-SMA_conv  109207.94 100659.281 117756.597 0.0000000
WMA_conv-SMA_conv  21731.91  13183.250  30280.566 0.0000000
WMA_inv-SMA_conv   87476.03  78927.373  96024.689 0.0000000
WMA_conv-SMA_inv  -87476.03 -96024.689 -78927.373 0.0000000
WMA_inv-SMA_inv   -21731.91 -30280.566 -13183.250 0.0000000
WMA_inv-WMA_conv   65744.12  57195.464  74292.781 0.0000000
```

The mean differences of all categories imply that the MA style one chooses to trade with and whether they adhere to conventional or inverse rule matters, as does whichever trading pair they choose. The mean of all inverse SMA profit figures was $109,207 greater than the conventional SMA profit figures. Of course, there were SMA strategies on both sides that were profitable, but the mean difference finding suggests that a trader might have better luck by doing the literal opposite of a trend following strategy. The same is true for the other two MA styles, though the mean differences are slightly lower. Table 20 displays the means of each 'rule' category.

Table 19: Reduced output of Tukey's HSD.

```
                      diff        lwr        upr     p adj
SMA_inv-SMA_conv  109207.94 100659.281 117756.597 0.0000000
WMA_inv-WMA_conv   65744.12  57195.464  74292.781 0.0000000
EMA_inv-EMA_conv   38873.94  30325.281  47422.597 0.0000000
SMA_conv-EMA_conv -35167.00 -43715.658 -26618.342 0.0000000
WMA_conv-EMA_conv -13435.09 -21983.750  -4886.434 0.0001107
WMA_conv-SMA_conv  21731.91  13183.250  30280.566 0.0000000
```

Table 20: The means of all signal/control pairs for each MA/rule combination.

MA/Rule	Mean Profit
SMA_conv	-54604
WMA_conv	-32872.1
EMA_conv	-19437
SMA_inv	54604
WMA_inv	32872.1
EMA_inv	19437

Conclusion

This analysis employed programmatic back-testing methods with Python to ES price data to estimate the relative performances of simple, weighted, and exponential moving average crossover strategies. It was done to measure the historical profitability of 7500 rule pairs within a range of period length (1,50) from which to permute. The final visualizations provided a challenge that led to some interesting insights, such as the clustering tendency of winning and losing strategies and the idea that one can profit from a losing strategy by doing the opposite of what it predicts. The buy and hold strategy for this period would have profited $143,325 and the most profitable MA crossover strategy had an SMA signal of period 27, an SMA control of period 23, was an inverse crossover rule, and would have profited $327,437 over the span of the data. This was a discovery of profitability in surprising and obscure trading rule configurations. The SMA had 217/2500 trading rule pairs beat the market, the WMA had 135/2500 pairs beat the market, and the EMA had only 4/2500 perform better than buy and hold. It also appears that cross-over strategies overall would be a much more promising prospect with more deliberate, specific, and researched code instructions to a trading algorithm to help it avoid sideways markets and false signals.

The following is a reiteration of the limitations and recommendations from the abstract of this research report. This system does not make predictions based on sentiment or fundamental indicators, but rather on an assumption that price trends persist more often than they change direction at any timeframe. There is a whole world of causation to price behavior that has not been considered in this work. Running price data that is more granular than a `1min` timeframe through the Python script may result in extremely high demand on memory storage and processing. The data spans over seventeen years and four months and includes 104,439 hours of open, high, low, and close (OHLC) prices. Over this amount of time, it is reasonable to assume that this algorithm would encounter, and account for, a comprehensive variation of technical situations which would be expressed in the performances of these MA strategies. As such, it may also be reasonable to assume that the strategies that did well within this span of time might continue to do so over any span of time that is similar in length. Some of the most profitable strategies were the inverses of failing strategies. A willingness to utilize a counter-intuitive application of MA cross-over based on these findings may result in an algorithm that outperforms the market in the long-run.

There is yet much exploration to be done on moving averages and in what scenarios they can be most effective in profiting from trading. The reasons why some of these strategies were able to outperform the market over this span of time but not others remain a mystery to uncover, as does the effect of technical environment on moving average profitability. A similar analysis could be conducted on tick or volume-based data in an attempt to discover how these moving averages perform differently than when used on time-based price data. This descriptive analysis of ES price data and how moving average strategies performed against it is only the first step in the effort to build a machine learning algorithm that will hopefully one day trade in futures markets. This machine learning algorithm would be trained to change its trading rules according to recent evolving technical price structures as time moves on.

Reference Material

1. Will Kenton. “What Is a Broad-Based Index, and What Are Some Broad Index Funds?.” 2022, What Is a Broad-Based Index, and What Are Some Broad Index Funds? (investopedia.com)
2. James Chen. “What Are Index Futures? Definition, Types, and How to Profit.” 2022, What Are Index Futures? Definition, Types, and How to Profit (investopedia.com)
3. Timothy Smith. “How to Trade Futures: Platforms, Strategies, and Pros and Cons.” 2022, How to Trade Futures: Platforms, Strategies, and Pros and Cons (investopedia.com)
4. Margin - Futures Trading Platform - Tradovate
5. Barclay Palmer. “An Introduction to Trading Types: Technical Trading.” 2022, An Introduction to Trading Types: Technical Trading (investopedia.com)
6. Adam Hayes. “Technical Analysis: What It Is and How to Use It in Investing.” 2022, Technical Analysis: What It Is and How to Use It in Investing (investopedia.com)
7. Shobhit Seth. “Basics of Algorithmic Trading: Concepts and Examples.” 2023, Basics of Algorithmic Trading: Concepts and Examples (investopedia.com)
8. Caroline Banton. “Moving Average, Weighted Moving Average, and Exponential Moving Average.” 2022, Moving Average (MA), Weighted MA, and Exponential MA (investopedia.com)
9. Cory Mitchell. “Understanding an OHLC Chart and How to Interpret It.” 2021, Understanding an OHLC Chart and How to Interpret It (investopedia.com)
10. Adam Hayes. “Simple Moving Average (SMA): What It Is and the Formula.” 2022, Simple Moving Average (SMA): What It Is and the Formula (investopedia.com)
11. Cory Mitchell. “How to Use a Moving Average to Buy Stocks.” 2022, How To Use a Moving Average to Buy Stocks (investopedia.com)
12. Troy Segal. “Fundamental Analysis: Principles, Types, and How to Use It.” 2023, Fundamental Analysis: Principles, Types, and How to Use It (investopedia.com)
13. CFI Team. “Market Efficiency.” 2023, Market Efficiency - Overview, Efficient Markets, Implications (corporatefinanceinstitute.com)
14. Martin Sewell. “History of the Efficient Market Hypothesis.” 2011, History of the Efficient Market Hypothesis (ucl.ac.uk)
15. J. B. Maverick. “The Weak, Strong, and Semi-Strong Efficient Market Hypotheses.” 2022, Forms of Market Efficiency: Weak, Strong, and Semi-Strong (investopedia.com)
16. Eero Pätäri and Mika Vilska. “Performance of moving average trading strategies over varying stock market conditions: the Finnish evidence.” 2014, Full article: Performance of moving average trading strategies over varying stock market conditions: the Finnish evidence (tandfonline.com)
17. Massoud Metghalchi, Juri Marcucci, and Yung-Ho Chang. “Are moving average trading rules profitable? Evidence from the European stock markets.” 2011, Full article: Are moving average trading rules profitable? Evidence from the European stock markets (tandfonline.com)
18. Ben R. Marshall, Nhut H. Nguyen, and Nuttawat Visaltanachoti. “Time series momentum and moving average trading rules.” 2016, Full article: Time series momentum and moving average trading rules (tandfonline.com)
19. James Chen. “Backtesting: Definition, How It Works, and Downsides.” 2021, Backtesting: Definition, How It Works, and Downsides (investopedia.com)
20. Zaw Thiha Tun. “Pros and Cons of a Passive Buy and Hold Strategy.” 2022, Pros and Cons of a Passive Buy and Hold Strategy (investopedia.com)