Research Paper:

The research question that my capstone project will answer is “Can Linear Regression Analysis of Bitcoin Time Series Price Data on Fractal Time Frames be used for Trade Optimization?”. The project will consist of running linear regressions on fractal time frames of Bitcoin’s publicly available price data. The intent is to produce a trading strategy that consists of comparing the linear regression slopes of these fractal time frames as an indicator to enter a trade. The comparison of the linear regression slopes will help determine the overall trend of the market, and if the fractal time frames all indicate the market is bullish, then a trade long (purchase of Bitcoin) will be executed. The trade is initiated with the intent of closing the trade (selling Bitcoin) at a later date for a higher market price, resulting in a profit, which is calculated at the future price minus the current price.

There are a few concepts to review and terminologies to define required for further discussion on the project. The first concept to define is that of price action and time frames, and why these are relevant to traders. Price action of an asset (in our case, Bitcoin) is defined as the delta in price from one moment in time to another. This delta can be negative if the price fell during this time, or positive if the price rose during this time. The starting and ending time are chosen by the trader, depending on their trading goals and strategies. Technically, price action is constantly changing on infinitesimally small time frames, and certain traders, specifically high frequency & quant traders, do trade on the micro and nano second time frame for example. For most traders, this is both impractical to execute on and difficult to conceptualize, thus price action is grouped into time frames. This helps traders quantify price movement by ignoring the noise of price action and focusing on the signal of price action, in the time frame that is relevant to the trader. This is relevant to a trader as it is their goal to determine both the direction and the strength of the trend of price action, and it is required to constrain this analysis to a time frame. To a data scientist, what I just elaborated on is easily conveyed as Time Series analysis, however it is relevant to elaborate from the ground up as the overwhelming majority of participants in the market do not approach trading from a mathematical or statistical perspective.

The second concept to define is that of fractal time frames. Rather than define this from a strictly mathematical perspective, we will define it from the perspective of a day-trader. Thus, the colloquial definition for fractal in our context is “Larger things are made up of a collection of identical smaller things” (Severson, 2020, p. 95). For example, let’s compare three time frames of price action: the 1-day chart, the 5-day chart, and the 25-day chart. This multi time frame analysis would be defined as fractal due to the fact that moving up and down the time frames requires the same scalar multiple. The scalar doesn’t have to be a specific integer, although 5x is the most common value chosen to compare trading time frames. But the concept of each time frame being related to the next time frame (in chronological order) by the same scalar multiple is what defines them as having a fractal relationship. To reiterate, this concept is not about strict mathematical principal adherence, but about a practical methodology to compare three or more time frames of summarized price action in a meaningful way. For example, comparing 4 charts of the 1-day, 5-day, 25-day, and 125-day, we intuitively understand that quarters consist of months, which consist of weeks, which consist of days. And though the math isn't exact, it's close enough to identify trends within trends, which is the ultimate goal of fractal time frame analysis. This concept is critical, and so I will elaborate further. Trends exist within trends, just as days exist within weeks. It is the job of the trader to not only determine the trend of the time frame they choose to analyze, but the larger and smaller trends as well. For example, Bitcoin may have risen in price by 5% this week. This information, in isolation, is of little use to the trader utilizing a systematic approach to entering and exiting the market. If the trader knew that the 25-day chart showed price rising by 15%, then the trader could determine entering in long is in confirmation with the larger trend and enter the market. But if the trader knew that the 25-day chart showed price declining by 15%, then the trader could determine entering in long is in conflict with the larger trend and that price is likely to continue declining after this mini rebound, thus the trader could stay out of the market. Without going into an entire breakdown of trading methodology, the point of fractal time frame analysis is to not just understand the trend of the current time frame being analyzed, but to understand the context to which this trend exists.

We will now review the time-series analysis performed on the fractal time frames. Bitcoin's daily price data for the last year will be retrieved and imported into a Google Colab Notebook, and will be processed into a series object, with the date as the index and the closing price as the value. This will allow us to perform time series analysis on the data in fractal time frames of 1-day, 5-days, and 25-days. Every Tuesday at 12:00 am, linear regression will be run on the three historical fractal time frames, and the slope of the regression lines will be calculated and compared. A positive trade signal will be defined as two or more neighboring fractal time frames having a linear regression with a positive slope, indicating a positive trend in the market. A negative trade signal will be defined as two or more neighboring fractal time frames having a linear regression with a negative slope, indicating either a negative or neutral trend in the market. Of minor note, the cryptocurrency market runs 24 hours a day, 7 days a week. Thus, trades can be opened or closed at any time on any day. Additionally, daily price data will discard the open, low, and high price, and focus only on the closing price. This is to consolidate daily price action from a range to a single point.

We will now review the experiment methodology. A trade will be defined as a long entry (buying Bitcoin) of one BTC into the market at Tuesday 12:00 am and closing the trade (selling Bitcoin) at Friday 12:00 am. The trade holding time, purchase time, and sell time are standardized across both groups. After the trade is completed, the start price will be subtracted from the end price, resulting in the price delta. Trading fees and other miscellaneous expenses will be ignored for this experiment. Two accounts of value 100,000.00 will be created, one for the control group and one for the experimental group. The control group will represent the trader following the null hypothesis, and ignore the linear regression indicators. They will place trades every Tuesday morning, resulting in 52 total trades. The experimental group will represent the trader following the alternative hypothesis. If the linear regression indicators show a trade should be placed, only then will they place a trade Tuesday morning, resulting in 52 - weeks skipped total trades. For each trade performed, the price delta will be either added to or subtracted from the account balance. As the experiment proceeds, the equity curve and the profit/loss statements can be compared for each group, and at the conclusion of the experiment the final account balance values will be compared to see which group performed better.

With the mechanics of the experiment covered, we will now state our hypothesis. The Null Hypothesis is that running the linear regression analysis on Bitcoin's price data over fractal time frames will not serve as an indicator to enter a trade long, and will have no influence on profits and losses due to trading activity. Trades that are part of the control group will always enter the market at 12:00 am every Tuesday morning, and will exit the market at 12:00 am every Friday. There will be a total of 52 trades in the control group, with one trade occurring a week. Since the control group is entering the market every week, regardless of our linear regression indicators, it will serve as evidence to either support or reject the Null Hypothesis. The Alternate Hypothesis is that running the linear regression analysis on Bitcoin's price data over fractal time frames will serve as a reliable indicator to enter a trade long, and that this success as an indicator to enter the market long will result in increased profits and minimized losses due to trading activity. Trades that are part of the experimental group will enter the market at 12:00 am every Tuesday morning, and will exit the market at 12:00 am every Friday, only if the linear regression indicators signal that a trade should be placed. If the indicators do not indicate a trade should be placed, then the week will be skipped and no trading activity will occur in the experimental group. There will be a total of 52 trades minus the weeks skipped in the experimental group, with one trade occurring a week, if a trade is placed at all. Since the experiential group is entering the market only if the linear regression indicators confirm a trade should be placed, it will serve as evidence to either support or reject the Null Hypothesis.

We will now review the data collection process. In brief review of the nature of the data, the daily price data of Bitcoin is public domain, and is available via multiple cryptocurrency brokers and financial websites. We will be utilizing Yahoo Finance as our source for Bitcoin daily price data. Access to this information, from both an availability and a legal perspective, has not presented an issue for the project use case. Additionally, since the data is of a financial nature, this safeguards against missing and/or corrupted data, as cryptocurrency brokers need to provide both timely and correct data to retail traders per US Securities and Exchange Commission regulations. We will still check for null data in our time series as a formality, however for this industry and use-case, null data is exceptionally rare. Finally, the data being analyzed is of type numeric and datetime, and requires minor formatting for our purposes. The price data is already typed as Float, which is the correct type to represent the dollars and cents of price. And the datetime data needs to be formatted into a datetime object via format '%Y-%m-%d'. Other than this minor formatting, data does not need to be transformed or encoded for our project purposes.

The data gathering methodology used is to directly pull the time series data of Bitcoin's price via API call from inside the project. One advantage of this is the convenience of accessing and processing the information. By using an API call from inside the project file, we can utilize the internet to request the financial information from Yahoo Financial and directly load it into our project file. This convenience also ensures our data is up to date, as the data is pulled at the time of project execution. One disadvantage of this is the reliance on internet connectivity to obtain the financial data. Since we do not have the financial data in an offline storage format like a CSV file, we cannot execute our project without a stable internet connection to source the data.

The data extraction and preparation process begins with the 'yfinance' package, which allows us to initiate an API call to Yahoo Financial to retrieve the Bitcoin price data. The first parameter specifies the asset data to be retrieved, which is 'BTC-USD', or Bitcoin to the United States Dollar. The start parameter is the beginning date of price action requested, and the end parameter is the end date of price action requested. We will request data starting a year back up to the current day. The 'yfinance' package returns a dataframe with the index preformatted as a DateTimeIndex, with each index representing one day. This dataframe comes with additional information other than the closing price, but for our purposes we will drop these extra features. We will be left with the DateTimeIndex and the closing price, completing the formatting of our time series data. The main reason we chose to format the data into a time series is the accessibility of the price values by the DateTimeIndex. Since the date is the independent variable, and price is the dependent variable, we construct our data array as a time series to reflect this relationship. Said another way, we as traders make sense of the change of price in relation to the progression of time. One advantage of formatting and preparing the data this way is the ease of accessing the price value by using the datetime as the index. The datetime fulfills the requirement of being an index, as all datetimes are unique and preserve chronological ordering. One disadvantage of creating a series object from the original dataframe object containing the data from the original API call is the loss of tangential data that was dropped to simplify our model. The original API call includes price data for the open, close, high, and low, as well as volume of trading activity. While this additional information is out of scope for our analysis, this data could be useful for additional research and future iterations of our model. All steps in the data extraction and preparation process can be referenced with the attached Google Colab Notebook.

We will now review the data analysis methodology and techniques. After construction of the time series data, we add a column indicating the row number in integer format. This is an unorthodox addition, so let me elaborate. Date time indexes, while needed for this particular analysis, are cumbersome to iterate through compared to standard indexing. There were times where it was required to iterate both forwards and backwards through the price values, and this column was added as a convenience to perform such iterations. For all intents and purposes, this additional column does not change the nature of the time series data, and will be ignored when it is not relevant.

The first item to review was the creation of the lists of lists utilized in this program. The first group identifies all the 'Mondays' and 'Thursdays' in our time series, and a list that contains only these days is created. This is because we initiate our trade open on Tuesday at 12:00 a.m. using Monday's closing price, as the closing price for Monday is the open price for Tuesday. The same logic applies for the Thursday list, as all trades are closed on Friday at 12:00 am. The next group of lists are the lag prices. We need the prices for 1, 5, and 25 days behind each Monday to calculate the slope of price for that fractal timeframe. Of note, the list 'Lag_Price_1_Day' has 2 values, while 'Lag_Price_5_Day' and 'Lag_Price_25_Day' have 5 and 25 days. This is because you can't calculate a slope from a single point. The next group of lists are the slopes for each fractal timeframe, which will hold the slope value once it's calculated from the lag price. Of note, it was necessary to create lists representing the x-axis for each time frame that would allow slope calculation. This is a convenience substitute for calculating the number of days in each slope calculation from date time indexes. Since we already know that a 5 day fractal time frame has 5 days, this was hard coded with the variable 'X_Lag_5" with the list value of [0, 1, 2, 3, 4].

To calculate the slopes for each fractal timeframe, we imported the linregress package from scipy.stats. This allows us to calculate the slope for each timeframe, which is then stored in the list belonging to that timeframe. Finally, each time frame of slopes was brought together into one list called 'Slopes_All_Fractal_Timeframes'. This list of lists allows us to access the 1-day, 5-day, and 25-day slope for each day we initiate a trade. The next list calculated is called 'Trade_Profit_Loss', and it contains the price delta for each trading week. The final list created is called 'Trade_Indicator'. This list is the end product of the indicator calculations, and for every trading week it indicates if the trader should buy that week or wait until the next week. This is calculated by comparing the slopes of each fractal timeframe, with the condition that two time frames need to show a positive slope for trade entry. This means the 5-day slope always has to be positive, and if either the 1-day or 25-day slopes are also positive, then the indicator tells us to buy. Finally, both the control group and the experiment group are run through their eligible trades, and the final account balance for each is printed. All steps in the data calculation and analysis process can be referenced with the attached Google Colab Notebook.

The main technique utilized in this analysis is linear regression, specifically the calculation of the slope of price data with the intent of using the direction and magnitude of the slope to forecast future price movement. The justification for this is that the slope of the line represents the signal of price data, while eliminating the noise that is not relevant to the trader. For this analysis, the trader is defined as a weekly trader, which would be analogous to a swing trader in real world terminology. What this means is that this type of trader is not looking to capture every penny of price movement like a high frequency trader, but capitalize on 80% of the overall trend. Most retail traders fall into this category, as most non-professionals are not going to manage their portfolio minute-by-minute. With this context in mind, the day-to-day price movements are not especially relevant, but the general direction of and strength of the trend is important. The slope line calculated by linear regression is the representation of this signal, thus why it was utilized in this analysis. As mentioned earlier, one advantage of this technique is that it captures the signal of the data, without getting lost in the minutia. This type of trader would see a per-trade profit of $495 and $505 as equivalent for reasons detailed previously. One disadvantage of this methodology & technique is that sudden market changes are not factored in, and could have deleterious conditions to a trader's equity curve. "Black Swan Events", which are market anomalies, will not be accounted for until the normal analysis cycle. This leaves the trader open to potential downside loss if there are not standard risk management practices implemented, like a stop loss on every position.

The results of the data analysis do not provide evidence to reject the null hypothesis. The control account made a profit of $9,870.04, while the experiment account made a profit of $4,140.86. While there are statistically significant differences between the two accounts, and we could perform significance tests to prove this, the results will be analyzed from the perspective of a trader and not a statistician. What is meant by this is that from the perspective of a money manager with a $100k account, when factoring in the risk of capital loss resulting from trading, as well as the fixed costs of trading that were excluded from this study (such as trading commissions, slippage, etc.), the real question is if either of these accounts constitute success, which they do not. A 9.8% or 4.1% return on account equity may be considered a success for more passive forms of investment, but not for an active trader. This is not highlighted to convolute the fact that the results of the experiment fail to provide evidence to reject the null hypothesis. The data clearly shows the linear regression methodology was clearly inferior to the control group. Rather, this elaboration is meant to highlight the broader context in which the data and its results exist. One limitation of this analysis is the strict time frames used for the trade entry and exit of three days. For review, all trades are entered on Tuesday 12:00 am, and exited on Friday 12:00 am. This strict entry and exit is required to standardize the trades across the control and experimental group, but is not how a trader would actually manage their position. As position management is outside the scope of this analysis, this limitation is accepted as part of the experimental design. There are two recommended courses of action, one based on the results and one based on standard practices for automated trading algorithms. Per the results, even though we did see validity in the linear regression comparison over fractal time frames, the next step would be to compare different fractal time frame linear regressions to determine which fractal time frame comparison achieves the optimal results, which is defined as the highest profit. While this analysis served as a solid A/B hypothesis test of the concept itself, further analysis will consist of multi group hypothesis testing of different iterations of the concept. This further analysis is not technically required in the most pedantic sense, however from a trading perspective neither the null or alternate hypothesis were constructed as actual trading methodologies, but rather as an effectiveness test of an indicator. The second recommended course of action is to incorporate best practice features into the automated trading algorithm that are tangential to the trading methodology itself. Features like risk control, automated stop loss calculation, and initiation of trades via API call are some of the features to be added in future iterations of the program. There are also two recommended directions for further study. The first direction for future study is the use of linear regression of fractal time series analysis as a trade exit signal. This analysis has demonstrated the effectiveness of the indicator to enter trades, but further research could be done on the viability of using the negative of the indicator as a sign to exit the trade. Essentially, the analysis would be done in reverse. If the indicator proved to be as effective when done in reverse, this would add value to the overall trading algorithm by both signaling the entrance and the exit of a trade, thus encompassing the entirety of the trade instead of just the initiation. The second direction for future study is the utilization of different fractal time frame scalar multipliers. For this experiment, a 5x scalar multiplier in between time frames was borrowed from the US Equities market. Since this market only trades Monday - Friday, a week is designated as 5 days, hence the 5x scalar multiplier. Further experimentation could be done with a range of scalar multipliers, seeing which scalar multiplier resulted in the highest gross profit and thus being the most optimized. Since the cryptocurrency market is a global market and trades 24/7/365, there are endless opportunities to choose a scalar multiplier, and multiple iterations could be tested while still conforming to a trader’s notion of a day, a week, a month, etc.

References:

Severson, D. (2020). *Fractal Energy Trading: Four Simple Rules to Profit In Any Market & Any Timeframe*. Independently Published.