Analyse High Frequency Trading Data

Michael Trimmel, Marks Tretzmueller, Viktor Ludwig

January 22, 2017

Abstract

This design study is focused on the visualisation of high frequency data of limit order markets. The dataset is provided by Lobster. We discuss several visualisation strategies for order book data and discuss our design choices. Our primary goal is to facilitate the exploration of order book data.

1 Introduction

1.1 Background

Modern stock markets operate as limit order markets. Market participants can place bid -, as well as ask - orders i.e. offers to buy or sell specified amount of stock at a specified price. Orders are either executed by market participants on the other side of the market or cancelled by the submitters who placed the order.

The order of execution takes into account the price and the time of placement. Orders are executed such that the market parcipant on the other side always receives the best price at that time. If there is more than one order at the same price, the order of execution is based on priority of the older submission.

1.2 High Frequency Trading

High frequency traders buy and sell within very short periods of time, sometimes fractions of seconds. This has two consequences. First, their downward risk is limited because in short periods, the price cannot decline very much. Second, a strategy does not need to be always profitable, as long as it performs well on average. If a strategy pays off in 60% of all cases, it is already a successful strategy.

There are many different types of strategies, such as mean-reversion (betting against movements), trend-following (betting on the continuation of movements) or pair-trading (betting on coherent movement of similar stocks). All those strategies need markets with a lot of liquidity, i.e. market many transactions per day. That's why high frequency trading can only be done with stock of companies with a high market capitalization and a wide-spread ownership.

1.3 Data

Lobster is a repository of high quality high frequency data, free for academic use. The Lobster data consists of two csv files, the order book and the message book.

Every row of the order book is a snapshot of the order book at a certain time, showing the best (most competitive) 5 bid and ask positions. The message book keeps track of every change in the order book and describes the event, that led to a change. An event can either be the submission of an order, an execution or a cancellation and is characterised by the direction in *buy* or *sell*. The timestamps in the order book and the message book correspond.

2 Motivation

This design study is concerned with the visualisation of limit order book data to assist researchers in studying the structure and relationships between quantities of limit order markets. The visualizations are designed for academic use, but might also be useful for traders because they also need to come up with a hypothesis about a general relationship in the market before deploying a trading strategy. Price changes are usually measured in return, i.e. the relative change in price, because it is irrelevant at what price the trade is conducted, instead it's important how the price develops after that.

Since time is measured in milliseconds in high frequency data, one day of trading results in a very large dataset. There are multiple dimensions to be taken into account, so visualisation of high frequency data is a non-trivial task.

3 Related Work

Visualising a limit order book means visualising several quantities, such as price/return, volatility and order book entries. In the case of the Lobster datasets, there are the five most competitive orders for both the bid and ask side at each point of time.

3.1 Limit Order Book Visualisation

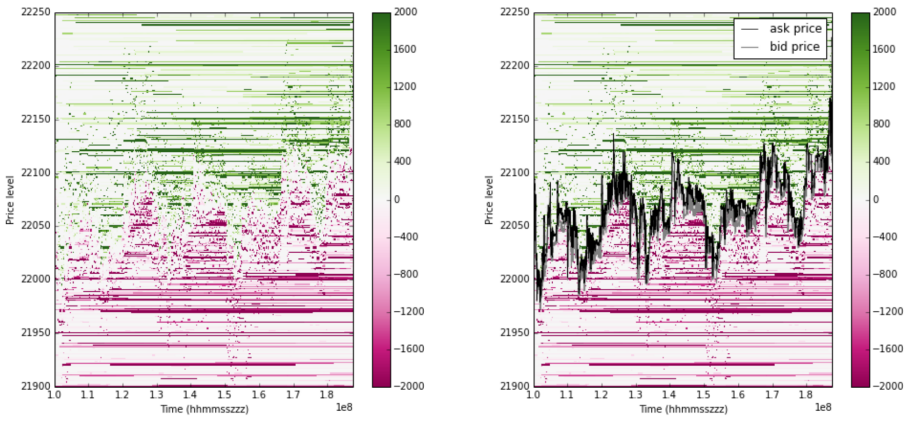

Figure 1: Schroeter 2014

This[1] is a neat and intuitive combination of the developments of the stock price and the order book. The vertical axis shows the price and horizontal axis the hours of the day. Limit orders are depicted as bars above and beneath the price. This representation is very intuitive. Whenever the price reaches a limit order, it is executed. The order size is visualised by a continuum of colors. The chart is good to get an overview of the order book developments of one day, but does not provide many details. Since the size of the orders is shown by a continuous colour scale, it is hard to guess the actual quantities. For our task, i.e. the exploration of orderflow and the development of hypothesis about the market, this visualisation is not appropriate.

The following plot shows the price of limit orders on the vertical axis and the size of the positions on the horizontal axis. This visualisation is already implemented in an R package[2].

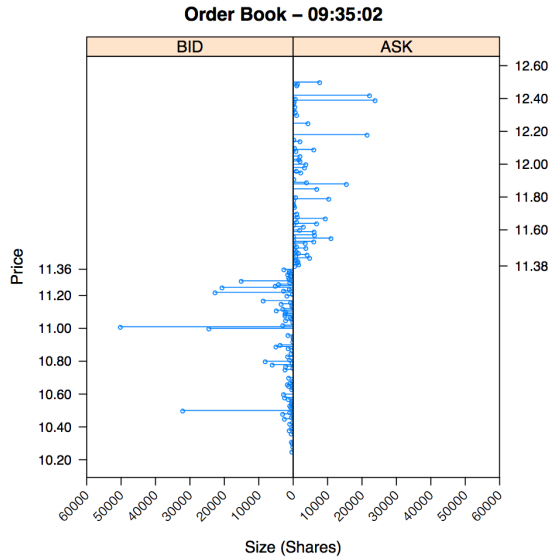

Figure 2: Bid Ask Plot

The advantage of this visualisation is that the bid and the ask side are seperated in an intuitive way, by going into opposite directions. The downside of this choice for visualisation is the missing comparability of the bars.

3.2 Volatility

Volatility is defined as the variance of returns. Variance can be visualised as bandwith around the mean.

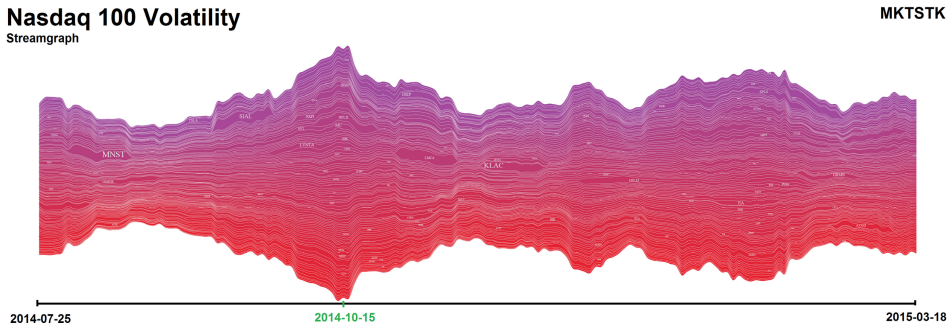

Figure 3: Volatility Streamgraph[3]

The lack of a common reference point makes it hard to compare different magnitudes of volatility.

4 Approach

While the previously discussed approaches are aesthetically appeasing, they cannot accurately communicate the details of the Lobster dataset. There is either too much information in the plot or the layout does not support the comparability of the respective quantities. This is why we have decided in favour of multiple linked views, which allow the researcher to maintain an overview of the trading data and to inspect the data in more detail at the same time, if needed.

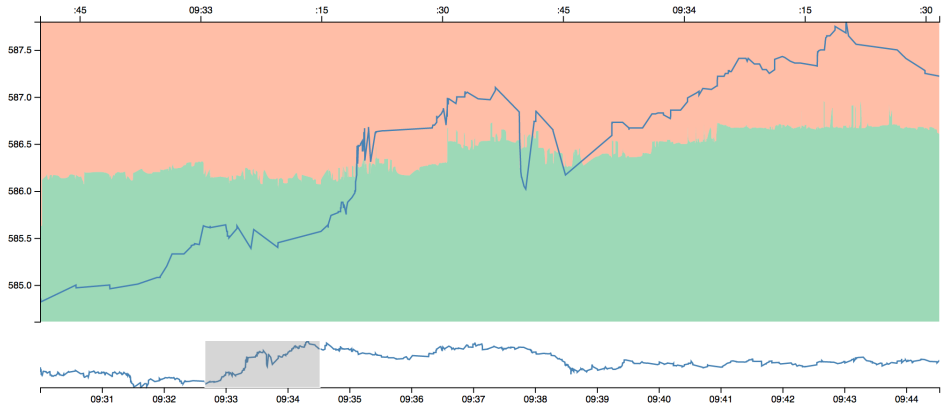

Figure 4: Price / Aggregated Order Flow

The area chart at the bottom provides an overview of price movements on the entire day. A sliding window allows the spectator to subset on a time slot and to view details on the selected time slot in the other three charts. The chart in the center of the dashboard depicts the price as a line chart and the aggregate order flow for both sides (bid and ask) as area chart in the background.

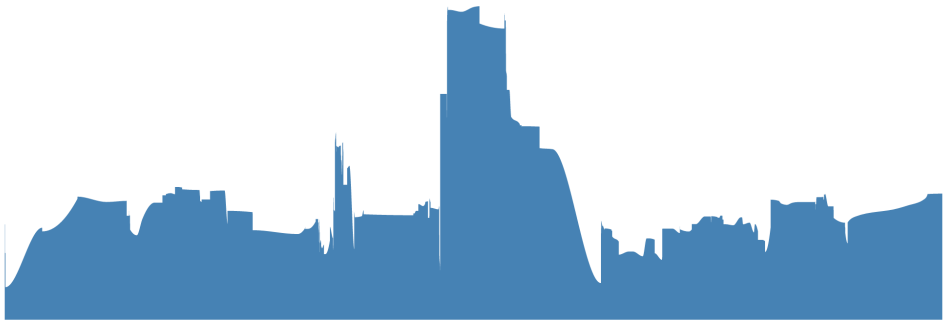

Figure 5: Volatility Plot

The chart at the top shows the volatility of the selected time slot. The volatility has been calculated based on a sliding window with 50 observations. The histogram on the right displays the volume of order flow types (submission, cancellation or execution) compared to the average volume during the day.

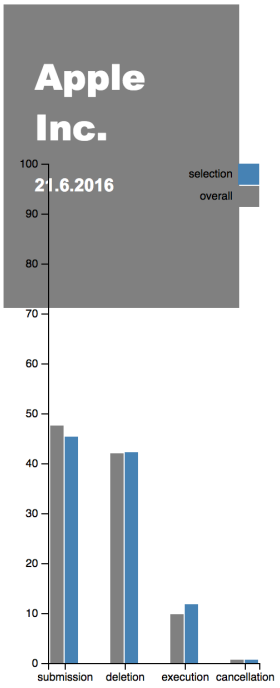

Figure 6: Cancellations, Submissions, Executions; Overall vs. Interval

5 Implementation

The first Hi-Fi prototype was based on the R library dygraphs, which is just an interface to the JavaScript charting library with the same name. While the prototype could be setup quickly, we faced several issues that made it impossible

to use dygraphs for the final visualisation. First, the R interface to dygraphs offers many templates for visualisations but does not allow finetuning in much detail. Second, the large size of the Lobster dataset made the rendering of the graphs too slow for extensive use. That's why we have decided to 'handcraft' the visualisation in the low level JavaScript charting library D3.

5.1 Issues

There were several issues that we faced, while working on the project. In the beginning, it took us quite a while to acquire the necessary background knowledge of the market mechanics and economics involved in high frequency trading. There are many sophisticated trading strategies, e.g. the submission of big amounts of asks and bids to beat down the price of stocks and for that get better position for executions. In order to design a visualisation in a meaningful way, a profound knowledge of this topic is required.

Another issue is the the size and format of the input data. Since high frequency trading is transacted in milliseconds, there is a huge amount of data for a small timespan. This data is a compound of the *messagebook* and the *orderbook*. The size of the input data is in our case 11.3Mb for the messagebook and 63.2Mb for the orderbook, which can diversify with the number of bids and asks levels requested.

Nevertheless, after a few seconds of initializing the environment, the visualization performs in view of the huge amount of data very good and the user gets fast feedback for each interaction with the tool.

6 Discussion

One of the first insights that we gain from the visualisation is that at any point of time during the trading day, only a small fraction of all orders are executed. The vast majority of orders are cancelled. This suggests that many market participants continuously shift their orders by cancelling and resubmitting their outstanding orders.

We can see that there is frequently more order flow activity on side than on the other side of the market. However, there doesn't seem to be any general pattern between imbalanced order flow and price change.

We want to thank Goegkhan Cebiroglu, Torsney-Weir Thomas, Torsten Mller and Michael Sedlmair for their advice and feedback to our implementations and ideas.

7 Who did What?

- Viktor Ludwig: Design, Report, Presentation - Markus Tretzmller: Design, Implementation D3 - Michael Trimmel: Design, Report, Presentation

References

[1] Julien Schroeter. *Limit Order Book reconstruction, visualzation and statistical analysis of the order flow*. 2014. URL: https://www1.ethz.ch/er/publications/Thesis_JulienSchroeter.pdf.

[2] *Orderbook visualization/Charting software*. https://cran.r-project.org/web/packages/orderbook/index.html. Accessed: 22-01-2017.

[3] *Visualizing Volatility with Streamgraphs*. https://mktstk.com/2015/03/18/visualizing-volatility-with-streamgraphs/. Accessed: 22-01-2017.