Designing a Market-Making Algorithm: A Predictive and Risk-Aware Approach

Luca Nicoli

February 2025

In this document, I present the design and analysis of a market-making algorithm tailored to efficiently manage liquidity provision in a trading environment. Leveraging a provided dataset of order book and trade data for a specific token, I explore a potential approach for market-making:

- **Market-Making Around a Predictive Fair Value:** This approach focuses on determining an optimal fair value for the token and providing bid and ask quotes around this value. The predictive fair value is derived based on an analysis of market dynamics, allowing for competitive yet efficient liquidity provision.

The document outlines the following components:

- The rationale behind the selected algorithm, considering its applicability and advantages in the given market context.
- The mathematical foundation and key assumptions underlying the algorithm.
- A comprehensive explanation of how the algorithm processes order book and trade data to generate quotes dynamically.
- An evaluation of potential risks and limitations of the approach, along with possible mitigation strategies.
- A detailed description of the expected behavior and output of the algorithm under various market conditions.

This work showcases a systematic approach to market-making, integrating theoretical insights with practical data analysis to propose an algorithm that balances profitability and risk management in a highly dynamic trading environment.

Introduction

A Market Maker (MM) is an entity that provides liquidity to the market by quoting buy and sell prices for an asset (in this case, a token) and profiting from the difference between the bid price (buy) and the ask price (sell). The market maker must balance:

1. Providing liquidity: Posting buy and sell orders at various price levels.
2. Risk management: Avoid situations where orders are not filled, leading to potential losses.
3. Profit from the spread: Making money from the difference between the price at which they buy (bid) and sell (ask).

Our goal is to design an MM that provides quotes around a predictive fair value for an asset. In the model we will use, the MM places *Limit Orders* (LOs) on the *Limit Order Book (LOB)* at strategic distances from the current midprice to maximize profits while managing the likelihood of their orders being executed.
To develop the algorithm, we will first need to derive a *predictive fair value* for the asset, which serves as the baseline for placing our quotes. The model we employ assumes that the market maker is a *risk-neutral trader* with *costless inventory management* and infinite patience and that all relevant information is public. The MM will observe the current LOB and output quotes (i.e., buy and sell limit orders) based on its computed *predictive fair value*. The approach assumes that the predictive fair value will guide the optimal placement of these LOs, ensuring the MM remains competitive.
Let's split the problem into two separate subtasks: evaluating the fair price and formulating the optimal quotes.

Fair price prediction

Several methods can be used to estimate a fair price in real-time trading. In the following, we will introduce the formula of two simple *fair price* estimators.

Mid-Price of the Best Bid and Ask

The mid-price is the average of the best bid and ask prices in the order book:

$$S^{mid} = \frac{\text{Best Bid} + \text{Best Ask}}{2} \tag{1}$$

Each of these quantities is evaluated at a specific time T. The mid-price reflects an actionable price by capturing the center of market activity. However, in thin or volatile markets, wide spreads can make the mid-price less reliable. Thus, it is best suited for liquid markets with tight spreads.

Weighted Mid-Price (Depth-Weighted)

This method takes into account both the best bid and ask prices, weighting them by the available depth (size) at each level:

$$S^{wmid} = \frac{(\text{Bid Price} \times \text{Bid Size}) + (\text{Ask Price} \times \text{Ask Size})}{\text{Bid Size} + \text{Ask Size}} \tag{2}$$

Each of these quantities is evaluated at a specific time T. The weighted mid-price adjusts for order book imbalances, providing a fairer estimate in uneven markets.
However, it can be affected by large orders or price manipulation.
Thus, it is useful for markets with frequent imbalances on one side of the order book.

For liquid markets, these two approaches are usually sufficient. In the following, we will employ the Weighted Mid-Price approach since if the market were perfectly balanced, it would converge to the mid-price estimate

Optimal quotes estimate

The problem of estimating the optimal quotes is of vital relevance for the MM. Indeed, Garman (1976) demonstrated that market makers require a bid-ask spread to remain viable by managing inherent imbalances between buy and sell orders.[1] These imbalances arise from varying trader needs and preferences, such as risk tolerance and market access, which create temporary disparities in order flow.

The need of MM's quotes spread

In Garman's model, the **market maker** acts as a monopolist, setting **bid and ask prices** to receive and process orders while aiming to maximize profit and avoid running out of cash or inventory—a condition that would cause them to fail. This model uses the **Gambler's Ruin Problem** to quantify the likelihood of a market maker going bankrupt:[2]

$$\Pr_{failure} = \left(\frac{\Pr_{loss} \cdot \text{Loss}}{\Pr_{gain} \cdot \text{Gain}}\right)^{\text{Initial Welth}} \tag{3}$$

Here, the market maker risks failure if their **probability of losing inventory or cash** exceeds their **probability of gaining** it, which depends on the relative probabilities of buy and sell order arrivals, denoted as λ_a for buys and λ_b for sells. Calling p_a , and p_b the ask and bid prices respectively, the Gambler's Ruin Problem can be expressed by the following equations:

$$\lim_{t\to\infty} \Pr_{failure}(t) \approx \mathcal{K} \in (0,1) \text{ , if } \lambda_b > \lambda_a$$
$$= 1 \text{ otherwise}$$

$$\lim_{t\to\infty} \Pr_{failure}(t) \approx \mathcal{J} \in (0,1) \text{ , if } \lambda_a p_a > \lambda_b pb$$
$$= 1 \text{ otherwise}$$

In order for a market maker to remain viable, the conditions from these equations must both be satisfied simultaneously. This requires that the following inequalities hold at the same time:

$$\lambda_b > \lambda_a \quad \text{and} \quad \lambda_a p_a > \lambda_b p_b$$

For these inequalities to be true together, it must always hold that $p_a > p_b$, which establishes the bid-ask spread. Thus, bid-ask spreads are vital conditions for an MM.
However, the problem now is to define the optimal prices the MM should employ to maximize its terminal wealth.

Formal Derivation

Given that market participants operate in a competitive, efficient environment, our MM algorithm will need to continually adjust her quotes based on incoming market data and her estimation of the predictive fair value. This ensures the MM remains competitive in the market while minimizing the risk of unfilled orders, leading to *optimal profitability*. Let's assume that the MM seeks a strategy $(\delta_t^{\pm})_{0\leq t\leq T}$ that maximizes cash at the terminal time T without penalizing uncertainty in her sales directly. At this terminal time, the MM will liquidate her remaining inventory Q_T using a *market order* (MO) at the midprice. Additionally, the MM restricts her inventory within specified bounds $q_{lim}^{\pm}$, ensuring that it does not exceed an upper limit $q_{lim}^{+} > 0$ or fall below a lower limit $q_{lim}^{-} < 0$, where both limits are finite. Penalties on inventory levels are also applied at terminal time through a coefficient $\alpha \geq 0$, representing the liquidity-taking fees and the impact of the MO as it moves through the LOB, and $\phi \geq 0$, the parameter penalizing running inventory levels. With this preamble, the MM seeks to solve this stochastic optimization problem to maximize its cash at terminal date T:[3]

$$H(t,x,S,q) = \sup_{\delta^{\pm}} \mathbb{E}\left[X_T + Q_T^{\delta}(S_T^{\delta} - \alpha Q_T^{\delta}) - \phi \int_t^T (Q_u)^2 du\right] \tag{4}$$

Where, for each time $t \in [0,T]$ we define:

1. S_t is the midprice which is a stochastic variable defined as

$$S_t = S_0 + \sigma W_t$$

   Where $\sigma > 0$, and W_t is a Brownian motion connected to price fluctuations;

2. $\delta_t^{\pm}$ are the depths at which the MM posts LOs (i.e. the spread of its quotes), and are the variables against which the MM needs to optimize;

3. X_t is the MM's cash process satisfying

$$dX_t^{\delta} = (S_t + \delta^+)dN_t^{\delta,+} - (S_t - \delta^-)dN_t^{\delta,-}$$

   Where $N_t^{\delta,\pm}$ denote the counting process for the MM's filled sell (+) and buy (-) LOs;

4. The MM's inventory satisfies

$$Q_t^{\delta} = N_t^{\delta,+} - N_t^{\delta,-}$$

We further assume that other participants buy (+) and sell (-) MOs arrive at Poisson times with rates $\lambda^{\pm}$. Equation 4 is a stochastic control problem and can be solved by employing the dynamic programming principle (DPP) and the related non-linear partial differential equation (PDE) known as the Hamilton-Jacobi-Bellman (HJB) equation also called the dynamic programming equation (DPE). The DPE for this class of problems read:[3]

$$\begin{aligned} 0 =& \partial_t H + \frac{1}{2}\sigma^2 \partial_{SS} H - \phi q^2 \\ &+ \lambda^+ \sup_{\delta^+} \left\{ e^{-\kappa^+ \delta^+} \left(H\left(t, x + \left(S + \delta^+\right), q - 1, S\right) - H \right) \right\} \mathbb{I}_{q > \underline{q}} \\ &+ \lambda^- \sup_{\delta^-} \left\{ e^{-\kappa^- \delta^-} \left(H\left(t, x - \left(S - \delta^-\right), q + 1, S\right) - H \right) \right\} \mathbb{I}_{q < \bar{q}}, \end{aligned} \tag{5}$$

Where $\mathbb{I}$ is the indicator function. Equation 5 it is representing a stochastic diffusion problem with control. Indeed, the first two terms $\partial_t H + \frac{1}{2}\sigma^2 \partial_{SS} H$ are typical of diffusion processes. Additionally, the term ϕq^2 captures the effect of penalizing deviations of inventories from zero along the entire path of the strategy, which represents the inventory cost due to misalignment with the target inventory. In the second line of the DPE, the supremum over δ^+ contains the terms corresponding to the arrival of a market buy order, which is filled by a limit sell order. Similarly, in the last line of the DPE, the supremum over δ^- contains analogous terms for the market sell orders that are filled by limit buy orders. In addition, $\kappa^{\pm}$ is a parameter that adjusts the likelihood of a LO being filled when a MO arrives, according to:

$$\Lambda_t^{\pm} = \lambda^{\pm} e^{-\kappa^{\pm} \delta_t^{\pm}} \tag{6}$$

Given my choice of developing an MM that provides quotes around a predictive fair value, we will make the following assumptions for the MM:

1. It does not impose a penalty on running inventories;
2. It does not incur a terminal inventory penalty (i.e., $\varphi = \alpha = 0$)
3. There are no constraints on the amount of inventory the strategy may accumulate (i.e., $|q_{lim}| \to \infty$);
4. At terminal time T, the inventory is unwound at the midprice.

Given these assumptions, the optimal MM's strategy, i.e. the solution to Eq. 4 becomes:[3]

$$\delta^{+,*}(t,q) = \frac{1}{\kappa^+} \quad , \text{and} \quad \delta^{-,*}(t,q) = \frac{1}{\kappa^-} \tag{7}$$

The strategy in Eq. 7 translates into the MM posting in the LOB, maximizing the probability of the LOs being filled.
However, $\kappa^{\pm}$, which in our case defines the whole MM's strategy, needs to be estimated. Even in this simple case, $\kappa^{\pm}$ is not easy to estimate, and the whole optimal strategy might depend upon several factors, such as the latest trading prices, market volatility, and order imbalances.[4]

MM implementation: *MarketMaker* class

The *MarketMaker* class simulates a market maker's process of generating bid and ask quotes. It takes order book data (bids and asks) and calculates a fair price for the asset using the volume-weighted midpoint of the best bid and ask. Based on this fair value, the class generates bids and asks for quotes by applying a spread, determining the prices at which the market maker is willing to buy and sell. The spread from the *fair price* is defined as

$$\delta^{\pm} = \pm c \cdot \Delta_t^{\pm}, \text{ where } c \in [0, 1] \tag{8}$$

Where $\Delta_t^{\pm}$ are the observed best ask/bid prices from the orderbook at time t. Although I know I shouldn't simulate the strategy, I found a kind of appealing, having only a couple of constants to define.

```python
market_maker.py

import pandas as pd
from typing import Optional

class MarketMaker:


    def calculate_fair_value(self, current_orderbook: pd.Series) -> float:
        """
        Calculate the fair value based on the selected method.

        Args:
            current_orderbook (pd.Series): A row from the orderbook containing bid and ask
                                           data.

        Returns:
            float: The predicted fair value.
        """
        if self.fair_price_method == 'wmid':
            # Weighted mid price (wmid)
            best_bid_price, best_bid_quantity = current_orderbook['bids'][0]
            best_ask_price, best_ask_quantity = current_orderbook['asks'][0]
            fair_value = (
                (best_bid_price * best_bid_quantity) + (best_ask_price * best_ask_quantity)
                ) / (best_bid_quantity + best_ask_quantity) if (best_bid_quantity +
                best_ask_quantity) > 0 else 0
        else:
            raise ValueError(f"Unknown fair price method: {self.fair_price_method}")
        return fair_value

    def generate_quotes(self, fair_value: float) -> dict:
        """
        Generate quotes for the market maker (bid and ask prices) around the fair value.

        Args:
            fair_value (float): The predicted fair value.

        Returns:
            dict: A dictionary with bid and ask quotes.
        """
        # Extract the best bid and ask prices from the orderbook
        best_bid_price = current_orderbook['bids'][0][0]  # The best bid price
        best_ask_price = current_orderbook['asks'][0][0]  # The best ask price
        # Calculate the spreads
        bid_spread = fair_value - best_bid_price
        ask_spread = best_ask_price - fair_value
        # Calculate quotes
        bid_price = fair_value  - (self.spread_pct/100) * bid_spread
        ask_price = fair_value  + (self.spread_pct/100) * ask_spread

        return {'bid': bid_price, 'ask': ask_price}
```

Figure 1: Shortened code snippet of the *MarketMaker* class.

Risks and limitations

While this model provides a structured approach to market-making, there are several limitations:

- **Unrealistic Assumptions**: The model assumes *infinite patience*, *no trading costs*, and *risk neutrality*, which may not reflect real-world conditions where transaction costs, inventory management, and risk considerations are present.
- **Market Conditions**: The model assumes that other market makers' actions are independent of the MM's decisions and that market orders follow a simple, predictable distribution. In practice, market dynamics can be more complex, with *information asymmetry*, sudden market shocks, and other factors that influence trading decisions.
- **Predictive Fair Value Accuracy**: The quality of the quotes depends on the accuracy of the *predictive fair value*. If the predictive model is inaccurate, the MM's quotes may be suboptimal, leading to either missed opportunities or excessive risk exposure.

Thus, my implementation of this model into the *MarketMaker* class described before has several limitations and risks. It assumes that the best bid and ask prices ($\Delta_t^{\pm}$) and their midpoint sufficiently represent the asset's fair value, which may not hold in volatile or illiquid markets where price movements are rapid or order book imbalances occur. The spread calculation, $\delta^{\pm} = \pm c \cdot \Delta_t^{\pm}$, uses a constant c and does not dynamically adapt to changes in volatility, liquidity, or market conditions, potentially resulting in inefficient pricing. Moreover, the model lacks inventory risk management, exposing the market maker to significant risks of accumulating unbalanced positions and adverse price movements. Additionally, the model relies only on the best bid and ask, ignoring order book depth, and does not account for the market maker's impact on prices, which can lead to signaling risks or inefficiencies. Furthermore, external factors such as news, macroeconomic data, or correlated asset prices are not incorporated, potentially undermining the accuracy of the fair value calculation. Dependency on accurate order book data exposes the model to risks from spoofing or manipulation. Key risks include inventory risk, where unbalanced positions due to imbalanced order flows can result in significant losses, and adverse selection risk, where informed traders exploit mispriced quotes, leaving the market maker at a disadvantage. Other risks include liquidity risk in shallow markets, volatility exposure due to rapid price changes, operational risks such as system errors or downtime, and regulatory risks associated with improper market-making behavior. To address these limitations, the model can incorporate dynamic spread adjustments based on market conditions, implement inventory risk management mechanisms, use deeper order book analysis, refine fair value calculations with external data.

Results

We first analyze the behavior of the fair price against the actual trade price (See Fig. 2) throughout the period under analysis.

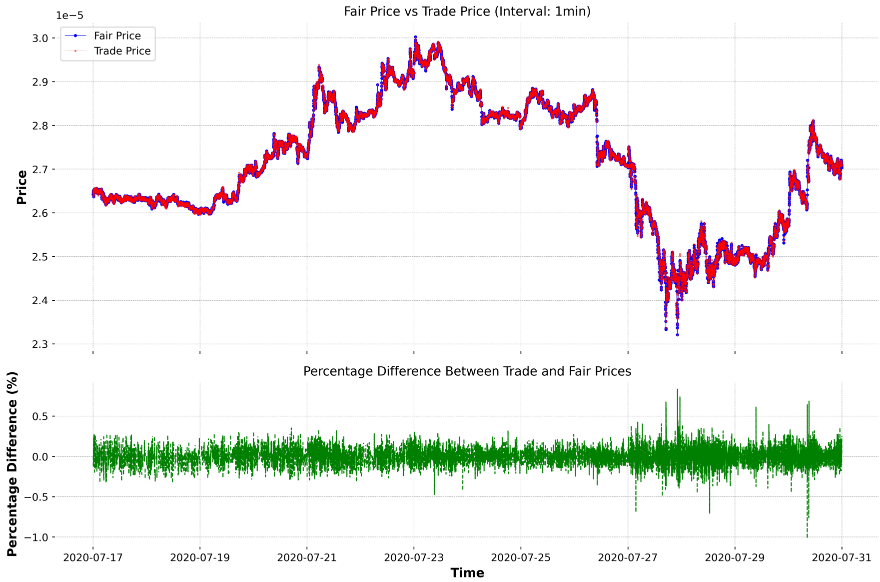

Figure 2: (Top) Fair price (blue) and actual trade price (red) as a function of time. (Bottom) Difference in percentage between the two prices.

As can be noticed, the difference between the prices is bound below 1.02% for all market events. Thus, our fair price is a good approximation of the true trade price.

Let's now analyze the market trend using a candlestick chart (See Fig. 3). After an initial slight downtrend and low-volatility market behavior (small candles and wicks), we observe a clear uptrend from July 19 to 23. This period is characterized by consecutive green candles, signaling strong buying pressure. On July 19, a bullish engulfing pattern initiates this uptrend, as a larger green candle completely engulfs the preceding red candle. The buy/sell volume ratio reaches 1.145 on July 19, suggesting strong buyer dominance. This is further supported by increased bid depth, indicating ample liquidity to sustain the upward movement (See Fig. 4).
A reversal pattern forms around July 23, with the market exhibiting more erratic price action due to low liquidity (evidenced by the long wicks in the candlesticks and the lowest cumulative depths in both the bid and ask orders, see Fig. 4). This lack of depth leads to large price fluctuations, and on July 27, a sharp red candle with long wicks indicates selling pressure. At this point, the buy/sell ratio drops to 0.864, confirming the dominance of sellers. The sudden price drop may be due to external factors such as news or market sentiment shifts, leading to a spike in volatility.
From July 28-29, green candles appear, signaling potential recovery.

An ideal MM should continuously adapt its strategy to the prevailing market conditions. In a low-volatility, down-trending market with small candles and wicks, the MM should focus on providing liquidity with tight bid-ask spreads while minimizing inventory risk by adjusting bids and asks frequently. During a bullish trend characterized by consecutive green candles and strong buying pressure, the MM should narrow the spread to capture increased trading volume while carefully accumulating inventory on the buy side. It is crucial to monitor the buy/sell volume ratio to gauge the strength of the trend and adjust the strategy accordingly. When a reversal occurs, and the market enters a downtrend, the MM should widen the spread to mitigate risk, reduce exposure to falling prices by unwinding positions, and employ hedging strategies to protect against downside risk. Finally, during periods of recovery, the MM should adjust the spread to reflect the renewed buying pressure but remain flexible, watching for signs of continuation or further reversal. By monitoring market indicators like buy/sell ratios, order book depth, and liquidity, the MM can continuously adjust its behavior, offering liquidity when possible while managing risk through dynamic spread adjustments, inventory management, and hedging. In this way, the MM maximizes profitability while minimizing risk across varying market conditions.

Our MM strategy is not ideal. In particular, the spread is set as a fixed proportion of the observed bid/ask spread in the order book ($\delta^{\pm} = \pm c \cdot \Delta_t^{\pm}$, with $c = 50\%$).

In a low-volatility phase, this approach is likely to provide liquidity efficiently, as the MM quotes close to the market spread. However, in a bullish market, the MM may find itself less competitive, as it may not capture trades at higher volumes if the spread narrows quickly. During a downtrend or reversal, the strategy could be less effective, as the MM might not widen its spread fast enough to account for increased volatility, leading to potential exposure to larger price movements. Finally, in a recovery phase, the strategy could perform better as the market stabilizes, but the MM must remain cautious of sudden reversals. Overall, while this strategy can provide liquidity in calm markets, it might not be robust enough in volatile or fast-moving conditions, potentially exposing the MM to risk or missed opportunities if it does not adjust dynamically to market changes.

This is consistent with the observed behavior of the MM in practice (See dotted lines in Fig. 3), where the quotes are consistently tight and close to the asset's closing price, reflecting the rigid spread strategy. However, as noted, this approach does not account for high-volatility periods, during which the MM would need to widen the spread to manage risks more effectively.

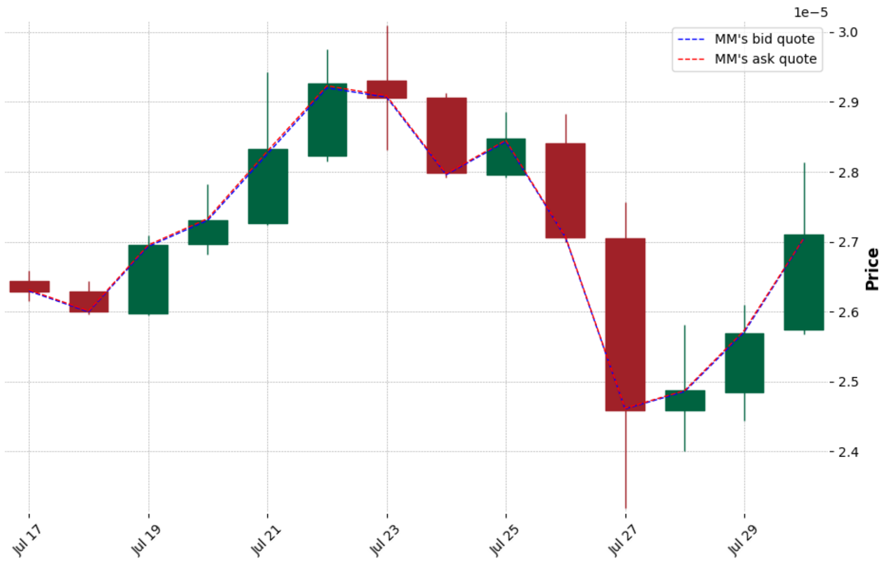

Figure 3: Daily candle stick graph of the price of the asset. The blue and red dotted lines represent the MM's bid and ask quote respectively.

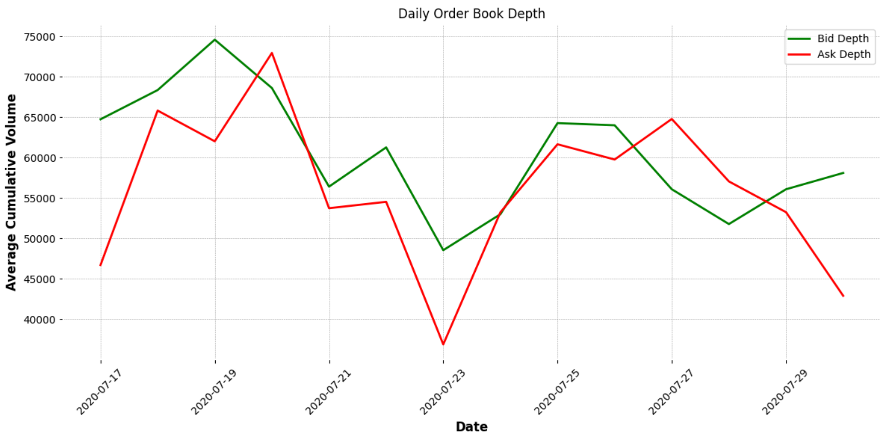

Figure 4: Daily Orderbook depth.

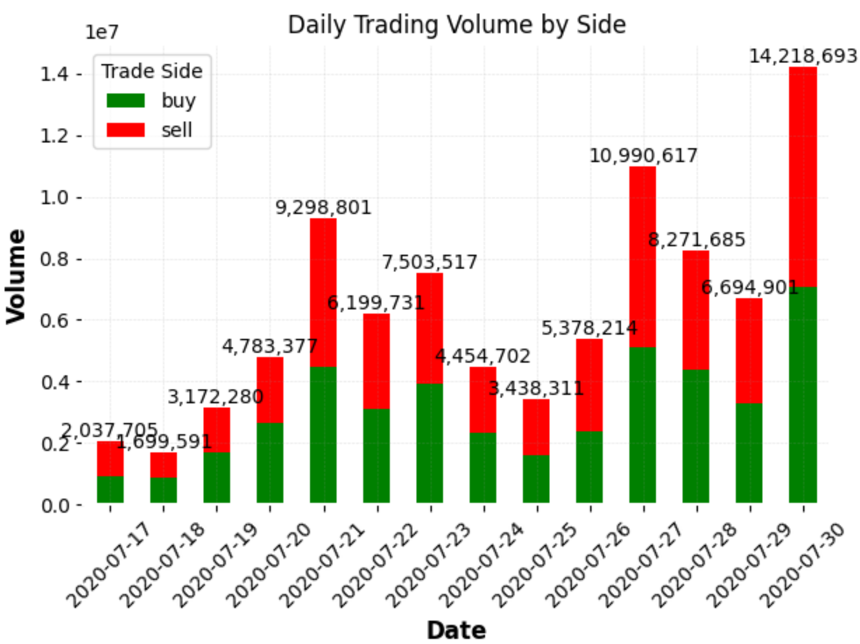

Figure 5: Daily Volume analysis.

Reflection on Machine Learning Models for Strategy Optimization

Although simulating the strategy was not part of the task, I got excited by the potential of using machine learning (ML) techniques to train a market-making strategy. Specifically, one could design a system involving two Long Short-Term Memory (LSTM) models trained on three-quarters of the provided dataset. The first LSTM model aims to simulate the dynamics of the order book, while the second model simulates trade execution within the order book. One could test and evaluate these models on the remaining quarter of the dataset. These models can be used as surrogate oracles to train a reinforcement learning (RL) strategy. In this setup, the MM is treated as a RL agent interacting with the oracle's order book, posting quotes that could potentially be filled by the trades generated by the trade-oracle. The RL agent learns an optimal strategy by minimizing a loss function that is properly defined. This approach enables the MM to adapt to the simulated market environment, refining its strategy over time by learning from trial and error.
Although exciting, at least for me, this methodology does have some limitations. One significant challenge is the complexity of accurately simulating the market dynamics, as real-world order books and trading patterns are influenced by a wide range of factors that might not be captured by the LSTM models. Additionally, the agent's performance may be highly dependent on the quality and representativeness of the training data, which can impact how well the strategy generalizes to unseen market conditions. Furthermore, while the RL agent can adapt to simulated environments, translating this to real-world market conditions might prove difficult, especially if the market experiences drastic changes or unforeseen events that are not part of the training data.
Nevertheless, it was fun thinking about it.

References

[1] Garman, M. B. *J. financ. econ.*, 3(3):257–275, **1976**.

[2] Aldridge, I. *High-frequency trading: a practical guide to algorithmic strategies and trading systems*, volume 604. John Wiley & Sons, **2013**.

[3] Cartea, Á., Jaimungal, S. and Penalva, J. *Algorithmic and high-frequency trading.* Cambridge University Press, **2015**.

[4] Xiong, Y., Yamada, T. and Terano, T. In *2015 Winter Simulation Conference (WSC)*, pages 324–335. IEEE, **2015**.