Interim Report

Radek Chramosil

WorldQuant University

E-mail: radek@keemail.me

This document is the Interim Report for the Capstone Project. It shows the basic structure for the database that holds data the data from crypto-currency exchange Binance and some initial data analysis.

Introduction

The structure of the database is described below. Additionally, I provide more details about the data analyses so far carried out as well as some suggestion for trading possible trading strategies. Some plots for second data set are provided. The asset I worked with is **TUSDBTC** as quoted on *Binance Exchange*. The asset is Bitcoin quoted in USD. The access to the full exchange is free and level 2 data is available for every tick. I was not sure how to set up and event that the code would run each time there is trade on the exchange.

Database Structure

I have included the below tables in this submission. The tables can be created by SQL commands in the "create_db.sql" file. The schema for the database is shows in Figure 1.

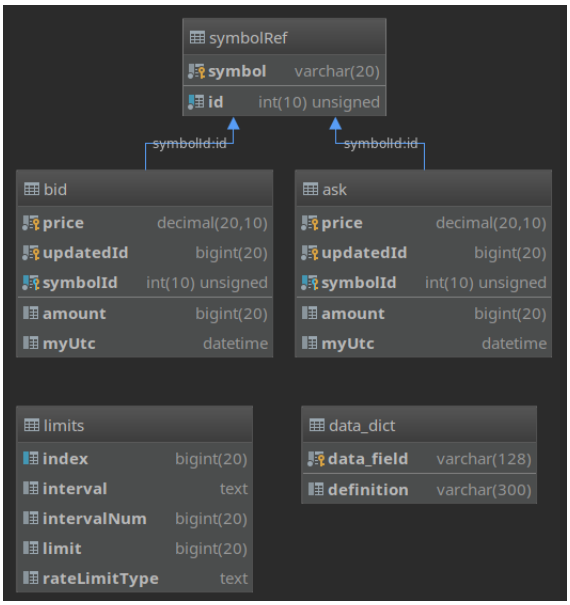


Figure 1: Schema

The tables can be dropped using the "rollback_db.sql" file. This file provides a logical inverse to the create operation. Additionally, I built functionality that can create table on fly using Pandas and SQL Alchemy however the default behaviours are not very good as it fails to create foreign keys and so on.

Tables

symbolRef. This is just a reference table for symbols. The symbols are strings, however it is more efficient to use integers for foreign keys and it is better to store integers than strings.

bid and ask. These to table are identical from data types point of view however the just store the bid and ask prices as well as my UTC time stamp and the server ‘updatedId‘ flag. This is a large integer that gets changed each time there is a trade.

limits. . Basically, the exchange give users request or order limits in time intervals. We can make 1200 data request in a minute or 10 orders in a second and 100 000 orders in a day. If we are creating a real trading application we would have to consider these closely.

data_dict. This is manually created table and holds descriptions for the data fields in the database.

Bid and Ask

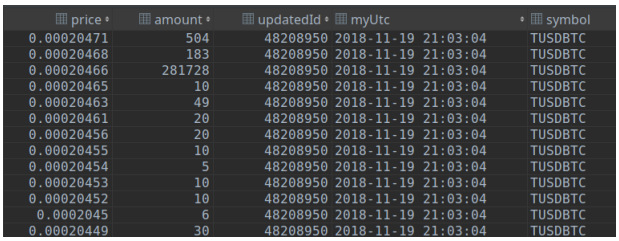
Bid and ask prices are split into two tables. The tables are identical. The tables from Figure 2 and Figure 3 carry the data for the symbol and specific Exchange ID as given by *updateId* column. My time stamps are in the *myUtc* column representing the time on my computer.

price	amount	updatedId	myUtc	symbol
0.00020497	1806	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020503	147	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020504	6	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020511	9	48208950	2018-11-19 21:03:04	TUSDBTC
0.0002052	4453	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020538	4453	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020546	18528	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020556	4453	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020558	1194	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020576	34	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020577	15	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020586	5	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020587	5	48208950	2018-11-19 21:03:04	TUSDBTC

Figure 2: Ask Price

Exchange API

It is important to handle the communication to the Exchange well. The class *DataEndPoint* hold all methods that can collect the data from the exchange as opposed to submit trades. I leave it



price	amount	updatedId	myUtc	symbol
0.00020471	504	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020468	183	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020466	281728	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020465	10	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020463	49	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020461	20	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020456	20	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020455	10	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020454	5	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020453	10	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020452	10	48208950	2018-11-19 21:03:04	TUSDBTC
0.0002045	6	48208950	2018-11-19 21:03:04	TUSDBTC
0.00020449	30	48208950	2018-11-19 21:03:04	TUSDBTC

Figure 3: Bid Price

to the reader to read over the comments for these methods to see what can be easily pulled out from the exchange.

Data Analysis - First Look

I collected 1 hour of trading data on per second basis. Figure 4 shows cumulative bid and ask prices with quantity. The way how I think about this is it is supply and demand curves. We see that there is a lot of volatility over 1 minute interval. This is good as it means that we will be able to create strategies.

The problem in the Figure 4 is that we do not see much details. So, I plotted the same time interval but with close look at what was happing around the spread in Figure 5. Again we see the volatility in the price within one minute.

We also have have look the maximum bid and minimum ask prices and plot them over the 1 hour interval. This is plotted in Figure 6. This shows that we do have trends and turning points happing over time again. We should use the trading strategies that we learning for this and we can starting simulating the trade at bid and ask prices.

Finally, we check the spread for the two currencies as in Figure 7. We immediately see that that we could create mean reversion strategy for the spread. It may not be viable on profit versus fee basis however we could tests this.

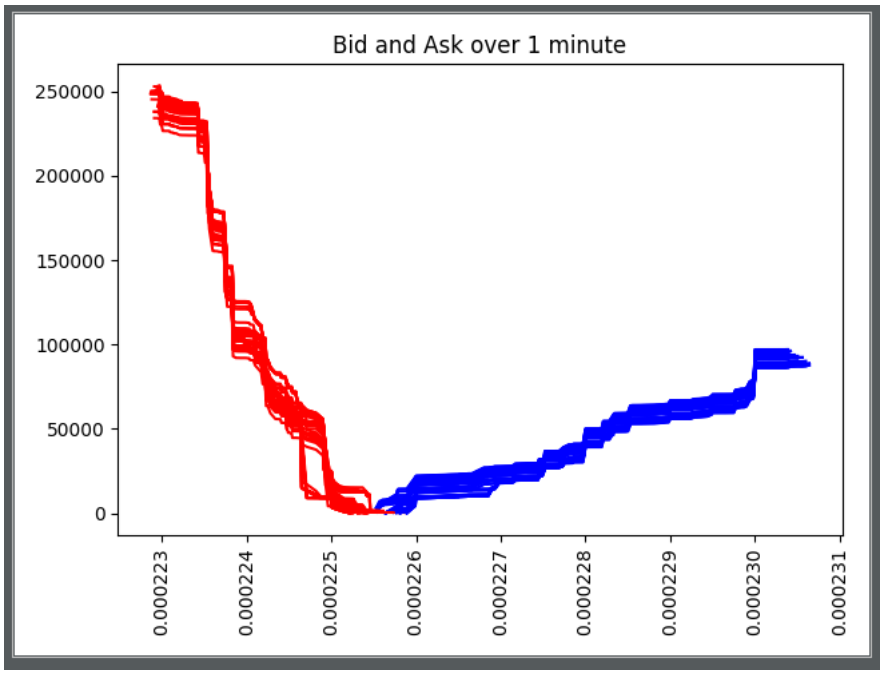


Figure 4: All bid and ask prices and quantities in 1 minute

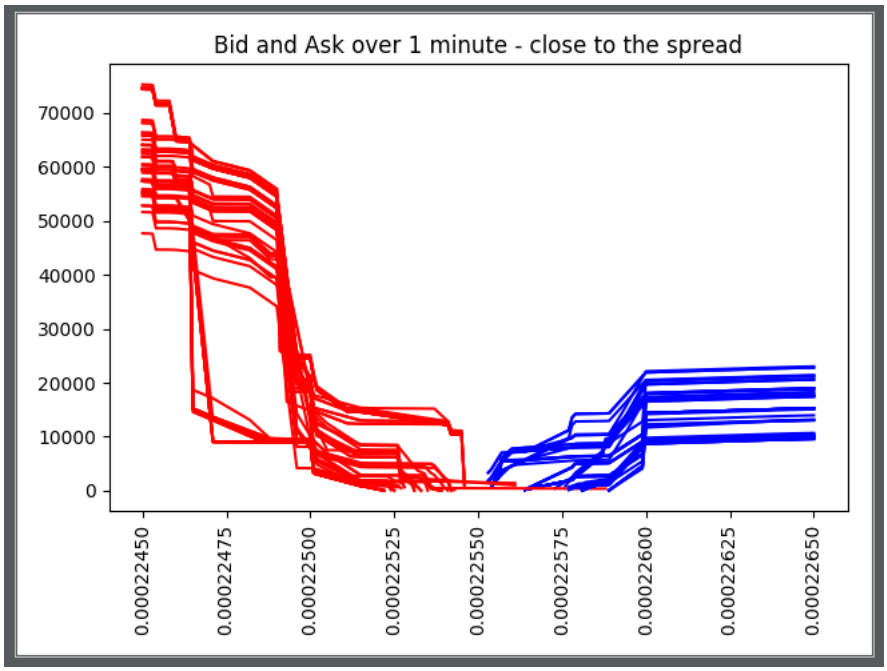


Figure 5: Detailed Price and quantity moments around the spread in 1 minute

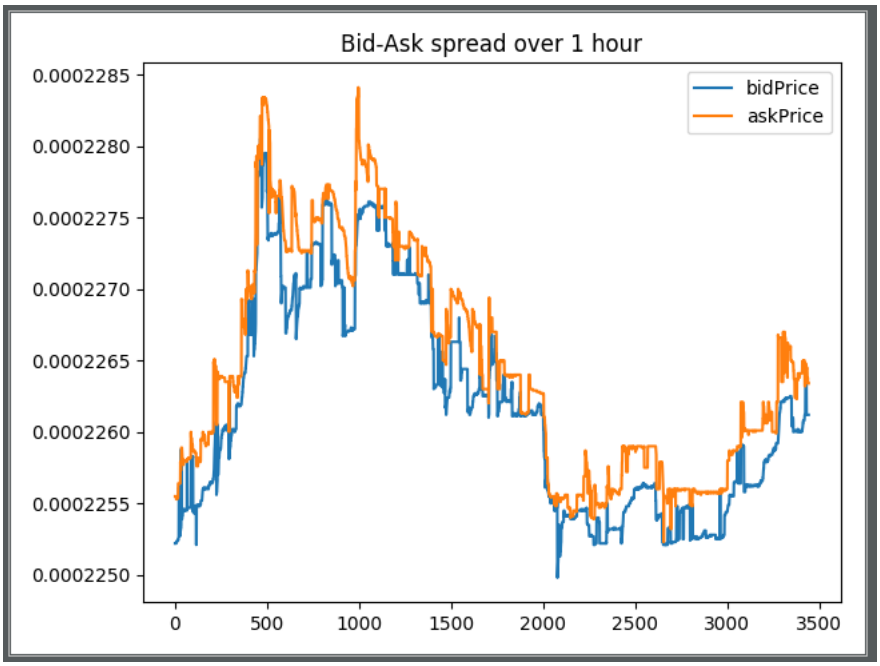


Figure 6: Max bid and min ask over 1 hour

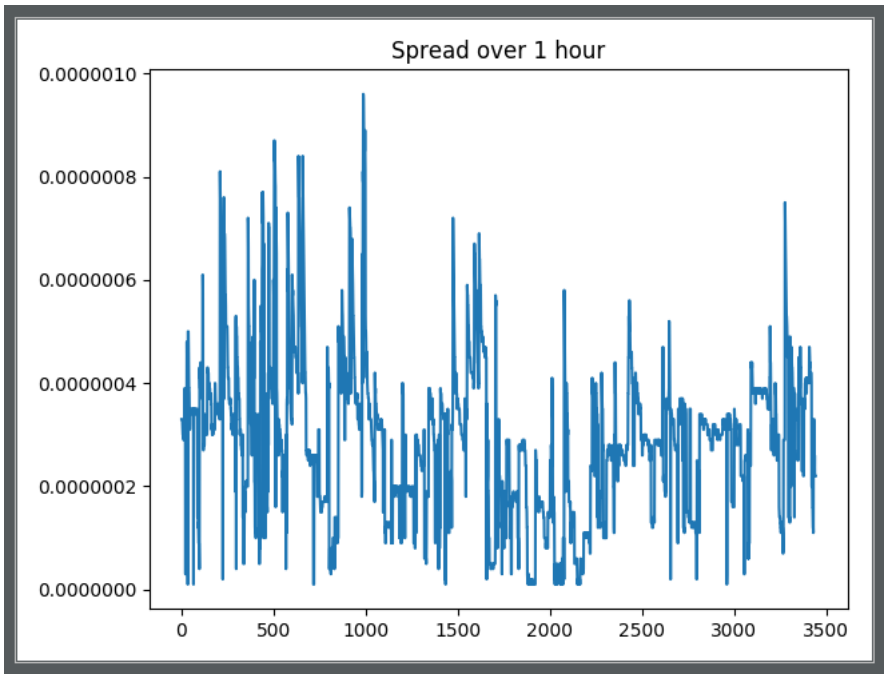


Figure 7: Max bid and min ask spread over 1 hour

Summary

Looking at the data, it is easy to spot the same patters as in the time frequencies however connecting to the exchange directly gives full details as to what is going in the market. We are not able to do this for equity or standard financial markets for free. Having a look at the data, we will be able to exactly model what is going in the market rather than second guess from some summary data such as OHLC from Yahoo.

Formatting Citations

Citations can be handled in one of three ways. The most straightforward (albeit labor-intensive) would be to hardwire your citations into your LaTeX source, as you would if you were using an ordinary word processor. Thus, your code might look something like this:

```
However, this record of the solar nebula may have been
partly erased by the complex history of the meteorite
parent bodies, which includes collision-induced shock,
thermal metamorphism, and aqueous alteration
({\it 1, 2, 5--7\/}).
```

Compiled, the last two lines of the code above, of course, would give notecalls in *Science* style:

> . . . thermal metamorphism, and aqueous alteration (*1, 2, 5–7*).

Under the same logic, the author could set up his or her reference list as a simple enumeration,

```
{\bf References and Notes}
```

```
\begin{enumerate}
\item G. Gamow, {\it The Constitution of Atomic Nuclei
and Radioactivity\/} (Oxford Univ. Press, New York, 1931).
\item W. Heisenberg and W. Pauli, {\it Zeitschr.\ f.\
Physik\/} {\bf 56}, 1 (1929).
\end{enumerate}
```

yielding

References and Notes

1. G. Gamow, *The Constitution of Atomic Nuclei and Radioactivity* (Oxford Univ. Press, New York, 1931).
2. W. Heisenberg and W. Pauli, *Zeitschr. f. Physik* **56**, 1 (1929).

That's not a solution that's likely to appeal to everyone, however — especially not to users of BIBTEX (***?***). If you are a BIBTEX user, we suggest that you use the `Science.bst` bibliography style file and the `scicite.sty` package, both of which we are downloadable from our author help site (http://www.sciencemag.org/about/authors/prep/TeX_help/). You can also generate your reference lists by using the list environment {`thebibliography`} at the end of your source document; here again, you may find the `scicite.sty` file useful.

Whether you use BIBTEX or {`thebibliography`}, be very careful about how you set up your in-text reference calls and notecalls. In particular, observe the following requirements:

1. Please follow the style for references outlined at our author help site and embodied in recent issues of *Science*. Each citation number should refer to a single reference; please do not concatenate several references under a single number.

2. Please cite your references and notes in text *only* using the standard LaTeX `\cite` command, not another command driven by outside macros.

3. Please separate multiple citations within a single `\cite` command using commas only; there should be *no space* between reference keynames. That is, if you are citing two papers whose bibliography keys are `keyname1` and `keyname2`, the in-text cite should read `\cite{keyname1,keyname2}`, *not* `\cite{keyname1, keyname2}`.

Failure to follow these guidelines could lead to the omission of the references in an accepted paper when the source file is translated to Word via HTML.

Handling Math, Tables, and Figures

Following are a few things to keep in mind in coding equations, tables, and figures for submission to *Science*.

In-line math. The utility that we use for converting from LaTeX to HTML handles in-line math relatively well. It is best to avoid using built-up fractions in in-line equations, and going for the more boring "slash" presentation whenever possible — that is, for `$a/b$` (which comes out as a/b) rather than `$\frac{a}{b}$` (which compiles as $\frac{a}{b}$). Likewise, HTML isn't tooled to handle certain overaccented special characters in-line; for $\hat{\alpha}$ (coded `$\hat{\alpha}$`), for example, the HTML translation code will return [$\hat{}(\alpha)$]. Don't drive yourself crazy — but if it's possible to avoid such constructs, please do so. Please do not code arrays or matrices as in-line math; display them instead. And please keep your coding as TeX-y as possible — avoid using specialized math macro packages like `amstex.sty`.

Displayed math. Our HTML converter sets up TeX displayed equations using nested HTML tables. That works well for an HTML presentation, but Word chokes when it comes across

a nested table in an HTML file. We surmount that problem by simply cutting the displayed equations out of the HTML before it's imported into Word, and then replacing them in the Word document using either images or equations generated by a Word equation editor. Strictly speaking, this procedure doesn't bear on how you should prepare your manuscript — although, for reasons best consigned to a note (*?*), we'd prefer that you use native TEX commands within displayed-math environments, rather than LATEX sub-environments.

Tables. The HTML converter that we use seems to handle reasonably well simple tables generated using the LATEX `{tabular}` environment. For very complicated tables, you may want to consider generating them in a word processing program and including them as a separate file.

Figures. Figure callouts within the text should not be in the form of LATEX references, but should simply be typed in — that is, `(Fig. 1)` rather than `\ref{fig1}`. For the figures themselves, treatment can differ depending on whether the manuscript is an initial submission or a final revision for acceptance and publication. For an initial submission and review copy, you can use the LATEX `{figure}` environment and the `\includegraphics` command to include your PostScript figures at the end of the compiled PostScript file. For the final revision, however, the `{figure}` environment should *not* be used; instead, the figure captions themselves should be typed in as regular text at the end of the source file (an example is included here), and the figures should be uploaded separately according to the Art Department's instructions.

What to Send In

What you should send to *Science* will depend on the stage your manuscript is in:

- **Important:** If you're sending in the initial submission of your manuscript (that is, the copy for evaluation and peer review), please send in *only* a PostScript or PDF version of the compiled file (including figures). Please do not send in the TEX source, `.sty`, `.bbl`, or other associated files with your initial submission. (For more information, please see the instructions at our Web submission site, http://www.submit2science.org/ .)

- When the time comes for you to send in your revised final manuscript (i.e., after peer review), we require that you include all source files and generated files in your upload. Thus, if the name of your main source document is `ltxfile.tex`, you need to include:

  - `ltxfile.tex`.
  - `ltxfile.aux`, the auxilliary file generated by the compilation.
  - A PostScript file (compiled using `dvips` or some other driver) of the `.dvi` file generated from `ltxfile.tex`, or a PDF file distilled from that PostScript. You do not need to include the actual `.dvi` file in your upload.
  - From BIBTEX users, your bibliography (`.bib`) file, *and* the generated file `ltxfile.bbl` created when you run BIBTEX.
  - Any additional `.sty` and `.bst` files called by the source code (though, for reasons noted earlier, we *strongly* discourage the use of such files beyond those mentioned in this document).

1. We've included in the template file `scifile.tex` a new environment, {`scilastnote`}, that generates a numbered final citation without a corresponding signal in the text. This environment can be used to generate a final numbered reference containing acknowledgments, sources of funding, and the like, per *Science* style.

Fig. 1. Please do not use figure environments to set up your figures in the final (post-peer-review) draft, do not include graphics in your source code, and do not cite figures in the text using LaTeX `\ref` commands. Instead, simply refer to the figure numbers in the text per *Science* style, and include the list of captions at the end of the document, coded as ordinary paragraphs as shown in the `scifile.tex` template file. Your actual figure files should be submitted separately.