Statistical Metrics of 100 Scandinavian Blue Chip Stock Across 4 Days

Technique for AWS Dataset Preprocessing Using $C++$

Mak Ho Wai

2024-09-16

Contents

Introduction

This report presents a comprehensive analytical study of statistical metrics derived from a dataset containing four days' worth of tick data for 100 Scandinavian blue-chip stocks. The dataset, sourced from AWS and structured as a large CSV file.

The preprocessing and metric extraction were implemented using C++, with the corresponding code attached at the end of this report (or on my GitHub: `https://github.com/HowaiMak/HFT-100-Scandi-BlueChip`). This study focuses on several critical metrics, including inter-trade times, tick changes, bid-ask spreads, and the round number effect.

By analysing these metrics, the study aims to identify patterns and potential inefficiencies in the market, offering insights relevant to both theoretical research and practical algorithmic trading applications. This task serves to enhance the understanding of the role while addressing real-world analytical challenges commonly encountered in quantitative analysis.

Investigation Outline

Data Overview

The dataset used in this analysis includes several columns that capture various statistical metrics for each stock. To ensure the accuracy of the results, periods with no trading activity are excluded. The columns in the dataset are described as follows:

- **Stock Name**: The stock name for each stock.
- **Mean Time Between Trades**: The average duration, in seconds, between consecutive trades for a specific stock.
- **Median Time Between Trades**: The median value of the time intervals between trades, providing a central tendency measure that is less susceptible to outliers.
- **Mean Time Between Tick Changes**: The average duration, in seconds, between observed changes in tick type for each stock.
- **Median Time Between Tick Changes**: The median interval between tick changes, offering a robust measure of typical market movement.
- **Longest Time Between Trades**: The maximum observed duration between consecutive trades, indicative of periods of low liquidity or market inactivity.
- **Longest Time Between Tick Changes**: The longest recorded interval between tick changes, reflecting periods of market momentum shifts or heightened volatility.
- **Mean Bid-Ask Spread**: The average difference between the bid price (the highest price a buyer is willing to pay) and the ask price (the lowest price a seller will accept), reflecting market liquidity and transaction costs.

- **Median Bid-Ask Spread**: The median bid-ask spread, providing a robust measure of typical spread values, less influenced by extreme fluctuations.

- **Round Number Effect**: A metric designed to capture the influence of round numbers on stock prices, typically reflecting psychological barriers or support levels in market trading behaviour.

Quantitative Techniques

To analyse the dataset, several quantitative techniques are employed to extract meaningful insights:

- **Descriptive Statistics**: Key metrics, including mean, median, and range, are computed for each stock to summarise the dataset. This encompasses measures of central tendency, dispersion, and extreme values to identify patterns, trends, and anomalies.

- **Correlation Analysis**: This technique explores the relationships between different metrics, with particular focus on the interaction between bid-ask spreads and time-based metrics (such as time between trades and tick changes). Correlation coefficients are computed to determine the strength and direction of associations between these variables.

- **Plot Analysis**: Graphical tools such as scatter plots, histograms, and scatter charts are utilised to visually represent data distribution and relationships between variables. These plots assist in detecting trends, clusters, and outliers that may not be immediately evident from descriptive statistics alone.

This methodological approach provides a structured framework for examining the dataset, facilitating a comprehensive understanding of the dynamics of Scandinavian blue-chip stocks, and supporting the development of data-driven trading strategies.

Analysing and Findings

Visualisations of Statistic Metrics

Mean and Median Time between Trade

Histograms are presented below to visualise the distribution of time-related metrics between trade executions.

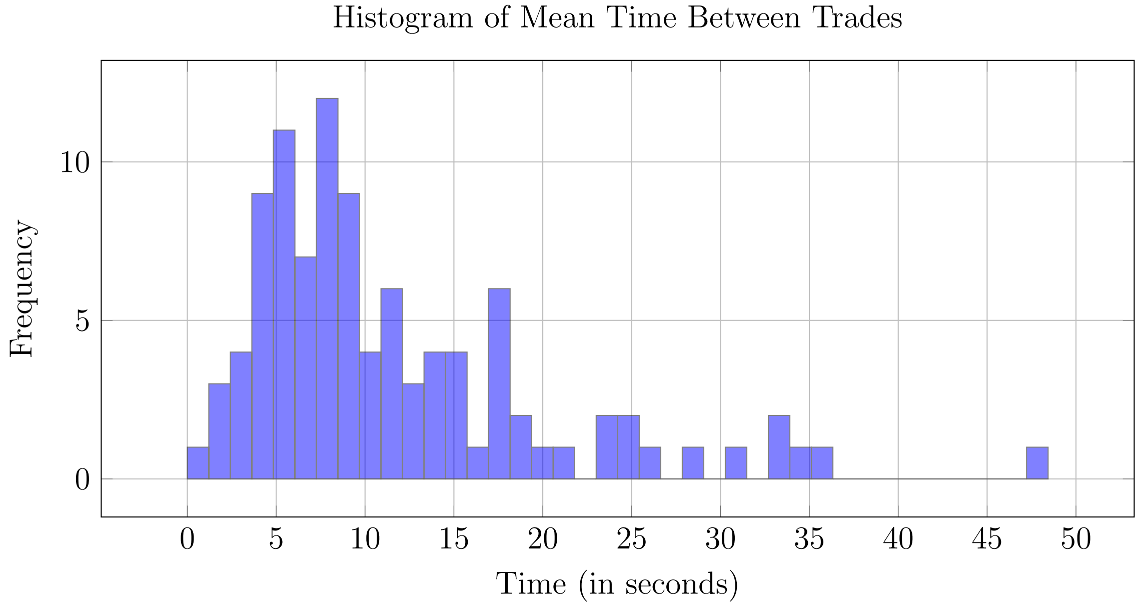

Figure 1: Distribution of Mean Time Between Trades

As illustrated in Figure 1, the distribution of mean inter-trade times for the 100 analyzed stocks exhibits a prominent peak within the range of 4 to 6 seconds, followed by a gradual decline as the inter-trade time increases. This behavior is characteristic of a stochastic process, closely resembling a Poisson distribution, which typically describes the probabilistic nature of discrete event occurrences, such as trade executions.

The concentration of trades within shorter time intervals suggests a high frequency of rapid trades, which is indicative of highly liquid markets. Such a pattern is consistent with the dynamics observed in high-frequency trading (HFT) environments, where market activity is dominated by algorithmic trading strategies executing large volumes of trades in rapid succession.

Notably the median inter-trade time appears to be zero across all stocks, potentially reflecting the limitations in data resolution, particularly with respect to trades that are recorded with identical timestamps. This issue is especially pronounced in HFT settings, where multiple trades may be executed within the same millisecond. Such data limitations pose challenges for both visual representation and statistical modeling of inter-trade times.

Moreover, the influence of occasional longer inter-trade intervals on the mean indicates a skewed distribution. While the majority of trades occur in quick succession, these infrequent yet extended gaps between trades increase the overall mean. Consequently, reliance on the mean as a summary statistic may obscure key aspects of trade intensity.

Mean and Median Time between Tick Changes

When a tick change occurs, it reflects a modification in the price or volume of either buy or sell orders. The time interval between successive tick changes provides valuable insights into market decision-making processes and the underlying volatility dynamics.

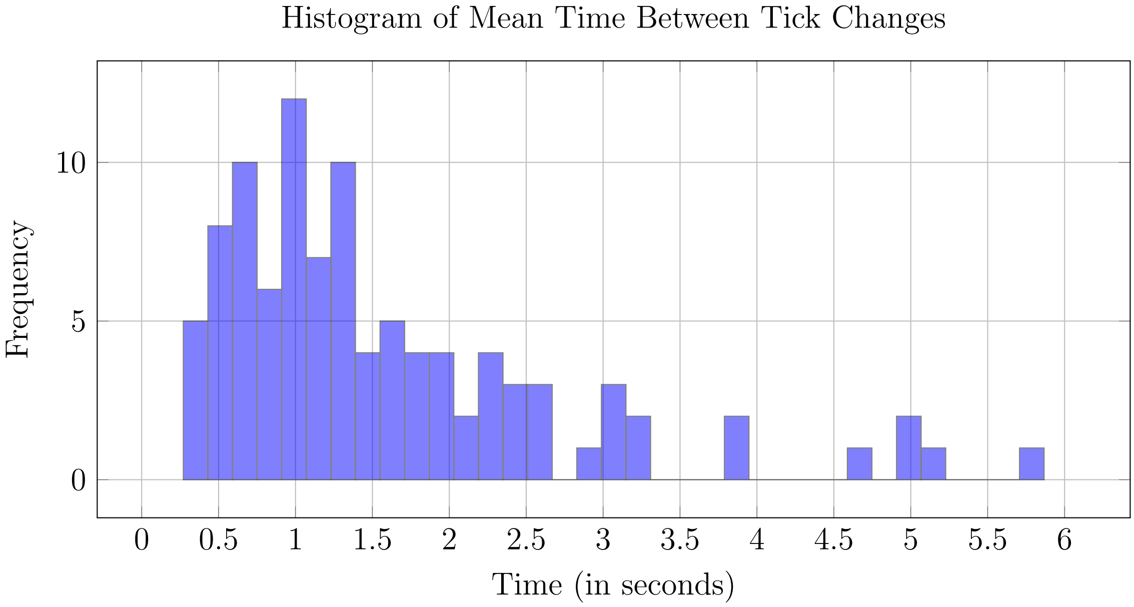

Figure 2: Distribution of Mean Time Between Tick Changes

In Figure 2, the distribution of the mean time between tick changes across the 100 investigated stocks peaks at approximately 1 second, followed by a gradual decline. This pattern resembles the trend observed in Figure 1, though with a generally shorter inter-event time. This indicates that the order book undergoes frequent updates across all observed stocks.

As anticipated, the alignment between the mean time between tick changes and the time between trades suggests a potential correlation between trade frequency and order-book update frequency, a relationship that will be explored in subsequent sections. This dynamic is consistent with the general principle that trading activity is largely driven by rapid buy/sell decisions in response to evolving market conditions.

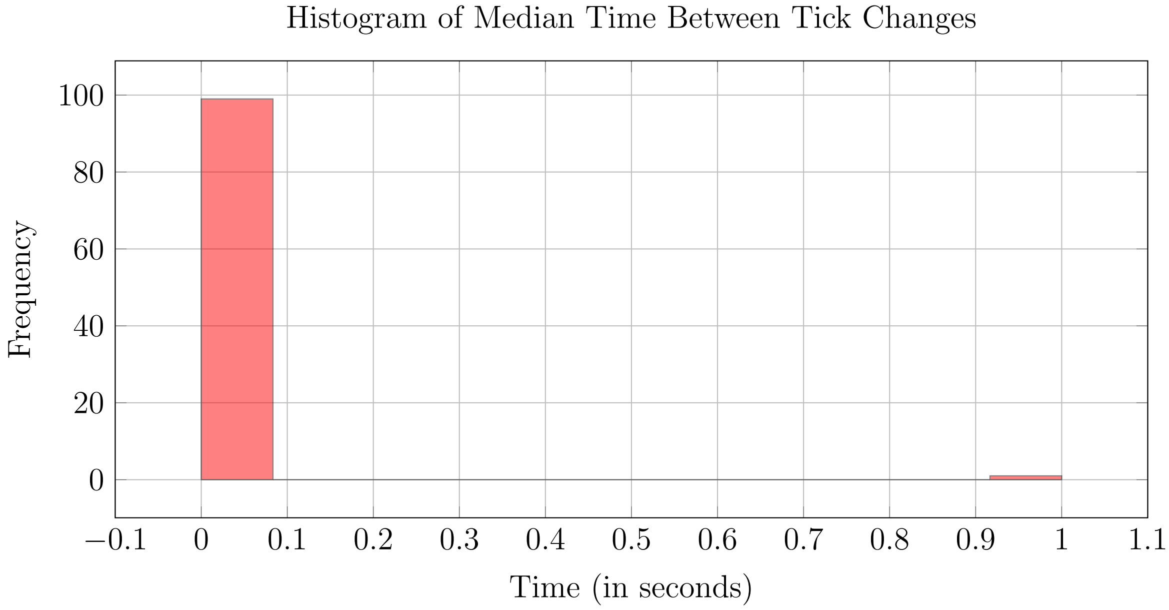

Figure 3: Distribution of Median Time Between Tick Changes

Similar to the distribution of trading times, the median time between tick changes is consistently reported as zero for 99% of the stocks analyzed. Even the longest observed intervals do not exceed 1 second. The prevalence of zero median tick times suggests that a substantial proportion of tick changes occur within the millisecond range or faster.

This observation indicates that the market experiences constant price and volume adjustments, which is characteristic of high-frequency trading (HFT) environments and potentially signals periods of heightened market uncertainty or stress.

Once again, the near-zero median time might reflect limitations in the data resolution, particularly in accurately capturing the precise intervals between tick changes at extremely short timescales. To obtain a more precise understanding of market volatility and dynamics, the use of higher-resolution data may be necessary to better capture the frequency and impact of these high-frequency events.

Longest Time between Trades and between Tick Changes

The two histograms representing the "Longest Time Between Trades" and "Longest Time Between Tick Changes" provide a detailed view of the distribution of outliers.

As discussed earlier, these visualizations highlight the frequency and range of extreme values, offering insights into potential anomalies and the underlying market behavior contributing to these deviations.

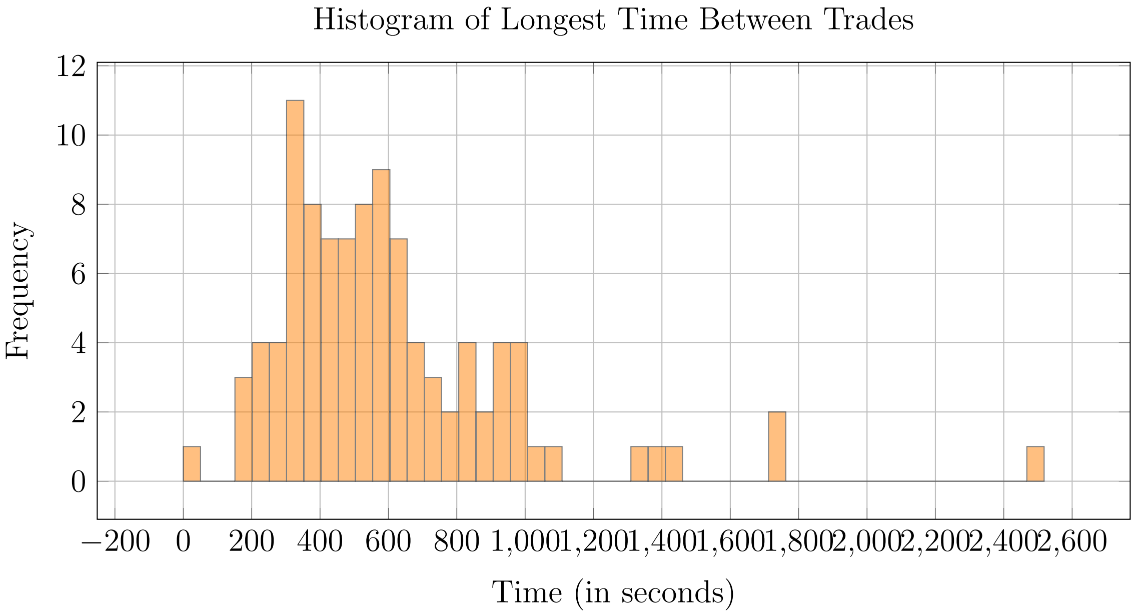

Figure 4: Distribution of Longest Time Between Trades

In Figure 4, the distribution of the longest inter-trade times across 100 stocks shows a concentration predominantly within the 180 to 1000-second range, with a subtle peak in the middle of this interval. These extended gaps between trades are responsible for the inflated mean inter-trade time observed in Figure 1, despite the median being effectively zero. This highlights the impact of extreme values on the overall mean, skewing the distribution towards higher values.

Interestingly, these outliers are likely associated with day traders, whose strategies typically involve holding positions for longer periods within a single trading session. Unlike high-frequency traders (HFTs), who focus on exploiting small, short-term price inefficiencies, day traders aim to capture larger price movements throughout the trading day. The longer inter-trade times observed here likely reflect their more deliberate trading approach, which contrasts with the continuous, rapid trading patterns characteristic of HFT strategies.

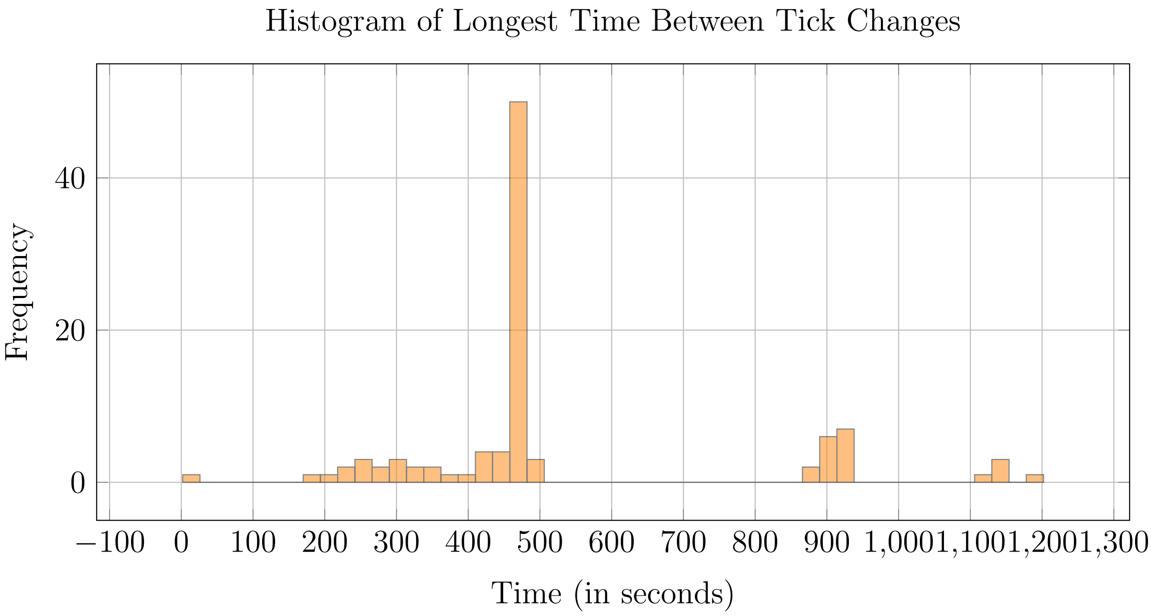

Figure 5: Distribution of Longest Time Between Tick Changes

Figure 5 reveals an unexpected peak at 480 seconds, corresponding to the mid-range of the longest inter-trade times observed in Figure 4. This graph indicates that approximately half of the stocks in the sample experience a maximum time of 480 seconds between order book changes.

The coincidence of this peak with the inter-trade time outliers suggests a potential connection between order book inactivity and prolonged trade intervals. Matching our prediction as previous. This could imply that both are driven by similar market dynamics, such as periods of market consolidation or low liquidity, where fewer trades and order updates occur. Understanding these patterns is critical for accurately modeling market behavior, especially during periods of low volatility or reduced activity.

Mean and Median of Bid-Ask Spread

The bid-ask spread is computed by subtracting the bid price from the ask price across the entire dataset. This measure represents the difference between the lowest price at which sellers are willing to sell (ask price) and the highest price at which buyers are willing to buy (bid price).

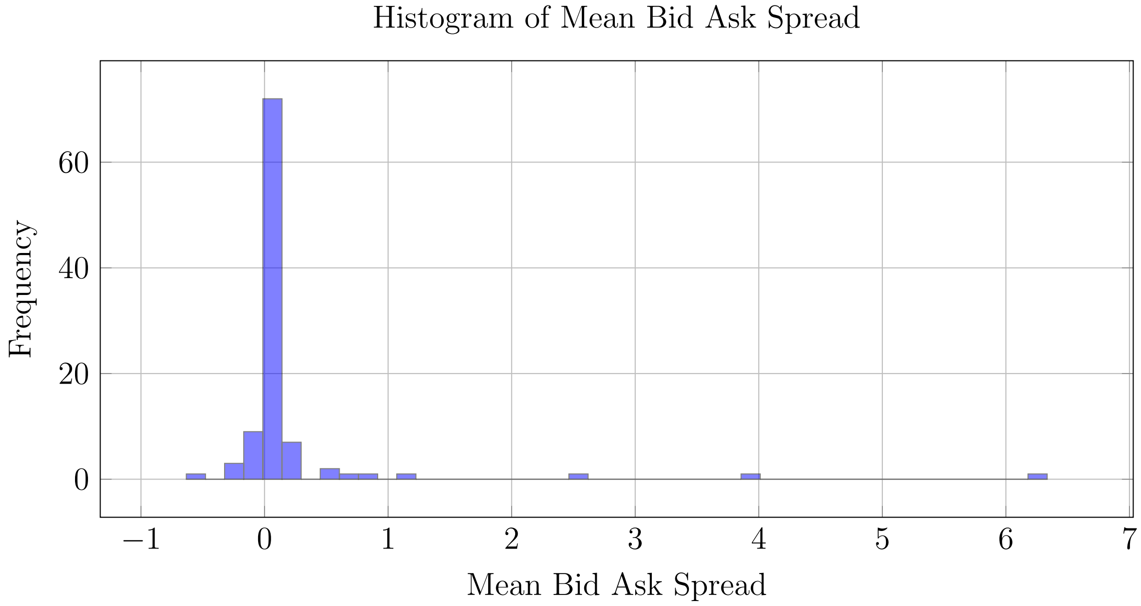

Figure 6: Distribution of Mean Bid Ask Spread

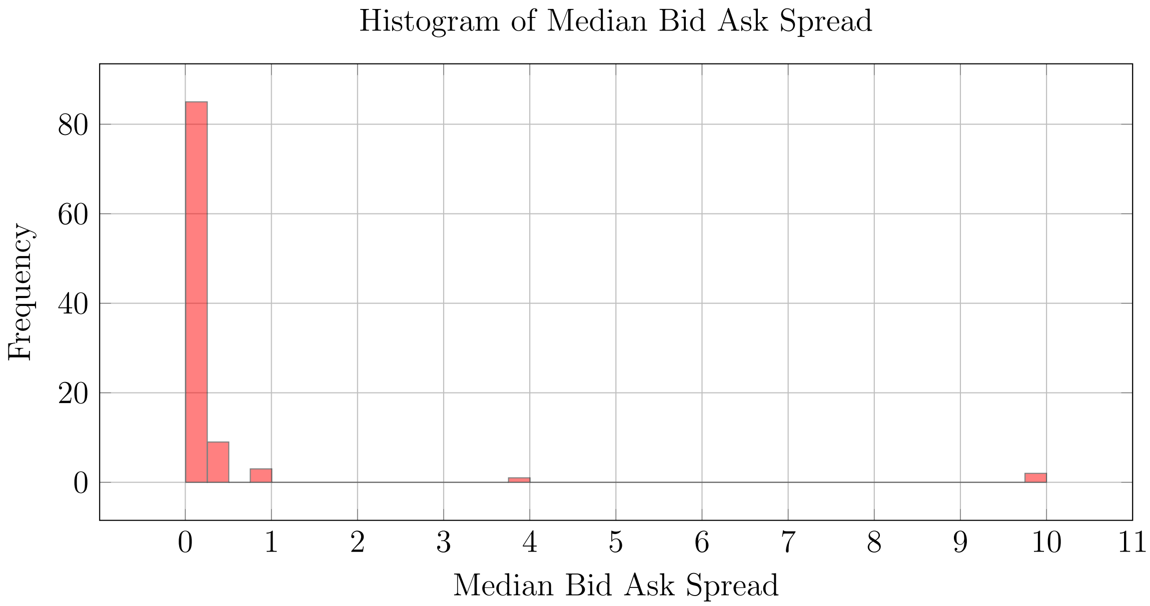

Figure 7: Distribution of Median Bid Ask Spread

Unlike the previous figures that depict the distributions of inter-trade and inter-tick change times, the distributions of both the mean and median bid-ask spread, as shown in Figures 6 and 7, reveal a distinct pattern. The values of the bid-ask spread are either exactly zero or very close to zero on the positive side (e.g., $0.1, $0.2). This suggests that, in most cases, the spread between the best bid and ask prices is minimal, indicating a highly liquid market where buy and sell orders are closely aligned.

Notably, there are instances of a negative mean bid-ask spread, which might reflect data anomalies or irregularities in market conditions, such as crossed markets. Additionally, a small number of cases exhibit higher bid-ask spread values (greater than $1, reaching up to $6 for the mean and up to $10 for the median). These larger spreads may correspond to periods of reduced liquidity or market stress, where the divergence between buy and sell prices widens.

Round Number Effects RNE

The following histograms examine the potential effects of the round number effect (RNE), specifically when the last digit of trade prices is 0, on both trade prices and volumes. These analyses offer valuable insights into market behavior near psychologically significant price levels, where traders may exhibit biases or preferences. By investigating this phenomenon, we can better understand how round numbers might influence trading patterns and decision-making processes in financial markets.

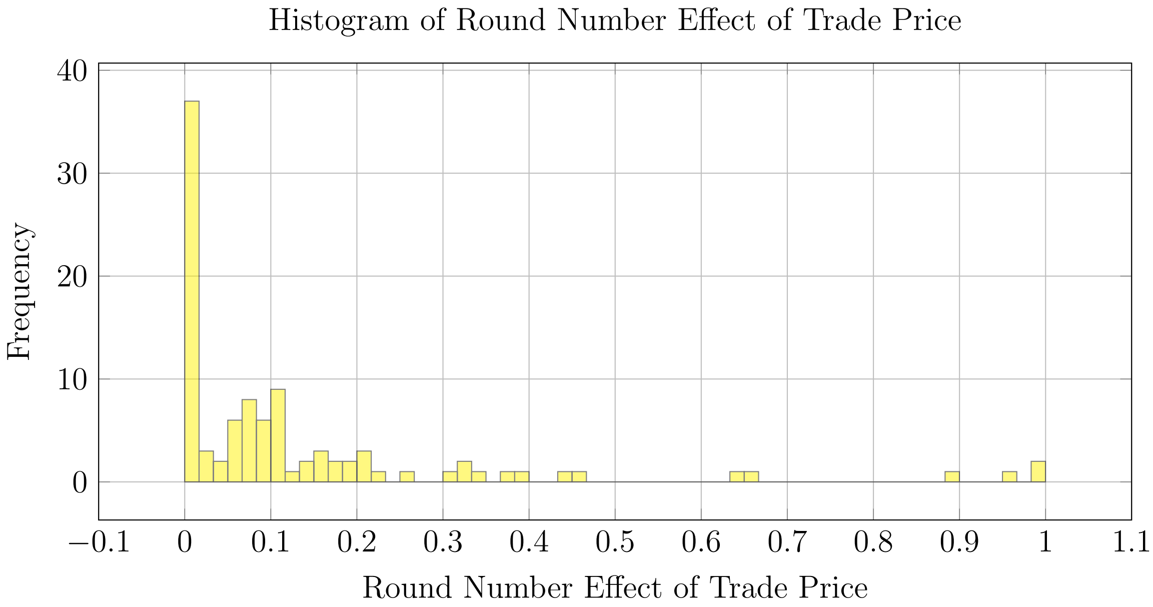

Figure 8: Distribution of RNE of Trade Price

The data in Figure 8 show that the round number effect (RNE) on trade prices occurs with relatively low frequency. The distribution is centered around zero as a percentage, suggesting that the impact of round numbers on trade prices is minimal for most stocks. This finding indicates insufficient evidence to support the hypothesis that the round number phenomenon strongly influences price levels. Thus, the overall effect on price movements appears weaker than previously assumed.

However, it is important not to overlook the RNE on specific stocks. For the 2-3 stocks in the 90% to 100% range of the distribution, almost all executed trade prices are round numbers. This unique characteristic could warrant further investigation, as focusing on these stocks might reveal opportunities to optimize trading strategies by leveraging the round number bias to maximize returns.

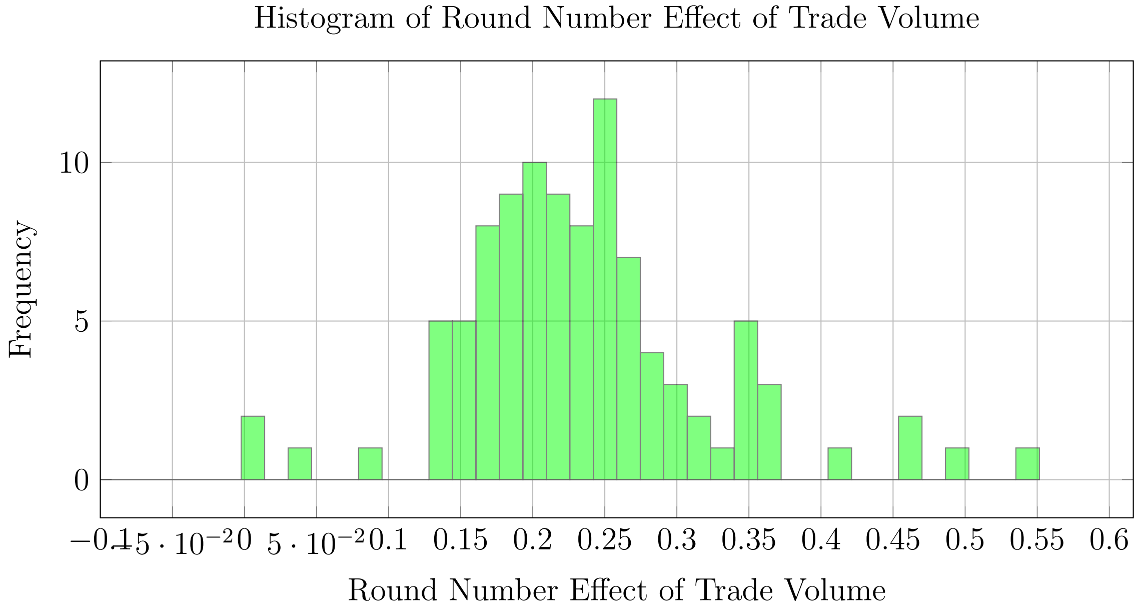

Figure 9: Distribution of RNE of Trade Volume

In contrast to trade prices, Figure 9 reveals a more pronounced round number effect on trade volumes, peaking between 20% and 25%. Although not overwhelmingly strong, there is a clear tendency for trade volumes to end in round numbers, with the effect being more consistent than in trade prices. Furthermore, a significant number of stocks (10-15) exhibit a round number effect in trade volumes in the higher percentage ranges (greater than 30% and up to 50%).

Additionally, it is noteworthy that only 3 out of the 100 stocks show no round number effect in their trade volumes, highlighting the widespread nature of this phenomenon across the dataset. This suggests that round numbers play a non-negligible role in shaping trade volume patterns for the majority of stocks, potentially influencing trader behavior and market liquidity.

Visualisation of Correlation Analysis

The following section will focus on exploring the correlations between various statistical metrics. Analyzing these relationships will provide valuable insights into how pricing dynamics interact within the market. For convenience, the color scale in the scatter plots represents data intensity, where red indicates lower intensity, and blue signifies higher intensity.

Mean inter-Trade and inter-Tick Time Correlations

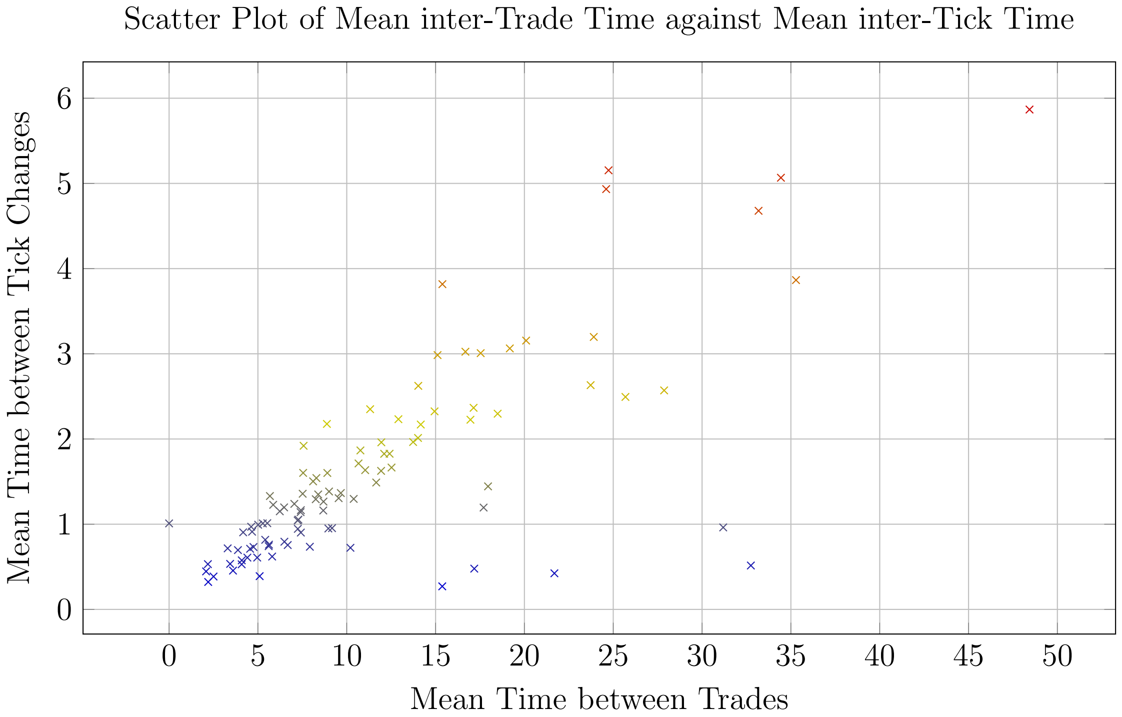

Figure 10: Mean inter-Trade and inter-Tick Time Relation

In Figure 10, a clear linear trend is observed, supporting the hypothesis of a positive correlation between inter-trade time and inter-tick-change time. This observation implies that higher trade frequency is often associated with shorter intervals between changes in the order book, suggesting a potential link between trade frequency and market volatility. To enhance the interpretability of the plot, it would be beneficial to apply a simple linear regression line as a visual aid.

This correlation aligns with market microstructure theory, where increased trading activity is typically indicative of heightened information flow or shifts in market sentiment, both of which contribute to greater volatility. In such scenarios, traders respond more quickly to new information, leading to more frequent price adjustments and, consequently, amplifying market fluctuations.

The dataset exhibits higher intensity at the lower tail, reinforcing the presence of high-frequency trading activity in this region.

It is important to note, however, that 5 to 6 external outliers deviate from the linear regression. Furthermore, trade frequency is not the sole factor influencing market volatility. Other key determinants include liquidity conditions, order book depth, and exogenous market events such as macroeconomic announcements or geopolitical developments.

Mean inter-Time Metrics and Round Number Effect (RNE) Correlations

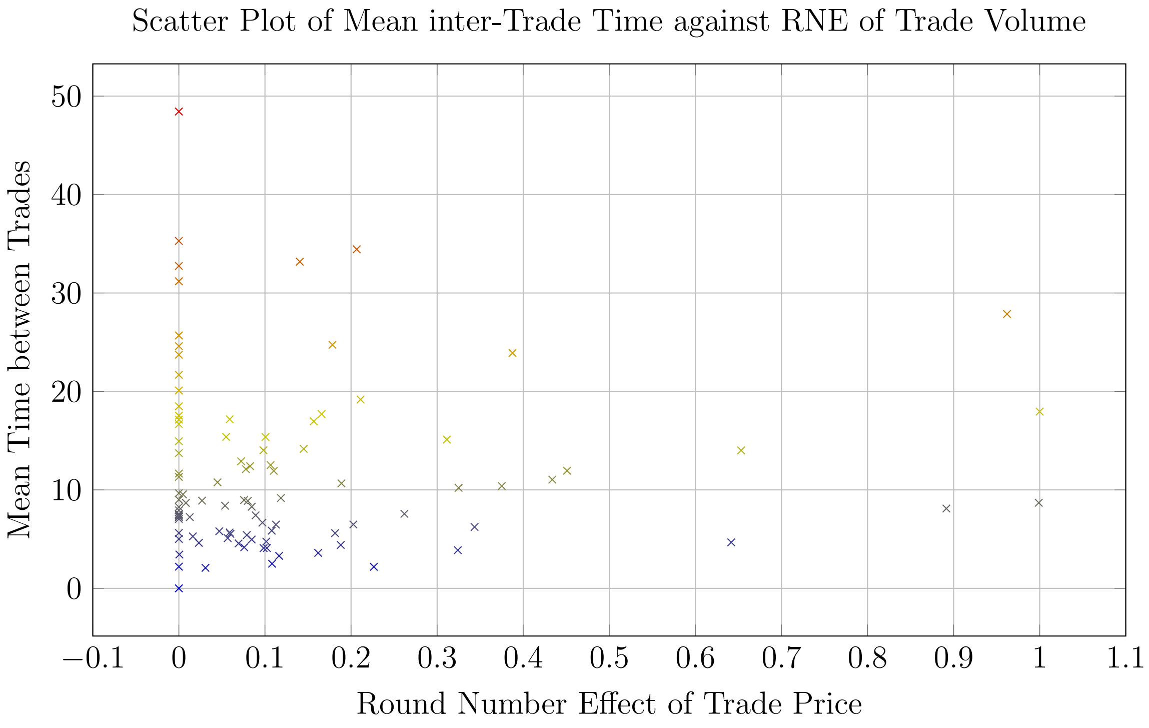

Figure 11: Mean inter-Trade Time against RNE of Trade Price Relation

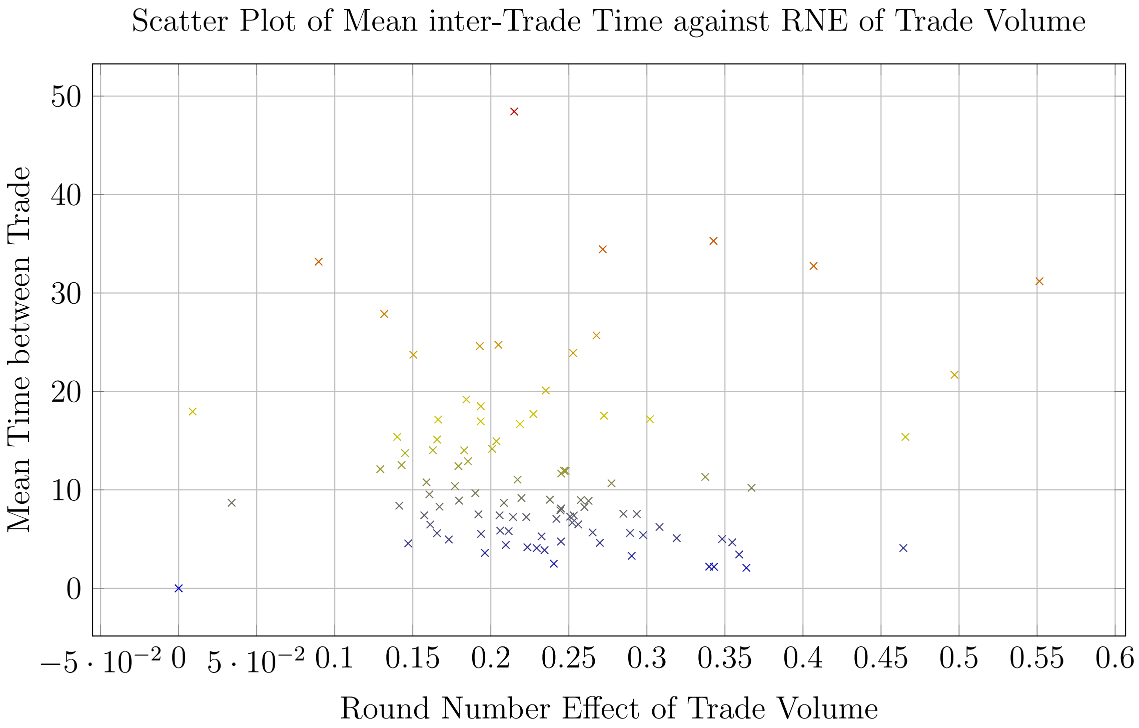

Figure 12: Mean inter-Trade Time against RNE of Trade Volume Relation

Analysis of Figures 11 and 12 reveals no significant correlation between the relative number of events (RNE) and the time intervals between trades (or tick changes, given their linear correlation). This absence of correlation suggests that the occurrence of round numbers does not influence either the timing of trades or the frequency of tick changes within the observed dataset.

This result challenges the hypothesis that round numbers might play a pivotal role in shaping trading patterns or affecting market microstructure. It is particularly unexpected, as psychological factors associated with round numbers are often thought to influence trading behaviour, potentially altering the timing and frequency of trades. However, the lack of such an effect in the data implies that other factors may exert a stronger influence on these trading metrics.

Conclusion

In conclusion, this analysis of inter-trade times, tick changes, bid-ask spreads, and the round number effect across 100 Scandinavian blue-chip stocks provides important insights into the intricate dynamics of high-frequency trading (HFT) markets. The predominance of rapid trades, as indicated by the distribution of mean and median inter-trade times, highlights the market's high liquidity and alignment with typical HFT characteristics. However, the skewness in mean inter-trade times due to occasional longer intervals emphasizes the need for caution when using the mean as a summary statistic, suggesting that more precise tick data (in milliseconds) may be required to capture the full picture of trading activity.

The close correlation between tick changes and trade frequency underscores the strong connection between market activity and updates in the order book. Additionally, the analysis of bid-ask spreads confirms the overall liquidity of the market, with most stocks exhibiting minimal spreads, though the occasional larger spreads signal periods of reduced liquidity or heightened market stress.

The round number effect (RNE) emerges more prominently in trade volumes than in prices, indicating that while round numbers may not heavily influence price levels, they do play a significant role in shaping trading behaviors, particularly with respect to trade volumes.

These findings shed light on the complex interplay between liquidity, market behavior, and trading strategies, particularly in the context of HFT. Future research could enhance these insights by utilizing even higher-resolution data and focusing on specific anomalies, such as the round number effect and extended inter-trade times. A deeper understanding of these nuances will be vital for refining market models and optimizing algorithmic trading strategies, especially under conditions of market volatility or diminished liquidity.

Appendix: C++ Code

```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <map>
#include <vector>
#include <cmath>
#include <algorithm>
#include <tuple>
#include <iomanip>

using namespace std;

// Function to split a string by a specific delimiter
vector<string> split(const string &line, char delimiter) {
    vector<string> tokens;
    stringstream ss(line);
    string token;
    while (getline(ss, token, delimiter)) {
        tokens.push_back(token);
    }
    return tokens;
}

// Function to compute the mean of a vector of doubles
double mean(const vector<double> &values) {
    if (values.empty()) return 0.0;
    double sum = 0.0;
    for (double v : values) sum += v;
    return sum / values.size();
}

// Function to compute the median of a vector of doubles
double median(vector<double> values) {
    if (values.empty()) return 0.0;
    nth_element(values.begin(), values.begin() + values.size() / 2, values.end());
    if (values.size() % 2 == 0) {
        nth_element(values.begin(), values.begin() + values.size() / 2 - 1, values.end());
        return (values[values.size() / 2 - 1] + values[values.size() / 2]) / 2.0;
    }
    return values[values.size() / 2];
}

// Function to compute the bid-ask spread
inline double computeBidAskSpread(double askPrice, double bidPrice) {
    return askPrice - bidPrice;
}

// Function to check if the condition indicates an auction period
bool exclusionPeriods(const double &bidPrice, const double &askPrice, const int &tradeVolume, const string &conditionCode) {
    return (tradeVolume == 0 || (bidPrice > askPrice && conditionCode != "XT" && conditionCode != ""));
}

int main() {
    // Open input CSV file containing tick data
    ifstream infile("scandi/scandi.csv");

    // Map to store stock data (keyed by stock ID)
    map<string, map<string, vector<tuple<double, double, double, int, int, string, int>>>> stockData; // Map of stockId -> date -> data vector
    // Dummy item
    string line;

    // Read and process CSV data line by line
    while (getline(infile, line)) {
        auto tokens = split(line, ',');

        double bidPrice = stod(tokens[2]); // Adjusted for Bid Price column
        double askPrice = stod(tokens[3]); // Adjusted for Ask Price column
        int tradeVolume = stoi(tokens[7]); // Trade Volume column
        string conditionCode = tokens[14]; // Condition codes column

        if (exclusionPeriods(bidPrice, askPrice, tradeVolume, conditionCode)) {
            continue;
        }

        string stockId = tokens[0]; // Bloomberg Code/Stock identifier
        string date = tokens[10]; // Date column
        double tradePrice = stod(tokens[4]); // Adjusted for Trade Price column
        int updateType = stoi(tokens[8]); // Update type
        int time = stoi(tokens[11]); // Time in seconds past midnight

        // Store data keyed by stock ID and date
        stockData[stockId][date].emplace_back(bidPrice, askPrice, tradePrice, tradeVolume, time, conditionCode, updateType);
    }

    // Open output CSV file
    ofstream outfile("output.csv");

    // Write header to output file
    outfile << "Stock ID,Mean Time Between Trades,Median Time Between Trades,Mean Time Between Tick Changes,Median Time Between Tick Changes,"
            "Longest Time Between Trades,Longest Time Between Tick Changes,Mean Bid Ask Spread,Median Bid Ask Spread,"
            "Price Round Number Effect,Volume Round Number Effect\n";

    // Process each stock's data for analysis
    for (const auto &stockEntry : stockData) {
        const string &stockId = stockEntry.first;
        const auto &dateData = stockEntry.second;

        vector<double> tradeTimes, tickTimes, bidAskSpreads;
        double zeroAsPriceLastDigit_Count = 0.0, nonzeroAsPriceLastDigit_Count = 0.0;
        double zeroAsVolumeLastDigit_Count = 0.0, nonzeroAsVolumeLastDigit_Count = 0.0;

        // Process data by date
        for (const auto &dateEntry : dateData) {
            const string &date = dateEntry.first;
            vector<tuple<double, double, double, int, int, string, int>> data = dateEntry.second;

            // Sort the data by time (4th element in tuple)
        sort(data.begin(), data.end(), [](const tuple<double, double, double, int, int, string, int>& a, const tuple<double, double, double, int, int, string, int>& b) {
            return get<4>(a) < get<4>(b);
        });

        double lastTradeTime = -1;
        double lastTickChangeTime = -1;

        for (size_t i = 0; i < data.size(); ++i) {
            const auto &curr = data[i];
            int currUpdateType = get<6>(curr);
            double currTime = static_cast<double>(get<4>(curr)); // time in seconds past midnight

            // Time difference between trades (when updateType is 1)
            if (currUpdateType == 1) {
                if (lastTradeTime >= 0) {
                    tradeTimes.push_back(currTime - lastTradeTime);
                }
                lastTradeTime = currTime;
            }

            // Time difference between tick changes (when updateType is 2 or 3)
            if (currUpdateType == 2 || currUpdateType == 3) {
                if (lastTickChangeTime >= 0) {
                    tickTimes.push_back(currTime - lastTickChangeTime);
                }
                lastTickChangeTime = currTime;
            }

            // Calculate bid-ask spread
            bidAskSpreads.push_back(computeBidAskSpread(get<1>(curr), get<0>(curr)));

            // Accumulate round number events (last digit being zero)
            double currTradePrice = get<2>(curr);
            int currTradeVolume = get<3>(curr);

            // Check if round number
            if (static_cast<int>(currTradePrice) % 10 == 0) {
                zeroAsPriceLastDigit_Count++;
                } else {
                    nonzeroAsPriceLastDigit_Count++;
                }

                if (currTradeVolume % 10 == 0) {
                    zeroAsVolumeLastDigit_Count++;
                } else {
                    nonzeroAsVolumeLastDigit_Count++;
                }
            }
        }

        // Calculate metrics across all days
        double meanTradeTime = mean(tradeTimes);
        double medianTradeTime = median(tradeTimes);
        double meanTickTime = mean(tickTimes);
        double medianTickTime = median(tickTimes);
        double longestTradeTime = tradeTimes.empty() ? 0.0 : *
           max_element(tradeTimes.begin(), tradeTimes.end());
        double longestTickTime = tickTimes.empty() ? 0.0 : *
           max_element(tickTimes.begin(), tickTimes.end());
        double meanBidAskSpread = mean(bidAskSpreads);
        double medianBidAskSpread = median(bidAskSpreads);

        double totalTradeCount = zeroAsPriceLastDigit_Count +
           nonzeroAsPriceLastDigit_Count;
        double priceRoundNumberEffect = totalTradeCount == 0 ? 0.0 :
            zeroAsPriceLastDigit_Count / totalTradeCount;
        double volumeRoundNumberEffect = totalTradeCount == 0 ? 0.0
           : zeroAsVolumeLastDigit_Count / totalTradeCount;

        // Write to output file
        outfile << stockId << "," << meanTradeTime << "," <<
           medianTradeTime << ","
                << meanTickTime << "," << medianTickTime << "," <<
                   longestTradeTime << ","
                << longestTickTime << "," << meanBidAskSpread << ","
                    << medianBidAskSpread << ","
                << priceRoundNumberEffect << "," <<
                   volumeRoundNumberEffect << "\n";
    }


    outfile.close();
    cout << "Processing complete. Output file closed.\n";
    return 0;

}
```

scandi.cpp