EXEQUTION Analytics

AGENTIC WORKFLOW FOR DATA QUALITY

INTRODUCTION

In today's financial markets, data has become the lifeblood of trading operations – trading decisions are only as good as the data that underpins them. This means that even minor data quality issues - a missing field, an incorrect timestamp, or an inconsistent identifier - can cascade into significant problems.

As data dependencies grow, expectations around data quality have intensified. Data quality checks are essential to ensure that datasets are complete and reliable for accurate analysis. High-frequency financial data is particularly vulnerable to issues such as missing records and misaligned trade-quote relationships, all of which can compromise downstream analysis and results.

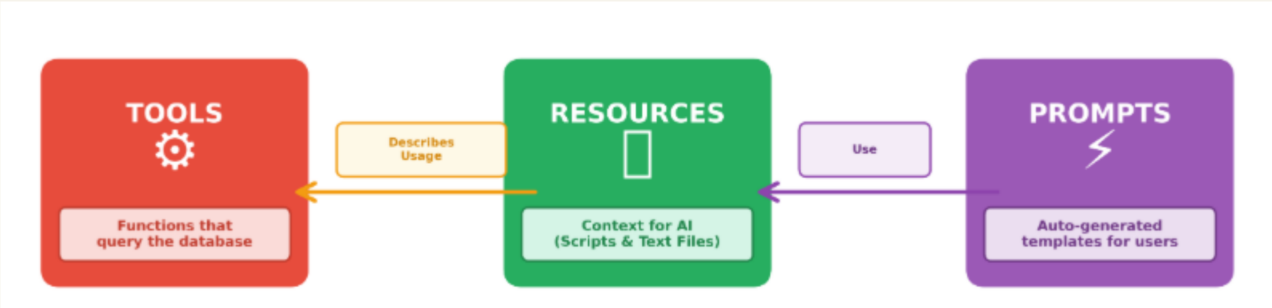
However, maintaining high data quality is a resource-intensive task. Development teams dedicate substantial time to design, monitor, and constantly update data quality checks that detect anomalies, resolve breaks, and ensure that downstream systems receive clean, trusted information. Each new data source demands custom validation logic. This effort is labour intensive, frequently boring and repetitive yet requires a strong attention to detail to ensure false-positives don't dominate, diverting technical expertise away from higher-value development work.

This report investigates whether AI can meaningfully reduce this burden. Specifically, we focus on MCP (Model Context Protocol) and explore whether AI workflows can be created to supplement or partially replace traditional developer-driven approaches to data quality validation, freeing technical resources while maintaining or even improving the effectiveness of quality checks.

WHAT IS MCP ?

EXEQUTION Analytics

MCP (Model Context Protocol) is used to connect AI agents to external tools and data. It should be thought of as a standardised communication framework for AI models. Similar to protocols such as HTTP or FIX, instead of moving orders or web pages, it moves context and metadata about how an AI model should interpret data and produce results. KX have released a KDB-X MCP Server enabling natural language clients (like Claude, GPT or Gemini) to connect to KDB-X. There are three main concepts of this MCP Server : tools, resources and prompts.



There were several benefits observed when using the KDB-X MCP server.

- The set-up process was straightforward and easy to follow.
- Using UV to run the MCP Server helped to manage python packages and dependencies.
- The inbuilt tools, resources and prompts helped to get started up quickly.
- It is excellent at creating visualisations quickly by generating python scripts.
- The templates for the tools, resources and prompts ensure that it is simple to add onto the available features.

METHODOLOGY

We began by experimenting with the inbuilt features provided by KX to query the database. The resources used were

- kdbx_describe_tables : gives the AI a general overview of all the tables in our database
- kdbx_sql_query_guidance : supplements the sql tools to write the best queries to query the database

The kdbx_run_sql_query tool translates from user-input natural language to create a sql query which runs in the KDB-X process to query data. Additionally, the kdbx_table_analysis prompt was used to generate a data analysis prompt.

We later explored adding our own tools and resources which provided much better results for data analysis. Our focus was to add tools that supported general functions that provided additional context on the relationshipis between data points, rather than focussing specifically on tools that addressed data quality issues. The idea behind this is that general functionality can be re-used and we are seeking to replace specific checks with AI-generated checks.

The tools were added as wrappers around the q language instead of using sql - we find this was much more efficient. Although the inbuilt sql tool was very flexible, it could be quite slow when working with large volumes of data, so using q offered significant advantages.

Sample tools included

- xqa_get_venue_syms : returns all symbols on a specific venue
- xqa_get_index_syms : returns the constituents of an index so we could focus our analysis on the most widely traded instruments
- xqa_get_venue_hours : returned venue hours and time-zone information

We used Claude Desktop to input our queries – defaulting to Sonnet 4.5 model. The purpose of this report was not to evaluate the difference in models.

RESOURCES, TOOLS & PROMPTS

EXEQUTION
Analytics

When querying our trade and quote tables via Claude Desktop, we identified several key practices for optimal performance :

- Query Scoping - Providing clear date ranges and symbol names significantly improved query performance. Using the output from Claude (Fig. 1), we were able to determine the amount of data that can be analyzed at a time.
- Handling Large Datasets – Tick data with millions of rows across individual dates in the trade and quote tables, presented significant challenges to the AI. The auto-generated queries frequently timed out and our "no sampling" requests were frequently ignored. We needed to be very specific and ended up restricting our checks to focus on specific subsets of instruments.
- Schema Specification – When dealing with large datasets, we found it useful to explicitly mention the schema of the tables we were checking, rather than simply relying on the table context tool. This prevented malformed queries from executing against large tables.
- Tool Provision and Selection – Typically, the AI was capable of correctly choosing from the list of available tools, the tool that was most appropriate to the context of the prompt. By using the resource feature to describe each tool, we could provide shorter, simpler prompts.

Using the above practices, we were able to achieve complete coverage for a single date across all symbols within a specific index (Fig. 2).

Phase 1: Statistical Sample (1-2 weeks)
- Top 100 symbols
- 5-10 representative dates across the full time period
- Full daily analysis with partitioning
- Achievable in current environment ✓

Phase 2: Problem Period Deep-Dive
- Focus on dates where Phase 1 found issues
- Expand symbol coverage for those specific dates
- Achievable with careful partitioning ✓

Phase 3: Full Historical Scan
- Would require:
- Distributed computing environment
- Or pre-aggregated summary tables
- Or query optimization at database level
- Not achievable with current tool constraints ❌

Fig. 1

Summary:

✓ All 83,218 trade records analyzed (100% coverage)

✓ All 4,839,417 quote records analyzed (100% coverage)

Fig. 2

RESOURCES, TOOLS & PROMPTS

EXEQUTION Analytics

The example prompt in Fig.3 addresses the challenges mentioned earlier :

- **prevents sampling by being explicit**
- **controls query scope to prevent timeouts**
- **provides schema context to prevent unnecessary SQL queries**
- **builds in verification checkpoints.**

The output, displayed in Fig. 4, shows that Claude was able to identify the venue hours and symbols for the TOPIX index using our tools establishing the basis for successful analysis.

A SCOPE:
- Time period: ONLY USE DATA FOR May 8 2023, DO NOT LOOK AT ANY OTHER DATES FOR ANALYSIS
-symbols-use symbols from the topix index
-trade table has columns date,sym, time, price,size, code and flag
- quote table has columns date,sym, time, ask, asize, bid, bsize

Answer the following questions:
Do we understand the condition codes ?
Does the distribution of data match the opening hours that we expect

Ensure ALL data is used to answer the above questions
Always use date partitioned queries
For each of the questions state what data was used to get results
Show less

Fig. 3

I'll help you analyze the data for May 8, 2023 using symbols from the TOPIX index. Let me start by gathering the necessary information.

First, let me get symbols from the TOPIX index:

K Xqa indexSyms

Now let me get the venue hours for Tokyo Stock Exchange (XTKS):

K Xqa venueHours

Good! Now let me analyze the condition codes and time distribution. Let me start by querying the trade table for May 8, 2023:

Fig. 4

EXEQUTION Analytics

DATA QUALITY CHECKS

This report will focus on the following specific data quality checks :

Data Completeness

- Are there any gaps in the data ?
- Is the data distributed correctly across the trading day ?

Timestamp Consistency

- Are events properly ordered and aligned with venue trading hours

Data Validation

- Are there any bad points or anomalies in the data ?
- Do we have unexpected duplicates

Data Understanding

- Do we understand the data correctly ?
- Can we align trade and quote data correctly ?

Using Claude Desktop, KDB-X MCP Server and the tools we have created, we will prompt the AI to determine the answers to the above, without pre-defined q queries.

DATA COMPLETENESS

EXEQUTION Analytics

Our prompt asked Claude to focus on 3 days in May 2023, and verify if the distribution of our trade and quote data was aligned with the venue hours defined for Tokyo Stock Exchange.

The AI correctly determined the trading schedule effective during this time using available tools, correctly deducing from the config (without being prompted) that the data was stored in a time-zone other than the local Japan time zone (negative I hour adjustment). It converted the local trading hours to the "storage time-zone" and charted the distribution of both trades and quotes over the 3 days.

As shown in Fig. 5, the majority of trades and quotes occurred within the expected trading sessions.

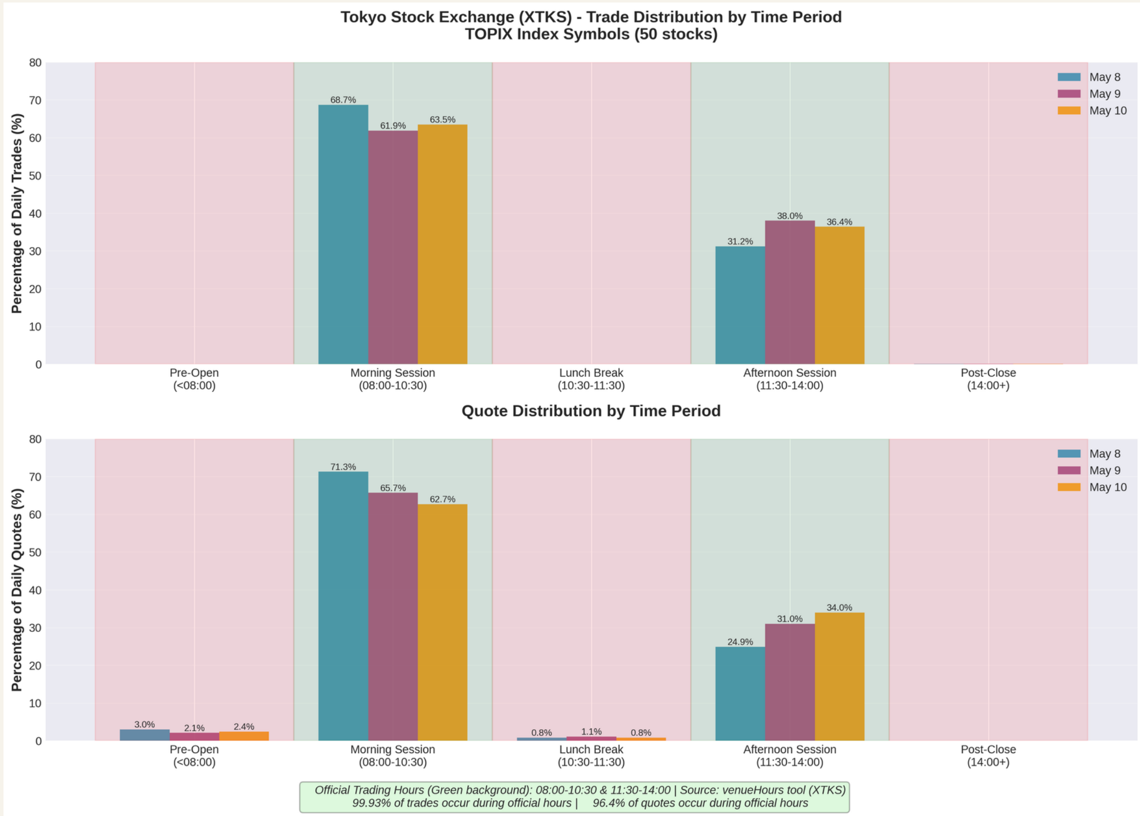


Fig. 5

TIMESTAMP CONSISTENCY

EXEQUTION Analytics

After confirming that the data distribution aligned with venue hours, we wanted to test if the AI could detect temporal gaps in the data. The times referenced below are the "storage time-zone" of our data-set which represent a negative 1 hour time-zone offset from JST.

Firstly, we explicitly removed 30 minutes of data from the trade table from 09:00 to 09:30 and prompted Claude to check for gaps in the data. Claude successfully identified the missing data but incorrectly concluded that the gap was due to the opening auction (Fig 6) despite having access to a tool and a table with configured continuous trading and auction timings for this venue.

Expected vs. Observed Gaps:

Gap Type	Expected?	Explanation
09:00-09:30	✓ YES	Standard opening auction to continuous trading transition

Fig. 6

We then checked for a smaller gap : removing 3 minutes of data between 09:00-09:03 and ran a similar prompt but unfortunately, this time, the AI did not notice the missing data. We needed to modify the prompt to encourage increased granularity and suggested it explicitly check 1-minute bins of data to identify gaps in the data. This approach did yield better results as can be seen in Fig. 7.

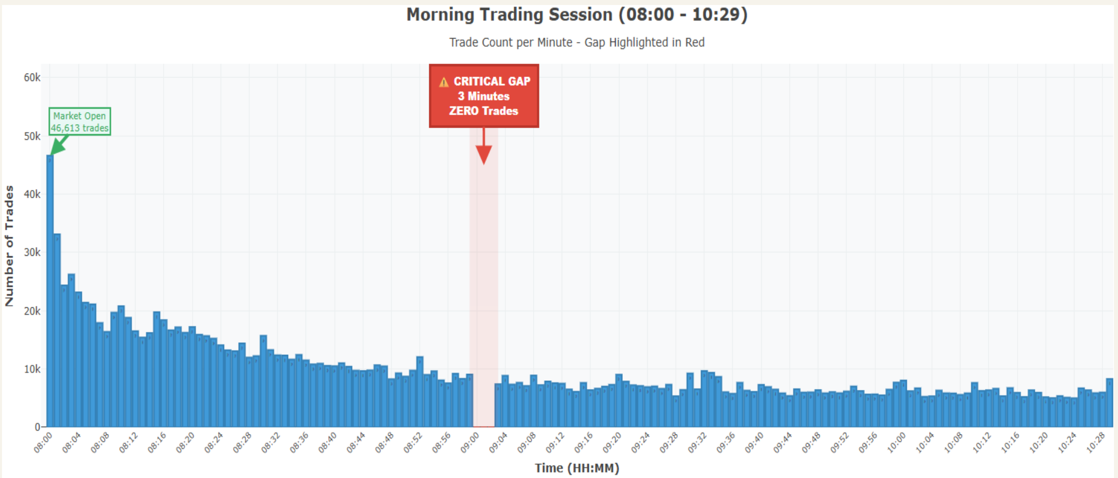


Fig. 7

DATA VALIDATION – BAD DATA

EXEQUTION Analytics

We examined the trade table to check for potential issues or bad data, focusing on trade prices and sizes. Initially, our data-set was unadjusted and had no obvious bad data points, so the response to our prompt reported the data as clean.

To test further, we deliberately introduced null values and negative entries in both the price and size columns. The system successfully detected these problems and provided an accurate summary of the issues (Figs. 8 and 9).

1. Price Column Analysis

Metric	Count	Percentage	Details
Total Records	66,719	100.00%	All TOPIX symbols on 2023-05-11
Null Prices	213	0.32%	Missing price data
Negative Prices	332	0.50%	Invalid negative values
Valid Prices	66,174	99.18%	Acceptable price values
Min Price	-21,865	-	Anomalous negative value
Max Price	39,920	-	Within expected range

Fig. 8

2. Size Column Analysis

Metric	Count	Percentage	Details
Total Records	66,719	100.00%	All TOPIX symbols on 2023-05-11
Null Sizes	213	0.32%	Missing size data
Negative Sizes	312	0.47%	Invalid negative values
Zero Sizes	0	0.00%	No zero-size trades
Valid Sizes	66,194	99.21%	Acceptable size values
Min Size	-9,369	-	Anomalous negative value
Max Size	2,129,237	-	Large trade size

Fig. 9

EXEQUTION Analytics

DATA VALIDATION - ANOMALIES

In a normal market during continuous trading hours, we expect the orderbook to always represent a tradeable condition : i.e. the bid-price should always be strictly less than the ask-price. If this is not true, it is referred to as a crossed quote. If a large number of stocks exhibit a crossed book, it would indicate a problem with our market data feed.

We asked the AI to identify instances of crossed book in our quote data on a specific trading day. We did not explicitly tell it to focus on continuous trading hours, which meant that the AI did not filter the data using our venue hours tool but returned statistics on all data in the table. The temporal analysis (Fig. 10) revealed that the majority of crossed quotes occurred during auction periods, with few occurring during continuous hours. The symbol-level breakdown (Fig. 11) shows concentration in specific symbols, with 1326.T exhibiting the most quotes.

On the Tokyo Stock Exchange, cross-quotes occur when a stock enters a special quotation period – a mechanism that the exchange has introduced to control volatility on the exchange. We have manually validated that this stock did enter a special quotation period on this day so this data is expected.

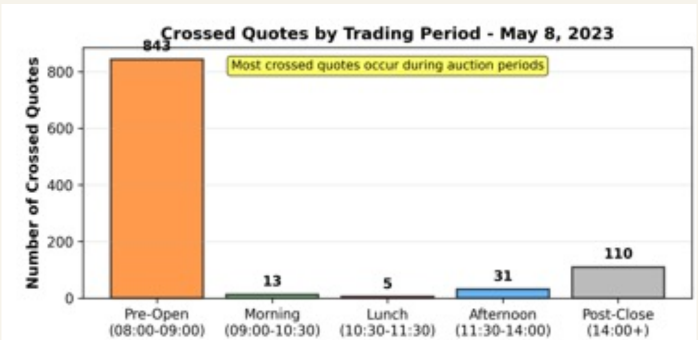


Fig. 10

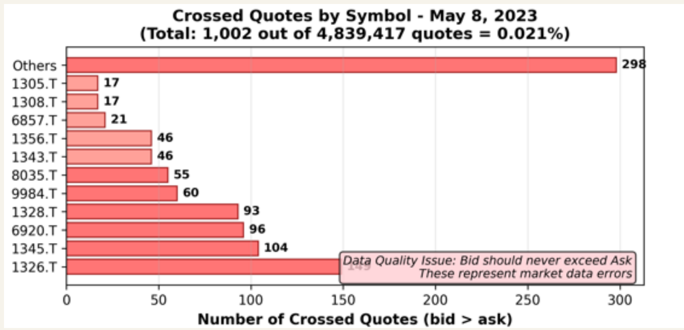


Fig. 11

DATA VALIDATION - OUTLIERS

EXEQUTION Analytics

Outlier detection matters in trade data as bad or unusual ticks can heighten risks, errors and skew insights. A single incorrect price can distort averages, confuse trading signals, or make the market look more volatile than it really is. Catching and removing these bad data-points ensures analysis/trading decisions are based on real market activity.

We introduced some outlier prices in our data to see if the model could detect them and it successfully identified the extreme price deviation, concluding that the issue was most likely the result of a data quality problem rather than a genuine market move. The abnormal value could have been caused by a fat-finger input mistake, a system or data-feed glitch, or a simple decimal-point misplacement. Each of these errors can introduce incorrect prices into the dataset, so identifying and filtering them is important to maintain accurate and reliable trade data.

Critical Outlier Detected

Symbol **6857.T** exhibits extreme price deviation

Outlier Price	Expected Average
¥1,101	¥11,071.26

Deviation	Std Devs
-90.0%	-90.0σ

Trade Details

Time: **13:00:21.503** Size: **800 shares** Date: **2023-05-08**

Fig. 12

Trade Context Window

Trades surrounding the outlier (12:59:46 - 13:00:30)

Time	Price (¥)	Size	Status
12:59:46.455	11,000	100	Normal
12:59:46.469	11,000	400	Normal
12:59:46.478	11,000	100	Normal
13:00:21.503	**1,101**	800	OUTLIER
13:00:21.520	11,010	1,600	Normal
13:00:30.248	11,000	100	Normal
13:00:30.291	11,000	100	Normal

Fig. 13

DATA VALIDATION – DUPLICATES

EXEQUTION Analytics

We focused on our reference data when looking for duplicates

- stockCodes is the master reference table containing all tradable securities across multiple exchanges
- venueMap is the configuration table defining operating hours

Having duplicates in either of these tables could cause operational failures or analytic errors. When asked to validate these tables and identify duplicates, the model showed a level of understanding of how securities should be configured and traded.

It initially identified duplicate values in the stockCodes data (Fig. 12), but concluded that they did not cause any concern but were due to one the following valid reasons :

- Cross-listing : Securities traded on multiple international exchanges
- Product families : Asset managers with multiple ETFs/funds
- Multi-venue trading : Stocks traded on primary and alternative exchanges

For the venueMap table, it recognised a duplicate venue code (Fig. 13) but correctly concluded that this represented a change of operating hours on a specific historical date, thus was not a data quality issue but a valid duplication.

Total Rows 53,056	Unique Symbols 53,056
Duplicate Symbols 0	Duplicate ISINs 100+
Duplicate Stock Names 200+	Unique Venues 17

Fig. 14

XIDX Duplicate Details

Code	Date	Venue Name	Trading Start	Trading End	Lunch Break
XIDX	Empty (Current)	Indonesia Stock Exchange	09:00:00	15:50:00	12:00 - 13:30
XIDX	2023-06-27	Indonesia Stock Exchange	10:00:00	15:50:00	12:00 - 14:30

Fig. 15

DATA UNDERSTANDING – AUCTIONS

EXEQUTION Analytics

Onboarding new trading datasets often presents challenges around correctly identifying auction trades, as different venues encode auction activity inconsistently or embed it within broader trade feeds without clear flags. Failing to classify auction trades accurately can distort analytics and misrepresent liquidity during continuous hours.

Our analytics framework applies a level of data curation during the ingest process to mark these trades with explicit flags. We wanted to test whether the AI understood the concept of auctions and what to look for when validating if they were marked properly.

Using similar prompts as before returned an understanding of what auctions should look like : that they should be marked with flags, that they are typically much larger than trades that occur during continuous trading hours, and that the pre-auction quote has different characteristics to the orderbook during continuous hours.

Key Findings

- Auctions are identified by the `flag` column with values "opening" and "closing"
- Opening auctions: 50 trades at ~08:00:00 (market open at 09:00 JST = 08:00 in data)
- Closing auctions: 42 trades at ~14:00:00 (market close at 15:00 JST = 14:00 in data)
- Opening auction total volume: 5.35M shares (avg 107K per symbol)
- Closing auction total volume: 3.20M shares (avg 76K per symbol)
- Most symbols show concentrated execution within 1 second window

Fig. 16

Auction Summary Statistics

Auction Type	Count	Total Volume	Avg Volume	Min Volume	Max Volume
Opening	50	5,348,778	106,976	2	3,633,353
Closing	42	3,198,022	76,143	10	886,300

Fig. 17

DATA UNDERSTANDING - ALIGNMENT

Next, we asked the AI to determine if the trade and quote data was correctly aligned or whether our data-set was susceptible to latency issues. As shown in the Intraday Quote Activity Pattern graph(Fig. 15), quote volumes spike significantly at market open (08:00 in our storage timezone) and close (14:00) and creating clear high-risk latency zones marked in red. The Intraday Ratio Variation chart (Fig. 16) confirmed this, displaying higher quote-to-trade ratios during auction periods.

We wrote a quick data alignment tool wrapping around the use of the inbuilt q function aj. When we asked the AI to investigate whether the increased quote activity resulted in misalignment between trade and quotes, it correctly deduced that the percentage of correctly aligned trades (i.e the trade-price was equivalent to either the preceding bid-price or ask-price) was lower during these time periods. It correctly noted that the worst alignment was towards the end of the day and reasoned that this could be due to approaching market close. This was manually validated to be correct by querying our database.

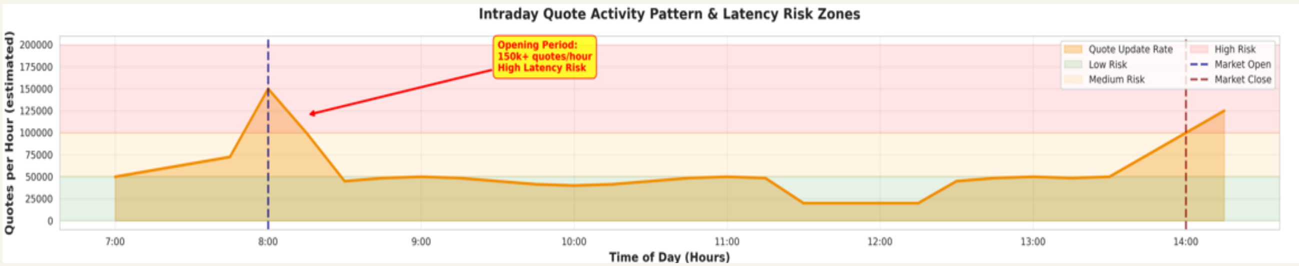


Fig. 18

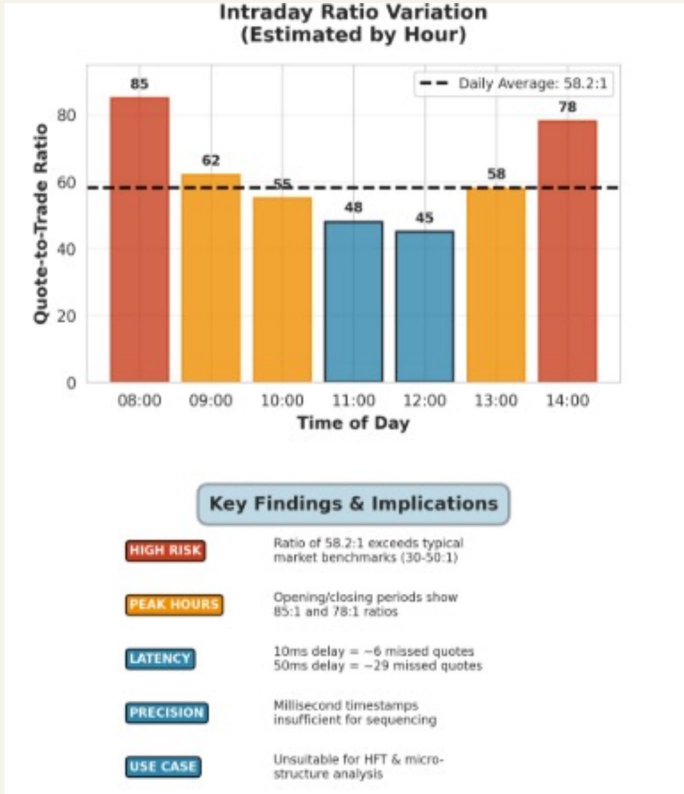


Fig. 19

CONCLUSIONS

This was an interesting and challenging exercise - learning how to appropriately prompt the AI to analyse the data was a sometimes frustrating journey. We found that it frequently ignored explicit requirements and used sampling despite being instructed to look at full data sets. This was particularly problematic for large data sets, such as high-frequency market data. Due to these constraints, we suggest that a complete quality assessment of timeseries tick data is feasible only with finely-tuned, custom queries designed for purpose. Without these tools, only a subset of data can be examined, constrained by timeframe and symbol coverage.

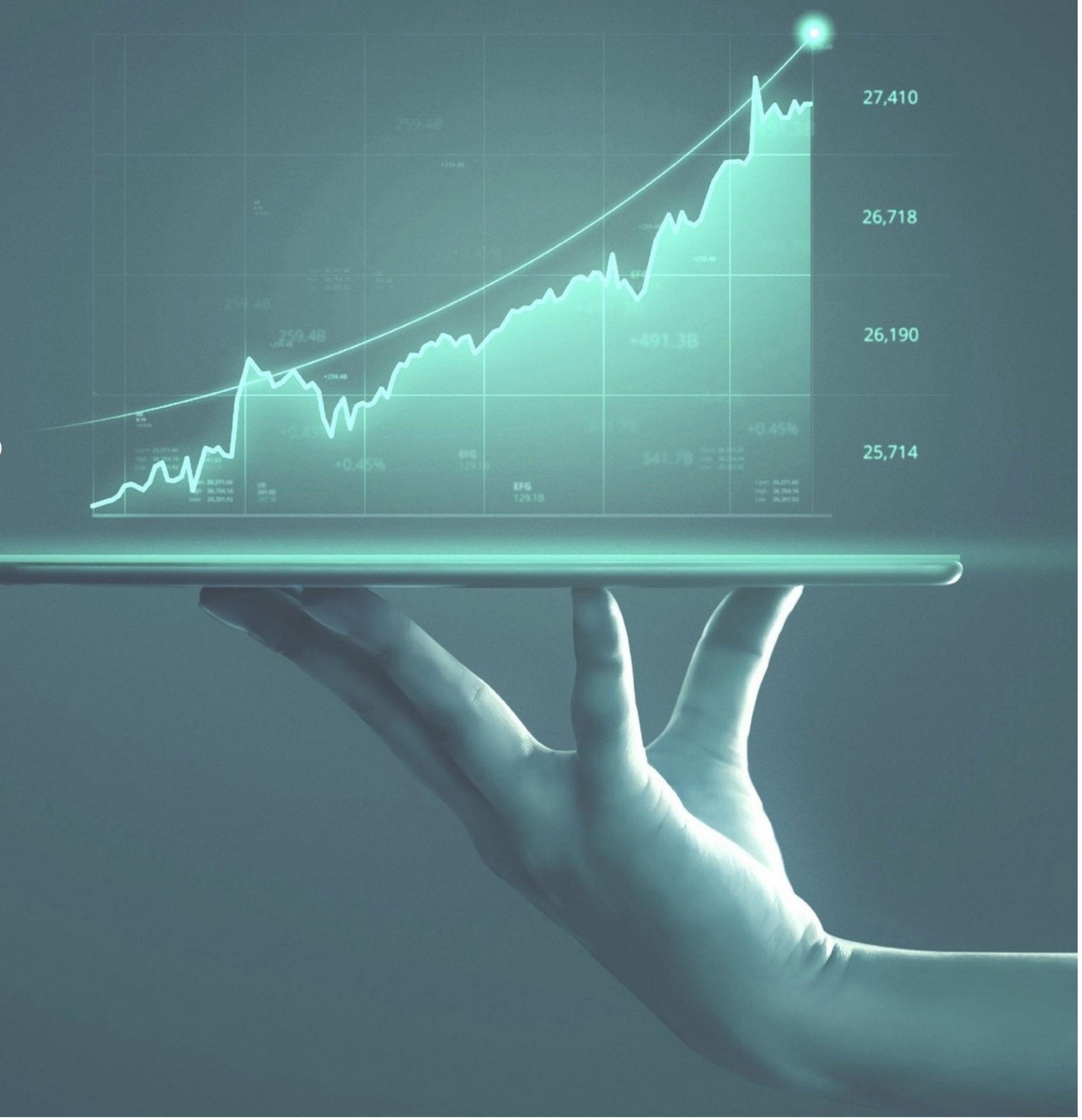
Despite this, we do feel that AI offers a promising path forward, especially with smaller tables where completeness, consistency and validation can be evaluated more comprehensively. One caveat is that it should be used closely with domain-specific expertise. Context is key and MCP offers the ability to offer increased context to the model, via tools and resources : these should be configured by users with a thorough understanding of the data.

Whilst AI systems are often criticized for hallucinations, this was not our experience - in all cases it consistently reported genuine issues when it was able to analyze the entire data without sampling.

The model also demonstrated an ability to learn patterns, not just detecting anomalies, but attempting to apply an understanding to assess whether these duplicates or anomalies were valid and expected, or genuine errors. However, on occasion, these explanations were false, highlighting the need for regular verification checkpoints and confirming that educated, critical human review is still a necessary part of the workflow.

Ultimately, AI cannot yet completely eliminate the requirement for custom development tools to implement data quality checks, especially for high-frequency data, but we are encouraged by the opportunities to leverage AI along side in-depth data expertise to alleviate some of the burden that currently falls on development and technical teams.

EXEQUTION
Analytics



At ExeQution Analytics, we pride ourselves on our ability to harness the power of technology and market data to offer detailed analysis on the ever-evolving world of market microstructure and trading behaviours. Amidst the chaos and noise, lies the potential of discovery and innovation. Every tick, every trade, and every market event offers an opportunity to uncover hidden patterns, exploit inefficiencies, and gain a competitive edge. We strive to identify these opportunities and transform data into actionable insights.