Arfima Financial Solutions

T2T Microstructure

Price Discovery and Lead-Lag Relationships

20th November, 2024

Arfima Financial Solutions
Calle José Bardasano Baos 9, 7
28016 Madrid, España
www.af-services.com
+34 91 129 2874

Arfima
Financial Solutions

Contents

Chapter 1

Introduction

This report aims to provide a comprehensive analysis of the dynamics underlying financial markets, focusing on key aspects of market microstructure.

The first chapter focuses on execution, latency, and the operational intricacies of trading systems. It provides an in-depth look at the practical aspects of market operations, including the measurement of various latencies encountered in the execution pipeline, the synchronization of tick-by-tick data with Audit Trails, and the implications of these factors for market efficiency and performance.

The second and third chapters delve into specific metrics that are instrumental in analyzing and comparing different markets, with a particular emphasis on identifying where price discovery occurs. In Chapter 2, we explore the lead-lag relationships and price discovery metrics, including the Hayashi-Yoshida correlation, which helps in understanding the temporal dynamics between different assets and how quickly various markets incorporate new information. Chapter 3 extends this analysis by introducing Hasbrouck's Information Share, a metric designed to quantify the contribution of different markets to price discovery, offering deeper insights into how information is reflected in asset prices across multiple trading venues.

Together, these chapters present a multi-faceted view of market microstructure, combining theoretical insights with practical considerations to offer a robust framework for analyzing the complex interactions that drive financial markets.

This document includes complementary code in the Market Microstructure repository. A description of the different scripts, notebooks and folders of the repository can be found on the Complementary Code Section

Chapter 2

Introduction to latencies

2.1 Foundation and objectives

Electronic markets are subject to extremely fast changes, which can be in the order of microseconds. These quick price variations make it difficult to achieve the price we exactly wanted when quoting an instrument. The software in this project helps ensure that executions run smoothly, but its uses go beyond execution and can improve the understanding of market microstructure issues. Some potential uses include:

- Measuring lead-lags across assets, including both liquid and less liquid markets.
- Measuring the time from signal to response, where the signal can be the timestamp of an event (such as an economic release or auction outcome) or a change in the prices of an asset.
- Merging data from Drop Copy files with information from insider information platforms (IIP) to investigate issues of market abuse.

This chapter gathers the tools and methods related to market microstructure, focusing on execution, latencies, and related concepts. By "execution," we refer to interactions with the market, which are reflected in the Audit Trail. Although the analysis here is based on a specific Audit Trail format, the methodology can be extended to other Audit Trails or even Drop Copy files from exchanges. Execution and quoting information from Audit Trails are combined with tick-to-tick exchange data.

The tools described here are closely related to analyses that can be used generally by anyone needing to measure latencies or analyze executions. While some

tools focus on specific data types, such as tick-to-tick (t2t) information, others integrate both Audit Trail and t2t data to provide a comprehensive view of market interactions.

The main parts of the project are:

- **Latency**: This includes code for measuring latencies at various steps of the execution pipeline and also synchronizing tick-to-tick market data with Audit Trail data. The latency code, like post-trade analysis, always requires Audit Trail information.
- **PostTrade**: This involves code aimed at analyzing real executions using Audit Trail and tick-to-tick information. It includes a visualization component, which shows bid, ask, and quotes for the displayed instrument, complemented by a reporting component that summarizes execution PnL, the price of payups, and latency information. Together, these components provide a detailed view of execution performance, leveraging Audit Trail information for both detailed reporting and effective visualization. PostTrade analysis always deals with Audit Trail information, both for the execution report and the visualization part.

Another important functionality of the module is the capability to synchronize the Audit Trail with the tick-to-tick (t2t) data. In many financial analyses, it is essential to integrate multiple sources of data to gain a comprehensive view of market activity. The Audit Trail provides detailed information about trades, while the tick-to-tick data offers a granular view of market dynamics on a very fine time scale. By combining these sources, we can verify trades more precisely, identify potential timing mismatches, and enhance the accuracy of trading algorithms. The timestamps present in these two datasets might be recorded at different times, rounded differently, or some data might be merged or coalesced. Therefore, for many applications, it is useful to match our trades or quotes from the Audit Trail with the tick-to-tick data, ensuring that the two datasets are properly synchronized.

2.1.1 Applications and Use Cases

In our specific use case, the general idea is to answer any question we have regarding t2t data at high-frequency scales. With this in mind, we developed a set of interconnected tools. For some questions, we might use the basic latencies modules. The PostTrade is used when there are real executions.

After executing our strategy, we will be interested in seeing what happened at the tick-to-tick level and within the order book. For this, we can use the PostTrade analysis to visualize the execution of our strategy. Moreover, we can see the breakdown of the latencies in the execution report, which can reveal if we didn't achieve the desired execution due to latency.

If latency is too high, we use the latencies module to check if this has historically occurred for the same product, exchange, or account. This analysis can also be performed with historical data to help understand execution.

2.1.2 Data requirements

We will be using tick-by-tick data with millisecond precision, and for some products, also microsecond or nanosecond precision. Data analysis will be restricted to the first layer of the order book, focusing on transactions, bids, and offers. A complementary source of data will be the Drop Copy files from the FIX protocol, and Audit Trails from various software vendors when Drop Copy is not available. The introduction of Audit Trail data provides a rich array of new information, including cancellations, refreshes, new quotes, fills, and cancels. This data is useful for understanding issues related to our own execution.

An important part of the project is understanding how to merge t2t data with Audit Trail data so that we can combine information from our execution with trades or quotes from the market itself, thereby improving our understanding of our interaction with the market.

2.2 The basics of the Audit Trail

An Audit Trail is a CSV file that contains detailed information on all trading activity for a given set of trades, on a specific day, and within a specific hour. Each row in the file represents an action taken by a trader, and the columns provide detailed information about that action. Typically, an Audit Trail file includes around 80 columns, capturing various aspects of each trading action.

Audit Trails are crucial for compliance and analysis, as they record every order, fill, cancellation, and modification made by traders. For example, an Audit Trail for a given set of traders on 03/05/22 between 7am and 8am might contain 112,952 rows of data, each corresponding to a distinct action taken during that hour.

In practice, an Audit Trail provided by an Independent Software Vendor (ISV) might be organized by day and hour. Files are usually stored in directories named

by the date (e.g., ddmmyy), and the files themselves are named according to the hour (e.g., ISVAudit-hhddmmyy.csv). This structure allows for easy retrieval and analysis of specific periods of trading activity.

The detailed information captured in an Audit Trail includes timestamps, order IDs, quantities, prices, and more. This rich dataset is essential for understanding the full context of trading activity, enabling firms to monitor compliance, analyze performance, and investigate any anomalies in market behavior.

2.3 Autospreaders: an example of a trading algorithm

In the context of trading algorithm analysis, one example of an advanced trading tool that can be studied is the autospreader. An autospreader is designed to create and manage synthetic spread markets by automatically coordinating and executing orders across multiple related instruments. This makes it a prime candidate for analyzing the effectiveness and behavior of such trading algorithms.

The autospreader allows traders to construct and trade custom spread strategies by linking orders in different legs of the spread, ensuring that the overall strategy is executed efficiently. When a synthetic spread order is placed, the autospreader submits quoting orders in the designated leg or legs, based on real-time market conditions in the hedge leg and the available liquidity. It calculates the optimal price level for these quoting orders to ensure that the spread is filled at the desired price. As market conditions change, particularly in the hedge leg, it automatically adjusts the quoting orders to maintain the intended spread.

By analyzing the performance of an autospreader, we can gain insights into how such algorithms manage execution and respond to dynamic market conditions, making it a valuable example for this analysis.

2.4 Introduction to latencies and types of latencies

The latency module is a collection of functions designed to measure different kinds of latencies in the execution pipeline. When working with such small time intervals, every millisecond counts, making it crucial to meticulously track all events. The example shown in Figure 1 illustrates this complexity. As we transition from working with a single machine to managing multiple time scales and clocks, the same event may occur at different times depending on the observer. In this example, we have a system with two clocks: one corresponding to the exchange and the other to our own server. The event is observed at different times by each system.

MINISTERIO DE CIENCIA E INNOVACIÓN
AGENCIA ESTATAL DE INVESTIGACIÓN
Plan de Recuperación, Transformación y Resiliencia
uc3m Universidad Carlos III de Madrid
Financiado por la Unión Europea NextGenerationEU
UNED

While these differences are minimal in a colocation setup, they remain important to consider, especially as the number of observers increases, which is particularly relevant for cross-exchange spreads.

Such latencies, and their potential market implications, have been studied extensively in the context of high-frequency trading. For example, Kirilenko et al. (2014) [6] highlighted how latency-sensitive strategies can significantly impact market behavior, particularly during events such as the 2010 Flash Crash. Understanding the role of latency in these scenarios provides valuable insights into the behavior of trading systems under stress and helps refine approaches to measuring and mitigating delays.

Let us guide you with an example. Assume that we have an algorithm operating in a system with two clocks: one for the exchange and one for our server. In this scenario, the same event might be observed at slightly different times by each system. This complexity is further illustrated in Figure 1, which highlights the various latencies computed by the latency module. Below is a list of the latencies that can be computed, along with the terminology used throughout the documentation:

- **Server to Exchange Latency**: Represented in the above scheme by the time from z_1 to t_1. This latency measures the time taken for an order to travel from our trading machine to the exchange.

- **Exchange Latency**: Represented by the time from t_1 to z_2. This latency accounts for the time the exchange takes to process an order, plus the time it takes for the order to reach our order connector. It can be calculated using the latency module, where a comparison of this latency between various order types is also provided.

- **Hedge Latency**: Represented by the time from z_0 to z_1. This latency measures the time it takes for a trading machine to send a hedge order after receiving a fill on the active leg of an autospreader.

- **Hedge Latency with Acknowledgment**: This latency is represented by the time from z_1 to z_2. It measures the time between receiving a fill message and being acknowledged that our new hedge order has arrived at the exchange. This latency can be computed in the same examples where "hedge latency" is calculated.

- **Requote Latency**: Also represented by the time from z_0 to z_1, but with a different context. Generally, it is the time between receiving a signal from the market and reacting to it. Currently, this code considers the case where the

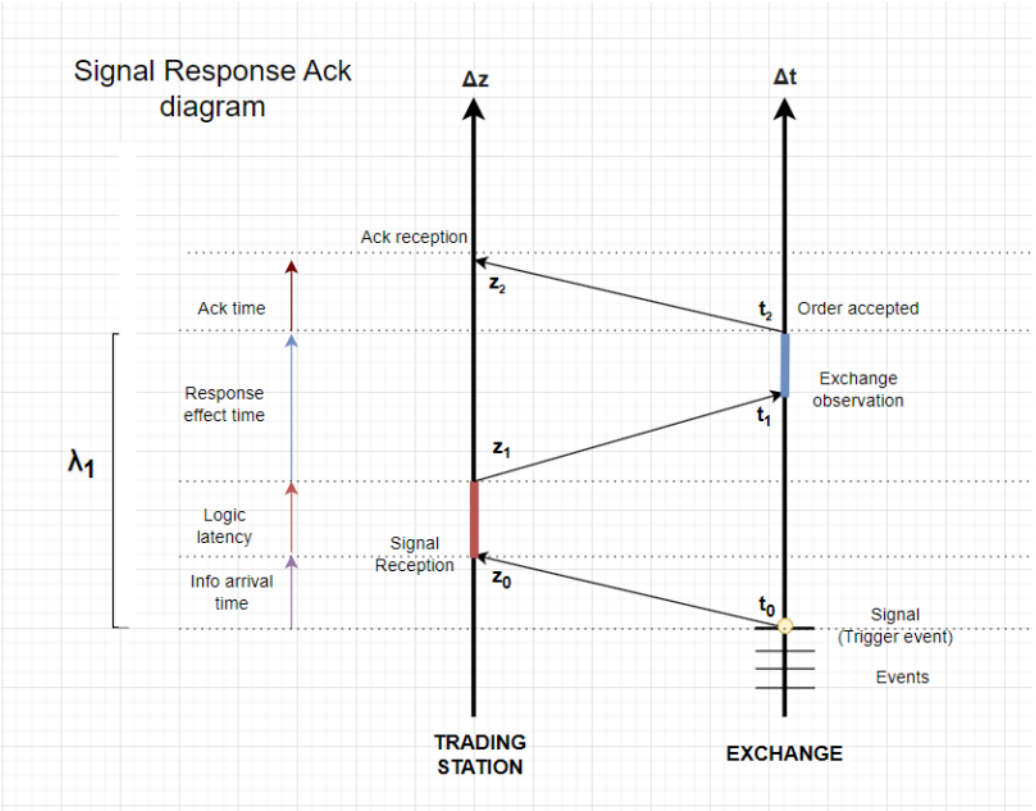


Figure 2.1: The timeline shows the flow of a signal from the exchange (trigger event) through its reception by the trading station, and subsequent actions, including order acceptance by the exchange and acknowledgment receipt by the trading station. The different latencies involved are highlighted along the timeline. The vertical axis represents the time taken for each step in the process. 'Ack' is an abbreviation for acknowledgment.

signals are updates of the price of the hedge leg of an autospreader, and the reactions are updates of our quotes in the active leg.

- **Payup Latency**: This latency measures the time between the first new hedging order and the first payup, as well as the time between subsequent payups. In the latency stats, which are part of the output, payup latency is divided into the first payup, the second payup, and the third and subsequent payups. Typically, in economic release environments, the first payups are slower, while the subsequent ones are faster. This variation is due to spikes in exchange latency around the time of economic releases.

2.5 Measuring the server to exchange latency

In this section, we analyze the latency between the server and the exchange, also looking at different exchanges and their specific characteristics. The following analysis was performed for a trader operating on CME and ICE, covering orders from January 1, 2023, to July 26, 2023. The analysis can be found in the Market Microstructure repository, specifically in the notebook `server_exch.ipynb`. The analysis highlights significant variations in latency across different exchanges, with a particular focus on one exchange where latency measurements frequently clustered around 40 milliseconds. We chose 40 ms as a possible threshold, but the analysis would be similar for other thresholds as well.

The analysis shows that for the first exchange, CME, there was a recurrent latency value of approximately 40 milliseconds. This observation was made through a detailed examination of the tick-by-tick data, which revealed a consistent pattern in the recorded latencies. This consistency in latency can be attributed to specific factors within the exchange's infrastructure or the configuration of the Independent Software Vendor (ISV) providing the data. In the specific case analyzed, this consistent latency was likely due to the travel time to Chicago, where the exchange's matching engine is located.

Additionally, it was observed that latencies for synthetic products were usually greater than those for non-synthetic products. Another key observation is that the server-to-exchange latency can only be computed for new orders, and not for replaces, cancels, or restates.

The occurrence of the 40-millisecond latency was far more prevalent in this exchange compared to others, such as the ICE exchange, where latency measurements were more dispersed and did not exhibit the same clustering.

In conclusion, the consistent 40-millisecond latency observed in the first exchange indicates a specific behavior in the server-to-exchange communication, likely influenced by the exchange's operational setup or the ISV's data handling processes. Understanding these latency characteristics is crucial for optimizing trading strategies and ensuring that latency-sensitive operations are effectively managed.

2.6 Measuring the exchange latency

As mentioned, this is the time between when the exchange receives an order and when it sends an acknowledgment or a fill.

In the examples checked in the already mentioned notebook, `server_exch.ipynb`, it is usually very low, the median is less than 1ms which is our maximum granularity for this type of latency. However, it also has high outliers, in the order of 1494 ms. This latency tends to have more outliers.

Talking about exchanges, in general exchange latency is slightly lower in EUREX than in CME.

2.7 Measuring the Hedge latency

In general, this latency has been found to be indistinguishable from 0 ms, as the logic of the algorithms employed is extremely fast. This typically occurs in intra-exchange scenarios, where all legs of the spread are executed within the same exchange. However, if some of the legs are in different exchanges, the hedge latency tends to be higher due to the additional time required to coordinate and execute orders across multiple exchanges.

2.8 Time measurements

This section provides a specification of the time measurements necessary to compute latencies. These time measurements are essential for understanding the sequence of events from order submission to processing.

The key time measurements obtained from the ISV API, in the sequence they occur, are as follows:

1. **Sent**: The time the ISV application sent the order.
2. **OCReceived**: The time the Order Collector (OC) received the order (following risk checking). The Order Collector (OC) in a trading system is a component that receives orders from the trading application, performs necessary validations like risk checks, and then routes the orders to the exchange. It also handles responses from the exchange, ensuring the order process is secure and efficient.
3. **OCSentToExchange**: The time the OC sent the order to the exchange.
4. **ExchTransactionTime**: The transaction time reported by the exchange.
5. **Received**: The time the OC received the response from the exchange.

6. **Processed**: The time the exchange response left the OC.

7. **SDKProcess**: The time the ISV SDK processed the order response message.

Additionally, the Audit Trail provides the following timestamps, which are useful for comparing with the ISV API times:

- **OriginalTime**: The time the order was initially submitted. For new orders, this is when the order was sent to the exchange.
- **ExchTime**: The time at which a fill is sent by the tier 1 (exchange gateway). In some exchanges, this represents the time the message leaves the gateway towards the exchange.
- **Time**: The time the transaction occurred, indicating when the fill was received and processed.
- **TimeSent**: The time, in nanoseconds, when the Order Connector routed the message within the ISV system.

There is a rough equivalence between the ISV API times and the Audit Trail times. The Audit trail is generated by the ISV, taking both its internal data and the exchange-provided data, and generating a table with the following columns:

- **ExchTime** $\approx$ **ExchTransactionTime**
- **TimeSent** $\approx$ **Received**
- **Time** $\approx$ **Received** (or **Processed**)
- **OriginalTime** $\approx$ **OCSentToExchange** (only for new orders)

Furthermore, the following timestamps can be retrieved from the FIX Drop Copy for detailed analysis:

- **fix_received_ts**: The time when the FIX message was received.
- **order_received_oc_ts**: Similar to **OCReceived**.
- **time_sent_exchange_ts**: Corresponds to **OCSentToExchange**.
- **transaction_time/execution_time**: Comparable to **ExchTransactionTime**.
- **received_from_exchange_ts**: Similar to **Received**.

- **TimeReceivedFromExchange**: Equivalent to **Received**.
- **time_sent_oc_to_fix_ts**: Analogous to **Processed**.
- **52= SendingTime**: Part of the FIX header, typically representing when the message was sent.

In conclusion, the ISV API and Audit Trail provide different types of data, and the inclusion and comparison of both enrich the analysis by offering a more comprehensive view of the order processing sequence and potential latency sources

2.9 Synchronization

In high-frequency trading and other time-sensitive financial applications, precise alignment of timestamps between different data sources is essential. Small discrepancies in timing can lead to misinterpretations of market events, inaccurate analysis of trade behavior, and flawed decision-making. Therefore, we perform this analysis to ensure that the timestamps recorded in the CME Audit Trail and those captured in the tick-to-tick (t2t) data are perfectly synchronized, enabling accurate downstream analyses.

The aim of this analysis is to evaluate the synchronization of timestamps between the trades recorded in the CME Audit Trail and those captured in the t2t data. Ensuring synchronization is crucial for accurately aligning trade information between these two sources, as discrepancies could impact downstream analyses. Therefore, the analysis seeks to investigate whether the current algorithm accurately matches trades in the Audit Trail with corresponding ticks in the t2t data, focusing on potential timing mismatches.

A specific analysis was performed for CME in two periods: from September 1, 2023, to September 25, 2023, and from July 1, 2023, to July 25, 2023. The objective was to study the distribution of offsets between the trades in the Audit Trail and those in the t2t data, with particular attention given to cases where the current algorithm matches Audit Trail trades to an earlier tick (i.e., positive offsets) or to a tick more than 1 ms away.

The analysis revealed that the current code matches all trades with offsets that never exceed 1 ms. This indicates that the timestamps in the Audit Trail and the t2t data are perfectly synchronized for CME. The hypothesis is that both timestamps represent the same FIX timestamp, which is rounded down to milliseconds by the

Independent Software Vendor (ISV) when providing the Audit Trail. This hypothesis is supported by the fact that no positive offsets were observed (i.e., the Audit Trail time is never larger than the t2t time).

Chapter 3

Lead Lag and Price Discovery

3.1 Introduction to Lead Lag and Price Discovery

A lead-lag relationship refers to the temporal dynamic between two instruments, where the price movements of one instrument (the leader) precede and potentially predict the movements of another instrument (the lagger). The liquidity of the market can play a significant role here.

In this chapter, we analyze different metrics to detect lead-lag relationships across assets. This analysis is key for understanding how information flows between assets, which can provide insights into the mechanisms of information transfer, the impact of market events, and the potential existence of insider information. By examining lead-lag relationships, we can identify which assets react first to new information and how quickly other assets follow. Such insights are valuable for traders, market makers, and analysts who need to predict price movements or assess the efficiency of information dissemination across different markets.

Additionally, this type of analysis could help detect potential insider behavior or uncover market abuses in trading practices. By identifying unusual patterns or timing mismatches in how information is reflected across assets, it becomes possible to pinpoint anomalies that may indicate manipulation or unfair trading activities.

The analysis primarily focuses on tick-to-tick (t2t) data, which provides high-frequency trade and quote information, allowing us to capture these relationships with a high degree of temporal precision. We will closely follow the methodology presented in the paper by Huth and Abergel [3].

3.2 Definition of the metrics used. Hayashi-Yoshida metric

In a formal setting, using stochastic calculus, we define the metrics for two Itô processes X, Y such that

$$dX_t = \mu_t^X dt + \sigma_t^X dW_t^X,$$
$$dY_t = \mu_t^Y dt + \sigma_t^Y dW_t^Y$$
$$d\langle W^X, W^Y \rangle_t = \rho_t dt,$$

and independent observation times $0 = t_0 \leq t_1 \leq \ldots \leq t_{n-1} \leq t_n = T$ for X and $0 = s_0 \leq s_1 \leq \ldots \leq s_{m-1} \leq s_m = T$ for Y, we can show that

$$\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij} \neq \emptyset\}},$$

where

$$O_{ij} =]t_{i-1}, t_i] \cap]s_{j-1}, s_j],$$
$$r_i^X = X_{t_i} - X_{t_{i-1}},$$
$$r_j^Y = Y_{s_j} - Y_{s_{j-1}},$$

is an unbiased and consistent estimator of $\int_0^T \sigma_t^X \sigma_t^Y \rho_t dt$ as the observation intervals become finer. In practice, this involves summing each product of increments whenever they share any time overlap. For constant volatilities and correlation, this method yields a consistent estimator for the correlation

$$\hat{\rho} = \frac{\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij} \neq \emptyset\}}}{\sqrt{\sum_i (r_i^X)^2 \sum_j (r_j^Y)^2}}.$$

It is possible and more useful to consider a lagged version for the correlation

$$\hat{\rho}(\ell) = \frac{\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij}^\ell \neq \emptyset\}}}{\sqrt{\sum_i (r_i^X)^2 \sum_j (r_j^Y)^2}}$$

where

$$O_{ij}^\ell =]t_{i-1}, t_i] \cap]s_{j-1} - \ell, s_j - \ell] \tag{3.1}$$

This is the key statistic from which derived metrics, such as the Lead-Lag Ratio (LLR) and the maximum lag, are calculated. We can think of it as a function of one

variable, specifically the lag.

The approach is related to Granger causality, but it offers additional insights. While Granger causality indicates which asset is leading the other in a given pair, these metrics go further by considering the strength and characteristic timing of the lead/lag relationship. In this context, the maximum level of the cross-correlation function and the specific lag at which it occurs are crucial factors to take into account [7]. Specifically, the Hayashi-Yoshida and the Lead-Lag Ratio, both measured with observation times of trades without conditioning by types of movements:

- **HY_lead_lag**: The lag (in milliseconds) that maximizes the absolute value of the lead-lag estimator, where the estimator is given by:

$$HY\_lead\_lag := \arg\max_{\ell} |\hat{\rho}(\ell)| \tag{3.2}$$

- **LLR**: The formula is

$$\text{LLR} := \frac{\sum_{i=1}^{p} \rho^2(\ell_i)}{\sum_{i=1}^{p} \rho^2(-\ell_i)}$$

  where index i refers to a discrete grid of lags we are arbitrarily choosing.

  Its output is always a positive one, and the interpretation is the following: X leads Y if $LLR > 1$, and Y leads X if $LLR < 1$.

  Note that this indicator tells us which asset is leading the other for a given pair. However, it is also important to consider the strength and the characteristic time of this lead/lag relationship. Therefore, the maximum level of the cross-correlation function and the lag at which it occurs must be taken into account. This is why both metrics complement each other. Nevertheless, as we will demonstrate in the examples, this metric tends to be noisier.

Relationships between the various statistics

In this section we present a summary of the statistics used and the key properties of each.

Object	*Formula*	*Input*	*Output*
HY function	$\hat{\rho} = \frac{\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij} \neq \emptyset\}}}{\sqrt{\sum_i (r_i^X)^2 \sum_j (r_j^Y)^2}}$	X, Y	correlation
HY lagged function	$\hat{\rho}(\ell) = \frac{\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij}^{\ell} \neq \emptyset\}}}{\sqrt{\sum_i (r_i^X)^2 \sum_j (r_j^Y)^2}}$	X, Y, ℓ	correlation of X,$Ylag$
HY lead/lag time	$\arg\max_{\ell} \lvert\hat{\rho}(\ell)\rvert$	X, Y	ρ-maximising lag time
LLR	$\frac{\sum_{i=1}^{p} \rho^2(\ell_i)}{\sum_{i=1}^{p} \rho^2(-\ell_i)}$	$\rho, X, Y, \{\ell_1, ..., \ell_p\}$	lead/lag indicator

Some interesting observations on these metrics are:

- The two time-series are not necessarily synchronous.
- The HY function is a way of measuring correlation between two asynchronous time series. This can be generalised to the lagged HY metric. This lagged metric can be considered as shifting the asset Y and then applying HY to the shifted intervals.
- The reason the HY function is chosen is to avoid being fooled by liquidity effects, yielding the most traded instrument automatically being a leader. Before the advent of the HY function the previous-tick correlation function was used to measure cross-correlation. The downside of this cross-correlation function was that a lead-lag effect was artificially created between two time-series when a difference in liquidity was present.
- Intuitively, the most heavily traded assets tend to incorporate information into prices faster than others, so they should lead. In general:

$$\text{Asset X more liquid} \implies \text{Asset X leads} \implies LLR > 1$$

However, this rule does not always hold. It is a heuristic rule, as the most liquid markets usually incorporate information the fastest, but this is not a universal truth.

3.2.1 Limitations of trade time

In the formal definition, we stated that X and Y are two Itô processes with independent observation times, but we have not yet specified what these times represent in practice. There are two logical alternatives:

- **Trade time**: consider a new time point whenever there is a new trade.
- **Tick time**: consider a new time point whenever there is a variation in the midquote.

These do not necessarily coincide: there can be variations in the midquote without trades actually occurring, and trades can happen without the midquote changing. Therefore, we can consider two different time indices for the analysis: one that changes every time there is a trade (we call this trade time) and one that changes every time there is a non-null variation in the midquote (we call this tick time).

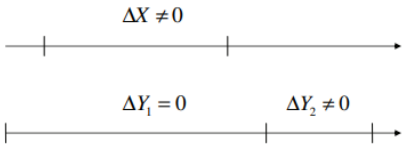


Figure 2: Difference between the trade time and tick time Hayashi-Yoshida correlations

In order to illustrate the difference between measuring correlation with tick time versus trade time, we will go through a simple example. We first construct an idealized scenario where the correlation $\hat{\rho}(\ell)$ for an asset Y differs depending on the chosen time framework. The example data is shown in Table 1, where the returns are already computed. On the right, we compute the returns only every time a trade occurs. If a trade does not occur, the return is zero. If a trade occurs, it accumulates the virtual returns of the contemporaneous time and all the virtual returns that occurred when there were no trades.

We can identify the virtual returns with tick time, that is, those are the returns measured with all the midquotes whenever there is a change in the bid or the ask.

As both columns are differently, it is clear that if we consider the HY metric that we introduce later in this article, we will not get the same results. But we are going to give a concrete example to see it more clearly.

Asset X

Period	r_{it} (Tick Returns)	r^0_{it} (Trade Returns)
1	0.01	0.01
2	0.02	0.02
3	0.03	0
4	0.04	0
5	0.05	0.12

Table 3.1: Tick Returns (r_{it}) and Trade Returns (r^0_{it}) for Asset X

Asset Y

Period	r_{jt} (Tick Returns)	r^0_{jt} (Trade Returns)
1	0.02	0
2	-0.01	-0.01
3	0.01	0.01
4	-0.02	0
5	0.03	0.01

Table 3.2: Tick Returns (r_{jt}) and Trade Returns (r^0_{jt}) for Asset Y

Hayashi-Yoshida correlation calculations

For simplicity, in this case we consider the both assets to be synchronous.

With Tick Time (Virtual Returns)

$$\begin{aligned}\hat{\rho}_{\text{tick}} &= \frac{\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij}\neq\emptyset\}}}{\sqrt{\sum_i (r_i^X)^2 \sum_j (r_j^Y)^2}} \\ &= \frac{(0.01\cdot 0.02)+(0.02\cdot -0.01)+(0.03\cdot 0.01)+(0.04\cdot -0.02)+(0.05\cdot 0.03)}{\sqrt{(0.01^2+0.02^2+0.03^2+0.04^2+0.05^2)\cdot(0.02^2+(-0.01)^2+0.01^2+(-0.02)^2+0.03^2)}} \\ &= \frac{0.001}{\sqrt{0.0055\cdot 0.0019}} = 0.309\end{aligned}$$

With Trade Time (Observed Returns)

$$\begin{aligned}\hat{\rho}_{\text{trade}} &= \frac{\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij}\neq\emptyset\}}}{\sqrt{\sum_i (r_i^X)^2 \sum_j (r_j^Y)^2}} \\ &= \frac{(0.01\cdot 0)+(0.02\cdot -0.01)+(0\cdot 0.01)+(0\cdot 0)+(0.12\cdot 0.01)}{\sqrt{(0.01^2+0.02^2+0^2+0^2+0.12^2)\cdot(0+0.0001+0.0001+0+0.0001)}} \\ &= \frac{0-0.0002+0+0+0.0012}{\sqrt{0.0149\cdot 0.0003}} = \frac{0.001}{\sqrt{0.00000447}} = 0.474\end{aligned}$$

We can observe that the correlation measured in trade time is bigger and introduces some spurious correlation.

3.2.2 Why can trade time be problematic

Intuitively, it is quite clear that tick time (remember, understanding tick time as the clock that increments each time there is a non-zero variation of the midquote between two trades) should be more precise than the trade time. We will later see this effect in a real market example.

If interested in more details behind this intuition, the reader can consult pages 84-98 of the book *The Econometrics of Financial Markets* by Campbell, Lo, and MacKinlay [1], which rigorously illustrate the problems in a more abstract setting.

3.3 Further metric definitions

We have already defined several metrics in the previous section. In the following, we will introduce additional metrics that will help us identify liquid assets.

- **Liquidity Statistics**: These help in the search for lead-lags as support:
  - The mean duration between consecutive trades during the event (**mean_duration_between_trades**). This gives us an idea of the frequency of information arrival.

$$\text{mean\_duration\_between\_trades} = \frac{1}{N-1}\sum_{i=1}^{N-1}(t_{i+1} - t_i) \tag{3.3}$$

    where t_i and t_{i+1} are the timestamps of consecutive trades, and N is the total number of trades during the event.
  - The mean bid-ask spread during the event (**mean_bid_ask_spread**).

$$\text{mean\_bid\_ask\_spread} = \frac{1}{N}\sum_{i=1}^{N}(a_i - b_i) \tag{3.4}$$

    where a_i is the ask price and b_i is the bid price at the time of the i-th trade, and N is the total number of trades during the event.
  - The percentage of trades that eat a level of the book. This can also be relevant, as these trades are precisely the ones that can propagate the signal of a news release.

$$\text{percentage\_of\_trades\_eating\_book} = \frac{\text{Number of trades displacing a bid or offer}}{N} \times 10 \tag{3.5}$$

    where N is the total number of trades during the event.

- **The ratio between the maximum dislocations of two contracts**: This does not really indicate the lead-lag, it would just correspond to a "beta" between the two contracts for that particular event.

3.3.1 Lead-Lag conditional to extreme events

In Section 3.2, we examined the Hayashi-Yoshida correlation function. Here, we explore a more nuanced version of this estimator.

Intuitively, larger movements of the leading asset tend to incorporate more information. Therefore, we may want to focus on the correlation between the future and the stock only when the future makes a sufficiently large move above a certain threshold, denoted as θ (where the units are measured in tick sizes). The threshold version of the Hayashi-Yoshida cross-correlation function, when asset X leads asset Y, is presented below.

$$\hat{\rho}(\ell) = \frac{\sqrt{N_\theta^X N_0^Y}}{N_\theta^{X,Y}(\ell)} \frac{\sum_{i,j} r_i^X r_j^Y 1_{\{O_{ij}^\ell \neq \emptyset\}} 1_{\{|r_i^X| \geq \theta\}}}{\sqrt{\sum_i (r_i^X)^2 1_{\{|r_i^X| \geq \theta\}} \sum_j (r_j^Y)^2 1_{\{|r_i^Y| \geq \theta\}}}} \tag{3.6}$$

where

$$N^\theta(k) = \sum_i \mathbf{1}\{|r_i^k| \geq \theta\}$$

$$N_\theta^{k,p}(\ell) = \sum_{i,j} \mathbf{1}\{|r_i^k| \geq \theta\} \mathbf{1}\{|r_j^p| \geq 0\} \mathbf{1}\{O_{ij}^\ell \neq \emptyset\}$$

A natural follow-up question is: What is the optimal threshold to consider? In other words, what threshold results in the most favorable lead-lag relationship between the leading asset and the lagging asset? While this is an important question, for now, we leave it aside and consider all moves, as this will still provide sufficient insight into market movements.

3.3.2 Explicit calculation of threshold cross-correlation function

In the following, we will use a toy example to explain the intuition behing the previous concept.

Asset X

Period	r_{it} (Tick Returns)
1	0.01
2	0.02
3	0.03
4	0.04
5	0.05

Table 3.3: Tick Returns (r_{it}) for Asset X

Asset Y

Period	r_{it} (Tick Returns)
1	0.02
2	-0.01
3	0.01
4	-0.02
5	0.03

Table 3.4: Tick Returns (r_{it}) for Asset X

$$\begin{aligned}\hat{\rho}(\ell) &= \frac{\sqrt{N_\theta^X N_\theta^Y}}{N_\theta^{X,Y}(\ell)} \frac{\sum_{i,j} r_i^X r_j^Y \mathbf{1}_{\{O_{ij}^\ell \neq \emptyset\}} \mathbf{1}_{\{|r_i^X| \geq \theta\}}}{\sqrt{\sum_i (r_i^X)^2 \mathbf{1}_{\{|r_i^X| \geq \theta\}} \sum_j (r_j^Y)^2}} \\ &= \frac{\sqrt{3 \times 5}}{15} \frac{(0.03 \times 0.01) + (0.04 \times -0.01) + (0.05 \times 0.03)}{\sqrt{(0.03^2 + 0.04^2 + 0.05^2)} \times \sqrt{(0.02^2 + (-0.01)^2 + 0.01^2 + (-0.02)^2 + 0.03^2)}} \\ &= \frac{1}{\sqrt{15}} \frac{0.0014}{0.707... \times 0.0435...} \\ &= 0.117 \end{aligned} \tag{3.7}$$

As we can see, this value is lower than the regular Hayashi Yoshida with Tick Time that we computed on section 3.2.1. Intuitively, this is telling us that the threshold θ of this example is removing a period of data (period 1) where the assets show high correlation.

Now, to further investigate how markets respond to lead-lag effects, we introduce lead-lag response functions.

3.3.3 Lead-Lag response functions

The cross-correlation provides insight into the presence of a lead-lag relationship. However, it does not reveal the properties of the lagger's response in terms of bid-ask spread or movement of the bid quote. The response of the lagger to the movement of the leading asset is crucial for constructing a successful trading strategy that overcomes the bid-ask spread and the associated transaction costs. Lead-lag response functions are defined as follows:

$$R_{v,\gtrless}(\ell, \theta) = \left\langle v_{t+\ell}^A - v_t^A \mid r_t^B \gtrless \theta \right\rangle \tag{3.8}$$

Here, A and B represent two different assets, where A is the lagging asset and B is the leading asset.

In this formula, v represents any relevant variable from the order book. For instance, v could be the bid-ask spread or the bid quote. Below we insert a very simple example of the calculation of a response function.

3.3.4 Explicit simple example of a response function

We first set the threshold in equation (12) to $\theta = 3$. Next, we consider two assets: a futures contract, F_I, on an index, and a stock, S, which is a component of the index. In this scenario, we assume that the futures contract F_I acts as the leader due to its higher liquidity. The evolution of the futures contract F_I is shown in the table below:

Table 3.5: Evolution of the Leader, F_I

Time	Midquote	Change (r_t^{future})
1	100.0	0
2	101.0	1
3	105.0	4.0
4	106.0	1.0
5	105.0	-1.0
6	106.0	1.0
7	104.0	-2.0
8	105.0	1.0

We observe that at time $t = 3$, there is a movement in the leader F_I with a magnitude greater than the threshold θ. Consequently, we compute the threshold response function starting from $t = 3$ for the lagger S. Assume that the lagger S evolves according to the following table:

Table 3.6: Bid-Ask Data and $R(\ell, \theta = 3)$ Values

Time	Bid	Ask	Bid-Ask Spread	ℓ	$R_{\text{Bid-Ask}}(\ell, \theta = 3)$
1	10.10	10.20	0.10	NA	NA
2	10.15	10.30	0.15	NA	NA
3	10.15	10.25	0.10	0	0.00
4	10.10	10.25	0.15	1	0.05
5	10.50	11.00	0.50	2	0.40
6	10.60	11.15	0.55	3	0.45
7	10.65	11.15	0.50	4	0.40
8	10.65	11.10	0.45	5	0.35
9	10.75	11.20	0.45	6	0.35

We can then see that the response function evolves as follows after the move of the leader at time $t = 3$.

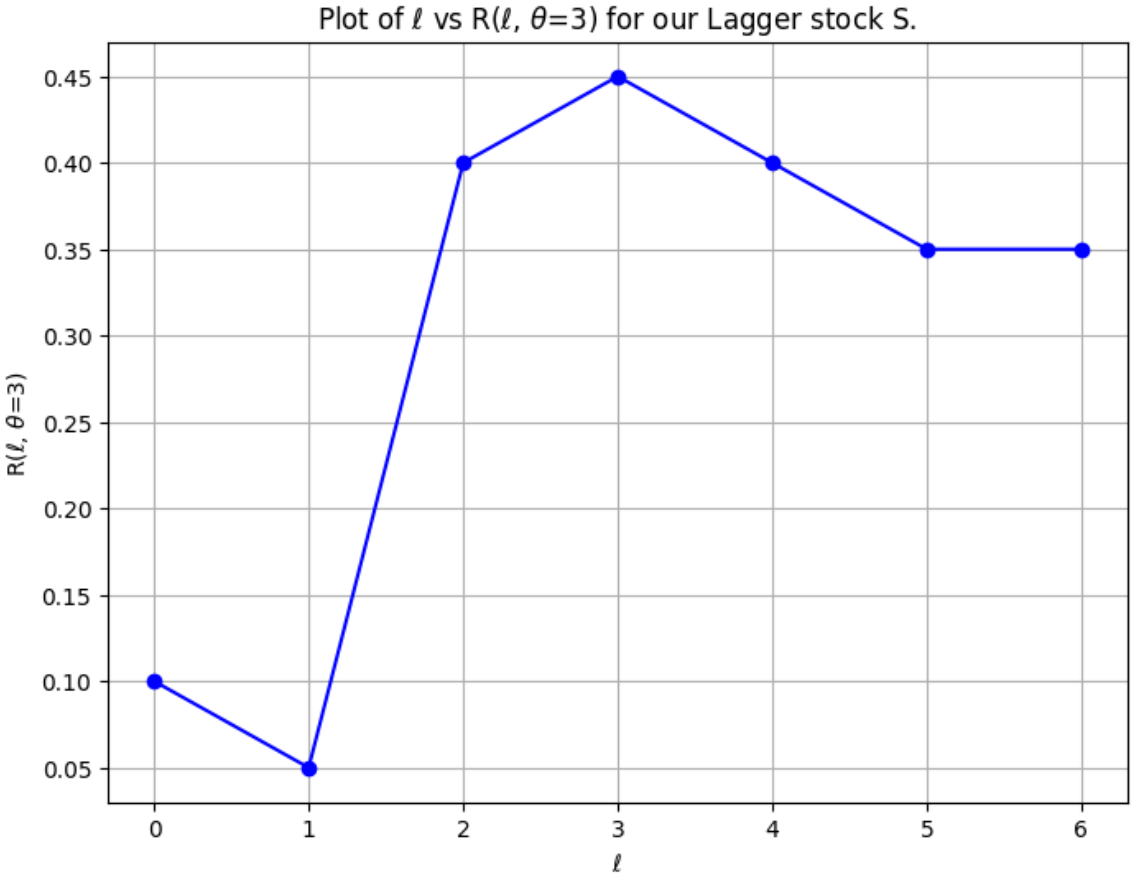


Figure 3.1: Response Function

3.3.5 SOFR vs FF response function example

In this section, we provide a detailed analysis of the lead-lag relationship between SOFR and FF following a news release, focusing on metrics and their implications. This serves as a comprehensive example, demonstrating how the toolbox can be used to calculate and interpret key statistics. By examining the metrics such as bid-ask spread, trade durations, and other liquidity-related statistics, we can evaluate the dynamics between these instruments.

Lead-Lag Relationship Analysis

We investigate the lead-lag relationship between SOFR and FF at 12:30 after a CPI release. Using the HY Yoshida methodology, we identify a lag of approximately 0.03 seconds in the first 5 seconds after the data is released. This lag suggests that SOFR leads FF due to its superior liquidity and faster reaction to the news. Additionally, we analyze the bid-ask spread and midquote-price behavior after the CPI release to understand the impact of this event on market dynamics.

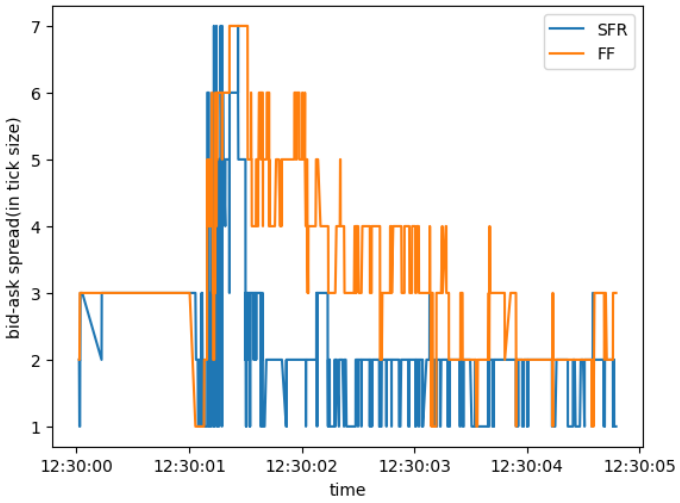


Figure 3.2: Plot of bid-ask spread of SOFR and FF in the five seconds after the CPI release on 12-03-2024. We observe a large spike in bid-ask spread in both SOFR and FF, corresponding to the increased volatility directly after the CPI release. The bid-ask spread of SOFR reverts more quickly than that of FF due to its superior liquidity.

Profitability Analysis of a Simple Strategy

Theoretically, if we could predict the movement of the lagger (FF) based on the leader (SOFR), we could profit by buying or selling the lagger when the leader

moves. However, to make this strategy viable, the updated lagger bid price must exceed the original lagged ask price (or vice versa). Below, we provide a plot demonstrating that in the given timeframe of 0.03 seconds, the bid price of FF never surpasses the original ask price. This shows that a simple directional strategy would not be profitable under these conditions.

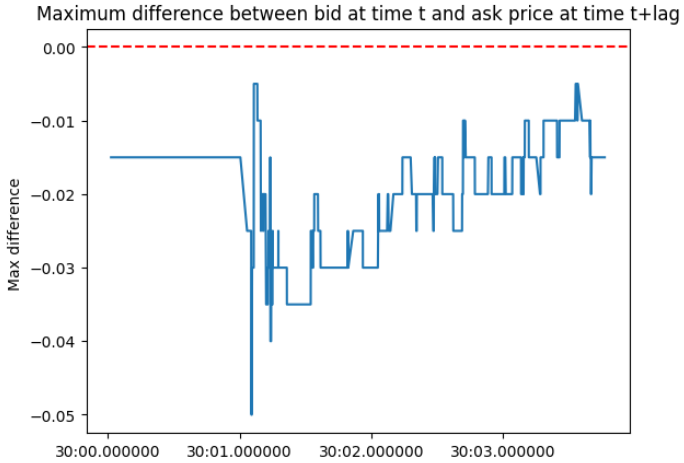


Figure 3.3: Plot depicting the difference between the bid at time t and the maximum ask in the next 30 microseconds (the lag of FF to SOFR). This simulates the strategy of selling when the leader decreases and buying when the lagger follows. The graph shows that a simple directional strategy is not profitable, as the values never exceed 0.

Detailed Metric Analysis

To further enrich this example, we calculate and interpret liquidity metrics for SOFR and FF:

Metric	Category	Value	Unit
FF_mean_bid_ask_spread	liquidity	0.006	bp
FF_mean_duration_between_trades	liquidity	847.701	ms
FF_median_duration_between_trades	liquidity	9.125	ms
FF_trades_wipeout_level_perc	liquidity	0.147	%
SOFR_mean_bid_ask_spread	liquidity	0.006	bp
SOFR_mean_duration_between_trades	liquidity	176.060	ms
SOFR_median_duration_between_trades	liquidity	2.074	ms
SOFR_trades_wipeout_level_perc	liquidity	0.267	%

Table 3.7: Liquidity metrics for SOFR and FF

Additionally, we calculate the ratio of maximum dislocations (betas) for SOFR

and FF across different spreads. This metric reflects the relative responsiveness of the two instruments:

Spread	Metric	Mean
Rois 2 M24_snp	beta_SFR_FF	1.385
Rois 2 U24_snp	beta_SFR_FF	1.137
Rois 2 Z24_snp	beta_SFR_FF	1.140

Table 3.8: Beta SFR FF for different spreads

Conclusion

This comprehensive example demonstrates the utility of the toolbox in evaluating lead-lag relationships, calculating key liquidity metrics, and assessing the feasibility of trading strategies. By combining theoretical insights with empirical data, this analysis highlights the practical applications of the metrics in understanding market dynamics.

3.4 Detection and measurement of misleading price reactions

Misleading price reactions occur when an asset's price initially moves in the opposite direction of the expected response to an event, only to reverse and align with the anticipated trend. Such anomalies can obscure market signals and complicate decision-making, making their detection and analysis critical. In this section, we investigate these misleading dynamics, their causes, and their implications for trading strategies and market efficiency.

These reactions are particularly relevant in the context of economic releases, where market participants often misinterpret the initial impact, leading to temporary mispricings. Moreover, misleading price dynamics could be intentionally induced through spoofing or other market-abusive activities, further highlighting the importance of identifying and understanding these behaviors.

Given an economic release, a **misled price reaction** is any initial movement in the opposite direction of the expected one, followed by a reversal and then the normal movement. We formally define it as follows:

Definition 1 (Misled Price Reaction) *Let $p(t)$ denote the price of an asset as a function of time $t \in [0, T]$, where T is the time horizon of interest. Given an economic event occurring at t_0, we define the following terms:*

- ***Reference Price:*** $p_0 = p(t_0^-)$, *the price just before the event.*
- ***Maximum Dislocation:*** $D_{\max} = \max_{t\in[t_0,T]} |p(t) - p_0|$, *the maximum absolute movement in the price after the event.*
- ***Normal Direction:*** *The direction of the maximum dislocation, defined as* $sgn(p(t_D) - p_0)$, *where* $t_D = \arg\max_{t\in[t_0,T]} |p(t) - p_0|$.

A ***misled price reaction*** *is defined as a movement in the price function* $p(t)$ *satisfying the following criteria:*

- ***Opposite Direction:*** *There exists a time interval* $[t_1,t_2]$ *with* $t_0 \leq t_1 \leq t_D$ *such that* $sgn(p(t_1) - p(t_0)) \neq sgn(p(t_D) - p(t_0))$.
- ***Duration:*** *The duration of the movement is less than one second, i.e.,* $t_2 - t_1 <$ 1 *second.*
- ***Number of Ticks:*** *The movement consists of at least* n *ticks, where* n *is a predetermined minimum number of ticks required.*
- ***Magnitude:*** *There exists* $t_1 \in [t_0,t_D[$ *such that:*

$$\begin{cases} p(t_1) - p_0 > M \cdot (p_0 - p(t_D)) & \text{if } sgn(p(t_D) - p_0) < 0, \\ p_0 - p(t_1) > M \cdot (p(t_D) - p_0) & \text{if } sgn(p(t_D) - p_0) > 0. \end{cases}$$

, where $M \in [0,1]$ *determines the size of the misled price reaction.*

Observations:

- In the examples provided here, 1 tick was the minimum requisite, but we believe a higher number of ticks would reduce uncertainty about whether the move is a genuine misled price reaction. It is worth investigating whether a fixed number of ticks should be used across markets, or if this should depend on the tick size of the particular market. For example, in Treasuries, 1 tick seems too small; a larger number of ticks would be necessary.

  We have also used a magnitude of $M = 0.05$ for our analysis.

- Stricter conditions on volume size could also be beneficial. It is not the same to "walk the book" (progressively creating new bid/ask offers and consuming the level) as it is to hit the market with current bid/ask offers. We expect a misled price reaction to occur more likely in the former scenario.

3.5 Metrics used

Of these price misleds, we are going to take three metrics:

- **Is_there_price_misled**: This metric indicates whether there has been a deception in a contract during a specific event, according to the definition in section 2. It is defined as the percentage of price misleds observed over the total number of events analyzed.

  Formally, let N be the total number of macro data points (events) considered.

  Let i index these events such that $i \in \{1,2,\ldots,N\}$. We consider

$$X_i = \begin{cases} 1 & \text{if there is a price misled during event } i \\ 0 & \text{if there is no price misled during event } i \end{cases}$$

  The total number of price misleds is then given by $\sum_{i=1}^{N} X_i$.

  The **Is_there_price_misled** metric is calculated as:

$$\text{Is\_there\_price\_misled} = \left(\frac{1}{N} \sum_{i=1}^{N} X_i \right) \times 100$$

  To illustrate, we have conducted an analysis with contracts from multiple curves (SFR, SFI, ER generics 3, 7, 11, and two-year treasuries from the US and Europe) during the CPI and NFP from September 2023 to May 2024. Here, $i = \{1,2,\ldots,N\}$ indexes these events.

  As we can see, the curves that suffer the highest percentage of deceptions are SOFR, treasuries (both US and EUR), and to a lesser extent FF and ER. Lastly, there is Sonia, which does not suffer deceptions (as we have defined them).

- **price_misled_nc**: This metric gives us the net change (from the reference price, in the time interval we are choosing) of the deception. We consider a price misled when the price "should go" the other way due to a news release, but it initially goes in the contrary direction. The time interval chosen for this analysis has been 3 seconds. For this metric, we have not required the last two conditions (at least one tick of movement (from the reference price), and movement greater than 5% of the maximum dislocation), as we want to measure deceptions in absolute value.

  Let t_0 be the initial time, t_1 be the time at which the maximum dislocation of the deception happens, and T_t be the time the analysis ends. The time t_1 is

Contract	Metric	Strategy	% of price misleds
SFRG3NR	SFRG_is_there_price_misled		0.368421053
SFRG7NR	SFRG_is_there_price_misled		0.368421053
SFRG11NR	SFRG1_is_there_price_misled		0.315789474
DUG1NR	DUG_is_there_price_misled		0.222222222
TUG1NR	TUG_is_there_price_misled		0.210526316
UXYG1NR	UXYG_is_there_price_misled		0.157894737
FFG5NR	FFG_is_there_price_misled		0.105263158
ERG11NR	ERG1_is_there_price_misled		0.052631579
ERG3NR	ERG_is_there_price_misled		0
ERG7NR	ERG_is_there_price_misled		0
SFIG11NR	SFIG1_is_there_price_misled		0
SFIG3NR	SFIG_is_there_price_misled		0
SFIG7NR	SFIG_is_there_price_misled		0

Table 3.9: Percentage of Price Misleds for Each Contract

defined as the time when the price reaches its maximum dislocation D in the direction opposite to the expected movement, where

$$t_1 = \underset{t \in [t_0, T_t] \text{ and } P(t) \text{ in the opposite direction}}{\arg\max} |P(t) - P(t_0)|$$

The net change of the deception, **price_misled_nc**, is then given by:

$$\text{price\_misled\_nc} = |P(t_0) - P(t_1)|$$

where $P(t)$ denotes the price at time t. For the same analysis above, we obtain:

This net change is very important as it measures how much we would need to space out entries (in a fading or lead-lag/momentum strategy) to avoid being deceived. We can look at the average, but the maximum is more indicative. Several points:

- In the seventh generic of SOFR (SFRG7NR), we observe a significant misled price reaction (6bp) during one of the events. This corresponds to the January CPI, which was high.
- We also have significant deceptions in US treasuries (in magnitude, similar to the SOFR above).
- Lastly, there are medium/low deceptions in European treasuries, ER,

Spread	Metric	Strategy	Mean	Max
UXYG1NR	UXYG_price_misled_nc		0.090460526	0.421875
TUG11NR	TUG_price_misled_nc		0.019942434	0.0859375
SFRG7NR	SFRG_price_misled_nc		0.015789474	0.06
SFRG11NR	SFRG1_price_misled_nc		0.012105263	0.05
SFRG3NR	SFRG_price_misled_nc		0.013684211	0.05
ERG11NR	ERG1_price_misled_nc		0.002631579	0.035
FFG5NR	FFG_price_misled_nc		0.002105263	0.025
DUG1NR	DUG_price_misled_nc		0.005833333	0.03
SFIG3NR	SFIG_price_misled_nc		0.003157895	0.02
SFIG7NR	SFIG_price_misled_nc		0.002368421	0.02
SFIG11NR	SFIG1_price_misled_nc		0.002368421	0.015
ERG7NR	ERG_price_misled_nc		0.000526316	0.01
ERG3NR	ERG_price_misled_nc		0.0003125	0.005

Table 3.10: Price Misled NC Metrics for Various Spreads

and Sonia. It may seem strange that Sonia appears in this table with deceptions >0, but this is due to the relaxation of the restrictions. Indeed, these deceptions last more than a second, where theoretically a machine could have closed the position before the "real" movement of the data (but not a human).

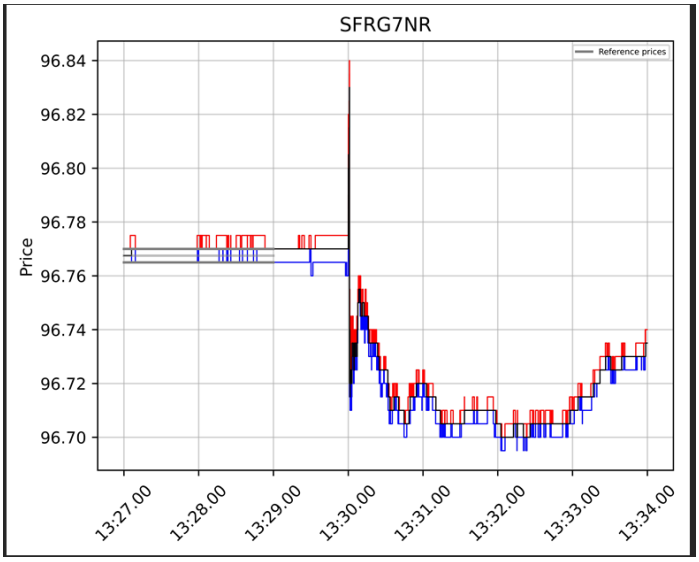


Figure 3.4: Price misled in the SOFR 7th generic contract following the January 2024 CPI release (a high CPI), showing more than 6 basis points of price misled. The initial reaction of this contract was upward. In fact, this price misled was as large as the downward movement:

- **price_misled_ratio**: As we have seen with the previous metric, there are deceptions so large relative to the maximum dislocation that it would be impossible not to be deceived no matter how much we separate the quotes. Therefore, the metric provides us with the ratio of the size of the deception to the size of the maximum dislocation.

Contract	Metric	Strategy	Max
DUG1NR	DUG_price_misled_ratio		0.888888889
SFRG7NR	SFRG_price_misled_ratio		0.888888889
ERG11NR	ERG1_price_misled_ratio		0.823529412
SFRG11NR	SFRG1_price_misled_ratio		0.740740741
SFIG3NR	SFIG_price_misled_ratio		0.727272727
SFRG3NR	SFRG_price_misled_ratio		0.642857143
TUG1NR	TUG_price_misled_ratio		0.637681159
SFIG11NR	SFIG1_price_misled_ratio		0.571428571
FFG5NR	FFG_price_misled_ratio		0.526315789
UXYG1NR	UXYG_price_misled_ratio		0.506024096
SFIG7NR	SFIG_price_misled_ratio		0.5
ERG7NR	ERG_price_misled_ratio		0.307692308
ERG3NR	ERG_price_misled_ratio		0.153846154

Table 3.11: Price Misled Ratio for Various Contracts

Formally, we can write

$$\max_{t\in[0,T]} \frac{M_t}{\text{MD}}$$

where

0 : time the news release is given.
T : time the maximum dislocation ends.
MD : size of the maximum dislocation.
M_t : maximum movement in the opposite direction.

3.6 Detection of Information Arrival for Each Product

Detecting when information from an event arrives is a crucial problem, as it allows us to compare contracts and identify **momentum opportunities** through relative value signals. While news events often have fixed timestamps, such as economic releases, relying solely on these timestamps may not accurately capture how the market processes and reacts to this information. Instead, focusing on actual market activity offers a more reliable measure of information reception and reaction.

The approach we have followed here is designed to detect information arrival based on market activity:

- We take an instrument during an **event**.
- For that instrument, we use the **trades** occurring between the start of the event (e.g., CPI at 2:30:00) and a specified window. For this analysis, the window is 3 seconds.
  - This approach benefits from the fact that the data is always released at a known time, typically at the start of the hour.
- We aggregate the traded volumes by milliseconds. This allows us to maintain granularity without sacrificing stability in the analysis.
- A **quantile** q is then calculated for these aggregated volumes by milliseconds. By default, we set this to the 80th percentile (i.e., $q = 0.8$).
- We define the start of the event (for this instrument) as the first timestamp where the traded volume is greater than or equal to this percentile.

By using market activity rather than relying solely on news release timestamps, we can better capture the actual moment when information is absorbed by the market. If signed trades are available, they could further refine this detection by allowing us to assess whether the direction of the first trades reflects the broader market reaction. For example, the initial burst of signed trades could provide an early signal of the market's intended direction, offering a valuable insight into price dynamics immediately following the information arrival.

Formal definition

Let $V(t)$ denote the traded volume of an instrument at time t, where $t \in [t_0, t_0 + \Delta t]$ and $\Delta t = 3$ seconds.

- **Aggregation:** Aggregate the traded volumes by milliseconds. Define $V_m(t_i)$ as the aggregated volume at millisecond t_i, where $t_i \in [t_0, t_0 + \Delta t]$ and t_i are discrete timestamps at millisecond intervals.

$$V_m(t_i) = \sum_{t \in [t_i, t_i + 1 \text{ ms})} V(t)$$

- **Quantile Calculation:** Calculate the quantile q (default $q = 0.8$) of the aggregated volumes $V_m(t_i)$. Let Q_q denote this quantile value.

$$Q_q = \inf\{x \in \mathbb{R} \mid P(V_m \leq x) \geq q\}$$

- **Detection of Information Arrival:**

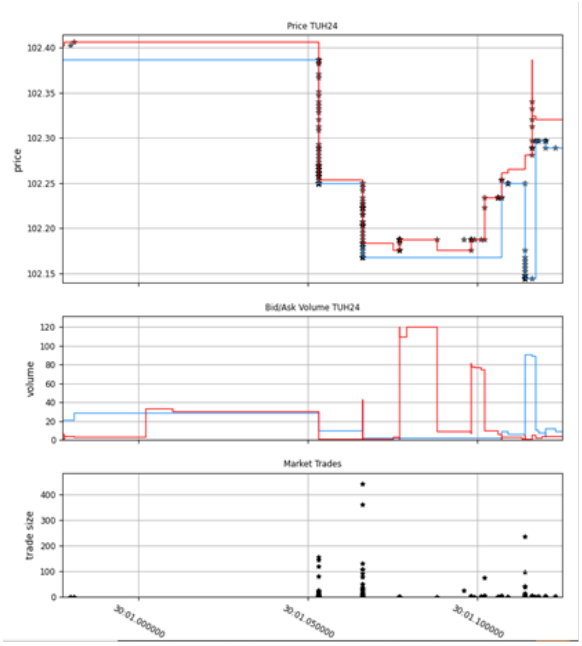
  This metric is needed for identifying the precise moment when the market begins to react to new information, moving beyond merely relying on event timestamps. This metric helps to pinpoint the actual time at which information is absorbed by the market. Such a measure is essential for comparing reactions across instruments, evaluating market efficiency, and uncovering potential momentum opportunities.

  Define the start of the event (for this instrument) as the first timestamp t^* where the aggregated volume exceeds or is equal to the quantile Q_q:

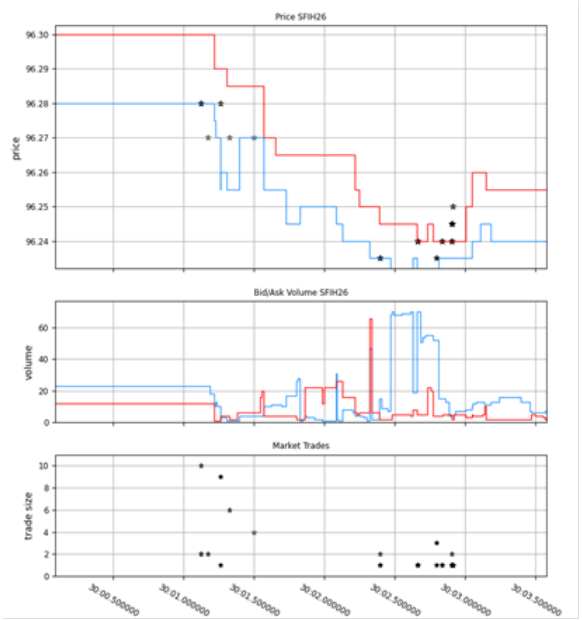
$$t^* = \min\{t_i \mid V_m(t_i) \geq Q_q\}$$

With this methodology, reminiscent of the one proposed in section 3.4 of the previously mentioned paper, the important aspect is the choice of q:

- A q that is too high only selects the millisecond with the most trades during the event. This is ideal for more **liquid markets** where large participants react first to the data, and where there may be low-volume trades that mislead the metric. Therefore, for US treasuries and SOFR, we choose $q = 0.98$. Here, we see an example of the volume pattern (trade size graph) in the two-year.



- A lower q is useful for more illiquid markets where the highest volume does not necessarily concentrate at the arrival of information. Therefore, for Sonia, Euribors, CORRA, and Canadian treasuries, we choose $q = 0.8$. Here, we see an example in Sonia:



To illustrate the methodology and compare the arrival of information across different products, we analyze the March CPI release as an example. This example highlights the application of the lead-lag detection method, focusing on how

quickly various products respond to the release of economic data. The sample period includes a set of financial instruments traded across different markets, such as futures on various US Treasuries, SOFR futures, and other key instruments listed in the table, enabling us to evaluate their relative reactions. The purpose of this analysis is to identify patterns in the timing of information absorption.

Using the timestamps obtained from our methodology, we order the products based on the arrival time of information. This allows us to identify the sequence in which the data reaches various markets.

The obtained results for the March CPI release are as follows:

Product	UXYM24	SFR26	SFR4	SFR25	GCM4	TUM24	CNM4	FFF25	DUM24	O
Time	1.028	1.056	1.056	1.056	1.056	1.056	1.06	1.084	1.085	1.0

Table 3.12: Lead-Lag Results for US CPI Mar 24 Products

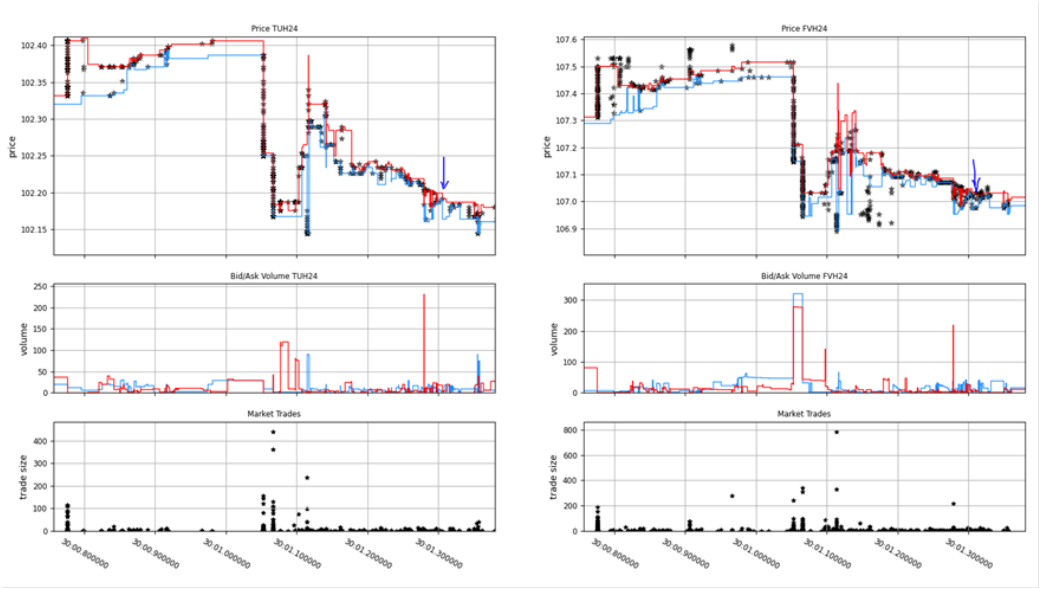
The timestamps are presented in the format "Seconds.Milliseconds," representing the time elapsed after the release of the CPI data.

- **US products lead:** As expected, US-based instruments—such as gold (GC), US treasuries, and SOFR—receive the information first. Notably, there is no significant timing difference observed among these products, suggesting simultaneous receipt of information at the millisecond level.
- **Intermediate responses:** The next group to respond includes Federal Funds (FF) and Canadian treasuries. These instruments react slightly later than their US counterparts but still exhibit relatively fast absorption of the data.
- **European markets lag:** Information reaches European instruments, such as treasuries (DU and OE), with a noticeable delay compared to US and Canadian products. Within Europe, the reaction starts with treasuries and is followed by Sonia and Euribors.
- **Slowest reactions:** CORRA and the 5-year Canadian treasury (XQ) are the last to respond. For CORRA, the delay can largely be attributed to illiquidity, as its reaction was based on minimal trades and movements primarily observed in the order book rather than actual trades.

This example highlights the importance of the chosen quantile q in accurately detecting the arrival of information, especially in distinguishing between liquid and illiquid markets. It also provides insights into the flow of information across

different markets, emphasizing the role of market structure and participant behavior in determining reaction times.

We have compared the obtained times with the times provided by a trading firm for the start of the events. In this case, the trading firm receives the data via AlphaFlash and records the timestamp on its London machine. A significant example is the February CPI. They record the event at 13:30:01.316, while our metric detects it at 13:30:00.776. As we see in the following graph, the event start is well-detected by our metric (the vertical column of sell trades), while the trading firm receives the event at the blue arrow.



3.7 Conclusions and further work regarding lead-Lag relationships

We have explored numerous metrics throughout this study. Determining which metrics will act as leaders, which will serve as support, and which should be discarded remains an area for future work.

One surprising finding is the high percentage of price misleds occurring around important news releases according to our definition. Future work includes refining and imposing stricter conditions on volume and tick movements to eliminate false positives.

Another area for further investigation is understanding why, according to our definition, price misleds are not observed in certain markets, such as SONIA, while they are prevalent in others, like SOFR and TU. Liquidity may be a contributing

factor, but the pronounced differences suggest that our definition of price misled may need adjustment. In the lead-lag part, some observations are from the already mentioned distinction between trade time and tick time. It may be interesting to look for more lags, not only the one that maximizes the metric. Perhaps a mean of, for example, the last five largest lags could help minimize noise in the outputs. The exact number of lags to use should also be discussed.

Chapter 4

Hasbrouck's metric for price discovery

Price discovery has been extensively analyzed in the literature through various methodologies. Hasbrouck introduced his Information Share (IS) measure, which quantifies the contribution of each market to the efficient price. Hasbrouck's IS measure evaluates the proportional contribution of each market's innovations to the variance of the efficient price. It provides a way to quantify the relative importance of different markets in the price discovery process. This measure has been widely applied in studies examining the role of various trading venues in reflecting new information.

The structure of the chapter is as follows: In the first six sections, we introduce the theoretical framework underlying Hasbrouck's model. Next, we analyze the behavior of the metric using simulated cointegrated data. Once we have developed a solid understanding of this example, we will apply the model to a real-world case in the futures market and draw conclusions from the analysis.

4.1 Cointegration in a Vector Autoregression (VAR) model

4.1.1 Vector Autoregression Model(VAR)

A Vector Autoregression (VAR) model is a statistical model used to capture the linear interdependencies among multiple time series. In a VAR model, each variable is modeled as a linear function of its own lagged values, as well as the lagged values of all other variables in the system.

For a vector of n time series variables $X_t = \begin{pmatrix} x_{1t} \\ x_{2t} \\ \vdots \\ x_{nt} \end{pmatrix}$, a VAR($p$) model can be written as:

$$X_t = A_1 X_{t-1} + A_2 X_{t-2} + \cdots + A_p X_{t-p} + u_t$$

where:

- A_i are $n \times n$ coefficient matrices for $i = 1, \ldots, p$.
- u_t is an $n \times 1$ vector of white noise error terms, which are assumed to be normally distributed with zero mean and constant covariance matrix Σ_u.

4.1.2 Cointegration in a VAR Model

Intuitively, cointegration refers to a situation where multiple time series are individually non-stationary, but a linear combination of them is stationary. In other words, even though the individual series may wander widely over time, there exists a long-term equilibrium relationship between them.

In the context of a VAR model, cointegration can be analyzed using the concept of a Vector Error Correction Model (VECM), which is a reparameterization of the VAR model that explicitly incorporates the cointegrating relationships.

4.1.3 Vector Error Correction Model (VECM)

The VECM representation for a VAR(p) model is given by:

$$\Delta X_t = \Pi X_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \Delta X_{t-i} + u_t$$

where:

- $\Delta X_t = X_t - X_{t-1}$ represents the first difference of the series.
- $\Pi = \sum_{i=1}^{p} A_i - I$ is an $n \times n$ matrix that contains information about the long-run relationships between the variables.
- Γ_i are $n \times n$ matrices capturing short-run dynamics.
- u_t is the vector of error terms.

The key component of the VECM that indicates cointegration is the matrix Π. If the rank of Π is reduced (i.e., $0 < \text{rank}(\Pi) = r < n$), then there are r cointegrating relationships among the n variables in X_t [8].

- If Π has full rank (i.e., $r = n$), all variables are stationary.
- If Π has zero rank (i.e., $r = 0$), there are no cointegrating relationships, and the VAR model is in differences (a standard VAR model).

When $0 < r < n$, the matrix Π can be decomposed as:

$$\Pi = \alpha\beta'$$

where:

- β is an $n \times r$ matrix of cointegrating vectors, representing the long-term equilibrium relationships between the variables.
- α is an $n \times r$ matrix representing the adjustment coefficients, indicating how the variables adjust towards the long-term equilibrium.

As we will see, the methodologies we will study place special importance on the α coefficient.

4.1.4 Simple example: bivariate case

For a bivariate system $X_t = \begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix}$, suppose $r = 1$ (i.e., there is one cointegrating relationship). Then:

$$\Pi = \alpha\beta' = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \begin{pmatrix} \beta_1 & \beta_2 \end{pmatrix}$$

This implies:

$$\Pi X_{t-1} = \alpha_1(\beta_1 x_{1,t-1} + \beta_2 x_{2,t-1}) + \alpha_2(\beta_1 x_{1,t-1} + \beta_2 x_{2,t-1})$$

The expression $\beta_1 x_{1,t-1} + \beta_2 x_{2,t-1}$ represents the cointegrating relationship (the long-term equilibrium), and α_1 and α_2 represent how each variable adjusts to deviations from this equilibrium.

4.2 Cointegration in the Moving Average (MA) representation

Better insight into the concept of cointegration can be grasped if we move to the MA, and express the process as function of innovations that drives it, see [8].

4.2.1 Moving Average representation(VMA)

For a vector of n time series variables $X_t = \begin{pmatrix} x_{1t} \\ x_{2t} \\ \vdots \\ x_{nt} \end{pmatrix}$, the Moving Average (MA) representation can be expressed as:

$$X_t = C(L)u_t$$

where:

- $C(L)$ is a matrix polynomial in the lag operator L.
- u_t is an $n \times 1$ vector of white noise errors (innovations).

This can be rewritten to its integrated form as:

$$X_t = \mu + C(1)\sum_{i=1}^{t} u_i + C^*(L)u_t$$

where:

- μ is a deterministic term (possibly including a drift).
- $C(1)$ is the long-run impact matrix, representing the cumulative effect of shocks on the levels of the series. It is the sum of the moving average coefficients.
- $C^*(L)$ represents the transitory components, capturing the effects that dissipate over time. $C^*(L)$ is a matrix polynomial in the lag operator L.

4.2.2 Common stochastic trends

In the MA representation, the term $C(1)\sum_{i=1}^{t} u_i$ represents the common stochastic trends in the system. These trends are responsible for the long-term non-stationarity in the individual series.

If X_t is cointegrated, the rank of the matrix $C(1)$ will be less than n. Specifically, if there are r cointegrating relationships, the rank of $C(1)$ will be $n - r$. This implies that there are $n - r$ common stochastic trends driving the non-stationarity in X_t, and r linearly independent cointegrating vectors that form stationary combinations.

Thus, cointegration in the Moving Average representation indicates that there are fewer common stochastic trends than the number of series in the system, and that these series are bound together by long-term equilibrium relationships. We will later discuss the decomposition of the cointegrated process into permanent and transitory components, but first, we will briefly examine the implications of cointegrated processes in financial prices and Hasbrouck's methodology.

4.3 Implications of cointegrated processes in financial prices

Cointegration in financial prices is crucial when dealing with related assets like spot and futures contracts, stocks of related companies, or currency pairs. It implies a long-term equilibrium relationship, even though individual price series may show short-term non-stationary behavior.

This equilibrium relationship is maintained by economic forces such as arbitrage, which prevent prices from diverging indefinitely. The concept is closely tied to the Error Correction Mechanism (ECM), which explains how deviations from equilibrium are corrected over time.

For risk management, cointegration provides confidence in portfolio stability, as related assets are less likely to experience large price divergences. It also highlights the limitations of traditional lead-lag analysis, which can be misleading when assets are cointegrated.

Finally, cointegration has implications for price discovery and market efficiency, as it ensures that prices across different markets or instruments for the same asset reflect the same underlying information, maintaining market efficiency.

4.4 Lead-lag analysis in the presence of cointegration:

In the following, we demonstrate how lead-lag analysis in the presence of cointegration might lead to spurious conclusions. We illustrate this with a simple process. Assume we have two cointegrated price series P_t and Q_t, representing the

prices of the same asset in two different markets. The Vector Error Correction Model (VECM) for these two series is given by:

$$\Delta P_t = \alpha_1(Q_{t-1} - P_{t-1}) + \epsilon_{1t} \tag{1a}$$

$$\Delta Q_t = \alpha_2(Q_{t-1} - P_{t-1}) + \epsilon_{2t} \tag{1b}$$

where:

- $\Delta P_t = P_t - P_{t-1}$ and $\Delta Q_t = Q_t - Q_{t-1}$ are the changes in the prices.
- $Q_{t-1} - P_{t-1}$ is the cointegrating relationship (i.e., the error correction term).
- α_1 and α_2 are the adjustment coefficients for P_t and Q_t, respectively, which describe how each price adjusts to deviations from the long-term equilibrium.
- ϵ_{1t} and ϵ_{2t} are white noise error terms.

4.4.1 Misspecification of models

To understand how this leads to a misleading interpretation of lead-lag relationships, we apply recursive substitution to equation $(1b)$, which describes ΔQ_t.

First, recall equation $(1b)$:

$$\Delta Q_t = \alpha_2(Q_{t-1} - P_{t-1}) + \epsilon_{2t}$$

Now, expand Q_{t-1} and P_{t-1} in terms of their lagged values:

$$Q_{t-1} = Q_{t-2} + \Delta Q_{t-1}$$

$$P_{t-1} = P_{t-2} + \Delta P_{t-1}$$

Substituting these into the error correction term $(Q_{t-1} - P_{t-1})$ gives:

$$Q_{t-1} - P_{t-1} = (Q_{t-2} + \Delta Q_{t-1}) - (P_{t-2} + \Delta P_{t-1})$$

This simplifies to:

$$Q_{t-1} - P_{t-1} = (Q_{t-2} - P_{t-2}) + (\Delta Q_{t-1} - \Delta P_{t-1})$$

Now substitute this back into the expression for ΔQ_t:

$$\Delta Q_t = \alpha_2[(Q_{t-2} - P_{t-2}) + (\Delta Q_{t-1} - \Delta P_{t-1})] + \epsilon_{2t}$$

Expanding this:

$$\Delta Q_t = \alpha_2(Q_{t-2} - P_{t-2}) + \alpha_2(\Delta Q_{t-1} - \Delta P_{t-1}) + \epsilon_{2t}$$

This process can be continued by expanding $Q_{t-2} - P_{t-2}$ in terms of Q_{t-3} and P_{t-3}:

$$Q_{t-2} - P_{t-2} = (Q_{t-3} + \Delta Q_{t-2}) - (P_{t-3} + \Delta P_{t-2})$$

Substituting this back into our expression for ΔQ_t:

$$\Delta Q_t = \alpha_2[(Q_{t-3} - P_{t-3}) + (\Delta Q_{t-2} - \Delta P_{t-2})] + \alpha_2(\Delta Q_{t-1} - \Delta P_{t-1}) + \epsilon_{2t}$$

Simplifying:

$$\Delta Q_t = \alpha_2(Q_{t-3} - P_{t-3}) + \alpha_2(\Delta Q_{t-2} - \Delta P_{t-2}) + \alpha_2(\Delta Q_{t-1} - \Delta P_{t-1}) + \epsilon_{2t}$$

This shows that lead-lag analysis is misleading. The recursive substitution shows that ΔQ_t is influenced by past differences between Q_t and P_t, as well as past changes in Q_t and P_t. However, these influences are not just one-way (i.e., P_t leading Q_t)—they are driven by the cointegrating relationship. In other words, ΔQ_t is responding to the disequilibrium between Q_{t-1} and P_{t-1}, and the adjustments involve terms that appear to "lag" or "lead" but are in fact part of a joint adjustment process to restore equilibrium.

This finite-lag model assumes convergent representations where none exist. We know that only lags up to order 2 are relevant in the true model, but in the current representation, many more lags appear. This can lead to spurious correlations among the coefficients, making some past lags seem more relevant than they actually are. Accurate modeling requires considering infinite lags, which are impractical for empirical analysis but essential for correct specification.

To summarize, addressing misspecification requires recognizing cointegration and using VECMs to incorporate both short-term and long-term relationships. This approach avoids misleading inferences and provides a more accurate understanding of price dynamics in multiple markets.

4.4.2 Hasbrouck's framework

Hasbrouck's argument is that in the presence of cointegration, using autoregressive models to assess lead-lag relationships can be misleading. The recursive expansion of the VECM shows that the price changes are influenced by multiple past values of both series, reflecting their interdependence due to the cointegrating relationship. This undermines the simple interpretation of causality that lead-lag analysis might suggest. Instead, the changes in both series are better understood as part of a system where both are moving together to maintain a long-term equilibrium relationship.

Hasbrouck's analysis of price discovery in cointegrated markets focuses on understanding how much each market contributes to the formation of the efficient price, which is the price implied by the long-term equilibrium relationship.

- **Information Share (IS)**: Hasbrouck's Information Share (IS) metric quantifies the contribution of each market to the variance of the innovations in the common factor (the efficient price).
- **Permanent-Transitory Decomposition**: The efficient price can be seen as the permanent component of the price process, while deviations from this price are transitory and tend to revert over time.

4.5 Permanent-Transitory decomposition using the integrated Moving Average form

Starting from the Vector Error Correction Model (VECM):

$$\Delta X_t = \alpha \beta' X_{t-1} + \sum_{i=1}^{k} \Gamma_i \Delta X_{t-i} + u_t$$

This VECM can be transformed into its Vector Moving Average (VMA) representation:

$$\Delta X_t = C(L) u_t$$

where $C(L)$ is a matrix polynomial in the lag operator L. By summing the moving average coefficients, the integrated form of this moving average representation (which accumulates the shocks) is given by:

$$X_t = C(1)\sum_{i=1}^{t} u_i + C^*(L)u_t \tag{4.1}$$

Here:

- $C(1)$ is the long-run impact matrix, which captures the cumulative effect of shocks on the system.
- $C^*(L)$ represents the transitory components, which have only a temporary impact on X_t.

The equation can be interpreted as follows: The permanent component, $W_t = C(1)\sum_{i=1}^{t} u_i$, represents the cumulative effect of past shocks on the system, which causes permanent (non-reverting) changes in the price series X_t. This component corresponds to the long-term equilibrium that the system will converge to, and it is equivalent to the efficient price or common factor in Hasbrouck's framework. On the other hand, the transitory component, $G_t = C^*(L)u_t$, represents the short-term deviations from the equilibrium that dissipate over time. These effects are temporary and do not contribute to the long-term level of the price series.

Now, Johansen [**?**] proves that

$$C(1) = \beta_\perp \left(\alpha'_\perp \left(I - \sum_{i=1}^{k} \Gamma_i \right) \beta_\perp \right)^{-1} \alpha'_\perp. \tag{4.2}$$

So equation 2.1 can be rewritten as

$$X_t = \beta_\perp C \left(\sum_{i=1}^{t} u_i \right) + C^*(L)u_t, \tag{4.3}$$

with

$$\psi = \left(\alpha'_\perp \left(I - \sum_{i=1}^{k} \Gamma_i \right) \beta_\perp \right)^{-1} \alpha'_\perp. \tag{4.4}$$

The final step in calculating the information share (IS) involves removing the contemporaneous correlation in u_t. This is achieved by generating a new set of errors:

$$u_t = Fe_t, \tag{4.5}$$

where Ω is the covariance matrix of the innovations e_t and F is the lower triangular matrix such that $\Omega = FF'$. The proportion of the innovation variance attributed

to e_j is calculated as:

$$IS_j = \frac{([\psi F]_j)^2}{\psi \Omega \psi'}, \tag{4.6}$$

where $[\psi F]_j$ represents the jth element of the row matrix ψF. Denote

$$F = \begin{pmatrix} f_{11} & 0 \\ f_{12} & f_{22} \end{pmatrix} = \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sigma_2(1-\rho^2)^{1/2} \end{pmatrix}.$$

Simplifying, we obtain:

$$IS_1 = \frac{(\gamma_1 f_{11} + \gamma_2 f_{12})^2}{(\gamma_1 f_{11} + \gamma_2 f_{12})^2 + (\gamma_2 f_{22})^2},$$

$$IS_2 = \frac{(\gamma_2 f_{22})^2}{(\gamma_1 f_{11} + \gamma_2 f_{12})^2 + (\gamma_2 f_{22})^2}.$$

where γ_1 and γ_2 are components of $\alpha_\perp$, and f_{ij} are elements of the matrix F.

Intuitively, the information share measures the extent to which innovations in a particular market contribute to the variance of the efficient price. A higher information share indicates that the market plays a more significant role in incorporating new information into the security price.

As we can see, the information share for each market can then be computed using the elements of Ω and α. In the original paper (1995) [4], Hasbrouck does not estimate the metric with the VECM model, but with its corresponding Vector Moving Average (VMA) representation. As Baillie et al. (2002) [5] point out, it is easier to estimate the metric with the VECM representation, since in the end we only need the vector $\alpha_\perp$ and the matrix of error correlations.

A significant advantage of Hasbrouck's approach is that it does not require the direct estimation of the VMA, which can sometimes be challenging. All necessary calculations can be performed by estimating the VECM. With the vector $\alpha_\perp$ and the matrix of correlation of errors, we are well-equipped to calculate the information share.

In summary, Hasbrouck's information share provides a metric for understanding the contributions of different markets to price discovery. By quantifying the proportion of the efficient price variance attributable to each market, this measure tries to identify where new information is most rapidly and accurately reflected in security prices.

4.6 Hasbrouck's canonical example with simulated data

In this canonical example to test the Information Share (IS) of Hasbrouck, we generate the time series as follows. We start with two initial time series P_1 and P_2, both set to an initial value of 30, for example. The time series are defined by the following equations:

$$P_{1,t} = P_{1,t-1} + w_t$$

$$P_{2,t} = P_{1,t-2} + \epsilon_t$$

where w_t and ϵ_t are random noises that follow a normal distribution with a mean of zero and a standard deviation of one.

The series P_1 is generated such that each value is the previous value plus a random noise term w_t. The series P_2 is generated such that each value depends on the value of P_1 from two periods prior, plus a random noise term ϵ_t.

Both series are cointegrated, with a cointegration vector of (1, -1). This can be shown by considering the difference between the two series:

$$P_{1,t} - P_{2,t} = (P_{1,t-1} + w_t) - (P_{1,t-2} + \epsilon_t)$$

Simplifying the expression, we have:

$$P_{1,t} - P_{2,t} = w_{t-1} + w_t - \epsilon_t$$

Since w_t and ϵ_t are both stationary (i.e., random noise terms), the difference $P_{1,t} - P_{2,t}$ is also stationary. This proves that the cointegration relationship between P_1 and P_2 is indeed given by the vector (1, -1).

The results of the Johansen Cointegration Test with a 10% significance level are presented in the table below. The Johansen Cointegration Test is used to determine whether a group of non-stationary time series are cointegrated, meaning they share a long-term equilibrium relationship despite being individually non-stationary. This test identifies the number of cointegrating vectors among the series. It is performed sequentially to determine the number of cointegration vectors. The test starts with the null hypothesis of zero cointegration vectors. If rejected, the test proceeds to the next rank, continuing until the null hypothesis is no longer rejected, indicating the maximum number of cointegration vectors.

The trace statistic tests the null hypothesis that the number of cointegration vectors is less than or equal to a certain rank r against the alternative hypothesis that

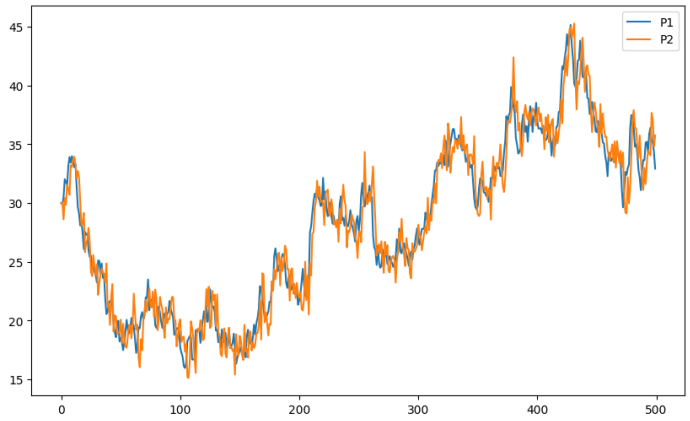


Figure 4.1: Synthetically generated cointegrated time series P1 and P2

the number is greater than r. It is calculated as:

$$\text{Trace Statistic} = -T \sum_{i=r+1}^{n} \ln(1 - \lambda_i)$$

where T is the number of observations and λ_i are the eigenvalues.

	Trace Statistic	Critical Values (5% level)
Rank 0	493.665	15.494
Rank 1	2.159	3.842

Table 4.1: Johansen Cointegration Test Results

The trace statistic for rank 0 (493.66465317) is significantly greater than its critical value (15.4943), indicating that we can reject the null hypothesis of no cointegration. For rank 1, the trace statistic (2.15881617) is less than its critical value (3.8415), indicating that we cannot reject the null hypothesis of having at most one cointegrating relationship.

Therefore, the test suggests there is exactly one cointegrating relationship between the time series P_1 and P_2, meaning the series are cointegrated with a rank of 1. This implies that there is a long-run equilibrium relationship between the two time series. We get that there is cointegration, like we expect for such an example.

The information share results from Hasbrouck's method are presented below. We conduct the analysis and calculate the metric twice, changing the order of the variables each time, since the metric is known to be asymmetric.

The results clearly indicate that market 1 is the one that incorporates the information, while market 2 reacts to it, as expected.

Table 4.2: Hasbrouck's Information Share

Vector 1	Vector 2
0.999	0.006
0.0007	0.993

4.7 5 Year German bond vs Euribor futures example

Next, we compute the metric for a real world case. we will apply the methodologies to compare the futures on German Bonds with a maturity of 5 years (commonly referred to as OE) and the futures on Euribor (ER). The German 5-Year Bond is the most secure debt in Europe and serves as a key indicator of the state of the economy, reflecting investor sentiment and economic expectations. In contrast, Euribor futures represent the expected interest rates in the Eurozone. Unlike the previous case, we lack a strong prior intuition regarding the relationship between these two contracts. This scenario is particularly valuable as it allows us to observe the metrics' performance in a less predictable context. The selection of the dataset in this study is driven by the need to thoroughly validate the methodologies employed and to explore their applicability across a range of financial contexts.

We have tick-by-tick data for the 5-Year German Bond futures, provided by Refinitiv. Its exchange ticker is OE. We also analyze Euribor futures, with ticker ER. We are choosing the first expiration for the OE futures, which is very liquid. For Euribor futures, we conducted a preliminary analysis to identify the most liquid contracts. It became clear that the contracts in the middle of the curve are the most liquid, so we decided to use the December 2025 (Z25) and March 2026 (H26) contracts. The dataset pertains to March 12, 2024, covering a 15-second interval from 12:30:00 to 12:30:15 UTC. We chose to analyze the sample period just following the US Consumer Price Index (CPI) release, where relevant information gets incorporated into the assets. While it might seem more intuitive to analyze the Eurozone CPI, in practice, the market tends to react more strongly to the US CPI. The slightly larger time window was chosen because these securities are generally less liquid than SOFR futures.

We first plot the bid prices of the event to confirm that there is sufficient liquidity to perform the analysis.

Hasbrouck's metric provides clear results in this case. The output of the information shares, obtained by changing the order of the time series, is as follows:

This indicates that the futures on the German 5-Year Bond lead the Euribor futures, because the information shares of the bond is very close to 1 in both cases.

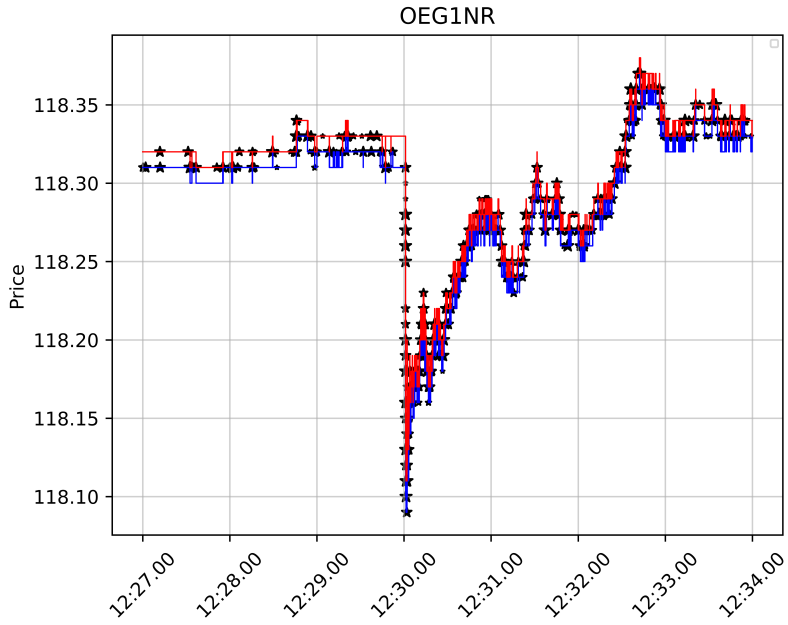


Figure 4.2: Bid and ask prices of the German 5-Year Bond futures before and during the event. Stars indicate that a trade has occurred.

Table 4.3: Hasbrouck's Information Share

Market		
Euribor	2.62×10^{-5}	1.69×10^{-5}
German Bond	0.999	0.999

The analysis was conducted by calculating the metric twice, changing the order of the variables each time.

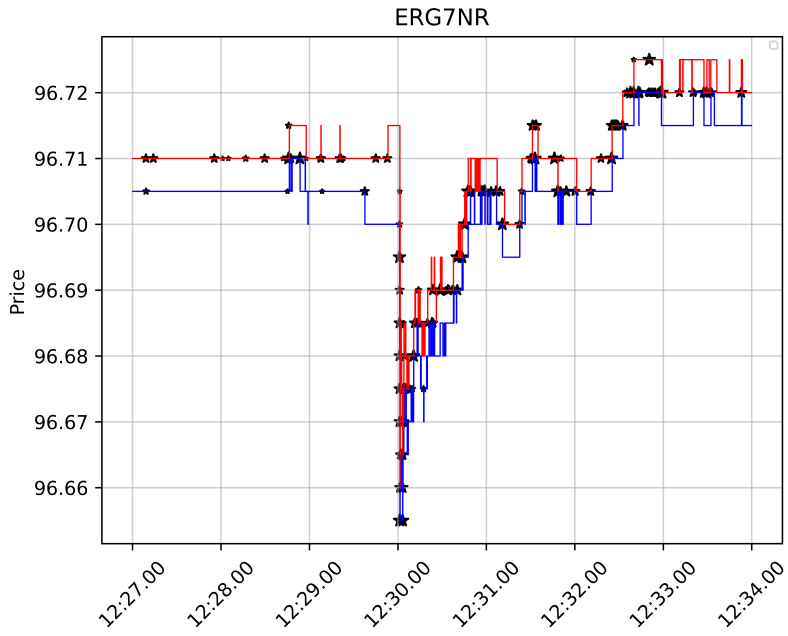


Figure 4.3: Bid and ask prices of Euribor futures before and during the event. Stars indicate that a trade has occurred.

Bibliography

[1] John Y. Campbell, Andrew W. Lo, and A. Craig MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, 1997.

[2] Takahiro Hayashi and Nakahiro Yoshida. *On covariance estimation of non-synchronously observed diffusion processes*. Bernoulli, 11(2):359–379, 2005.

[3] Nicolas Huth and Frédéric Abergel. *High Frequency Lead/Lag Relationships: Empirical Facts*.

[4] Joel Hasbrouck. *One Security, Many Markets: Determining the Contributions to Price Discovery*. The Journal of Finance, 50(4):1175–1199, 1995.

[5] Richard Baillie, Geoffrey Booth, Yiuman Tse, and Tatyana Zabotina. *Price discovery and common factor models*. Journal of Financial Markets, 5(3):309–321, 2002.

[6] Andrei Kirilenko, Albert S. Kyle, Mehrdad Samadi, and Tugkan Tuzun. *The Flash Crash: The Impact of High-Frequency Trading on an Electronic Market*. MIT Sloan School of Management Working Paper, May 5, 2014. Original version: October 1, 2010.

[7] Clive W. J. Granger. *Investigating Causal Relations by Econometric Models and Cross-Spectral Methods*. Econometrica, 37(3):424–438, 1969.

[8] Søren Johansen. *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models*. author=Søren Johansen, Oxford University Press, 1996.

Chapter 5

Annex

5.1 Complementary code

5.1.1 Lead-Lag Analysis

The `lead_lag/lead_lag` directory is focused on exploring lead-lag relationships in financial data.

- `lead_lag_impl.py`: Core implementation for lead-lag analysis.
- `conditioned_lead_lag.ipynb`: Jupyter notebook for lead/lag response functions conditioned on a movement threshold.
- **Data Files**: Includes various CSV files containing trade and spread data relevant to the analysis, such as `FFV24.csv`, `SFRU24.csv`, and `spread_data.csv`.

5.1.2 Hasbrouck Measure

- `hasbrouck.py`: Implements Hasbrouck's measure to analyze market price discovery.

5.1.3 Metrics Implementation

These scripts help evaluate different metrics, implementing them based on the analyzed data.

- `opportunity.py`: Framework for trading metrics. This is the parent class that all metrics chiild classes should inherit from.

- `leadlag_opportunity.py`: Specialized script for metrics within lead-lag contexts.
- `metric.py`: Class to store different microstructure metrics.

5.1.4 Latency Computation

These scripts help evaluate different market latencies and implement them based on the analyzed data. The `analysis_metrics/base` directory contains all the implementations.

- `server_exch.ipynb`: The time that an order takes to go from our hypothetical trading machine to the exchange.
- `hedge_latency.ipynb`: The time taken by our trading machine to send a hedge order after receiving a fill on the active leg of an autospreader.

5.1.5 Utilities

- `util_funcs.py`: A collection of utility functions used throughout the project.
- `HY_leadlag_example.py`: An example script showcasing lead-lag analysis.
- `Hasbrouck_example.ipynb`: An example script showcasing the Hasbrouck metric.

5.1.6 Documentation and Dependencies

- `README.md`: This documentation file.
- `requirements.txt`: A list of Python dependencies needed to run the scripts in this repository.