Expert Talk

Optimise trading costs and comply with regulations leveraging LSEG Tick History – Query for Transaction Cost Analysis

Transaction Cost Analysis (TCA) is an important part of a toolkit for firms that need to fulfil regulatory obligations and to reduce trading costs. Creating your own TCA platform enables you to customise this analysis to your own trading performance. Historically, this has required the time-consuming necessity of maintaining your own internal tick database. Today, financial firms can instead query vast amounts of tick data stored with market depth in LSEG Tick History – Query, without having the time and costs of maintaining their own internal tick databases. Furthermore, because it is cloud-based, compute is scaled when required.

LSEG DATA & ANALYTICS

Use TCA to reduce trading costs and improve returns

Traders seek to maximise some sort of return metric. This metric can, for example, be annualised or risk-adjusted returns. If our trading signal is more accurate, it should help to improve our returns. However, perhaps a more overlooked component of returns is the cost of trading. If trading costs are substantial, it can take a big chunk out of our returns. As we increase the frequency of our trading, these trading costs can contribute an ever-increasing drag on our performance. If we are executing very large amounts in the market, we need to note that our trading costs can be a nonlinear function of the size we are executing.

Reducing trading costs should be a key objective for any trader, because it can increase underlying trading returns. Regulators are now stipulating that showing best execution is necessary for buy-side participants. Hence, financial firms have multiple reasons for undertaking transaction cost analysis, not only to improve returns, but also to fulfil regulatory obligations.

How can we do transaction cost analysis?

There are number of different ways of doing TCA. One approach is to use a vendor, who will perform the calculations on your behalf and send you a report summarising your transaction costs. However, many market participants will want to create their own TCA platform. Such a TCA platform can be fully customised to their trading requirements. Having your own TCA solution allows traders to delve much deeper into their own trading activity to find ways of reducing costs.

Writing your own TCA library can be a complex exercise. Alternatively, to speed up the process, market participants can use existing open source TCA software, such as tcapy, and modify it to their own requirements.

Complexities of managing your own tick database for TCA

Once the software development process is complete, a TCA platform needs to be deployed to a live environment. Running a TCA platform live often results in another complexity, namely the necessity to manage vast amounts of high-frequency market data, which needs to be done on an ongoing basis.

When performing TCA, at its most basic level, you compare your own fills against a benchmark derived from high-frequency tick data. You can calculate metrics including:

– Slippage (the difference between your fill and a market benchmark such as the mid price)

– Market impact (how much the market moves after your fill, or by estimating using a model) etc.

By looking at these various metrics on an aggregated basis, you can understand which factors impact your trading costs. We can do liquidity analysis to, for example:

– Analyse how trading costs differ across liquidity providers

– Do venue analysis to see how our orders are being filled in different venues, to understand the best way to route an order

– Understand the conditions that different trading algorithms work with best

You can also flag specific trades where the trading costs were an outlier, to investigate in more detail with your trading teams and liquidity providers.

The market data required can be very granular. For more sophisticated analysis, you cannot simply rely on top of book bid/ask data. You will also need to have the volumes associated with quotes. Furthermore, you'll need to have market depth data, in other words the ability to see quotes across multiple levels of the order book. Level 2 and 3 data enable you to view quotes that are placed many levels away from the top of book. These more granular datasets would be required if you wanted to create your own proxy benchmarks for a risk transfer price, volume-weighted average price (VWAP) or other estimates to compare against your own fills.

The file sizes of such a tick dataset can be huge on disk, particularly if you are trading many instruments. If you run all this locally, you will also have to manage scaling the hardware, as the computational and storage demands increase, particularly if you are storing level 2 and 3 data. The tick database also needs to be continually kept up to date. This can be done by running processes that stream live market data to disk, or by regularly downloading market data from a provider. If any of these processes fail, you'll also need to backfill missing parts of your database.

It is also likely that this same process of managing a tick database is repeated across multiple parts of a trading firm, where different teams look after tick databases for different asset classes or use cases. Say, for example, our back office team is monitoring TCA primarily for compliance purposes, and our trading team is also doing TCA but primarily for reducing trading costs. If they use different tick databases, they can end up with different results, and hence, it can be challenging to reconcile the results between the different reports.

In short, managing the storage of data in your own tick database can end up being a hugely time consuming and very expensive proposition, especially if it is being done by multiple teams.

How LSEG Tick History – Query can make it easier to do TCA

How can we make it easier for you to run your own TCA platform without having to spend the time and money required to manage large tick databases? Can we simply use a tick database that is already maintained elsewhere? How can we avoid the replication of work, where multiple teams in different asset classes maintain their own tick database? How can we use more granular level 2 and 3 data to get a much better picture of market liquidity, but avoid the significantly increased computation and storage costs?

On Google Cloud Platform (GCP), it is possible to access a vast quantity of high-frequency market tick data on LSEG's Tick History product, via BigQuery. It covers multiple asset classes and millions of instruments, meaning it can serve as your single source of high-frequency data for your whole firm, including the trading team and compliance team. It also offers more granular level 2 and 3 datasets.

It is no longer necessary to download vast amounts of market data locally, and to manage your own tick database. Instead, market participants can use BigQuery and familiar SQL to directly query LSEG Tick History stored in GCP to perform queries do to TCA. BigQuery is also accessible via APIs, including one in Python, allowing it to integrate into your own TCA library.

In other words, rather than moving your data to where you run those computations (e.g., downloading locally to number crunch on local servers), we simply move our computation to where the data is stored on GCP. We can number crunch vast quantities of tick data very quickly with BigQuery, and simply retrieve the results. We also don't need to worry about managing and scaling hardware resources as well, given it's all in the cloud. Compute power is scaled according to user needs automatically, and you just pay for the compute that you use.

Saeed Amen

Saeed Amen is the founder of Cuemacro. Over the past 15 years, Saeed Amen has developed systematic trading strategies at major investment banks including Lehman Brothers and Nomura. He is also the author of Trading Thalesians: What the ancient world can teach us about trading today (Palgrave Macmillan) and is the co-author of The Book of Alternative Data (Wiley). Through Cuemacro, he now consults and publishes research for clients in the area of systematic trading.
He has developed many Python libraries, including finmarketpy and tcapy for transaction cost analysis. His clients have included major quant funds and data companies. He has presented his work at many conferences and institutions, which include the ECB, IMF, Bank of England and Federal Reserve Board. He is also a visiting lecturer at Queen Mary University of London and a co-founder of the Thalesians.