INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Market making model analysis in High Frequency Trading for the north American stock market

Daniela Fernanda Martinez Vargas

Master in Finance

Supervisor:

Ph.D., José Carlos Gonçalves Dias, Full Professor,

ISCTE Business School, Department of Finance.

October 2023

BUSINESS SCHOOL

Department of Finance

Market making model analysis in High Frequency Trading for the north American stock market

Daniela Fernanda Martinez Vargas

Master in Finance

Supervisor:

Ph.D., José Carlos Gonçalves Dias, Full Professor,

ISCTE Business School, Department of Finance.

October 2023

Sumário

Este trabalho pretende testar um modelo de market making num High Frequency Trading para diferentes acções, utilizando um código ou algoritmo que ajude-nos a compreender o comportamento do modelo com diferentes variáveis como a latência, o número de simulações durante o dia, o coeficiente de aversão ao risco e a margem.

Isto é conseguido utilizando preços de compra e venda fictícios, para criar as diferentes ordens que tornarão possível esta simulação.

Palavras Chave: algorithm trading, python, trading de alta frequência, modelo de criação de mercado, mercados electrónicos, preços de compra e venda, EMA.

Classificação JEL: G12

Abstract

This work wants to test a market making model on a High Frequency Trading for different stocks, using a code or algorithm that help us to understand the behavior of the model with different variables as latency, number of simulations during the day, risk aversion coefficient, and margin.

This is accomplished by using fictional bid and ask prices, to create the different orders that will make possible this simulation.

Keywords: algorithm trading, python, High Frequency Trading, market-making model, electronic markets**,** bid and ask prices, EMA.

JEL Classification: G12

Contents

List of Graphs

List of Tables

Glossary of Acronyms

AAPL	Apple Inc.
MSFT	Microsoft Corporation
NVDA	NVIDIA Corporation
LSE	London Stock Exchange
NYSE	New York Stock Exchange
HFT	High Frequency Trading
NASD	National Association of Securities Dealers
AQ	Automated Quotation
AT	Algorithmic Trading
RTD	Real Time Data
SMA	Simple Moving Average
EMA	Exponential Moving Average
MACD	Moving Average Convergence/Divergence Rules
RSI	Relative Strength Index
OBV	On Balance Volume

1. Introduction

The nature of trading has changed over the years and the introduction of technology in daily life has enabled new forms of trading, such as High Frequency Trading and algorithmic trading, which aim to enforce strategies to make profits by placing positions in extremely short periods.

These types of strategies have led to the emergence of market makers, which provide liquidity to the market by placing regular bids and ask orders. These market makers can be developed using a variety of models such as statistical models, market impact, order book, inventory models, and algorithmic trading models that could be used individually or mixed, depending on the investor risk tolerance and preferences. However, in this dissertation, we will apply the model proposed by Avellaneda and Stoikov (Avellaneda & Stoikov, 2008), who implemented an optimization model for the market making pricing model through algorithmic trading to be studied in this project.

This model is applied because does not limit the stock size and allows the market maker to trade regardless of his position; however, this increases the risk, so it is better to incorporate a stock control that gives more flexibility to the model to ensure profit.

Besides the model, we will apply not only algorithmic trading, but also technical analysis of the data to demonstrate how accurate could be this methodology to help the investor to predict the market trends using reference period 400 for the 3s latency and 20 observations in the 1m latency, meaning 20 minutes of time-lapse.

The methodology will be applied to three financial instruments, from the same industry in the North American market, especially in the U.S. where we can find intraday data, to Apple Inc. (AAPL), Microsoft Corporation (MSFT) and NVIDIA Corporation (NVDA)

The data will be used to perform the statistical analysis that will help us to summarize and organize the information, allowing us to identify patterns and trends in the data, moreover a technical analysis methodology will be done over the data to understand how the market conditions are and if the technical analysis is accurate to predict the trends of the instruments.

Once the theoretical part is completed, then we will use the 1-minute frequency market prices to test the algorithm developed by creating a factorial to create a fictional bid and ask price through a technical analysis that allows us to create a strategy, and then replicate it in 3s timestamp evaluating if the model and methodology used accomplish the expectative.

2. Objectives

2.1 General Objective

This project aims to implement and analyze High Frequency Trading on a market making model. The goal is to develop an algorithm that allows us to understand what is the impact of using this model on trading with high latency.

This will be applied to some American stocks trying to see the performance of the model for ensuring profitability using a no restricted model that gives the liberty to adapt the inventory to get different kinds of results by adapting it to market price data and bid and ask data. The idea of developing this master's project is to mark a starting point for the creation of an algorithm that allows one to understand in an easier way a subject that is not so studied as High Frequency Trading.

2.2 Specific Objectives

- Create an algorithm under the market making model to be applied to more accessible data such as historical intraday data with only market price information available.
- Test the model over the market price data sample.
- Analyze the output generated by the two different tests performed to determine if the different kinds of data could be truthful and with this be able to conclude if the algorithm and the model is efficient.

3. History of Financial and Electronic Markets

To understand and grasp how the market making model works in the High Frequency Trading model, we must first explain the history of electronic markets and how they work today, what market making is, and its optimal method to then move on to analyze the model and its implications for High Frequency Trading.

Over the years, financial markets have evolved and adapted to current needs. With the development of new technologies, markets have made great efforts to obtain algorithms and methods that allow them to buy and sell as fast as possible. To better understand this need and evolution, it is important to look to the past and see how the market was created and has evolved.

For historians, the development of systems goes back to ancient times, and both Mesopotamia and Southeast Asia played a very important role in the history of barter as a system, as they were among the first societies of mankind and developed as trading societies that allowed the exchange of goods and services that we know today as commodities, with the exchange of natural resources such as metal, olive oil, horses and other types of animals, spices or fabrics; so the market itself could determine their value based on the quantity demanded at a given time.

After nearly a millennium and a half, Afro-Eurasian exchange markets retained their basic structures, but trade, as we know it today, began its creation process in the 16th century, and was defined in the 19th century. Then it is in the 17th century that the exchange market began to take shape. Although the exact place of its emergence is still disputed, there is information about several starting points such as Amsterdam in 1602, when the United East India Company made the first public offering of shares in the exchange market, or in Japan in 1603, which claims the origin, where the first exchange took place in the Yodoya rice market. All this led to the development of local markets in different parts of the world, from the Tea houses in London, where expeditions to the New World were organized, later also called the London Stock Exchange (LSE), to the New York Stock Exchange (NYSE), which began with the financial collectors and traders under the Buttonwood tree.

These examples can be understood as the beginning of what we know as exchange markets. However, we also need to look at the importance and impact of technology on these markets, because it is this electronification of trading that will give rise to High Frequency Trading (HFT), which began in the 19th century with the invention of the telegraph, that allowed messages to be transmitted in less time but was not as agile or economical as Morse code later in 1867 or telephone communications in the early 1900s.

In 1925, Translux Corporation introduced the electronic board, and as early as 1959, a predecessor of Quotron Systems introduced the electronic quote terminal. This marks an important step in the evolution of financial markets, as Grody and Levecq note in their article "Past, Present, and Future: The Evolution and Development of electronic financial markets" (Grody & Levecq, 1993); the development and evolution of electronic financial markets, the evolution along a thread that begins with the placing of orders first by telegraph, then by telephone, and now digitally. The speed of order execution has been revolutionized, from the days when it was done by telegraph, to minutes or seconds, to the advent of electronics at a speed of more than a million gigabytes per second or nanosecond (Smith, 2009).

The electronification of securities began more than 50 years ago in the United States, when the National Association of Securities Dealers (NASD) began requesting computerized support through Automated Quotation (AQ), creating what we know today as NASDAQ; unlike Europe, where computerized systems were not fully introduced until the 1990s.

In this way, the human trading process is gradually being replaced by electronic systems that allow not only the acquisition of financial instruments around the world but also greater agility through automatic quotes and stop-loss orders, giving rise to models such as algorithmic trading (AT) and HFT.

Algorithmic trading has changed the traditional way of communication between investors and their brokers by generating individual trades without human intervention. However, the more detailed study and the incorporation of market models in this type of algorithm have made it possible to better control the buying processes. On the other hand, we must note that there is a difference between HFT and AT, although they are quite similar because while AT deals mainly with the execution of orders, HFT involves the use of trading strategies.

One of the most important changes in trading instruments was the way it was done. Before electronic markets, trading was done physically and required brokers to stand out among the rest, with the skills most in demand like running faster or speaking louder than others, so agencies looked for people who stood out in the market to make their trading intentions known and get the best deals. But with high-speed trading, the skill most in demand became the ability to receive and deliver data in the shortest time possible.

As explained above, AT and HFT have both similarities and differences, which are explained in detail (Gomber et al., 2012) in their paper High Frequency Trading:

<table>
<tr><th colspan="2">Common Characteristics of AT & HFT</th></tr>
<tr><td colspan="2">- Pre-designed trading decisions
- Observing market data in real-time
- Automated order management
- Use of direct market access
- Used by professional traders.
- Automated order submission
- Without human intervention</td></tr>
<tr><th>Specific Characteristics of AT</th><th>Specific Characteristics of HFT</th></tr>
<tr><td>• Agent trading
• The goal is to achieve a particular benchmark.
• Working an order through time and across markets
• Minimize market impact (for large orders)
• Holding periods days/weeks/months</td><td>• The extremely high number of orders
• Proprietary trading
• No significant position at the end of the day (flat position)
• Extracting exceptionally low margins per trade
• Use of co-location/proximity services and individual data feeds
• Rapid order cancellation
• Profit from buying and selling (as an intermediary)
• Short holding periods
• Low latency requirement
• Focus on high-liquid instruments</td></tr>
</table>

Table 1 Common characteristics of algorithm trading and High Frequency Trading.

4. High Frequency Trading

As it was mentioned before the introduction and generalization of electronic markets through new technologies and the automation of the procedures and financial transactions have improved the decision-making process, this kind of trading allows to perform multiple orders during the day, through the development of an algorithm; however, it is important to mention that not all AT occurs in HFT but HFT always require AT, trying to increase the profits from and investment or trading strategy.

HFT aims to make a profit by taking advantage of small discrepancies between different markets or by reacting to market news as faster as possible, being one step in front of the other traders, normally HFT uses short-term strategies, holding positions for a few microseconds.

Some authors like Aldridge (2013) ensure that commonly high frequency traders tempt to use different strategies based on their profile, these could be grouped into 4 branches:

1. Arbitrage: This type of strategy seeks to move away from deviation from long-term equilibrium prices, and usually includes multiple asset classes as well as multiple markets. Due to this diversification, many traders seek to trade the same asset in different markets to generate price arbitrage.
2. Directional event-based trading identifies short-term or momentum trends, this kind of strategy is based on predictable short-term price movements.
3. Automated market making: is one of the most traditional trading strategies, encompassing automated market making.
4. Liquidity detection, where speculation is the center of the strategy.

These kinds of models normally have a life cycle that should be updated quite frequently, this cycle process tracks the following five phases:

- Planning: the purpose of this phase is to plan the goals of the projects in terms of strategies, operating models, and requirements.
- Analysis: once the project is planned, then is necessary to analyze the scope of the project.
- Design: In the design of the model, is where the specifications of the model and rules are incorporated.
- Implementation: now this phase includes the development of the code with all the analysis and design performed before.

- Maintenance: now that the program is working and implemented, is necessary to constantly maintain the program to be aware of bugs or troubleshooting.

4.1 Data Construction

High frequency data or Tick data, are records of live market activity, meaning every order that a customer, a dealer, or another market participant places, ones this market orders are incorporate into the tick data, then it's possible to review it at different kind of levels, going from nanoseconds to daily data.

For this project, the data used should request the movement over the financial instruments in faster intervals, for this some codes will give the data generated for the price over some stocks with short intervals of time. Two kinds of data were found, one of them on the trading platform ThinkorSwim (TOS), through the Real-Time Data (RTD) interface where Excel can pull price data from the platform at about a 3-second update rate but with the limitation of just providing the closing market price.

And the other one with a timestamp of 1-minute delay founded on Quandl, a data science application to provide available data, both free and premium, all of them from the web and delivered via Nasdaq Data Link's industry-leading data delivery platform. This data is completed regarding volumes and prices; however, this data is not historically available, meaning that the only sample is the last month before the searching period on the tool.

For the data found with 3-second latency, the information will be grouped by week, starting every Sunday at 17h EST and ending every Friday at 18h EST, and for the 1-minute data the selected period will not be for weeks but will be June of 2023. This information is over the follow three stocks on the same industry as follows.

CODE	NAME
AAPL	Apple Inc.
MSFT	Microsoft Corporation
NVDA	NVIDIA Corporation

Table 2 Data list of financial instruments

One important thing to have in mind is that all transactions occurred between 9.30H and 16H from Monday to Friday. This means that for the selected data we have the following observations:

WEEK	OBSERVATIONS
Data week ending August 07th of 2020	38956
Data week ending November 27th of 2020	38950
Data week ending August 20th of 2021	38994
Data week ending December 03rd of 2021	38954
Data week ending September 16th of 2022	38976
Data week ending May 13th of 2022	38938
Starting on June 09th to June 23rd of 2023	7289

Table 3 Data sample observations - sample size

The variables recorded before 2023 include for each observation a time stamp and a traded price, and for 2023 the open price, high, low, close, and volume for each stock.

Due to the differences in the data, two analyses will be performed. One over the 3 seconds latency, where the main calculation will be the standard deviation calculated on the three-second latency data to determine how much the price could be dispersed around the average, and the stock data return that allows to understand how the behavior of the selected instruments is.

And a second analysis over the most recent but with less latency data found, where besides the prior statistical analysis it will also study how the exponential moving average behaves for the different stocks and how the error will be impacted by changing the number of observations or N.

4.2 Preliminary Data Analysis

In this section, we will review the behavior of the selected financial instruments for the different samples. To perform this, due to the large amount of data, it is necessary to calculate the return of each instrument using a logarithmic return.

Normally this kind of return is accurate when the data follow different characteristics as being a short-term investment horizon or intraday data Therefore, the following formula could be used:

$$r_{t_1} = ln\left(\frac{P_{t1}}{P_{t0}}\right) \qquad (1)$$

Once the return is calculated, it is possible to realize the statistical analysis over the return, since the return provides a better representation of the stock´s performance than prices.

This statistical analysis will be performed on one sample week per year for us to have a brief panorama of how the performance of every stock was by year and if this behavior is constant or at least quite similar.

- **2020**

For 2020 the week ending on August 7th was chosen. In this week the three financial instruments analyzed had a small standard error, which means that the amount of error that would be expected in the estimated value is accurate, thus giving low uncertainty for these stocks.

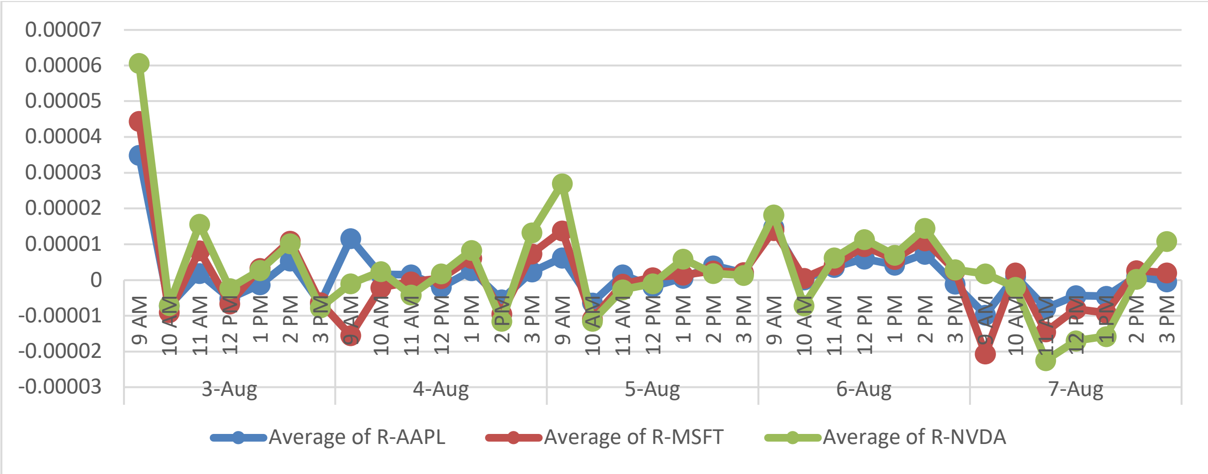
R-AAPL		***R-MSFT***		***R-NVDA***	
Mean	1.14E-06	Mean	9.09E-07	Mean	1.38E-06
Standard Error	1.04E-06	Standard Error	1.22E-06	Standard Error	1.11E-06
Median	0.00E+00	Median	0.00E+00	Median	0.00E+00
Mode	0.00E+00	Mode	0.00E+00	Mode	0.00E+00
Standard Deviation	2.05E-04	Standard Deviation	2.41E-04	Standard Deviation	2.18E-04
Sample Variance	4.21E-08	Sample Variance	5.82E-08	Sample Variance	4.76E-08
Kurtosis	1.74E+03	Kurtosis	8.42E+03	Kurtosis	3.61E+02
Skewness	1.85E+01	Skewness	5.90E+01	Skewness	5.99E+00

Table 4. Statistical analysis over the sample week ending on August 7th, 2020.

For the sample week of 2020, it is easy to check that from the 3 stocks, the one with the lowest return is Microsoft; however, when we analyzed the standard deviation, it is possible to notice that all of them are close to 0, meaning that the information analyzed haven't had big volatility on its returns, something that is expected since the data analyzed is data with 3 seconds of difference.

Another important analysis that should be taken into count, is the kurtosis and the skewness of the three stocks, in the case of the Kurtosis is easy to see that Microsoft is the one with the heaviest tails and with more positive skewness meaning that will tend to the right. This suggests that the stocks could be potentially profitable with also a higher level of risk and potential price movements.

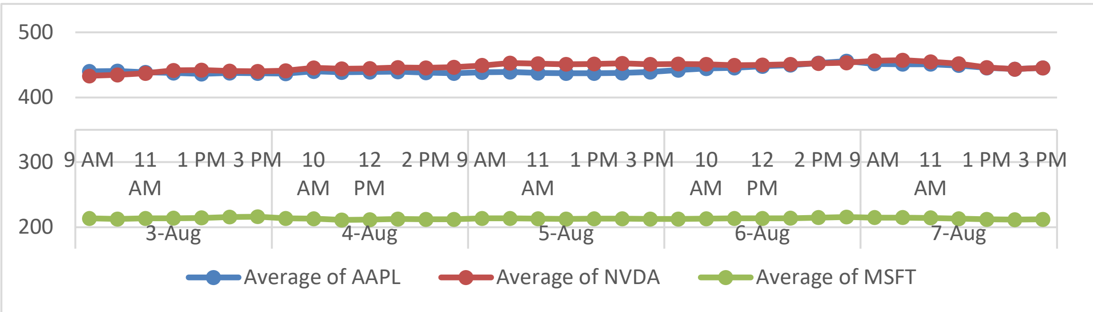
In graph 1, is possible to see the behavior of the returns during the sample week by hour to better understand what was analyzed on the previous chart.



Graph 1. The average return for the sample week ending on August 7th of 2020 – by hour.

The previous graph shows the dispersion of the returns for the one of sample weeks selected for 2020. For this week, it is easy to notice that the biggest variation is daily, meaning between the closure of the market and the opening on the next day.

Even though the statistical analysis is performed under the stock return, and both the standard deviation and the returns have a lot of movement during the different days, is different when we talk about prices, because the main purpose of the HFT is to trade and generate arbitrage or profit from the variations on the intraday data, so the return showed more variation than prices as it is possible to see in the following graph.



Graph 2 Average price for the sample week ending on August 7th, 2020.

- **2021**

For 2021, the market was more stable; however, some companies such as NVDA and AAPL had some remains from the COVID impact that were reflected in the market and the stock statistics. For 2021 we chose the week starting on August 16th and ending on August 20th from the sample for that year, and as for 2020, the returns and the statistics are shown in Table 5.

R-AAPL		***R-MSFT***		***R-NVDA***	
Mean	-8.64E-08	Mean	9.62E-07	Mean	8.95E-07
Standard Error	8.49E-07	Standard Error	6.91E-07	Standard Error	1.74E-06
Median	0.00E+00	Median	0.00E+00	Median	0.00E+00
Mode	0.00E+00	Mode	0.00E+00	Mode	0.00E+00
Standard Deviation	1.68E-04	Standard Deviation	1.37E-04	Standard Deviation	3.44E-04
Sample Variance	2.81E-08	Sample Variance	1.86E-08	Sample Variance	1.18E-07
Kurtosis	8.65E+02	Kurtosis	1.11E+03	Kurtosis	6.41E+02
Skewness	-1.09E+01	Skewness	1.77E+00	Skewness	7.06E+00

Table 5 Statistical analysis over the sample week ending on August 20th, 2021.

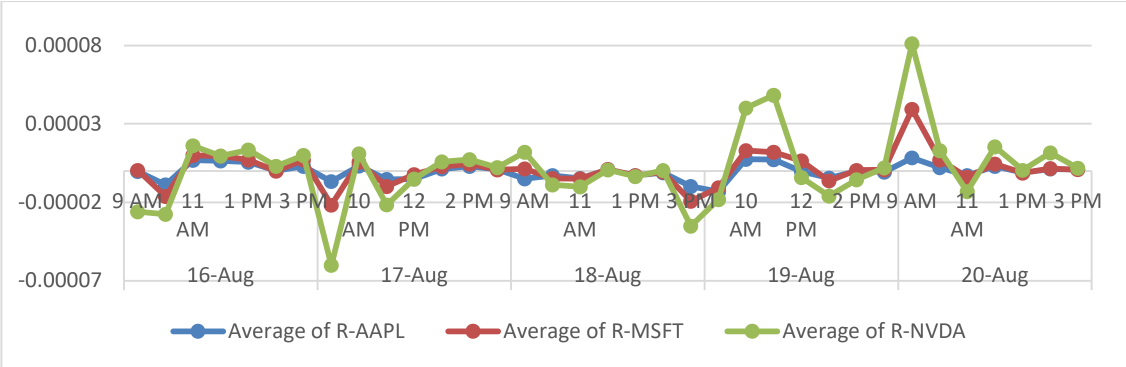
For 2021 the stock behavior was different, since in contrast to the last year, MSFT had the highest mean return followed by NVDA, and then AAPL with the lowest mean, with a negative return.

Similarly to 2020, the variability of the data is low since their standard deviation is close to 0.

In this year all, of them had a positive kurtosis, meaning that the distribution of their returns has stronger tails and had a peaked shape compared to a normal distribution. This could mean that there were some extreme fluctuations under the low range of prices.

Regarding the skewness coefficient, the stocks changed and for 2021 AAPL has a negative skewness meaning that is left-skewed compared with the other two which had positive skewness. What could let us deduce that for 2021 it was better to invest in MSFT and NVDA instead of AAPL.

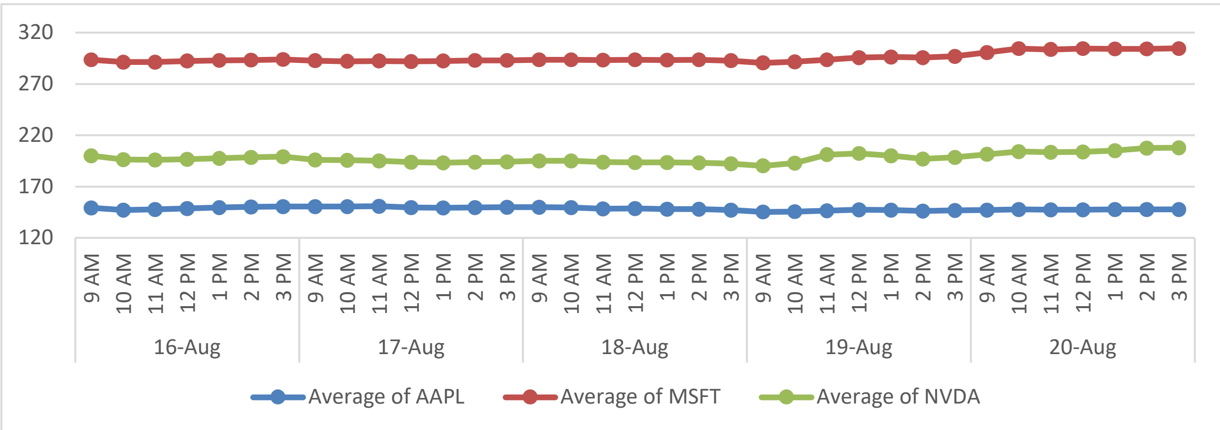
In Graph 3, it is possible to see the behavior of the returns during the sample week by hour to better understand what was analyzed on the previous chart.



Graph 3 Average return for the sample week ending on August 20th of 2021 – by hour.

The previous graph shows the dispersion of the returns for the one of sample weeks selected for 2021. For this week it is easy to notice that the biggest variation is daily, meaning between the closure of the market and the opening on the next day.

Even though the statistical analysis is performed under the stock return, and both the standard deviation and the returns have a lot of movement during the different days, we can notice that 2021 was more stable than the week analyzed for 2020. And it is possible to see it on prices fluctuation, in the following graph.



Graph 4 Average price for sample week ending on August 20th, 2021.

For 2021, both NVDA and AAPL had a decrease in their price as it was mentioned before because of the pandemic impact, but MSFT recovered becoming one of the most profitable for this period.

- **2022**

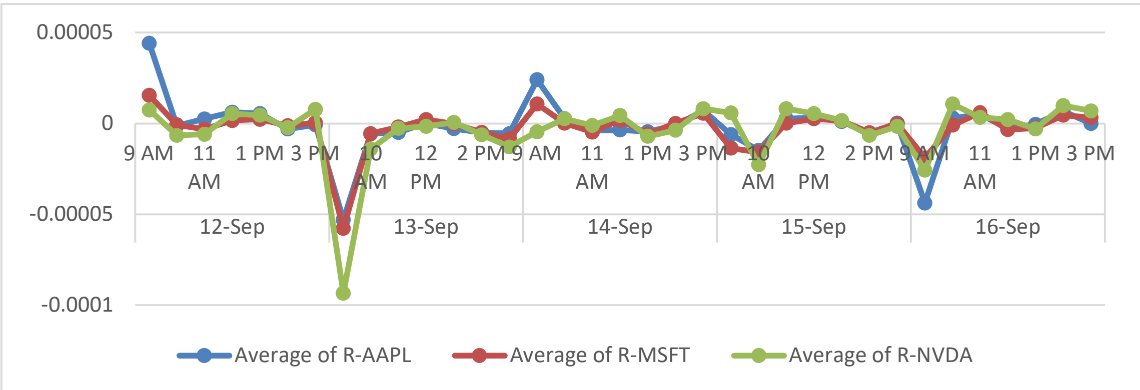
In 2022, the global economic consequences started to be visible in the statistics, and for this year the three of them had negative mean, with NVDA presenting the lowest mean return followed by MSFT and AAPL, as is shown in Table 6.

R-AAPL		***R-MSFT***		***R-NVDA***	
Mean	-1.14E-06	Mean	-1.99E-06	Mean	-2.20E-06
Standard Error	1.14E-06	Standard Error	1.19E-06	Standard Error	2.06E-06
Median	0.00E+00	Median	0.00E+00	Median	0.00E+00
Mode	0.00E+00	Mode	0.00E+00	Mode	0.00E+00
Standard Deviation	2.25E-04	Standard Deviation	2.35E-04	Standard Deviation	4.06E-04
Sample Variance	5.06E-08	Sample Variance	5.51E-08	Sample Variance	1.65E-07
Kurtosis	2.81E+03	Kurtosis	6.82E+03	Kurtosis	5.82E+03
Skewness	-2.00E+01	Skewness	-5.40E+01	Skewness	-4.78E+01

Table 6 Statistical analysis for sample week ending on September 16th of 2022.

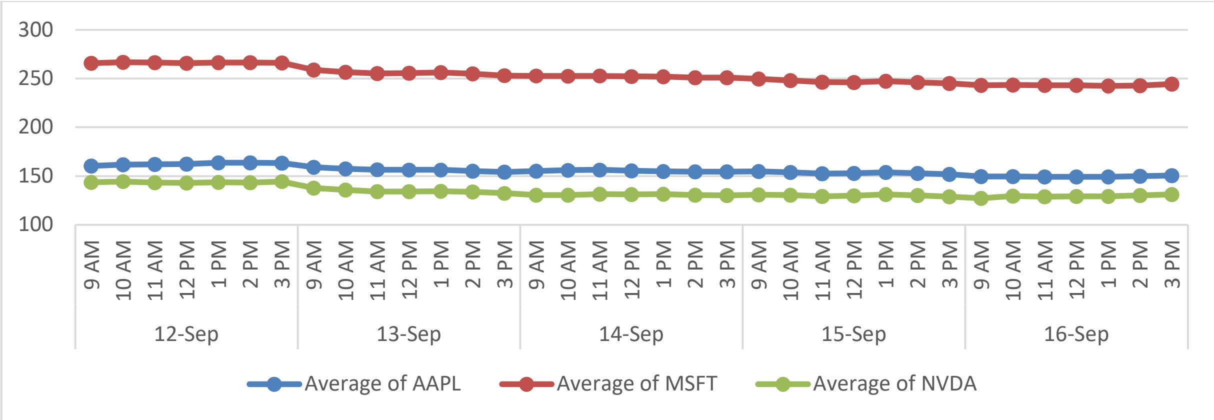
In the analyzed period, the three stocks have high kurtosis values suggesting that their distribution has heavier tails than a normal distribution; however, the skewed is negative, indicating that the distribution is left-skewed.

The next graph shows the dispersion of the returns for one of the two sample weeks for 2022, showing that same as the previous data analyzed, the biggest movement was between the closure and opening on the next day.



Graph 5. The average return for the sample week ending on September 16th of 2022 - by hour.

For 2022 the average market price for 2022 on the 2 sample weeks had a small decrease except for NVDA which has been struggling in the analyzed years. This performance could be related to concern about global economic growth or inflation that could lead to increased volatility in the stock markets.



Graph 6 Average price for sample week ending on September 16th of 2021.

- **2023**

In 2023, the market starts to show recovery for those three companies, with a positive standard error and standard deviation, as is shown in the following table.

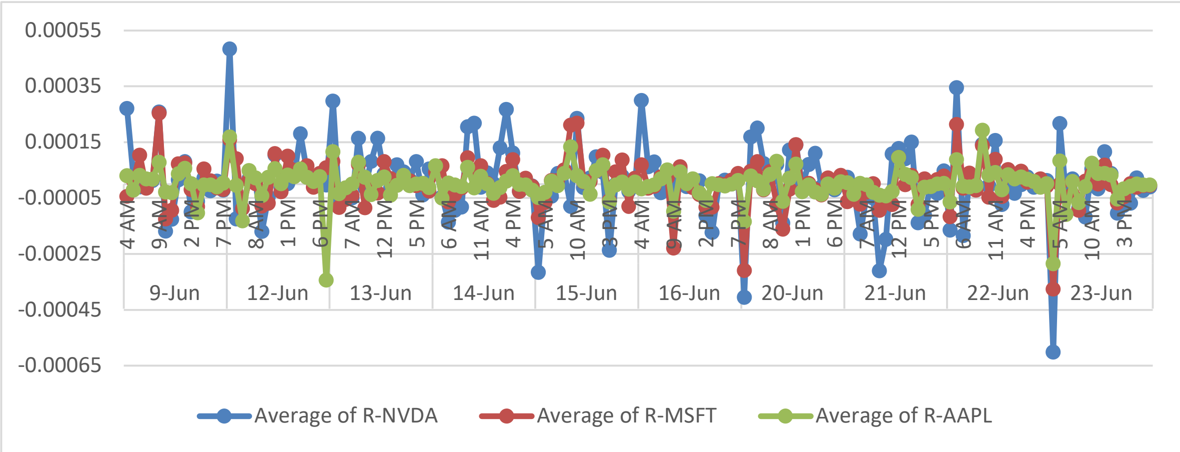
R-AAPL		***R-MSFT***		***R-NVDA***	
Mean	4.3E-06	Mean	4.0E-06	Mean	1.0E-05
Standard Error	4.7E-06	Standard Error	6.6E-06	Standard Error	1.1E-05
Median	0.0E+00	Median	0.0E+00	Median	0.0E+00
Mode	0.0E+00	Mode	0.0E+00	Mode	0.0E+00
Standard Deviation	4.3E-04	Standard Deviation	6.0E-04	Standard Deviation	9.8E-04
Sample Variance	1.8E-07	Sample Variance	3.6E-07	Sample Variance	9.5E-07
Kurtosis	3.9E+01	Kurtosis	3.2E+01	Kurtosis	7.7E+01
Skewness	1.1E+00	Skewness	-4.1E-01	Skewness	-3.0E-03

Table 7 Statistical analysis for the selected period from June 9 to June 23 of 2023

For these stocks in the analyzed period, the three stocks have high kurtosis values suggesting that their distribution has heavier tails than a normal distribution; however, the skewed is negative for MSFT and NVDA, indicating that the distribution is left-skewed.

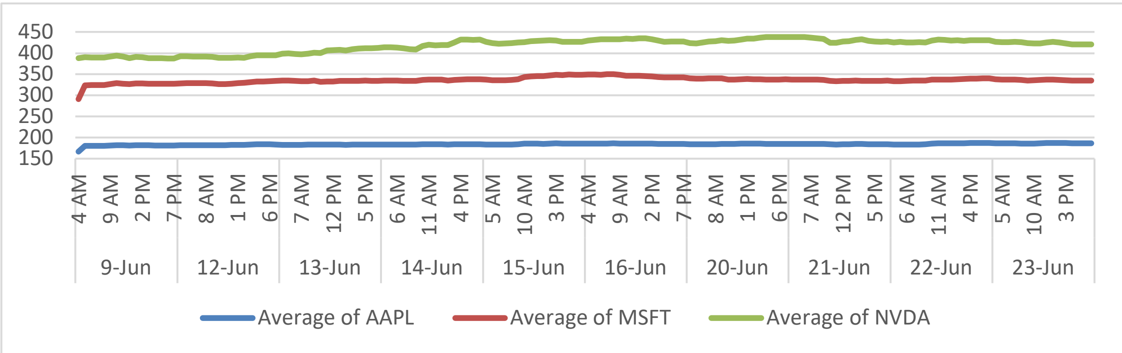
The next graph shows the dispersion of the returns for one of the samples selected in 2023, showing that same as the previous data analyzed, the biggest movement was between the closure and opening the next day.

For 2023 the Average market price had an increase on NVDA though AAPL and MSFT remain



Graph 7 Average return for the sample selected starting on June 9 and ending on June 23 - by hour.

constant. This performance could be related to the concern about global economic growth or inflation that could lead to increased volatility in the stock markets.



Graph 8 Average price for the sample period starting on June 9 and ending on June 23

5. Market Making Model

Now that we have distinguished the concept of HFT and AT, it is important to understand what market making is and how it relates to these concepts. Market making refers to the strategy of quoting a limited number of buy/sell orders for a financial instrument almost simultaneously to make a profit margin between the bid-ask prices.

This type of strategy is based on three main concepts:

- The cost of order handling[1].
- The cost of being averse selected in a bid or ask quote.
- The market makers are premium risk averse, which requires a risk above the price in nonzero positions.

It is a fact that market makers prefer to operate at lower costs, and in markets that are handled by humans, it is difficult to compete with human-generated costs, which justifies the importance of electronic markets and the implementation of algorithms (Menkveld, 2013).

The market makers have an important role in the market because more of them provide liquidity to all the market participants through the quoting of prices no matter the traded financial instrument, which could be equities, commodities, currencies, derivatives, or others. These orders are frequently placed by using quoting machines, which are programs that generate, update, and delete them according to the investor strategy.

We know that technology has drastically changed the way trading is done, mainly by reducing human trading, which at the same time significantly reduces costs and makes it possible to develop a market making strategy through fast matching to quickly update the quote with available public information and reduce risk.

The market making strategy is to create symmetric bid/ask orders in the mid-price range, but this type of move carries a higher risk of putting the trader in a disadvantageous position. Authors such as Avellaneda and Stoikov (Avellaneda & Stoikov, 2008) have developed an optimal market making model to

[1] To get more information about this topic, please read (Cartea et al., 2015) chapters 1 and 2

control inventory risk and find the optimal bid/ask spreads through two formulas that help market makers solve these problems, as explained below.

Reservation price:

$$r(s, q, t) = s - q\gamma\sigma^2(T - t) \tag{2}$$

Optimal bid & ask spread:

$$\delta^a + \delta^b = \gamma\sigma^2(T - t) + \frac{2}{\gamma} ln \left(1 + \frac{\gamma}{\kappa}\right) \tag{3}$$

where:

S = current market mid-price

q = quantity of assets in the inventory of base asset (positive/long or negative/short)

σ = market volatility

T = closing time or maturity

t = current time

$\delta b, \delta a$ = bid/Ask spread, since this in this model is symmetrical then $\delta a = \delta b$

γ = inventory risk aversion parameter

κ = order book liquidity parameter

For convenience, we will consider risk as a constant that does not change over time and wealth and an inventory equal to 1; and, therefore, an exponential utility function to test the monetary value for a market making strategy approaching our objective to perform a profit by placing different orders at the same time.

The project will be implemented using as a main reference the model proposed by Avellaneda & Stoikov (2008) where the inventory size is flexible and optimize the price as a bid-ask quote setting strategy, maximizing the expected exponential utility at time T. To do this it would be necessary to use a formula for the pricing and another one to complement its focus on the inventory that allows the model to reach the maximum level permitted where is still able to keep trading and have earnings:

Pricing: Using the optimal market making model under the exponential utility function.

$$u(s, x, q, t) = \max_{\delta^a, \delta^b} E_t\left[-e^{-\gamma(X_T + q_T S_T)}\right] \qquad (4)$$

Where:

δ^a, δ^b are the bid and ask spreads.

γ is the risk aversion parameter.

X_T is the cash at time T.

q_T is the inventory at time T (as it was said before, this will be 1).

S_T Is the stock price at time T.

Making the assumptions that the risk-free rate is zero and the mid-price follows the standard Brownian motion $dS_t = \sigma dW_t$ when $S_0 = s$ and the standard deviation σ.

Once these market orders are placed at the best price and are executed on the market, the trader will accumulate the traded financial instruments. These are known as inventory and are variables that the trader should manage by using two options. One could be by keeping track of the instruments to avoid accumulating them; or managing the inventory to ensure enough profitability.

Inventory Control[2]: this formula will be able to keep trading and earning by adjusting the order size based on the current position:

$$\phi_t^{bid} = \{\phi_t^{max} \ if \ q_t < 0 \ \phi_t^{max} * e^{-\eta q_t} \ if \ q_t > 0 \qquad (5)$$

$$\phi_t^{ask} = \{\phi_t^{max} \ if \ q_t > 0 \ \phi_t^{max} * e^{-\eta q_t} \ if \ q_t < 0 \qquad (6)$$

[2]To get more information is possible to consult the paper Optimal High-Frequency Market Making (Fushimi et al., 2018)

where:

ϕ_t^{bid}, ϕ_t^{ask} are the bid and ask order sizes at time t

ϕ_t^{max} is the maximum order size at time t.

η is a shape parameter, which could be -0.005 to control not only the inventory but the risk on smaller orders.

This not only will be helpful to obtain dynamic order sizer frameworks but also will be effective to control the inventory risk.

6. Algorithm on Trading

As was said before, HFT refers to the use of algorithms and computer programs to trade financial instruments as fast as possible, to take advantage of small price differences in securities traded on different exchanges.

Many kinds of algorithms are commonly used, such as:

- Statistical arbitrage, which looks for price discrepancies between two or more securities that are highly correlated.
- Trend following, where the algorithm looks for the trends in the market and buys or sells the financial instruments based on the market trend.
- News-based trading, where the algorithm is trained to scan news articles and social media to recap information that could affect the price in the market of any instrument.
- Scalping, which attempts to profit from small price movements by quickly buying and selling securities, often uses limit orders.
- Market making, which looks to provide liquidity to the market by generating both buy and sell orders for an instrument. These algorithms are designed to quickly adjust the price of the orders based on market conditions and level of demand.

For this paper, as it was explained, we will analyze the performance of a market making algorithm.

Aldridge (2013) exposes market making as the following process:

1. The market maker place limit (buy or sell) orders, depending on the market condition and in their portfolio. It is important to mention that the market maker could choose to either place only buy, only sell, or both kinds of limit orders.
2. Then the market reaches the investor buy limit order with the highest price. This price becomes in the best bid or put and is distributed, in the case of the sell orders it works quite similarly when instead of reaching the highest price reach the lowest price, then his order become the best offer or best ask.
3. Market makers' orders are executed by matching with income market orders of the opposite direction. It is important to remark that as soon as the orders are executed is the function of the market maker to manage their inventory to ensure sufficient profitability, and to track and respond to information to avoid buying or selling more than is capable.

These market makers face two types of risk that, as was mentioned before, could be due to inventory risk or risk-averse selection.

6.1 Basics of the Algorithm

As it was mentioned, under the market making model, an algorithm places buy and sell orders in a specific instrument to earn a small profit on each transaction. One way to optimize this trading process could be through the following steps:

1. Choose a security that has sufficient liquidity and trading volume that allows input multiple trades during the selected period.
2. Calculate the bid-ask spread for the financial instrument by analyzing information such as historical data, current market conditions, and competitor behavior.
3. Establish pricing limits for the orders; this will depend on the metric used and the current market conditions.
4. Monitoring continuously the market for changes in pricing, trading volume, and competitor behavior that may affect the ability to make profits.
5. In the case of market conditions changes, it is necessary to adjust the pricing limits to ensure that profitability will be maintained.
6. Ensure that the technology use is advanced and that it is possible to have high-speed connections to minimize the time it takes to process orders and receive market data.
7. Manage the risk to limit losses with risk management strategies like stop-loss orders.
8. Have a continuously tracking performance using data analysis tools to identify areas to improve and optimize the trading strategy.
9. With the algorithm performance, improve and refine constantly the algorithms reducing risk and improving profitability.

Now that we establish the steps to start performing the algorithm, we need to define the metric that will be used to perform the model.

There are multiple methodologies to invest and make informed decisions as fundamental analysis, value investing, growth investing, technical analysis, buy and hold strategies, diversification, income investing, and risk management, however these methodologies are not mutually exclusive, since is possible to use different combinations of them to reach the investor objective.

We will now analyze some of the different methodologies mentioned above and focus on the most appropriate one, considering the data obtained and the model to be implemented.

Fundamental analysis: this methodology aims to measure the intrinsic value of a security, as known as the value of the investment based on the market and economic conditions of a company, through the analysis of different external factors such as economic and financial, relying on financial information reported by the companies from the instruments that will be analyzed, using ratios and metrics that indicate how is the performance of the selected instrument against comparable companies, all of this by quantitative and qualitative metrics.

Value investing: this investment strategy involves buying securities that are cheap relative to the intrinsic value linked to financial statement variables. The target of this methodology is to buy financial instruments when the price is temporarily low to perform gains when it recovers in the market through making assumptions about market behavior and investing, through the analysis of factors such as low price-to-earnings-ratio, low price-to-book ratio, and high dividend yield.

Growth investing: in this methodology, the focus is on identifying companies or assets with high growth potential. They analyze different factors and indicators such as revenue growth, earning growth market share, and industry trends to identify investments that can deliver above-average growth rates.

Buy and hold strategy: this strategy involves selecting investments with a long-term perspective and holding onto them for an extended period regardless of short-term market fluctuations. In this methodology, the approach is in long-term growth and aims to benefit from compounding returns over time.

Technical analysis[3]: This methodology aims to analyze the trend of financial instruments through the study of past price movements, patterns, and trends to predict future price movements from the historical price and volume data.

Due to that the common information obtained by the market, the data banks contain only the market price for the different equities without taking into consideration the bid and ask, therefore the most appropriate methodology will be the technical analysis; however, the most common as it was said before,

[3] To have more information related please check the paper Fundamental and Technical analysis: substitutes or complements (Bettman et al., 2009)

is to incorporate more than one methodology which could be in this case fundamental and technical analysis, to recognize their potential as complements.

To use the technical analysis as methodology and metrics, it is important to check the different technical indicators such as the simple moving average (SMA), exponential moving average (EMA), moving average convergence/divergence rules (MACD), relative strength index (RSI) and on-balance-volume (OBV).

Since we cannot use volumes in our analysis, then we will use the moving average indicators to perform the model, specifically the exponential moving average (EMA) to help us to identify trends in the market and provide signals for potential entry or exit points.

The EMA[4] is a technical analysis indicator that is widely used to analyze price trends and generate trading signals as was said above; however, opposite to the SMA, this indicator places more weight on recent price data, making it more responsive to market conditions.

$$EMA = Price_{(t)} * k + EMA_{(y)} * (1 - k) \tag{7}$$

Equation (7) is used for EMA calculation, as follows:

1. Determine the period: based on the time frame and investor’s preferences, the specific number of periods will be selected. Since the basis of the project is intraday data, it will be selected 400 observations, which will be equivalent to 20 minutes.
2. Calculate the smoothing factor (k): also known as the smoothing multiplier, it will determine the weight given to the most recent data point, is calculated using the chosen period, through formula (8), where n represents the number of periods and is always a value between 0 and 1, where the closest to 0 suits more into long-term trading and the closest to 1 the most suitable to short-term trading.

$$k = \frac{2}{(N + 1)} \tag{8}$$

[4]Is possible to check more detailed information in chapter 2.3 of paper Technical analysis in Financial Markets (Griffioen, 2003)

3. Calculate the initial EMA: for the initial period, the SMA serves as the first EMA by totaling the most recent closing prices divided by the number “n” of periods in the calculation average.

$$SMA = \frac{\sum prices}{n} \qquad (9)$$

This metric will help to predict the price tendency for each instrument. Then to determine which will be the best EMA lapse it is necessary to perform a regression that allows us to quantify the performance of it in both frequencies (1-minute and 3-second latency) and with different periods.

In table 8 it is possible to check the regression analysis for both and, as expected, the smaller the number of observations, the smaller the error. Meaning that will be preferable to work between the 20-minute to 50-minute gap for both data.

Regression Statistics 1m

	20Min	***40Min***	***50Min***	***1H***	***1.5H***	***2H***	***2.5H***
Multiple R	0.996	0.991	0.989	0.987	0.987	0.974	0.968
R Square	0.992	0.983	0.979	0.974	0.974	0.949	0.937
Adjusted R Square	0.992	0.983	0.979	0.974	0.974	0.949	0.937
Standard Error	0.158	0.224	0.250	0.273	0.273	0.378	0.416
Observations	7270	7250	7240	7230	7230	7170	7140

Regression Statistics 3s

	20Min	***40Min***	***50Min***	***1H***	***1.5H***	***2H***	***2.5H***
Multiple R	0.997	0.993	0.991	0.989	0.982	0.976	0.969
R Square	0.994	0.986	0.983	0.979	0.965	0.952	0.938
Adjusted R Square	0.994	0.986	0.983	0.979	0.965	0.952	0.938
Standard Error	0.349	0.509	0.575	0.637	0.801	0.931	1.036
Observations	38577	38177	37977	37777	37077	36577	35977

Table 8. Regression Statistics over 1-minute and 3-second data

The main idea of the algorithm is to place an order based on the behavior provided by the EMA, that for this project will be 20 minutes where the error is smaller, and once the order is exercised, immediately the algorithm should create a selling position at or above the entry price or the call price.

Now that the EMA has been calculated and analyzed, it is important to establish the other variables to perform the model, as are the bid and ask prices. Due to the constraints on the data, it will be necessary to create a factor that allows us to create fictional bid and ask prices.

This will help the strategy and the code to be able to work not only with a factor but also with real values if the data is found to then replicate the coding with the investor's information and necessities.

For this project, the bid and ask prices will be just the mid-price times 0.006 over and below respectively to establish the limit orders for both buy and sell updated at no cost. These limit orders will help to determine the priority of execution and the limit price where the order should be executed.

To implement the model, we should develop the algorithm in three main steps, first then choose the parameters to run the simulation. These parameters are established by the authors as follows: $T = 1$, $\sigma = 2$, $q = 0$, $\gamma = 0.1$, $k = 1.5$ and $A = 140$; after this, the second step will be calculating the variables to allow the model to create the Bid and Ask offer price, they conclude that since the idea of the model is to create symmetrical bid and ask orders around the market mid-price, and to avoid the skewing in just one direction, it was important to create a reservation price (Formula 10) that help to solve this inventory problem.

$$r(s, q, t) = s - q\gamma\sigma^2(T - t) \tag{10}$$

And then a calculation of the optimal bid and ask spread as it was mentioned in Formula 3, means the optimal position the market makers should be to increase profitability.

As it was mentioned before, and now that the structure of the model is explained, then is necessary to recap the main parameter or factors described by Avellaneda-Stoikov:

1. Inventory position (q): belongs to the number of assets in the inventory of base asset, could be positive/negative for long/short positions, and is a percentage parameter.

2. Time until the trading session ends (T-t): where T will be normalized to 1 will be the number of days and t will be the range between the initial date and the final time to adjust the period where the code is located during the day.
3. Risk factor (ϒ): since this is proportional to the difference between mid-price and reserved price, we could say that the smaller the symmetric and less biased, this could be calculated or given.
4. Order book depth (k): similar to the Risk factor, this could be calculated or in this case given, however, is important to know that a significant value will mean that the order book is more liquid so will have a smaller spread since there is more demand on the market, and the other way around in case of a smaller k.

The third and final step will be combining the prior calculations, for us to create the market order by using the reservation price as follows:

$$\text{Bid offer price} = \text{reservation price} - \frac{\text{optimal spread}}{2} \tag{11}$$

and

$$\text{Ask offer price} = \text{reservation price} + \frac{\text{optimal spread}}{2} \tag{12}$$

6.2 Analysis of the Processed Data

As mentioned before, in this project was chosen to apply the model performed by Avellaneda and Stoikov. To accomplish this purpose, we will test the model in two different stages: the first one trying to replicate the model with the same data from their paper to ensure that the model fits with their proposal; and then with the data from this work.

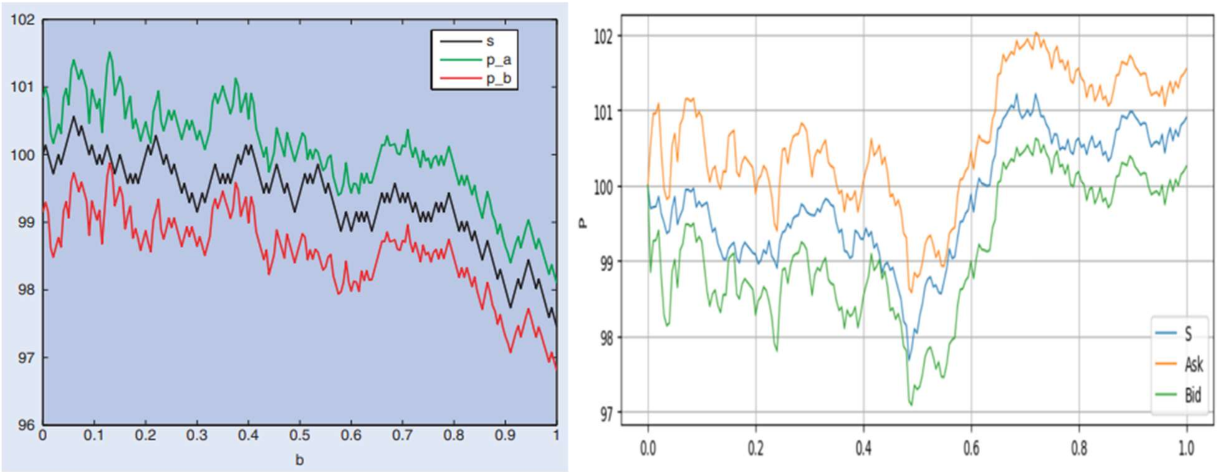
6.2.1 Replication of the model with the data provided by Avellaneda and Stoikov.

There are multiple camps as Hummingbot which is a non-for-profit organization that facilitates decentralized maintenance and governance of different codes focusing on cryptocurrencies.

Here it was possible to find a similar code under the model selected to use as the skeleton for the replication of the paper used as a reference.

The code creates 1000 Monte Carlo simulations for the same parameters selected and mentioned before, allowing us to compare with the provided results in the studied papers.

The results are quite accurate, In the figure shown below it is possible to determine that even though the behavior of the price is different, in Avellaneda and Stoikov's paper the price tends to go down, and the relationship between the Optimal offer bid and ask price to share the same behavior with the code used in this dissertation.



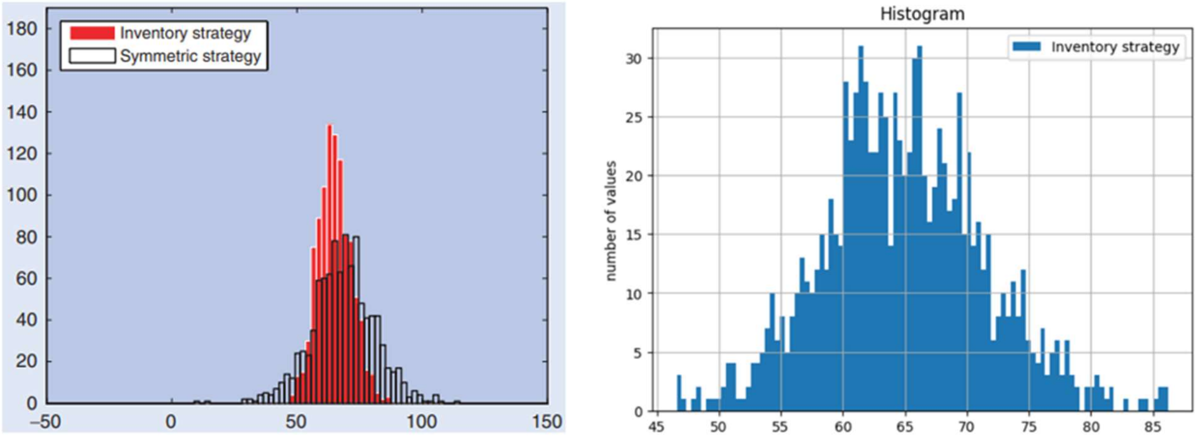
Avellaneda & Stoikov results

Code used to test the parameters

Graph 9 comparison for the price results between reference paper and code used.

Also, when we compared the inventory strategy against the PnL obtained by the total of observations during the selected period we realized that the information is like the one proposed by the authors as we can confirm in Graph 10.

This allows us to conclude that since the results follow the same trend, the code is accurate and fits perfectly with the model proposal.



Avellaneda & Stoikov results *Code used to test the parameters*

Graph 10 Comparison of the inventory strategy results between reference paper and code used.

6.2.2 Application of the proposed model on real data

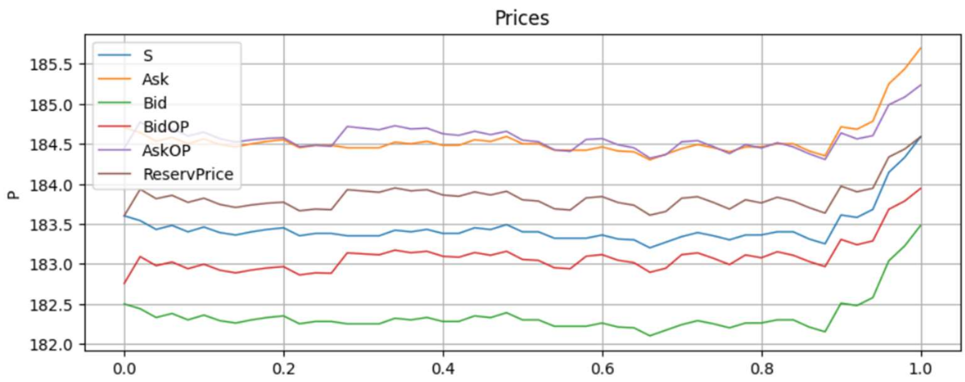
Now that we were able to validate that the code used fits with the information provided on the reference paper, then it was possible to adapt it with real data.

Contrary to what is proposed in the reference paper, for the analysis of this project we used the different intraday data found (one minute and three seconds latency), instead to create our simulations based on an initial price of S = 100.

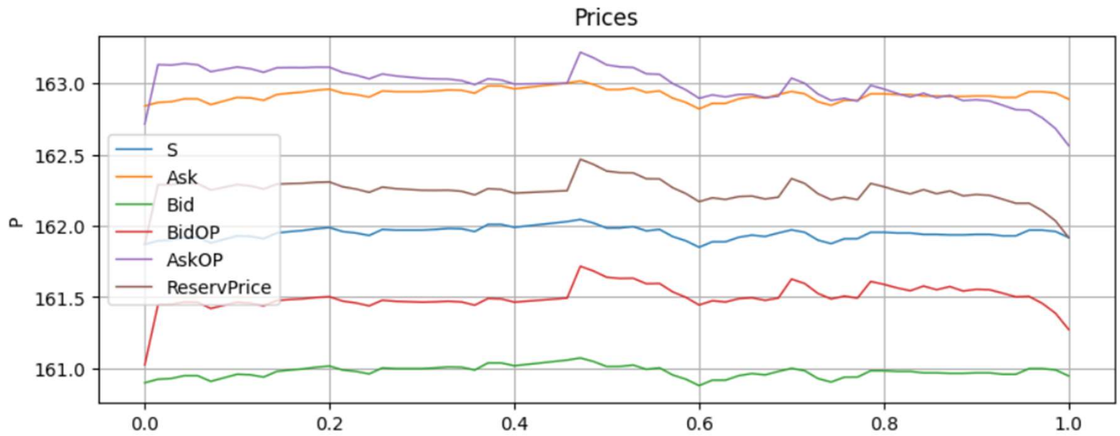
Another important remark to be included in the code created is the number of simulations and the number of steps, this should be proportional to the different number of observations provided.

For this exercise, we use two different latencies and observations to check how the model works on both and perform different results as prices, position, inventory strategy, and profitability.

In graphs 11 and 12 we can notice that the model is similar no matter the latency, the model works in the same way respecting the stock movements.

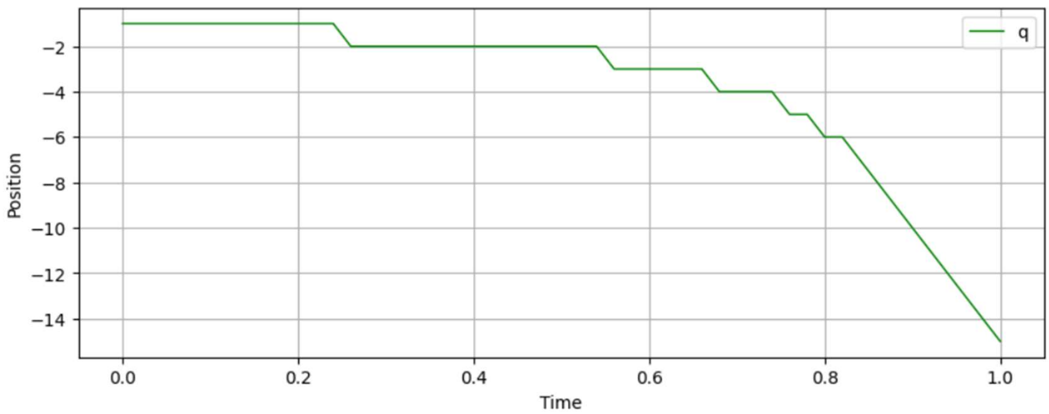


Graph 11 Prices 1-minute latency

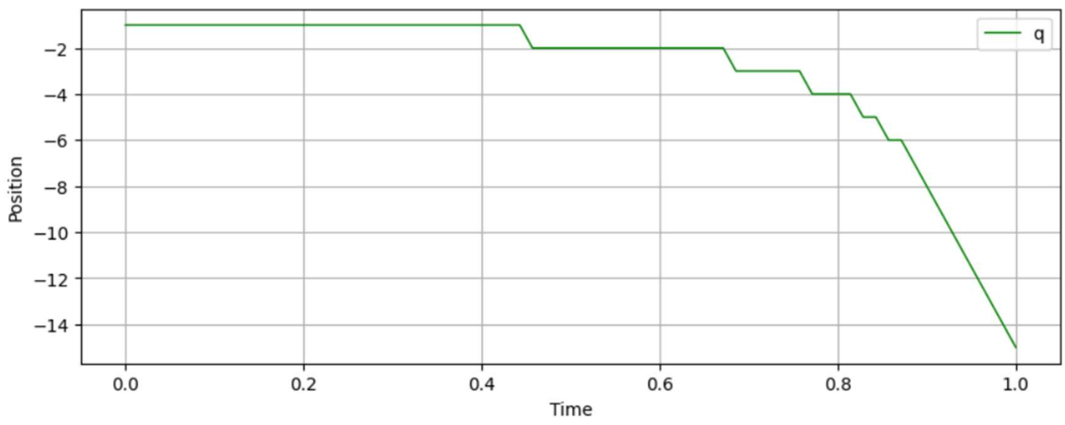


Graph 122 Prices 3 Seconds latency

In both cases and due to the high latency of the model, no matter the frequency of the data we realized that the tendency of the inventory should go in short as could be visible in graphs 13 and 14.

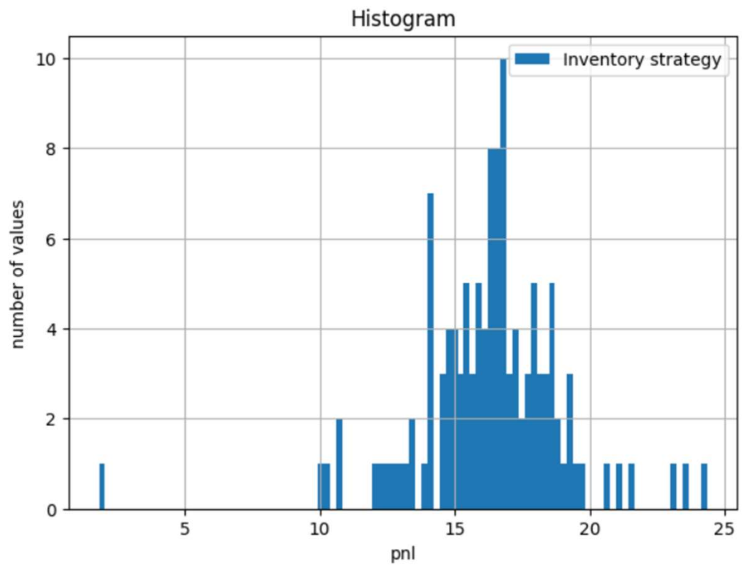


Graph 13 Position through time 1-minute latency

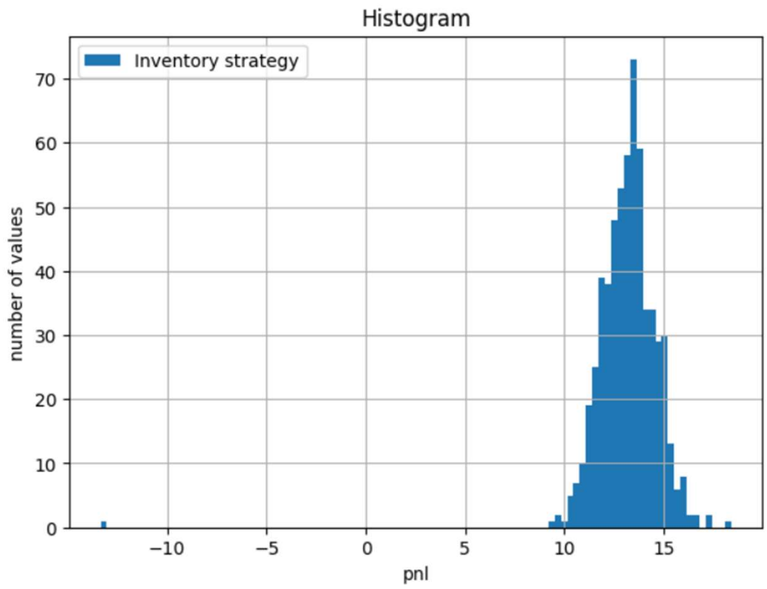


Graph 14 Position through time 3 Seconds latency

As a third indicator, the model is accurate in giving a symmetrical inventory, even though is skewed to the right, for this kind of indicator between more observations analyzed more focused the results and the inventory is more accurate.



Graph 15 Histogram inventory strategy 1-minute latency



Graph 16 Histogram inventory strategy 3 seconds latency

Finally, due to the high latency of the data, more simulations are run in the model with more probabilities of having losses as is possible to see in graphs 17 and 18.

Graph 17 Profit and losses 1-minute latency

Graph 18 Profit and losses 3 seconds latency

7. Concluding Remarks

This project aims to implement the high frequency market making pricing strategy proposed by Avellaneda and Stoikov (2008), and it allows us to conclude the following:

- The inventory control model used by them in the strategy is possible to achieve profitability.
- The number of simulations had a big impact on the profit and the inventory control due to their high latency, the bigger the difference between the simulation and the time steps or orders the lower the profit obtained.
- The idea of the model analyzed is to minimize inventory risk with constant risk aversion coefficients.
- Even though the results were calculated with assumed constant parameters, one way to improve the model proposed and the code created is by automatizing the calculation of them.
- Every simulation ends with a small position on average indicating that the inventory management was successful.
- The highest the frequency and the order placed, the margin error is smaller, is recommended to use a 20-minute interval to calculate the EMA and also as a maximum interval to place each order.

References

Aldridge, I. (2013). *High-frequency trading: a practical guide to algorithmmic strategies and trading systems* (Second Edition). John Wiley & Sons , Inc. www.WileyFinance.com.

Avellaneda, M., & Stoikov, S. (2008). High-frequency trading in a limit order book. *Quantitative Finance*, *8*(3), 217–224. https://doi.org/10.1080/14697680701381228

Bettman, J. L., Sault, S. J., & Schultz, E. L. (2009). Fundamental and technical analysis: Substitutes or complements? *Accounting and Finance*, *49*(1), 21–36. https://doi.org/10.1111/j.1467-629X.2008.00277.x

Cartea, A., Jaimungal, S., & Penalva, J. (2015). *ALGORITHMIC AND HIGH-FREQUENCY TRADING*.

Fushimi, T., Rojas, C. G., & Herman, M. (2018). *Optimal High-Frequency Market Making*.

Gomber, P., Arndt, B., Lutat, M., & Uhle, T. (2012). *High-Frequency Trading*. http://ssrn.com/abstract=1858626Electroniccopyavailableat:https://ssrn.com/abstract=1858626

Griffioen, G. (Gerwin A. W. (2003). *Technical analysis in financial markets*. Thela Thesis].

Grody, A. D., & Levecq, H. (1993). *Past, Present and future: The evolution and development of electronic financial markets*.

Menkveld, A. J. (2013). *High Frequency Trading and the New-Market Makers*. http://ssrn.com/abstract=1722924Electroniccopyavailableat:http://ssrn.com/abstract=1722924 Electroniccopyavailableat:https://ssrn.com/abstract=1722924

Smith L, R. (2009). *Premodern Trade in World History*.

Non-Academic Sources: Reports and other Sources

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, *53*(4), 3007–3057. https://doi.org/10.1007/s10462-019-09754-z

Lopez de Prado, M. (2011). *ADVANCES IN HIGH FREQUENCY STRATEGIES*. http://ssrn.com/abstract=2106117https://ssrn.com/abstract=2106117Electroniccopyavailableat:https://ssrn.com/abstract=2106117

Falces Marin, J., Díaz Pardo de Vera, D., & Lopez Gonzalo, E. (2022). A reinforcement learning approach to improve the performance of the Avellaneda-Stoikov market-making algorithm. *PloS One*, *17*(12), e0277042. https://doi.org/10.1371/journal.pone.0277042

Yahoo Finance: https://finance.yahoo.com/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAHlUigwaN9SuEtcgpp3BwkpEUgESIqJbDYmWnC5dytbYQHtZKiQJS9j6k1COCSVPGLvbdwo_k6j8dRgtRcWZHx6zGNJSImjltb66fTYC3ZTd17kc9K-gVrXNJ8DbC3r2GiyVs9ewg3_G-jODj981fJ8Qf1p546KPJadRcU24T3Lu

Hummingbot Blog: https://blog.hummingbot.org/2021-04-avellaneda-stoikov-market-making-strategy/

Kaggle Intraday market data: https://www.kaggle.com/datasets/brtnsmth/intraday-market-data

Nasdaq data base: https://data.nasdaq.com/

Annexes

Annex A – Complementary Tables

Table 1A Sample of return calculation 3 seconds latency September 2022

Day	Hour	AAPL	MSFT	NVDA	R-AAPL	R-MSFT	R-NVDA
16-Sep-22	3:59:59 PM	150.55	244.73	132.03	-0.07968%	-0.07556%	-0.00470%
16-Sep-22	3:59:56 PM	150.67	244.915	132.0362	-0.02986%	-0.00612%	0.01985%
16-Sep-22	3:59:53 PM	150.715	244.93	132.01	-0.00332%	-0.02556%	-0.03030%
16-Sep-22	3:59:50 PM	150.72	244.9926	132.05	-0.00332%	0.02147%	0.01515%
16-Sep-22	3:59:47 PM	150.725	244.94	132.03	0.00332%	-0.00816%	0.02272%
16-Sep-22	3:59:44 PM	150.72	244.96	132	0.02986%	0.03471%	0.00000%
16-Sep-22	3:59:41 PM	150.675	244.875	132	-0.02323%	-0.00613%	0.00758%
16-Sep-22	3:59:38 PM	150.71	244.89	131.99	0.01327%	-0.00817%	-0.01136%
16-Sep-22	3:59:35 PM	150.69	244.91	132.005	-0.03318%	0.01784%	-0.00311%
16-Sep-22	3:59:32 PM	150.74	244.8663	132.0091	-0.01990%	0.01891%	0.00689%
16-Sep-22	3:59:29 PM	150.77	244.82	132	-0.03316%	-0.04900%	-0.02651%
16-Sep-22	3:59:26 PM	150.82	244.94	132.035	0.00000%	-0.00408%	-0.00371%
16-Sep-22	3:59:23 PM	150.82	244.95	132.0399	-0.01326%	0.01021%	0.02643%
16-Sep-22	3:59:20 PM	150.84	244.925	132.005	-0.01326%	-0.03878%	-0.01894%
16-Sep-22	3:59:17 PM	150.86	245.02	132.03	-0.02651%	-0.00449%	-0.04165%
16-Sep-22	3:59:14 PM	150.9	245.031	132.085	-0.00663%	-0.02816%	-0.00757%
16-Sep-22	3:59:11 PM	150.91	245.1	132.095	0.04553%	0.04081%	0.03786%
16-Sep-22	3:59:08 PM	150.8413	245	132.045	0.00749%	0.00000%	0.02749%
16-Sep-22	3:59:05 PM	150.83	245	132.0087	0.00000%	0.00612%	0.00659%
16-Sep-22	3:59:02 PM	150.83	244.985	132	-0.06296%	-0.04693%	-0.07618%
16-Sep-22	3:58:59 PM	150.925	245.1	132.1006	-0.01391%	-0.02040%	0.00045%
16-Sep-22	3:58:56 PM	150.946	245.15	132.1	0.03379%	-0.00408%	0.00000%
16-Sep-22	3:58:53 PM	150.895	245.16	132.1	0.01518%	0.02040%	0.02271%
16-Sep-22	3:58:50 PM	150.8721	245.11	132.07	0.00139%	0.00408%	0.03029%
16-Sep-22	3:58:47 PM	150.87	245.1	132.03	-0.00663%	-0.00612%	-0.03408%
16-Sep-22	3:58:44 PM	150.88	245.115	132.075	-0.01902%	0.00861%	-0.01136%
16-Sep-22	3:58:41 PM	150.9087	245.0939	132.09	-0.01080%	-0.01065%	0.00000%
16-Sep-22	3:58:38 PM	150.925	245.12	132.09	0.01657%	-0.05302%	0.00757%
16-Sep-22	3:58:35 PM	150.9	245.25	132.08	-0.00663%	0.00000%	-0.00757%
16-Sep-22	3:58:32 PM	150.91	245.25	132.09	0.03977%	0.02447%	0.04543%
16-Sep-22	3:58:29 PM	150.85	245.19	132.03	0.01989%	0.00816%	0.00000%
16-Sep-22	3:58:26 PM	150.82	245.17	132.03	0.00000%	0.00816%	0.00000%
16-Sep-22	3:58:23 PM	150.82	245.15	132.03	0.00000%	0.00000%	0.02272%
16-Sep-22	3:58:20 PM	150.82	245.15	132	0.02321%	0.02040%	0.00379%
16-Sep-22	3:58:17 PM	150.785	245.1	131.995	-0.00332%	-0.00408%	0.00379%
16-Sep-22	3:58:14 PM	150.79	245.11	131.99	0.00809%	0.01224%	-0.00758%
16-Sep-22	3:58:11 PM	150.7778	245.08	132	0.01181%	0.02040%	0.00000%
16-Sep-22	3:58:08 PM	150.76	245.03	132	-0.00663%	0.00457%	0.00379%
16-Sep-22	3:58:05 PM	150.77	245.0188	131.995	-0.00663%	0.00359%	0.01894%
16-Sep-22	3:58:02 PM	150.78	245.01	131.97	0.02653%	0.01629%	0.00000%
16-Sep-22	3:57:59 PM	150.74	244.9701	131.97	0.01327%	-0.00200%	0.00341%
16-Sep-22	3:57:56 PM	150.72	244.975	131.9655	0.00995%	-0.00204%	-0.01099%
16-Sep-22	3:57:53 PM	150.705	244.98	131.98	0.00664%	0.00816%	0.00379%
16-Sep-22	3:57:50 PM	150.695	244.96	131.975	0.00332%	0.01225%	0.01137%
16-Sep-22	3:57:47 PM	150.69	244.93	131.96	-0.00332%	-0.00817%	-0.01515%
16-Sep-22	3:57:44 PM	150.695	244.95	131.98	0.00332%	0.00408%	0.00000%
16-Sep-22	3:57:41 PM	150.69	244.94	131.98	0.00664%	0.02042%	0.02652%
16-Sep-22	3:57:38 PM	150.68	244.89	131.945	0.00996%	-0.01225%	0.01736%
16-Sep-22	3:57:35 PM	150.665	244.92	131.9221	0.00000%	-0.00408%	-0.05146%

Table 2A Sample of return calculation 3 seconds latency August 2021

TimeStamp	TimeStamp	AAPL	MSFT	NVDA	R-AAPL	R-MSFT	R-NVDA
20-Aug-21	3:59:59 PM	148.1701	304.23	208.18	-0.00668%	-0.00329%	-0.01441%
20-Aug-21	3:59:56 PM	148.18	304.24	208.21	-0.04723%	0.00329%	-0.00240%
20-Aug-21	3:59:53 PM	148.25	304.23	208.215	0.02024%	0.00000%	-0.01201%
20-Aug-21	3:59:50 PM	148.22	304.23	208.24	-0.00452%	-0.01315%	0.00480%
20-Aug-21	3:59:47 PM	148.2267	304.27	208.23	0.00452%	-0.00329%	-0.02401%
20-Aug-21	3:59:44 PM	148.22	304.28	208.28	-0.00337%	-0.00164%	-0.00480%
20-Aug-21	3:59:41 PM	148.225	304.285	208.29	0.03711%	0.00657%	0.05282%
20-Aug-21	3:59:38 PM	148.17	304.265	208.18	-0.00675%	-0.00493%	-0.01441%
20-Aug-21	3:59:35 PM	148.18	304.28	208.21	-0.00337%	-0.00657%	0.00961%
20-Aug-21	3:59:32 PM	148.185	304.3	208.19	-0.01012%	-0.00986%	-0.01441%
20-Aug-21	3:59:29 PM	148.2	304.33	208.22	0.01350%	0.01314%	0.00000%
20-Aug-21	3:59:26 PM	148.18	304.29	208.22	0.00000%	0.00329%	0.00000%
20-Aug-21	3:59:23 PM	148.18	304.28	208.22	0.00553%	0.00000%	-0.01681%
20-Aug-21	3:59:20 PM	148.1718	304.28	208.255	0.00796%	-0.01249%	-0.00960%
20-Aug-21	3:59:17 PM	148.16	304.318	208.275	0.00000%	0.00000%	0.00000%
20-Aug-21	3:59:14 PM	148.16	304.318	208.275	0.00675%	-0.00394%	0.01200%
20-Aug-21	3:59:11 PM	148.15	304.33	208.25	0.00000%	0.00329%	-0.00826%
20-Aug-21	3:59:08 PM	148.15	304.32	208.2672	-0.01350%	0.00329%	-0.01095%
20-Aug-21	3:59:05 PM	148.17	304.31	208.29	0.00000%	-0.00657%	-0.01440%
20-Aug-21	3:59:02 PM	148.17	304.33	208.32	0.00000%	0.00329%	0.00000%
20-Aug-21	3:58:59 PM	148.17	304.32	208.32	0.00000%	-0.00329%	0.00480%
20-Aug-21	3:58:56 PM	148.17	304.33	208.31	-0.00337%	0.00654%	0.01920%
20-Aug-21	3:58:53 PM	148.175	304.3101	208.27	0.00000%	0.00000%	0.00000%
20-Aug-21	3:58:50 PM	148.175	304.3101	208.27	-0.00337%	0.00332%	0.00720%
20-Aug-21	3:58:47 PM	148.18	304.3	208.255	-0.00675%	-0.01314%	0.00240%
20-Aug-21	3:58:44 PM	148.19	304.34	208.25	0.00337%	-0.00657%	0.01441%
20-Aug-21	3:58:41 PM	148.185	304.36	208.22	-0.00337%	-0.00821%	-0.00960%
20-Aug-21	3:58:38 PM	148.19	304.385	208.24	0.00000%	-0.00164%	0.01921%
20-Aug-21	3:58:35 PM	148.19	304.39	208.2	-0.02024%	0.00000%	0.00961%
20-Aug-21	3:58:32 PM	148.22	304.39	208.18	0.02362%	0.00986%	0.01441%
20-Aug-21	3:58:29 PM	148.185	304.36	208.15	-0.00337%	-0.00657%	-0.00961%
20-Aug-21	3:58:26 PM	148.19	304.38	208.17	-0.01687%	-0.01314%	0.00000%
20-Aug-21	3:58:23 PM	148.215	304.42	208.17	0.00000%	0.00000%	0.00000%
20-Aug-21	3:58:20 PM	148.215	304.42	208.17	-0.01687%	-0.02628%	-0.03285%
20-Aug-21	3:58:17 PM	148.24	304.5	208.2384	-0.01349%	0.00000%	-0.00077%
20-Aug-21	3:58:14 PM	148.26	304.5	208.24	-0.01012%	-0.01970%	-0.00480%
20-Aug-21	3:58:11 PM	148.275	304.56	208.25	-0.00337%	0.00000%	-0.05281%
20-Aug-21	3:58:08 PM	148.28	304.56	208.36	0.00000%	-0.00328%	0.00518%
20-Aug-21	3:58:05 PM	148.28	304.57	208.3492	0.00000%	0.00000%	0.00000%
20-Aug-21	3:58:02 PM	148.28	304.57	208.3492	0.00000%	0.00000%	0.00000%
20-Aug-21	3:57:59 PM	148.28	304.57	208.3492	0.00337%	0.00154%	0.00922%
20-Aug-21	3:57:56 PM	148.275	304.5653	208.33	0.00337%	-0.00154%	0.00000%
20-Aug-21	3:57:53 PM	148.27	304.57	208.33	0.00000%	0.00328%	0.00480%
20-Aug-21	3:57:50 PM	148.27	304.56	208.32	0.00378%	-0.00657%	0.01680%
20-Aug-21	3:57:47 PM	148.2644	304.58	208.285	0.00297%	0.00000%	-0.00240%
20-Aug-21	3:57:44 PM	148.26	304.58	208.29	0.00000%	0.00000%	0.00000%
20-Aug-21	3:57:41 PM	148.26	304.58	208.29	0.00675%	0.00328%	0.02036%
20-Aug-21	3:57:38 PM	148.25	304.57	208.2476	0.00337%	-0.00932%	0.00567%
20-Aug-21	3:57:35 PM	148.245	304.5984	208.2358	0.00675%	-0.00053%	0.00250%

Table 3A Sample of return calculation 3 seconds latency August 2021

TimeStamp	TimeStamp	AAPL	MSFT	NVDA	R-AAPL	R-MSFT	R-NVDA
07-Aug-20	3:59:58 PM	444.365	212.4	448	0.00563%	0.03292%	0.01674%
07-Aug-20	3:59:55 PM	444.34	212.3301	447.925	0.01125%	0.01418%	0.01005%
07-Aug-20	3:59:52 PM	444.29	212.3	447.88	0.01351%	0.01413%	0.01116%
07-Aug-20	3:59:49 PM	444.23	212.27	447.83	0.00009%	0.02356%	0.03238%
07-Aug-20	3:59:46 PM	444.2296	212.22	447.685	0.01792%	0.00005%	0.02620%
07-Aug-20	3:59:43 PM	444.15	212.2199	447.5677	0.00225%	-0.00476%	-0.03201%
07-Aug-20	3:59:40 PM	444.14	212.23	447.711	0.01126%	-0.00942%	-0.01988%
07-Aug-20	3:59:37 PM	444.09	212.25	447.8	-0.02026%	0.00000%	0.02680%
07-Aug-20	3:59:34 PM	444.18	212.25	447.68	0.02477%	0.00942%	0.01787%
07-Aug-20	3:59:31 PM	444.07	212.23	447.6	-0.03154%	0.00471%	0.00670%
07-Aug-20	3:59:28 PM	444.2101	212.22	447.57	-0.00223%	0.00471%	-0.02681%
07-Aug-20	3:59:25 PM	444.22	212.21	447.69	-0.03376%	-0.02766%	-0.04020%
07-Aug-20	3:59:22 PM	444.37	212.2687	447.87	0.00450%	-0.00532%	0.00000%
07-Aug-20	3:59:19 PM	444.35	212.28	447.87	-0.00450%	-0.00005%	0.04243%
07-Aug-20	3:59:16 PM	444.37	212.2801	447.68	0.04277%	0.00005%	0.00476%
07-Aug-20	3:59:13 PM	444.18	212.28	447.6587	0.02252%	0.02827%	0.00000%
07-Aug-20	3:59:10 PM	444.08	212.22	447.6587	0.01576%	0.00000%	0.00865%
07-Aug-20	3:59:07 PM	444.01	212.22	447.62	0.00000%	0.01414%	0.01564%
07-Aug-20	3:59:04 PM	444.01	212.19	447.55	-0.01606%	-0.03298%	-0.04691%
07-Aug-20	3:59:01 PM	444.0813	212.26	447.76	-0.00421%	0.00000%	-0.00011%
07-Aug-20	3:58:58 PM	444.1	212.26	447.7605	0.00450%	-0.02120%	0.00458%
07-Aug-20	3:58:55 PM	444.08	212.305	447.74	-0.00450%	0.00707%	-0.00223%
07-Aug-20	3:58:52 PM	444.1	212.29	447.75	0.01351%	0.00000%	-0.01563%
07-Aug-20	3:58:49 PM	444.04	212.29	447.82	-0.00225%	0.00311%	-0.00893%
07-Aug-20	3:58:46 PM	444.05	212.2834	447.86	0.00676%	0.00160%	0.01563%
07-Aug-20	3:58:43 PM	444.02	212.28	447.79	-0.00450%	0.00471%	0.00000%
07-Aug-20	3:58:40 PM	444.04	212.27	447.79	0.00448%	-0.00942%	0.00558%
07-Aug-20	3:58:37 PM	444.0201	212.29	447.765	-0.01799%	-0.00236%	-0.00112%
07-Aug-20	3:58:34 PM	444.1	212.295	447.77	0.00901%	0.00236%	0.00071%
07-Aug-20	3:58:31 PM	444.06	212.29	447.7668	0.00901%	0.00000%	0.00040%
07-Aug-20	3:58:28 PM	444.02	212.29	447.765	-0.01239%	0.00471%	-0.00335%
07-Aug-20	3:58:25 PM	444.075	212.28	447.78	0.00563%	0.00471%	0.00893%
07-Aug-20	3:58:22 PM	444.05	212.27	447.74	0.00676%	0.00857%	-0.00223%
07-Aug-20	3:58:19 PM	444.02	212.2518	447.75	0.00144%	0.00085%	-0.00670%
07-Aug-20	3:58:16 PM	444.0136	212.25	447.78	-0.02171%	-0.02355%	0.00447%
07-Aug-20	3:58:13 PM	444.11	212.3	447.76	-0.00662%	-0.00471%	0.01787%
07-Aug-20	3:58:10 PM	444.1394	212.31	447.68	0.00761%	0.01762%	0.00894%
07-Aug-20	3:58:07 PM	444.1056	212.2726	447.64	-0.00099%	0.01300%	0.00447%
07-Aug-20	3:58:04 PM	444.11	212.245	447.62	-0.00901%	-0.02596%	-0.01787%
07-Aug-20	3:58:01 PM	444.15	212.3001	447.7	0.00225%	0.00005%	-0.00447%
07-Aug-20	3:57:58 PM	444.14	212.3	447.72	0.00901%	0.00942%	0.00824%
07-Aug-20	3:57:55 PM	444.1	212.28	447.6831	0.00000%	0.00942%	0.01633%
07-Aug-20	3:57:52 PM	444.1	212.26	447.61	-0.00676%	0.00942%	0.01117%
07-Aug-20	3:57:49 PM	444.13	212.24	447.56	0.00450%	-0.00471%	-0.00670%
07-Aug-20	3:57:46 PM	444.11	212.25	447.59	0.00000%	0.00471%	-0.00894%
07-Aug-20	3:57:43 PM	444.11	212.24	447.63	0.00000%	-0.00471%	0.00447%
07-Aug-20	3:57:40 PM	444.11	212.25	447.61	-0.01351%	0.00942%	0.00085%
07-Aug-20	3:57:37 PM	444.17	212.23	447.6062	-0.03377%	0.00000%	-0.00882%
07-Aug-20	3:57:34 PM	444.32	212.23	447.6457	-0.00113%	0.00471%	-0.01883%

Table 3A Sample of return calculation 1 minute latency June 2023

TimeStamp	AAPL	MSFT	NVDA	R-AAPL	R-MSFT	R-NVDA
6/23/2023 19:55	186.48	334.6501	420.6	0.00000	0.00000	0.00000
6/23/2023 19:54	186.48	334.65	420.6	0.00000	0.00000	0.00000
6/23/2023 19:53	186.48	334.65	420.6	0.00000	0.00000	0.00012
6/23/2023 19:52	186.48	334.65	420.55	0.00000	0.00000	-0.00002
6/23/2023 19:51	186.48	334.65	420.56	0.00000	0.00000	-0.00010
6/23/2023 19:49	186.48	334.65	420.6	0.00011	0.00000	-0.00012
6/23/2023 19:48	186.46	334.65	420.65	-0.00011	0.00000	0.00000
6/23/2023 19:47	186.48	334.65	420.65	0.00000	0.00000	0.00021
6/23/2023 19:45	186.48	334.65	420.56	0.00011	0.00000	0.00014
6/23/2023 19:44	186.46	334.65	420.5	0.00000	0.00000	-0.00005
6/23/2023 19:43	186.46	334.65	420.52	0.00000	0.00000	-0.00021
6/23/2023 19:42	186.46	334.65	420.61	0.00000	0.00000	-0.00010
6/23/2023 19:41	186.46	334.65	420.65	0.00000	0.00000	-0.00012
6/23/2023 19:40	186.46	334.65	420.7	0.00000	0.00000	-0.00012
6/23/2023 19:37	186.46	334.65	420.75	0.00000	0.00000	0.00000
6/23/2023 19:36	186.46	334.65	420.75	0.00000	-0.00024	-0.00024
6/23/2023 19:34	186.46	334.73	420.8499	0.00000	0.00000	-0.00017
6/23/2023 19:33	186.46	334.73	420.9199	-0.00027	0.00000	0.00028
6/23/2023 19:31	186.51	334.73	420.8	0.00000	0.00009	-0.00019
6/23/2023 19:30	186.51	334.7	420.88	0.00000	0.00000	-0.00014
6/23/2023 19:29	186.51	334.7	420.94	-0.00005	0.00000	0.00012
6/23/2023 19:28	186.52	334.7	420.89	0.00011	0.00000	-0.00002
6/23/2023 19:26	186.5	334.7	420.9	0.00000	0.00000	0.00000
6/23/2023 19:25	186.5	334.7	420.9	0.00000	0.00000	0.00005
6/23/2023 19:23	186.5	334.7	420.88	0.00011	0.00000	-0.00010
6/23/2023 19:22	186.48	334.7	420.92	0.00000	0.00000	0.00010
6/23/2023 19:18	186.48	334.7	420.88	0.00005	0.00000	0.00000
6/23/2023 19:17	186.47	334.7	420.88	0.00000	0.00000	0.00010
6/23/2023 19:15	186.47	334.7	420.84	-0.00016	0.00000	0.00012
6/23/2023 19:10	186.5	334.7	420.79	0.00000	0.00000	0.00010
6/23/2023 19:09	186.5	334.7	420.75	0.00000	0.00000	-0.00026
6/23/2023 19:07	186.5	334.7	420.86	0.00000	0.00000	0.00000
6/23/2023 19:06	186.5	334.7	420.86	0.00000	0.00000	0.00000
6/23/2023 19:03	186.5	334.7	420.86	0.00000	0.00000	0.00026
6/23/2023 19:02	186.5	334.7	420.75	0.00011	0.00000	0.00017
6/23/2023 19:00	186.48	334.7	420.68	-0.00005	0.00000	-0.00017
6/23/2023 18:55	186.49	334.7	420.75	0.00000	0.00000	0.00017
6/23/2023 18:54	186.49	334.7	420.68	0.00000	0.00000	-0.00029
6/23/2023 18:49	186.49	334.7	420.8	0.00000	0.00000	0.00012
6/23/2023 18:46	186.49	334.7	420.75	0.00000	-0.00015	-0.00014
6/23/2023 18:45	186.49	334.75	420.81	0.00000	0.00000	0.00026
6/23/2023 18:43	186.49	334.75	420.7	0.00000	0.00000	-0.00002
6/23/2023 18:42	186.49	334.75	420.71	0.00000	0.00000	-0.00005
6/23/2023 18:41	186.49	334.75	420.73	0.00000	0.00000	-0.00019
6/23/2023 18:40	186.49	334.75	420.81	0.00000	-0.00081	0.00000
6/23/2023 18:39	186.49	335.02	420.81	0.00000	0.00000	-0.00021
6/23/2023 18:38	186.49	335.02	420.9	0.00000	0.00000	0.00011
6/23/2023 18:37	186.49	335.02	420.8555	-0.00005	0.00000	-0.00034
6/23/2023 18:35	186.5	335.02	420.9999	0.00000	0.00000	0.00000

Table 4A Sample of EMA calculation 3 seconds latency May 2022

TimeStamp	TimeStamp	AAPL	EMA
13-May-22	3:59:57 PM	147.07	147.0162
13-May-22	3:59:54 PM	147.07	147.0175
13-May-22	3:59:51 PM	147.07	147.019
13-May-22	3:59:48 PM	147.0281	147.0203
13-May-22	3:59:45 PM	146.97	147.0216
13-May-22	3:59:42 PM	146.955	147.0232
13-May-22	3:59:39 PM	146.94	147.0251
13-May-22	3:59:36 PM	146.95	147.0269
13-May-22	3:59:33 PM	146.93	147.0286
13-May-22	3:59:30 PM	146.98	147.0305
13-May-22	3:59:27 PM	146.94	147.0323
13-May-22	3:59:24 PM	146.94	147.0341
13-May-22	3:59:21 PM	146.93	147.0357
13-May-22	3:59:18 PM	147	147.0373
13-May-22	3:59:15 PM	146.99	147.0388
13-May-22	3:59:12 PM	146.9977	147.0402
13-May-22	3:59:09 PM	146.94	147.0414
13-May-22	3:59:06 PM	146.94	147.0425
13-May-22	3:59:02 PM	146.97	147.0436
13-May-22	3:58:59 PM	146.9445	147.0448
13-May-22	3:58:56 PM	146.96	147.0461
13-May-22	3:58:53 PM	146.96	147.0475
13-May-22	3:58:50 PM	146.955	147.049
13-May-22	3:58:47 PM	146.97	147.0508
13-May-22	3:58:44 PM	146.965	147.0529
13-May-22	3:58:41 PM	146.977	147.0552
13-May-22	3:58:38 PM	146.96	147.0575
13-May-22	3:58:35 PM	146.9437	147.0596
13-May-22	3:58:32 PM	146.9437	147.0617
13-May-22	3:58:29 PM	146.92	147.0635
13-May-22	3:58:26 PM	146.895	147.0652
13-May-22	3:58:23 PM	146.8778	147.0667
13-May-22	3:58:20 PM	146.8745	147.0683
13-May-22	3:58:17 PM	146.84	147.0698
13-May-22	3:58:14 PM	146.84	147.0711
13-May-22	3:58:11 PM	146.78	147.0725
13-May-22	3:58:08 PM	146.765	147.0737
13-May-22	3:58:05 PM	146.82	147.0747
13-May-22	3:58:02 PM	146.8106	147.0761
13-May-22	3:57:59 PM	146.87	147.0774
13-May-22	3:57:56 PM	146.878	147.0786
13-May-22	3:57:53 PM	146.84	147.0798
13-May-22	3:57:50 PM	146.8098	147.0812
13-May-22	3:57:47 PM	146.82	147.0827
13-May-22	3:57:44 PM	146.79	147.0839
13-May-22	3:57:41 PM	146.775	147.0851
13-May-22	3:57:38 PM	146.785	147.0865
13-May-22	3:57:35 PM	146.8	147.0878
13-May-22	3:57:32 PM	146.77	147.0891

Table Sample of EMA calculation 3 seconds latency May 2022

TimeStamp	MSFT	EMA
3:59:57 PM	260.99	260.9536
3:59:54 PM	260.99	260.9553
3:59:51 PM	260.97	260.9572
3:59:48 PM	260.92	260.9592
3:59:45 PM	260.84	260.9607
3:59:42 PM	260.81	260.9617
3:59:39 PM	260.79	260.9629
3:59:36 PM	260.815	260.9638
3:59:33 PM	260.755	260.9649
3:59:30 PM	260.83	260.9656
3:59:27 PM	260.89	260.9662
3:59:24 PM	260.82	260.9662
3:59:21 PM	260.77	260.9661
3:59:18 PM	260.91	260.9664
3:59:15 PM	260.87	260.967
3:59:12 PM	260.7802	260.9678
3:59:09 PM	260.68	260.9689
3:59:06 PM	260.7	260.9698
3:59:02 PM	260.725	260.9707
3:58:59 PM	260.675	260.9717
3:58:56 PM	260.74	260.973
3:58:53 PM	260.67	260.9741
3:58:50 PM	260.68	260.9754
3:58:47 PM	260.68	260.9769
3:58:44 PM	260.68	260.9784
3:58:41 PM	260.66	260.9797
3:58:38 PM	260.66	260.9804
3:58:35 PM	260.67	260.981
3:58:32 PM	260.67	260.9817
3:58:29 PM	260.65	260.9824
3:58:26 PM	260.58	260.9832
3:58:23 PM	260.57	260.9841
3:58:20 PM	260.59	260.9854
3:58:17 PM	260.485	260.9866
3:58:14 PM	260.5	260.9884
3:58:11 PM	260.495	260.99
3:58:08 PM	260.46	260.992
3:58:05 PM	260.58	260.9943
3:58:02 PM	260.5255	260.9968
3:57:59 PM	260.6199	260.9992
3:57:56 PM	260.65	261.002
3:57:53 PM	260.62	261.0048
3:57:50 PM	260.575	261.0075
3:57:47 PM	260.58	261.0104
3:57:44 PM	260.52	261.0132
3:57:41 PM	260.51	261.016
3:57:38 PM	260.55	261.0181
3:57:35 PM	260.54	261.0205
3:57:32 PM	260.485	261.0224

Table 4A Sample of EMA calculation 3 seconds latency May 2022

TimeStamp	NVDA	EMA
3:59:57 PM	176.98	177.5488
3:59:54 PM	176.98	177.5512
3:59:51 PM	176.98	177.5539
3:59:48 PM	176.9	177.5565
3:59:45 PM	176.8176	177.5584
3:59:42 PM	176.76	177.5599
3:59:39 PM	176.77	177.5614
3:59:36 PM	176.8	177.5623
3:59:33 PM	176.79	177.5633
3:59:30 PM	176.78	177.564
3:59:27 PM	176.8	177.5646
3:59:24 PM	176.8	177.5646
3:59:21 PM	176.71	177.5642
3:59:18 PM	176.73	177.5647
3:59:15 PM	176.63	177.5653
3:59:12 PM	176.625	177.5661
3:59:09 PM	176.605	177.5673
3:59:06 PM	176.59	177.5685
3:59:02 PM	176.62	177.5697
3:58:59 PM	176.62	177.5707
3:58:56 PM	176.6275	177.5718
3:58:53 PM	176.66	177.5732
3:58:50 PM	176.66	177.5746
3:58:47 PM	176.65	177.5756
3:58:44 PM	176.66	177.5768
3:58:41 PM	176.685	177.578
3:58:38 PM	176.6401	177.5786
3:58:35 PM	176.6399	177.5791
3:58:32 PM	176.6399	177.5796
3:58:29 PM	176.59	177.5799
3:58:26 PM	176.59	177.5804
3:58:23 PM	176.52	177.5812
3:58:20 PM	176.52	177.5825
3:58:17 PM	176.43	177.5831
3:58:14 PM	176.4272	177.5843
3:58:11 PM	176.37	177.5854
3:58:08 PM	176.415	177.587
3:58:05 PM	176.52	177.5882
3:58:02 PM	176.495	177.5896
3:57:59 PM	176.57	177.591
3:57:56 PM	176.57	177.5928
3:57:53 PM	176.5401	177.5948
3:57:50 PM	176.515	177.5966
3:57:47 PM	176.52	177.5985
3:57:44 PM	176.5	177.6004
3:57:41 PM	176.4655	177.6027
3:57:38 PM	176.47	177.6052
3:57:35 PM	176.5	177.6079
3:57:32 PM	176.5072	177.6103

Table 5A Sample of 20-minute EMA calculation over 1 minute latency June 2023

TimeStamp	Close	EMA
6/23/2023 19:50	186.47	186.4803
6/23/2023 19:49	186.47	186.4814
6/23/2023 19:48	186.46	186.4826
6/23/2023 19:46	186.5	186.485
6/23/2023 19:45	186.48	186.4834
6/23/2023 19:35	186.5	186.4838
6/23/2023 19:34	186.46	186.4821
6/23/2023 19:33	186.46	186.4844
6/23/2023 19:32	186.46	186.487
6/23/2023 19:29	186.51	186.4898
6/23/2023 19:28	186.52	186.4877
6/23/2023 19:27	186.46	186.4843
6/23/2023 19:26	186.5	186.4868
6/23/2023 19:23	186.5	186.4854
6/23/2023 19:18	186.48	186.4839
6/23/2023 19:16	186.47	186.4843
6/23/2023 19:15	186.4799	186.4858
6/23/2023 19:12	186.47	186.4864
6/23/2023 19:05	186.47	186.4882
6/23/2023 19:02	186.48	186.4901
6/23/2023 19:01	186.49	186.4911
6/23/2023 19:00	186.48	186.4912
6/23/2023 18:59	186.48	186.4924
6/23/2023 18:54	186.49	186.4937
6/23/2023 18:52	186.47	186.4941
6/23/2023 18:51	186.47	186.4967
6/23/2023 18:48	186.4999	186.4995
6/23/2023 18:44	186.5	186.4994
6/23/2023 18:43	186.49	186.4994
6/23/2023 18:42	186.47	186.5003
6/23/2023 18:37	186.49	186.5035
6/23/2023 18:36	186.49	186.505
6/23/2023 18:35	186.49	186.5065
6/23/2023 18:32	186.49	186.5083
6/23/2023 18:31	186.5001	186.5102
6/23/2023 18:30	186.5	186.5113
6/23/2023 18:29	186.49	186.5125
6/23/2023 18:28	186.49	186.5148
6/23/2023 18:27	186.49	186.5174
6/23/2023 18:24	186.52	186.5203
6/23/2023 18:23	186.52	186.5204
6/23/2023 18:21	186.55	186.5204
6/23/2023 18:19	186.48	186.5173
6/23/2023 18:18	186.5	186.5212
6/23/2023 18:16	186.5	186.5234
6/23/2023 18:11	186.5001	186.5259
6/23/2023 18:09	186.51	186.5286
6/23/2023 18:07	186.5499	186.5306
6/23/2023 18:04	186.51	186.5285

Table 6A Sample of 40-minute EMA calculation over 1 minute latency June 2023

TimeStamp	Close	EMA
6/23/2023 19:50	186.47	186.4875
6/23/2023 19:49	186.47	186.4884
6/23/2023 19:48	186.46	186.4894
6/23/2023 19:46	186.5	186.4909
6/23/2023 19:45	186.48	186.4904
6/23/2023 19:35	186.5	186.4909
6/23/2023 19:34	186.46	186.4905
6/23/2023 19:33	186.46	186.492
6/23/2023 19:32	186.46	186.4937
6/23/2023 19:29	186.51	186.4954
6/23/2023 19:28	186.52	186.4947
6/23/2023 19:27	186.46	186.4934
6/23/2023 19:26	186.5	186.4951
6/23/2023 19:23	186.5	186.4948
6/23/2023 19:18	186.48	186.4946
6/23/2023 19:16	186.47	186.4953
6/23/2023 19:15	186.4799	186.4966
6/23/2023 19:12	186.47	186.4975
6/23/2023 19:05	186.47	186.4989
6/23/2023 19:02	186.48	186.5004
6/23/2023 19:01	186.49	186.5014
6/23/2023 19:00	186.48	186.502
6/23/2023 18:59	186.48	186.5031
6/23/2023 18:54	186.49	186.5043
6/23/2023 18:52	186.47	186.505
6/23/2023 18:51	186.47	186.5068
6/23/2023 18:48	186.4999	186.5087
6/23/2023 18:44	186.5	186.5092
6/23/2023 18:43	186.49	186.5096
6/23/2023 18:42	186.47	186.5106
6/23/2023 18:37	186.49	186.5127
6/23/2023 18:36	186.49	186.5139
6/23/2023 18:35	186.49	186.5151
6/23/2023 18:32	186.49	186.5164
6/23/2023 18:31	186.5001	186.5178
6/23/2023 18:30	186.5	186.5187
6/23/2023 18:29	186.49	186.5196
6/23/2023 18:28	186.49	186.5211
6/23/2023 18:27	186.49	186.5227
6/23/2023 18:24	186.52	186.5244
6/23/2023 18:23	186.52	186.5246
6/23/2023 18:21	186.55	186.5249
6/23/2023 18:19	186.48	186.5236
6/23/2023 18:18	186.5	186.5258
6/23/2023 18:16	186.5	186.5271
6/23/2023 18:11	186.5001	186.5285
6/23/2023 18:09	186.51	186.53
6/23/2023 18:07	186.5499	186.531
6/23/2023 18:04	186.51	186.5301

Annex B – Python Data

Table 1B Insert of historical data and variables

```
import pandas as pd
import matplotlib.pyplot as plt

url = 'https://docs.google.com/spreadsheets/d/1mhoAxN0zES4YMbW6pF4GNXF3OHDUdarc/edit?usp=drive_link&ouid=108155636531747605986&rtpof=true&sd=true'
online_url ='https://drive.google.com/uc?id=' + url.split('/')[-2]
excel_url = 'the 1 Min data.xlsx'

# First step
data_csv = pd.read_excel(online_url)
data = pd.DataFrame(data_csv, columns=['TimeStamp', 'Close', 'Bid', 'Ask'])

marginBA = float(input("Insert the margin for the bid and ask: "))

for j in range((data.shape[0]-1), -1, -1):
  data.iloc[j,2] = data.iloc[j,1] * (1 - marginBA)
  data.iloc[j,3] = data.iloc[j,1] * (1 + marginBA)
```

```
Insert the margin for the bid and ask: 0.006
```

Table 2B Setting parameters

```
import math
import numpy as np
import matplotlib.pyplot as plt
import random
from datetime import datetime


#Parameters for mid price simulation:

T = 1.0  #time
sigma = 2 #volatility
M = 110 #number of time steps
dt = T/M #time step
Sim = 120  #number of simulations
gamma = 0.1 #risk aversion
k = 1.5
A = 140
I = 1
current_date = datetime.now()
position = data.shape[0]-1

#Results:

AverageSpread = []
Profit = []
Std = []
```

```python
for i in range(1, Sim+1):

    S = np.zeros((M+1,I))
    Bid = np.zeros((M+1,I))
    Ask = np.zeros((M+1,I))
    BidOfferPrice = np.zeros((M+1,I))
    AskOfferPrice = np.zeros((M+1,I))
    ReservPrice = np.zeros((M+1,I))
    spread = np.zeros((M+1,I))
    q = np.zeros((M+1,I))
    w = np.zeros((M+1,I))
    equity = np.zeros((M+1,I))
    reserve_relation = np.zeros((M+1,I))

    q[0] = 0 #position
    w[0] = 0 #wealth
    equity[0] = 0

    #while(current_date.date() == data.iloc[position,0].date()):
      #position -= 1

    #current_date = data.iloc[position,0]

    for t in range(0, M+1):
```

```python
        #if(current_date.date() == data.iloc[position - t, 0].date()):
        S[t] = data.iloc[position - t, 1]
        Bid[t] = data.iloc[position - t, 2]
        Ask[t] = data.iloc[position - t, 3]
        ReservPrice[t] = S[t] - q[t-1] * gamma * (sigma ** 2) * (T - t/float(M))
        spread[t] = (gamma * (sigma **2) * (T - t/float(M))) + (2/gamma) * math.log(1 + (gamma/k))
        BidOfferPrice[t] = ReservPrice[t] - spread[t]/2.
        AskOfferPrice[t] = ReservPrice[t] + spread[t]/2.

        if Bid[t] > BidOfferPrice[t] and Ask[t] < AskOfferPrice[t] :
            q[t] = q[t-1] + 1
            w[t] = w[t-1] - Bid[t]

        if Bid[t] < BidOfferPrice[t] and Ask[t] > AskOfferPrice[t] :
            q[t] = q[t-1] - 1
            w[t] = w[t-1] + Ask[t]
        if Bid[t] < BidOfferPrice[t] and Ask[t] < AskOfferPrice[t] :
            q[t] = q[t-1]
            w[t] = w[t-1]
        if Bid[t] > BidOfferPrice[t] and Ask[t] > AskOfferPrice[t]:
            q[t] = q[t-1]
            w[t] = w[t-1] - Bid[t]
            w[t] = w[t] + Ask[t]

        equity[t] = w[t] + q[t] * S[t]
```

```
    position -= M
    AverageSpread.append(spread.mean())
    Profit.append(equity[-1])
    Std.append(equity[-1])
```

```
print ("                   Results               ")
print ("----------------------------------------")

print ("%14s %21s" % ('statistic', 'value'))
print (40 * "-")
print ("%14s %20.5f" % ("Average spread :", np.array(AverageSpread).mean()))
print ("%16s %20.5f" % ("Profit :", np.array(Profit).mean()))
print ("%16s %20.5f" % ("Std(Profit) :", np.array(Std).std()))



#Plots:

x = np.linspace(0., T, num= M + 1)

fig=plt.figure(figsize=(10,8))
plt.subplot(2,1,1) # number of rows, number of  columns, number of the subplot
plt.plot(x,S[:], lw = 1., label = 'S')
plt.plot(x,Ask[:], lw = 1., label = 'Ask')
plt.plot(x,Bid[:], lw = 1., label = 'Bid')
plt.plot(x,BidOfferPrice[:], lw = 1., label = 'BidOP')
plt.plot(x,AskOfferPrice[:], lw = 1., label = 'AskOP')
plt.plot(x,ReservPrice[:], lw = 1., label = 'ReservPrice')
plt.grid(True)
plt.legend(loc=0)
plt.ylabel('P')
plt.title('Prices')
plt.subplot(2,1,2)
plt.plot(x,q[:], 'g', lw = 1., label = 'q') #plot 2 lines
plt.grid(True)
```

```python
plt.legend(loc=0)
plt.axis('tight')
plt.xlabel('Time')
plt.ylabel('Position')


#Histogram of profit:

plt.figure(figsize = (7,5))
plt.hist(np.array(Profit), label = ['Inventory strategy'], bins = 100)
plt.legend(loc = 0)
plt.grid(True)
plt.xlabel('pnl')
plt.ylabel('number of values')
plt.title('Histogram')


#PNL:

plt.figure(figsize = (7,5))
plt.plot(np.array(equity), label = 'Inventory strategy')
plt.legend(loc = 0)
plt.grid(True)
plt.xlabel('pnl')
plt.ylabel('number of values')
plt.title('Profit')
```