orderbook

Andrew Liu, Khanh Nguyen and David Kane

Introduction

The limitob package provides functions for exploring and visualizing orderbook data. For example, we may want to study an orderbook immediately before a trade occurs, or maybe 50 orders before a trade occurs, or we want to visualize the change of an orderbook overtime as the price level fluctuates. The microstructure insights from this sort of analysis can help in researching short-term trading strategies.

Background

To demonstrate the capabilities of the limitob package, we consider the first 10000 orders for a single stock on June 8, 2010. The orders are in following form

```
A,31285893,1231884,11.49,200,ASK
R,31295779,1231884,150
T,31295779,1231884,11.49,50,BUY
C,31295781,1231884
```

where A, R, T, C mean Add, Replace, Trade, and Cancel order, respectively. The first number is the timestamp of the order in milliseconds after midnight of the users timezone, and the second number (or string) is the ID of the order. For a Replace the next number is the new size, while for Add and Trade price comes before size, followed by the type of order/transaction (BID/ASK or BUY/SELL).

```
> library(limitob)

> ob = orderbook(feed = "sample.txt")
> ob = read.orders(ob, 10000)
> ob

An object of class limitob
--------------------------
Current orderbook time:    09:35:02
Feed Index:                10000
Number of Bids:            631
Number of Asks:            1,856
Total Orders:              2,487
```

We create the orderbook object by setting the feed as our `sample.txt` data file. Then we read in the first 10,000 orders. We see that the 10000th order occurs at 9:35:02, and there are 631 outstanding bid orders, and 1,856 outstanding ask orders. However, sometimes the user may want to jump to a place in the orderbook by time; for example, it is interesting to view the orderbook when the market opens.

```
> ob = read.time(ob, "9:30:00")
> summary(ob)

Current time is 09:30:00

ASK price levels:   61
BID price levels:   39
Total price levels: 100
----------------------------
ASK orders:         87
BID orders:         79
Total orders:       166
----------------------------
Spread:             0.02

Mid point:          11.450
----------------------------
Inside market

Top Bid:            11.44
Size:               1,100.00

Top Ask:            11.46
Size:               657.00

> display(ob)

Current time is 09:30:00

                Price                   Ask Size
------------------------------------------------
                11.53                   100
                11.49                   100
                11.48                   800
                11.47                   100
                11.46                   657
------------------------------------------------
1,100                 11.44
100                  11.43
100                  11.42
100                  11.41
100                  11.40
------------------------------------------------
Bid Size          Price
```

Summary provides the user with a set of important summary statistics, while display by default shows the top five bid and ask price levels. Suppose we wanted to see the first trade after this time.

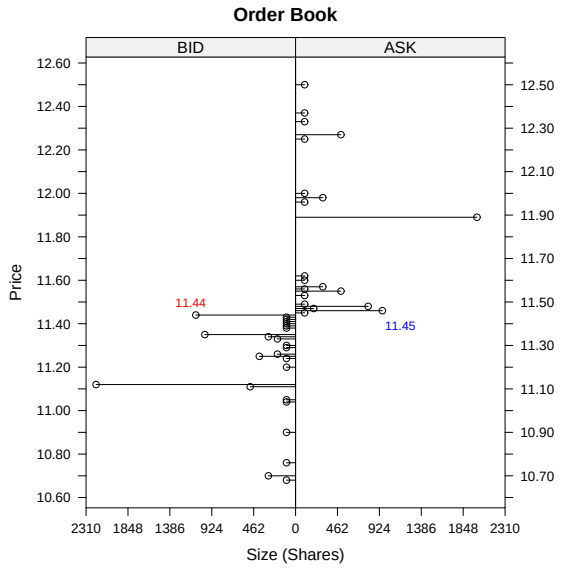
```
> ob = next.trade(ob)
> display(ob, n = 1)

Current time is 09:30:00

                Price                   Ask Size
------------------------------------------------
                11.45                   100
------------------------------------------------
1,100                 11.44
------------------------------------------------
Bid Size          Price
```

Note that to save space I set `n = 1` so that only the top three bid and ask price levels are displayed. Sometimes a graphical representation is more useful than simply seeing the price levels and order sizes in text. By default, the maximum and minimum price levels shown are 10% above and below the midpoint price, respectively.

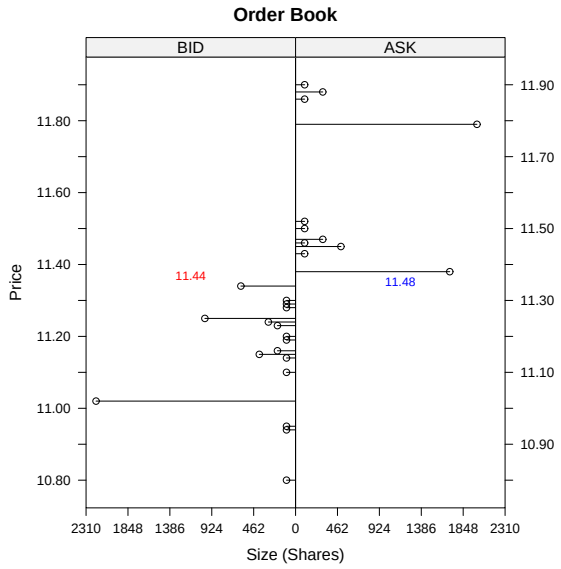
```
> plot(ob)
```



Users might want to view the orderbook fifty orders prior to the trade occurring. Here we also demonstrate setting the bounds to only 5% above and below the midpoint price.

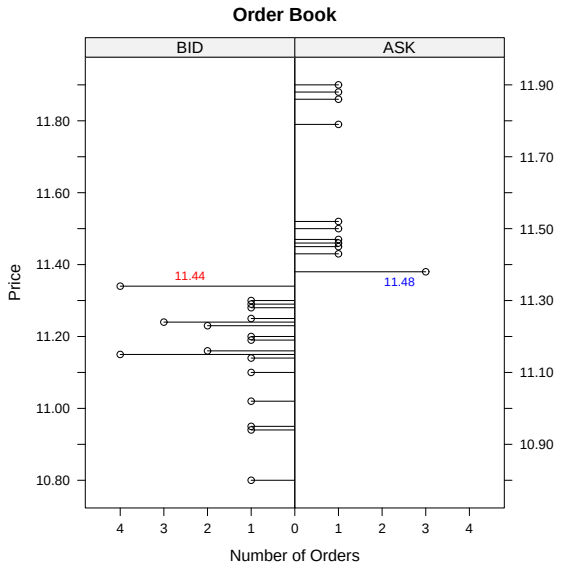
```
> ob.new = read.orders(ob, -50)

> plot(ob.new, bounds = 0.05)
```



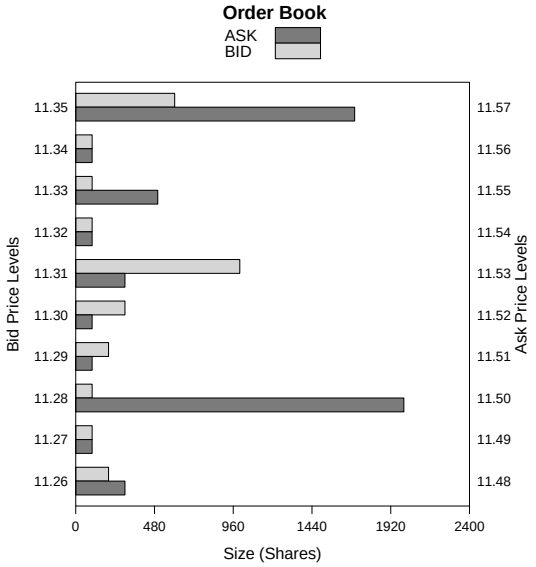
Note that there is a particularly large order around $11.03. It is helpful to know whether the volume at that price level is comprised of a single order, or several.

```
> plot(ob.new, bounds = 0.05, type = "o")
```



Viewing the orderbook with bids on one side and asks on another is useful, but sometimes users may want to view them side by side to more directly compare the supply and demand at each price level.

```
> plot(ob.new, type = "s")
```

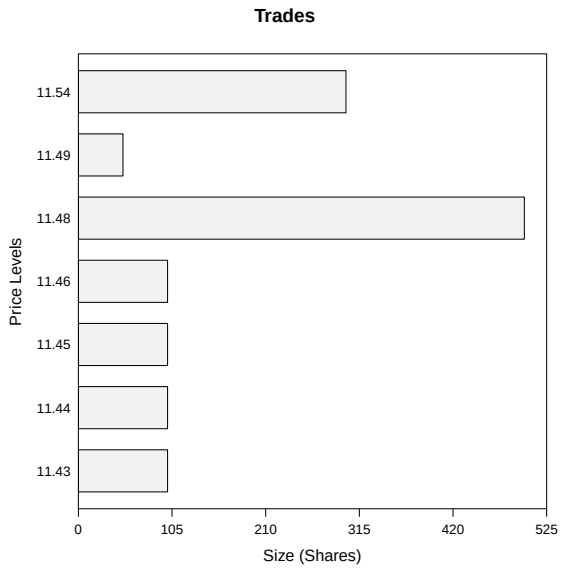


Users can also view the previous trade, and then pass in the time at each trade, along with the object, to view a simple animation between the two times.

```
> prev.ob = previous.trade(ob)
> animate.ob(ob, "9:30:00", "9:32:00")
```

Finally, users can easily plot the trade data by using `plot.trade`. This creates a simple bar graph of the number of shares traded at each price level.

```
> plot.trade(ob.new)
```



Simulation

The limitob object supports adding, replacing, and cancelling orders. To add an order, the user needs to specify the price, size, and type. Time and ID are optional, and will default to the maximum time and the maximum ID + 1, respectively. For replacing an order, only ID and size need to be given, and for cancelling an order, only ID is necessary. Market orders are also possible by specifying the size and side (BUY/SELL).

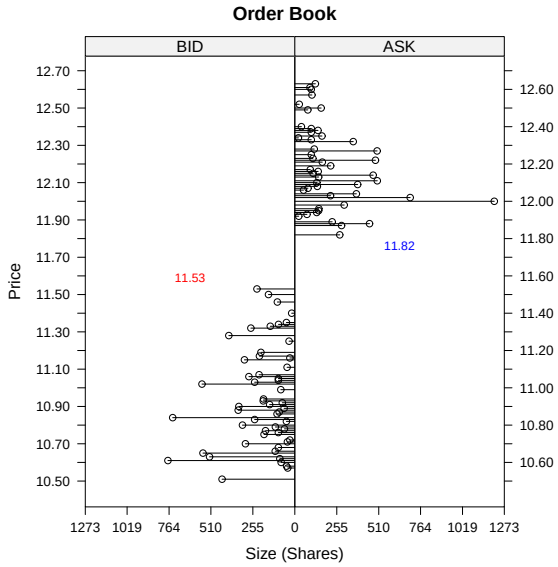
```
> display(ob)
> ob = add.order(ob, stuff)
> ob = remove.order(ob, stuff)
> ob = replace.order(ob, stuff)
> ob = market.order(ob, 200, "BUY")
> display(ob)
```

Using these tools, the user can write functions to simulate the movement of an orderbook.

In the following example, we consulted Gilles (2006). We simulate 1000 orders. In each iteration there is a 50% chance for a cancel order to be placed, 20% chance for a market order, and 30% chance for a limit order. Orders are cancelled completely randomly, and for a market order there is a 50-50 chance for a buy or sell order to be placed. The size of the market order always corresponds to the size of the best ask or bid at the front of the queue. When a limit order is placed, there is a 50-50 chance for it to be an ask or bid. Then there is a 35% chance for the price to be within the spread, in which case a price is chosen based on a uniform distribution. If the price is determined to be outside of the spread, a price is chosen using a power law distribution. The size follows a log-normal distribution.

```
> ob = simulate(ob)
```

```
> plot(ob)
```



Future work

There is still much room for improvement with `limitob`. Aside from additional optimization, we are looking into creating an "exchange" package that can better simulate trading. Ultimately, we would like to have the ability to backtest high frequency trading strategies.

Conclusion

In this short article, we have described the limitob package. limitob aims to provide user-friendly statistical and visualization tools for analyzing orderbooks. We demonstrated the functionality of the package through a series of examples. Users who deal frequently with orderbook data (i.e high frequency traders) will hopefully find the package useful. While the curent package is not complex enough to be a stand-alone platform for developing trading stategies, it can be useful for generating ideas for strategies.

Andrew Liu, Khanh Nguyen and David Kane
Kane Capital Management
Cambridge, Massachusetts, USA
`Andrew.T.Liu@williams.edu`, `knguyen@cs.umb.edu`, *and* `dave@kanecap.com`

Bibliography

D. Gilles. *Asynchronous Simulations of a Limit Order Book*. PhD thesis, University of Manchester, 2006.