obAnalytics Guide

Phil Nocturne

2015-09-03

Contents

Overview

Overview to the package and this guide.

Loading data

lala

Expected format

lala

Processing, saving and loading

lala

Preprocessed example data

lala

Trades

trades…

```
trades.ex <- tail(lob.data$trades, 10)
trades.ex$volume <- round(trades.ex$volume*10^-8, 2)
print(trades.ex, row.names=F)
```

timestamp	price	volume	direction	maker.event.id	taker.event.id	maker	taker
2015-05-01 04:59:27.503	235.73	0.01	buy	49630	49777	65619731	65619806
2015-05-01 04:59:27.532	235.79	0.02	buy	49672	49778	65619752	65619806
2015-05-01 04:59:41.568	235.77	0.02	buy	49802	49821	65619818	65619826
2015-05-01 04:59:55.877	235.77	0.02	buy	49803	49871	65619818	65619851
2015-05-01 04:59:59.217	235.77	0.38	buy	49804	49877	65619818	65619854
2015-05-01 05:00:08.361	235.77	0.12	sell	49878	49894	65619854	65619862
2015-05-01 05:00:08.395	235.58	0.21	sell	49406	49895	65619615	65619862
2015-05-01 05:00:08.424	235.01	0.07	sell	46221	49896	65618028	65619862
2015-05-01 05:00:10.108	235.79	0.02	buy	49816	49900	65619824	65619864
2015-05-01 05:03:13.566	235.45	0.05	sell	49992	50255	65619912	65620048

Visualisation

lala

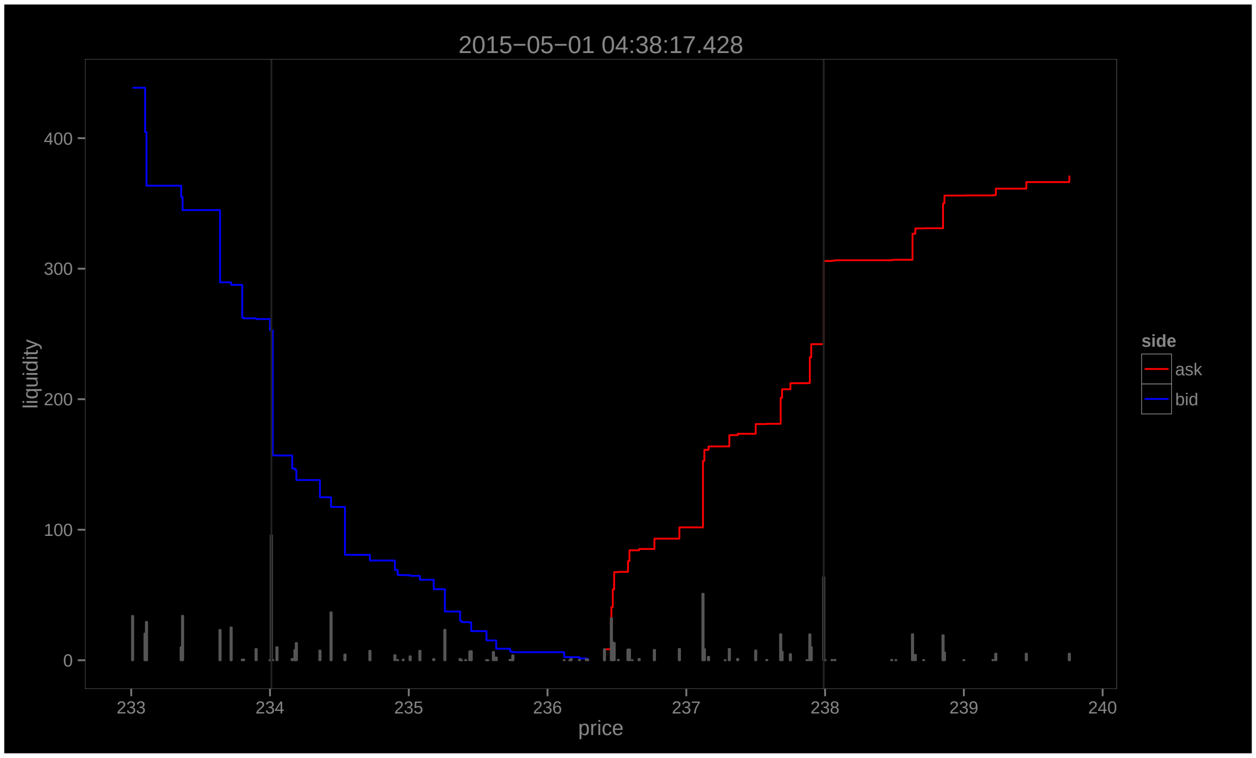
Order book shape

The purpose of the cumulative volume graph is to quickly identify the shape of the limit order book for the given point in time. The "shape" is defined as the cumulative volume available at each price level, starting at the best bid/ask.

Using this shape, it is possible to visually summarise order book imbalance and market depth.

```
# get a limit order book for a specific point in time, limited to +- 150bps
# above/below best bid/ask price.
lob <- orderBook(lob.data$events,
    tp=as.POSIXct("2015-05-01 04:38:17.429", tz="UTC"), bps.range=150)

# visualise the order book liquidity.
plotCurrentDepth(lob, volume.scale=10^-8)
```

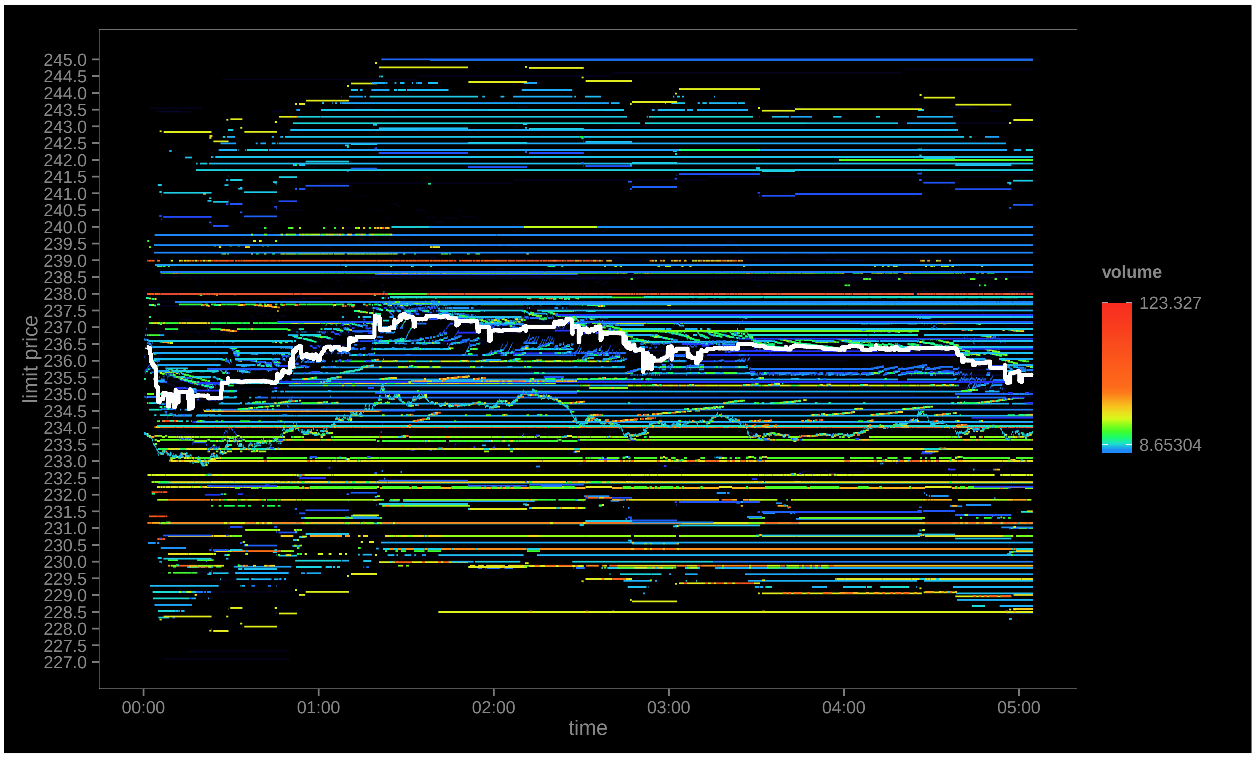


In the figure above, an order book has been reconstructed with the *orderBook* function for a specific point in time. The visualisation produced with the *plotCurrentDepth* function depicts a number of order book features. Firstly, the embedded bar chart at the bottom of the plot shows the amount of volume available at specific price levels ranging from the *bid* side on the left (blue) through to the *ask* side (red) on the right. Secondly, the blue and red lines show the *cumulative* volume of the bar chart for the bid and ask sides of the order book respectively. Finally, the two subtle vertical lines at price points $234 and $238 show the position of the top 1% largest limit orders.

Price level volume

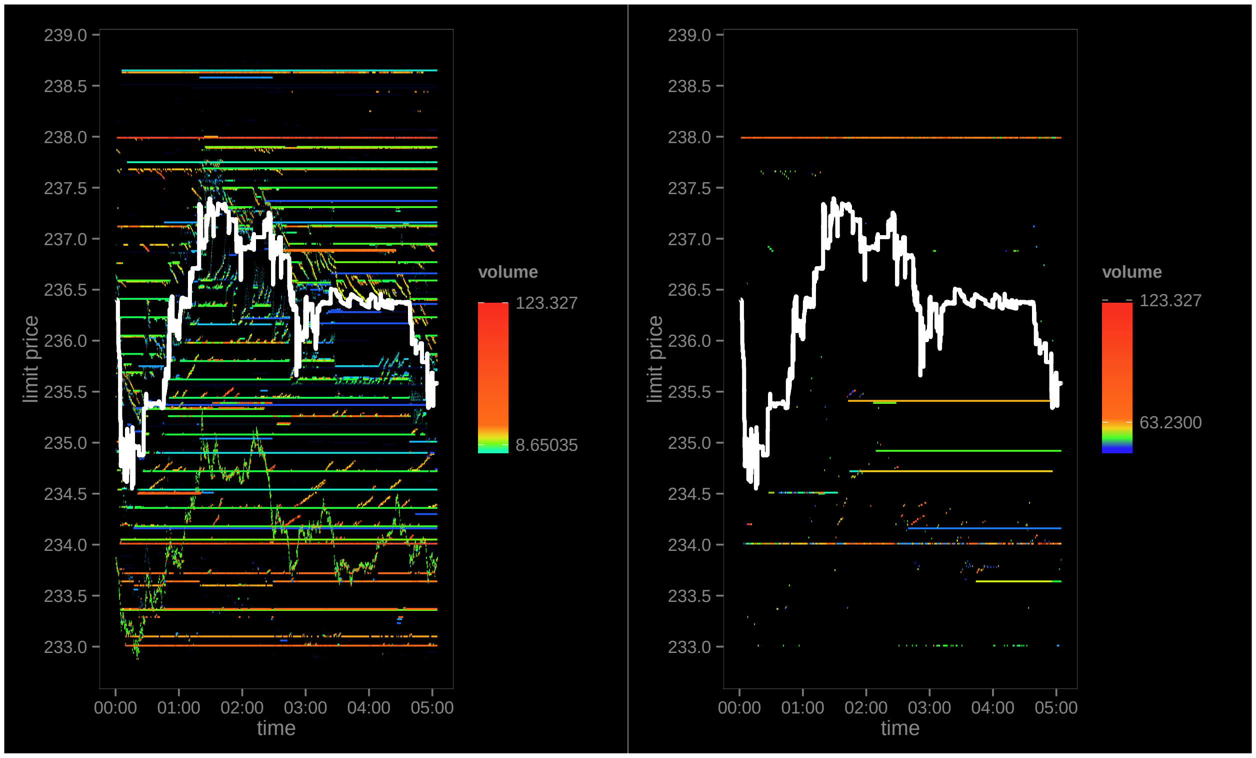
All lob.data:

```
# plot all lob.data price level volume between $247 and $245 and overlay the
# market midprice.
spread <- getSpread(lob.data$depth.summary)
plotPriceLevels(lob.data$depth, spread, price.from=227, price.to=245,
    volume.scale=10^-8, col.bias=0.25, show.mp=T)
```



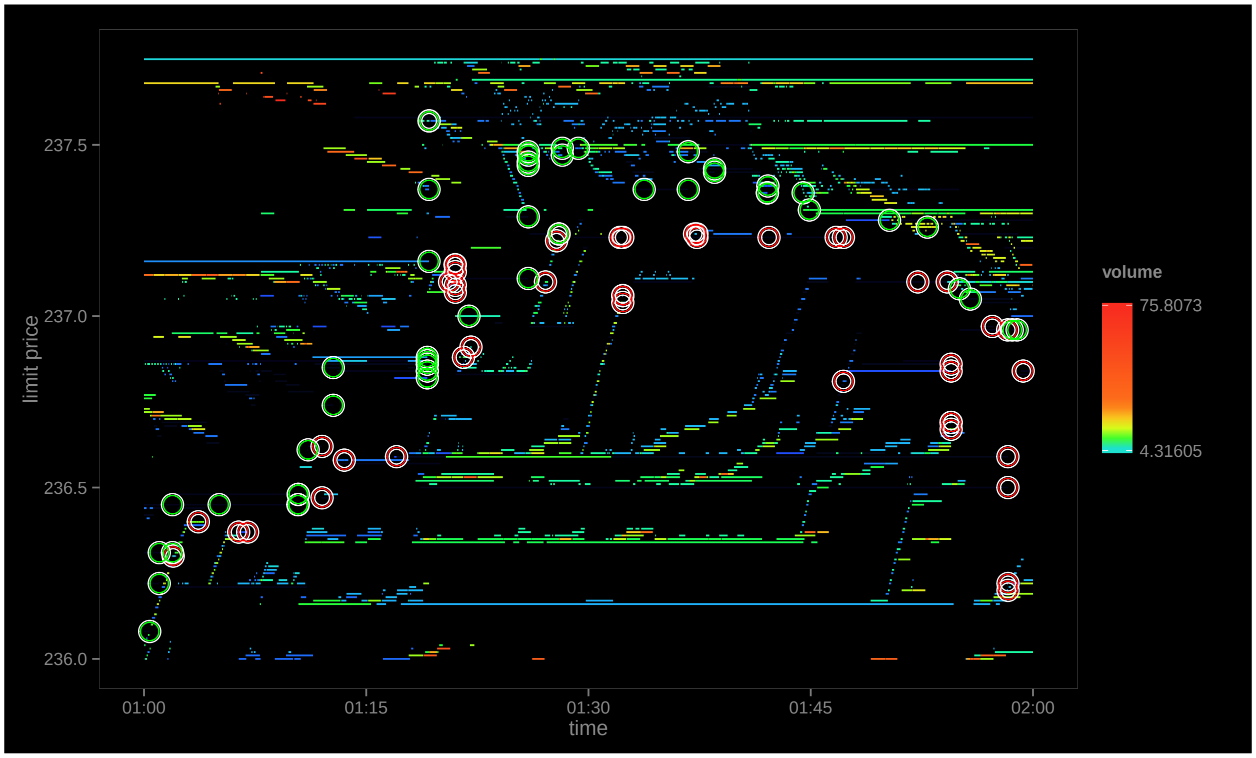
plot all depth levels, rescaling the volume by 10^-8. produce 2 plots side-by-side: second plot contains depth levels with > 50 units of volume.

```
spread <- with(lob.data, getSpread(depth.summary))
p1 <- with(lob.data, plotPriceLevels(depth, spread, col.bias=0.1, volume.scale=10^-8))
p2 <- with(lob.data, plotPriceLevels(depth, spread, col.bias=0.1, volume.scale=10^-8, volume.from=50))
library(grid)
pushViewport(viewport(layout=grid.layout(1, 2)))
print(p1, vp=viewport(layout.pos.row=1, layout.pos.col=1))
print(p2, vp=viewport(layout.pos.row=1, layout.pos.col=2))
```



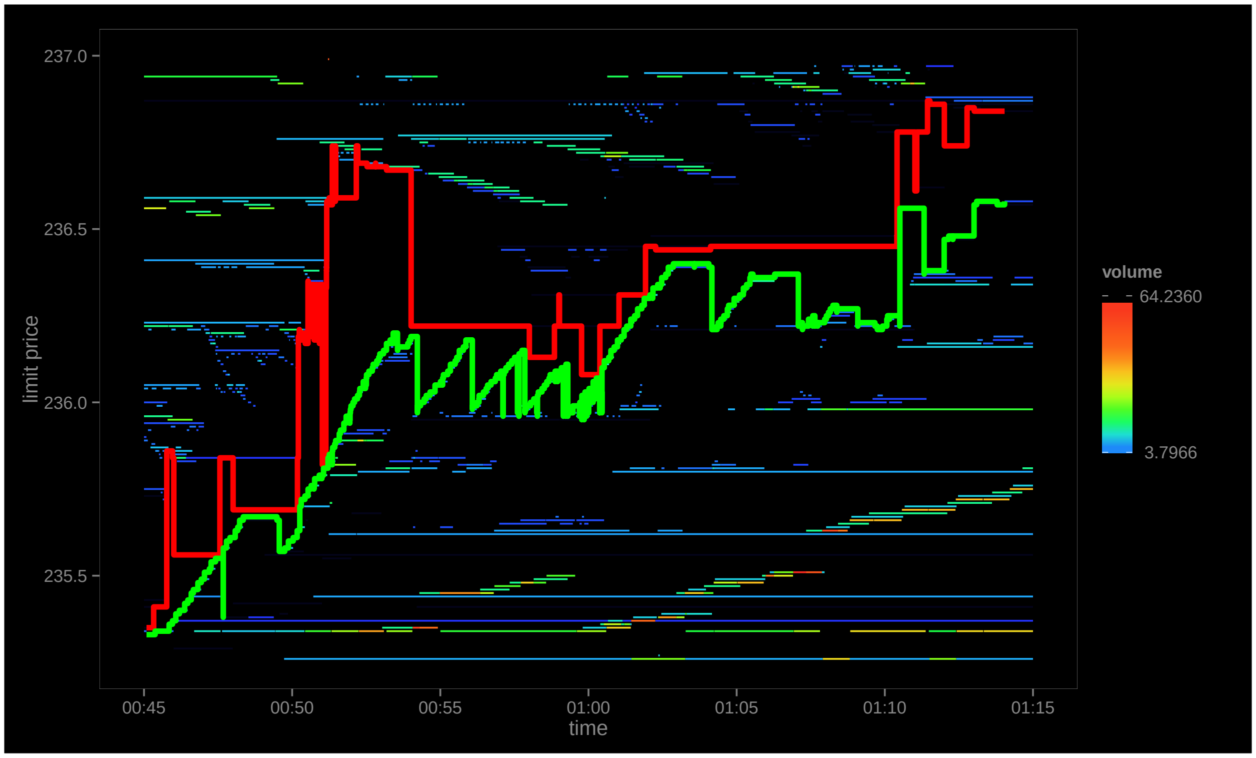
with trades:

```
# plot 1 hour of trades centred around the bid/ask spread.
plotPriceLevels(lob.data$depth, trades=lob.data$trades,
    price.from=236, price.to=237.75, volume.scale=10^-8, col.bias=0.2,
    start.time=as.POSIXct("2015-05-01 01:00:00.000", tz="UTC"),
    end.time=as.POSIXct("2015-05-01 02:00:00.000", tz="UTC"))
```



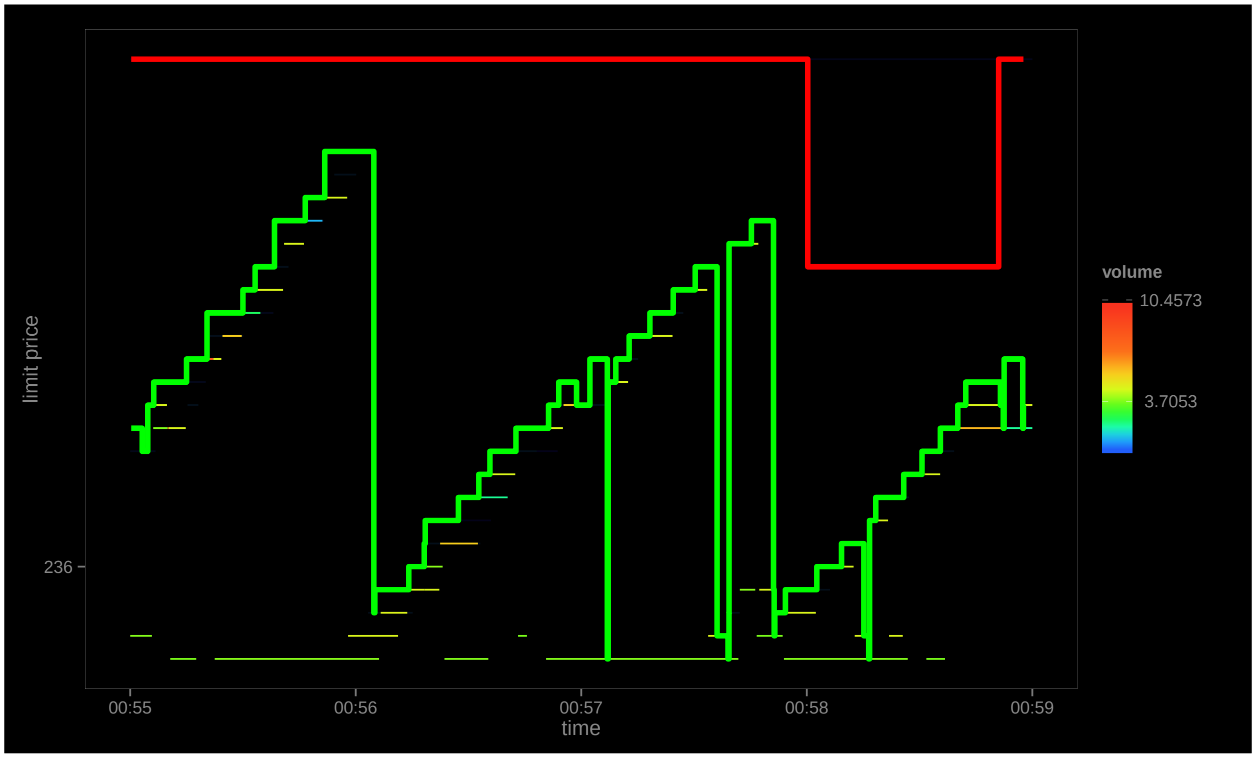
30 minute zoom:

```
# zoom in to 30 minutes of bid/ask quotes.
spread <- getSpread(lob.data$depth.summary)
plotPriceLevels(lob.data$depth, spread, price.from=235.25, price.to=237,
    start.time=as.POSIXct("2015-05-01 00:45:00.000", tz="UTC"),
    end.time=as.POSIXct("2015-05-01 01:15:00.000", tz="UTC"),
    volume.scale=10^-8, col.bias=0.5, show.mp=F)
```



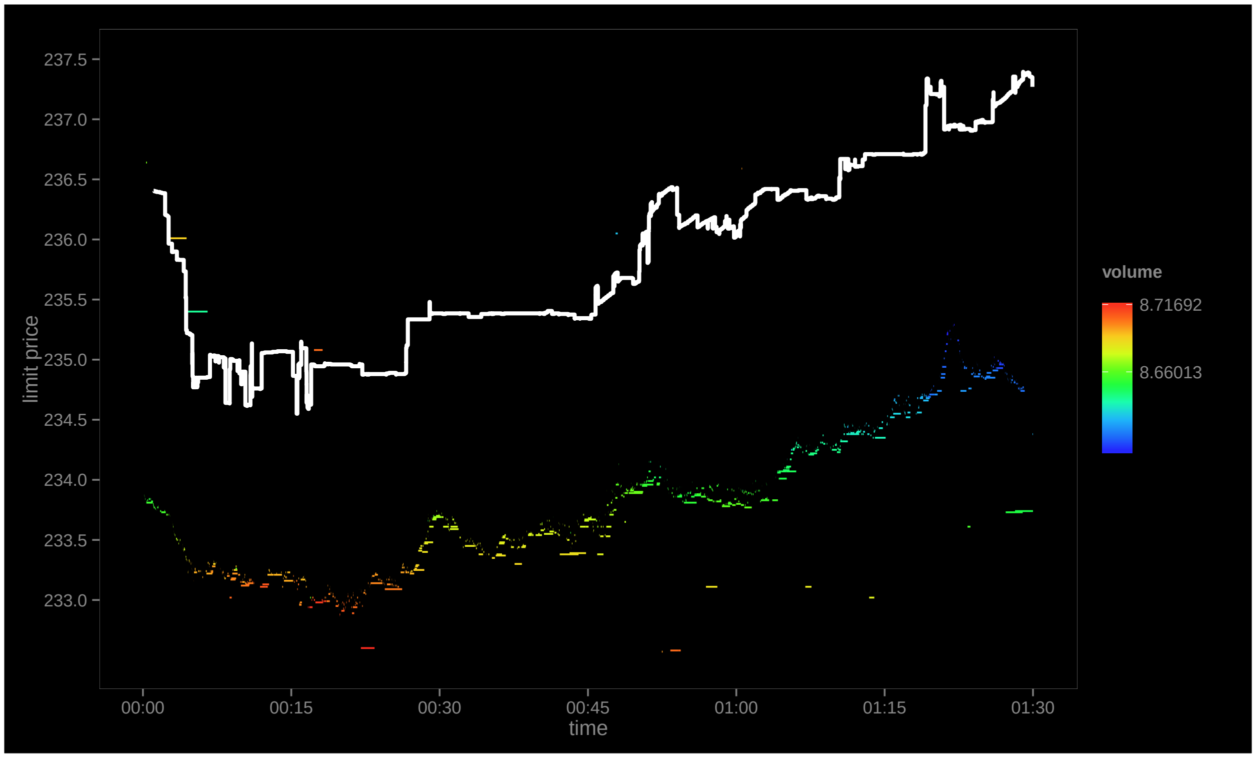
4 minute zoom:

```
# zoom in to 4 minutes of bid/ask quotes.
spread <- getSpread(lob.data$depth.summary)
plotPriceLevels(lob.data$depth, spread, price.from=235.90, price.to=236.25,
    start.time=as.POSIXct("2015-05-01 00:55:00.000", tz="UTC"),
    end.time=as.POSIXct("2015-05-01 00:59:00.000", tz="UTC"),
    volume.scale=10^-8, col.bias=0.5, show.mp=F)
```

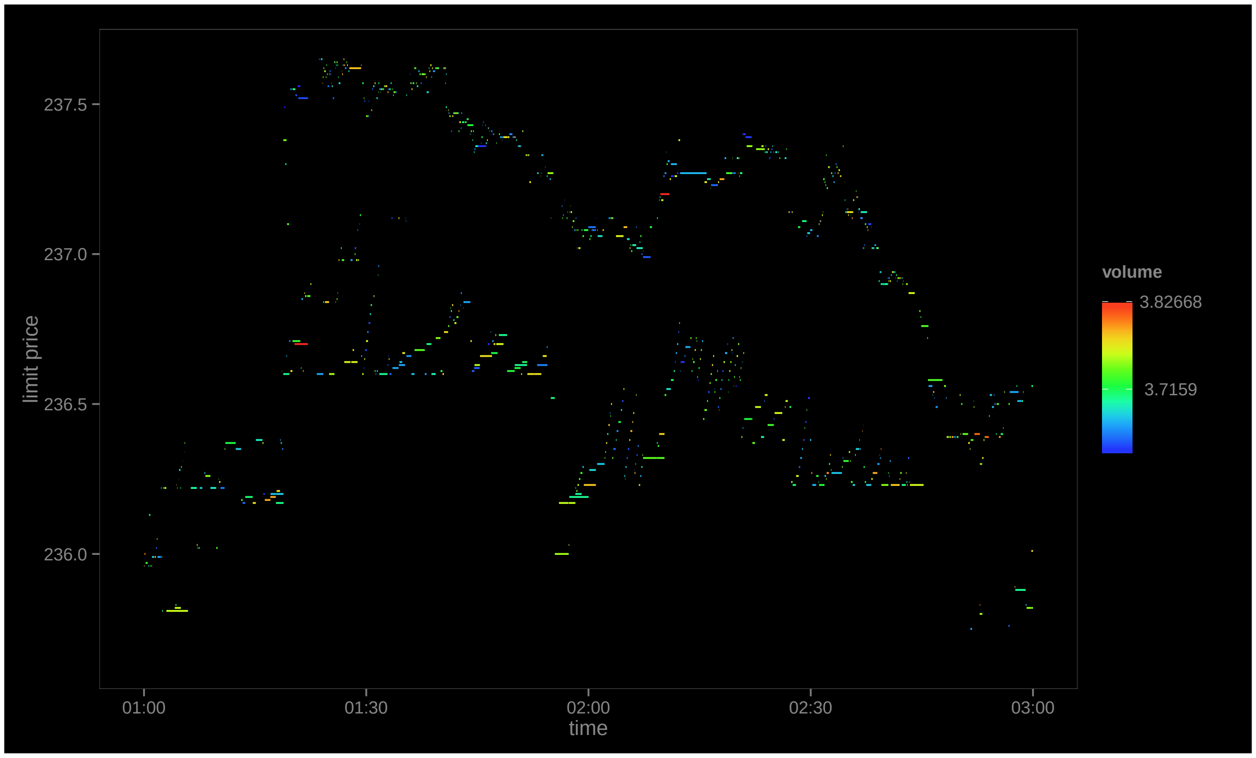


individual market paricipant:

```
#
spread <- getSpread(lob.data$depth.summary)
plotPriceLevels(lob.data$depth, spread, price.from=232.5, price.to=237.5,
    volume.scale=10^-8, col.bias=1, show.mp=T,
    end.time=as.POSIXct("2015-05-01 01:30:00.000", tz="UTC"),
    volume.from=8.59, volume.to=8.72)
```



```
#
plotPriceLevels(lob.data$depth, price.from=235.65, price.to=237.65,
    volume.scale=10^-8, col.bias=1,
    start.time=as.POSIXct("2015-05-01 01:00:00.000", tz="UTC"),
    end.time=as.POSIXct("2015-05-01 03:00:00.000", tz="UTC"),
    volume.from=3.63, volume.to=3.83)
```

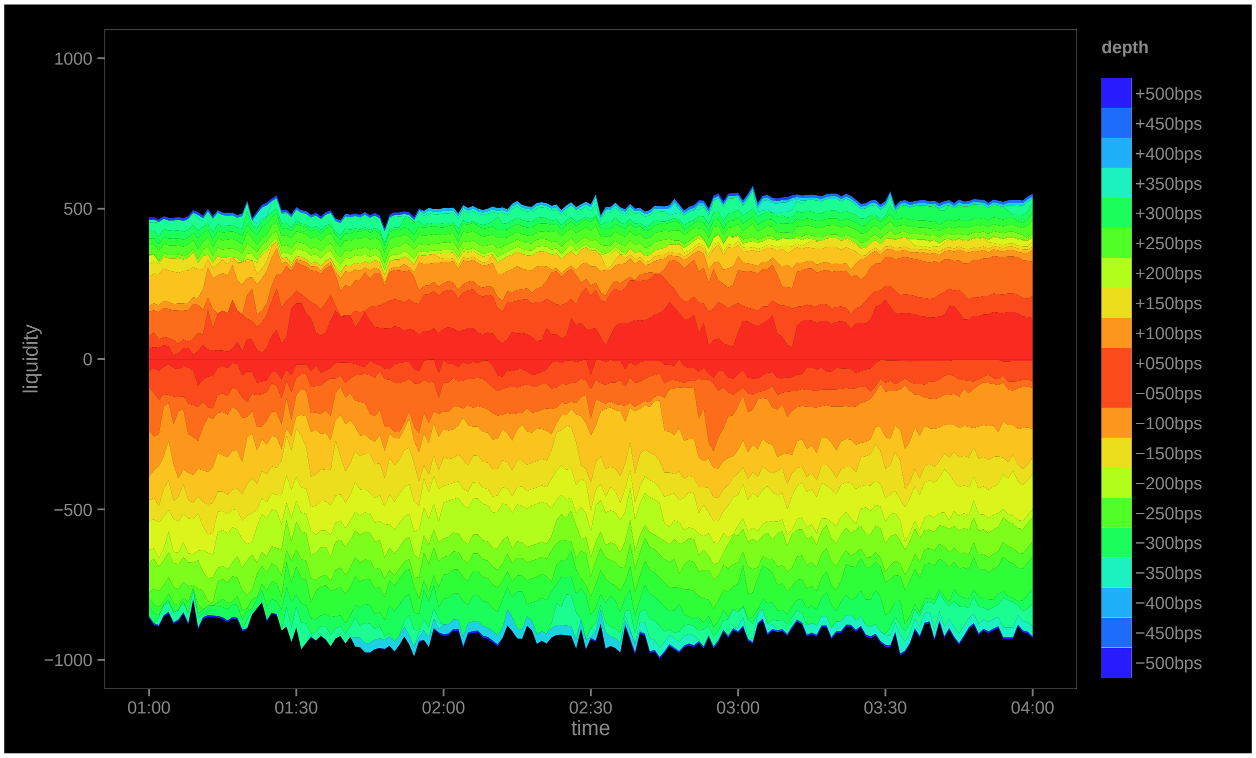


The available volume at each price level is colour coded according to the range of volume at all price levels. The colour coding follows the visible spectrum, such that larger amounts of volume appear "hotter" than smaller amounts, where cold = blue, hot = red. Since the distribution of limit order size exponentially decays, it can be difficult to visually differentiate: most values will appear to be blue. The function provides price, volume and a colour bias range to overcome this.

Liquidity

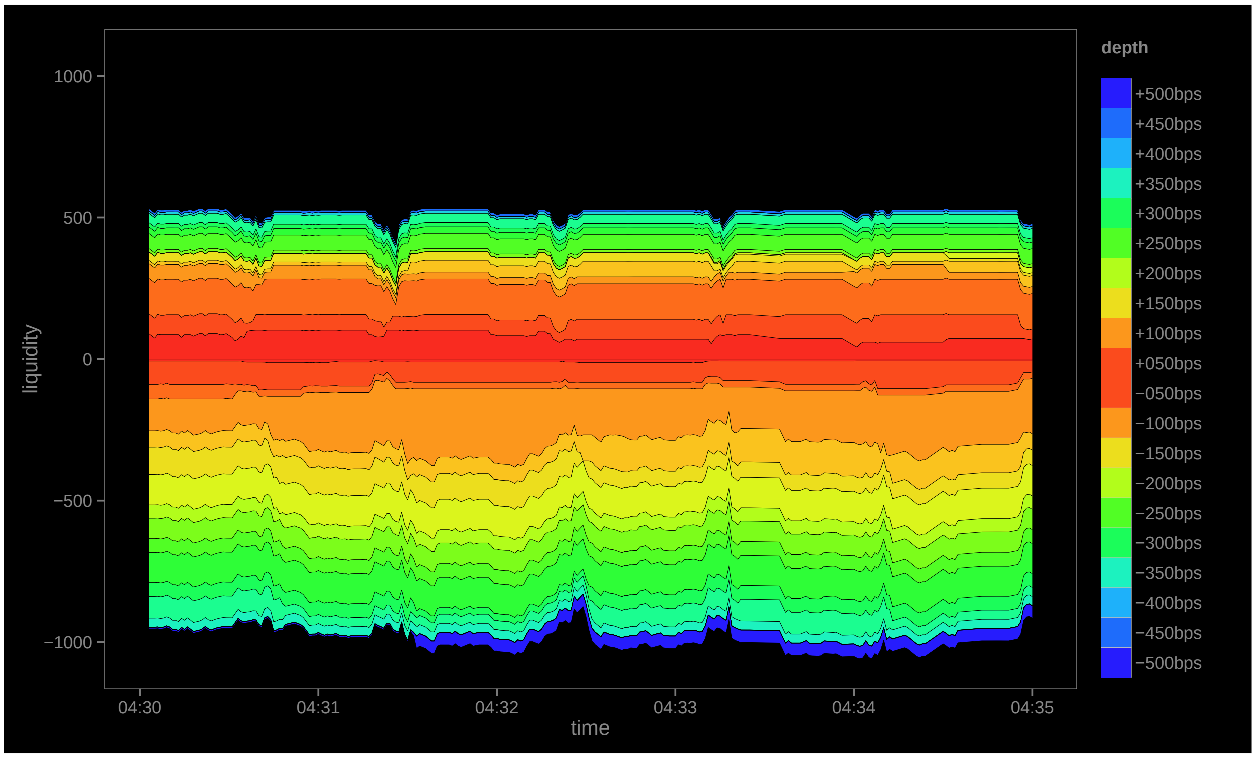
liquidity. . .

```
plotVolumePercentiles(lob.data$depth.summary, volume.scale=10^-8, perc.line=F, start.time=as.POSIXct("2
    end.time=as.POSIXct("2015-05-01 04:00:00.000", tz="UTC"))
```



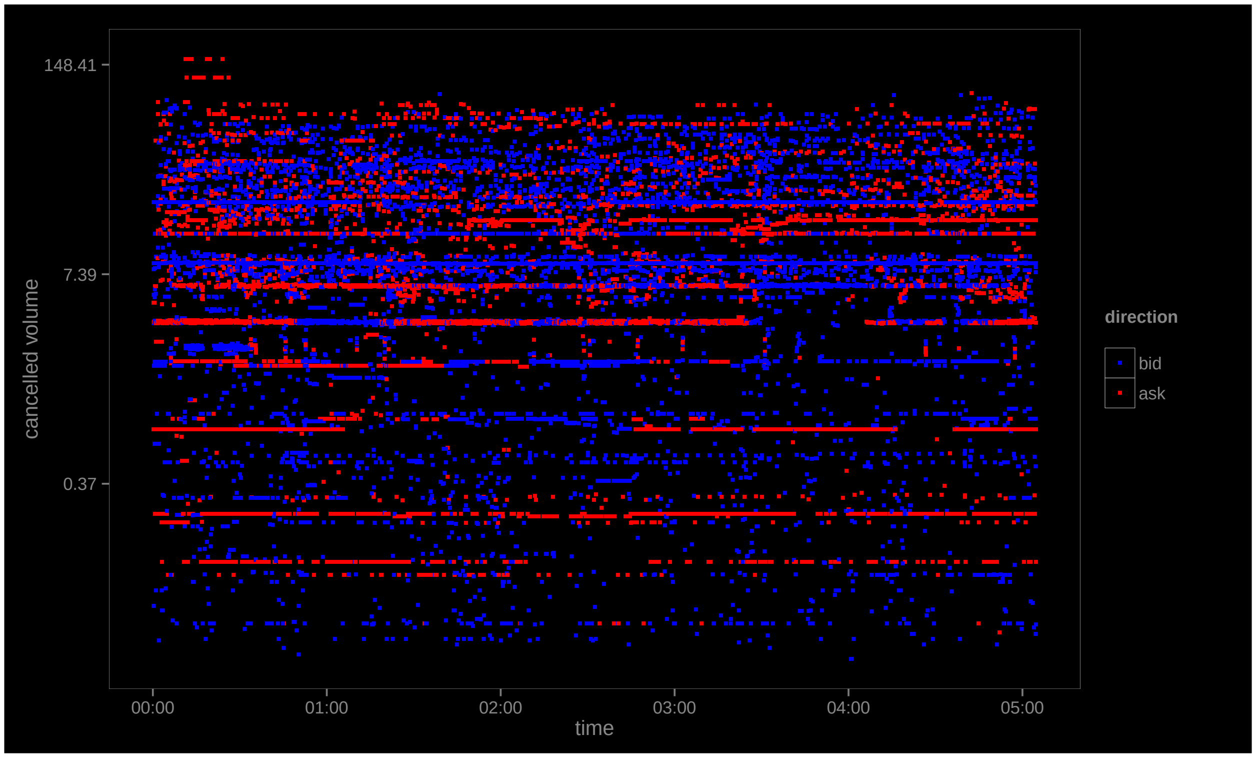
another. . .

```r
# visualise 15 minutes of order book liquidity.
# data will be aggregated to second-by-second resolution.
plotVolumePercentiles(lob.data$depth.summary,
start.time=as.POSIXct("2015-05-01 04:30:00.000", tz="UTC"),
end.time=as.POSIXct("2015-05-01 04:35:00.000", tz="UTC"),
volume.scale=10^-8)
```

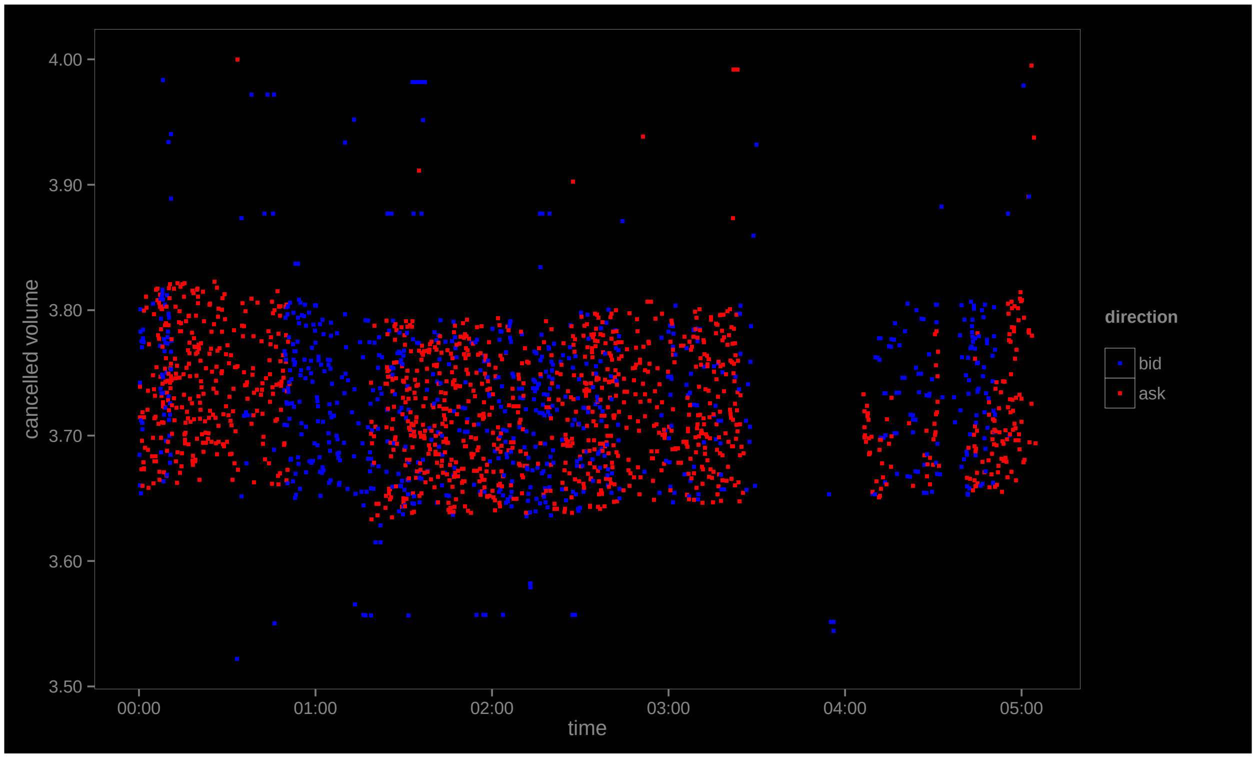


Fleeting orders

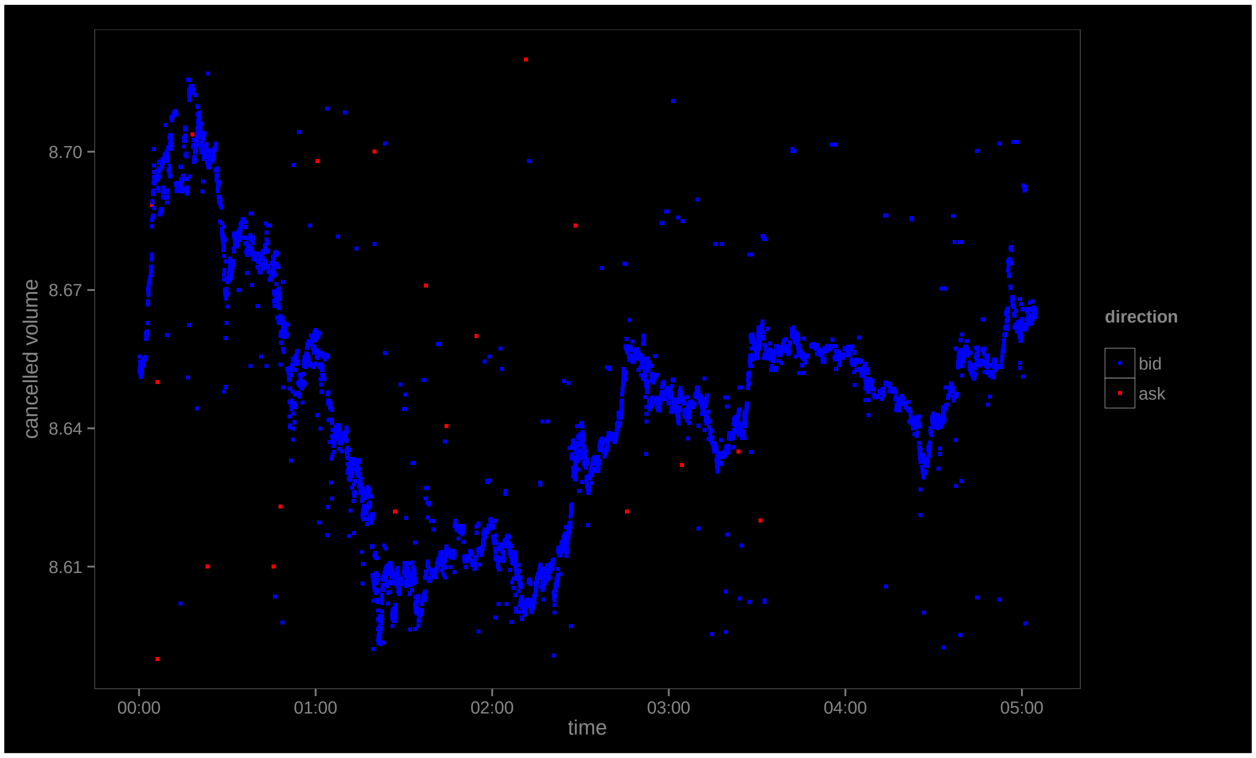
```
plotVolumeMap(lob.data$events, volume.scale=10^-8, log.scale = T)
```



```
plotVolumeMap(lob.data$events, volume.scale=10^-8, volume.from=3.5, volume.to=4)
```



```
plotVolumeMap(lob.data$events, volume.scale=10^-8, volume.from=8.59, volume.to=8.72)
```



Analysis

lala.

Order book reconstruction

lala.

```
tp <- as.POSIXct("2015-05-01 04:25:15.342", tz="UTC")
ob <- orderBook(lob.data$events, max.levels=10)
print(ob)
```

id	timestamp	liquidity	price	price	liquidity	timestamp	id
65619912	2015-05-01 05:03:13.566	0.16	235.45	235.71	3.91	2015-05-01 05:04:16.670	65620105
65620122	2015-05-01 05:04:26.395	1.10	235.12	235.71	7.70	2015-05-01 05:04:42.957	65620140
65620109	2015-05-01 05:04:18.993	2.03	235.10	235.72	7.91	2015-05-01 05:01:00.940	65619914
65618028	2015-05-01 05:00:08.424	4.57	235.01	235.80	21.11	2015-05-01 05:04:17.834	65620107
65619358	2015-05-01 04:54:21.109	4.67	234.95	235.81	34.31	2015-05-01 05:03:45.456	65620086
65598930	2015-05-01 00:39:56.799	4.80	234.92	235.84	50.29	2015-05-01 05:04:41.296	65620138
65620023	2015-05-01 05:02:33.711	5.88	234.74	235.85	73.11	2015-05-01 05:04:19.535	65620112
65620062	2015-05-01 05:03:28.263	16.86	234.73	235.87	79.90	2015-05-01 04:55:25.319	65619475
65619669	2015-05-01 04:57:31.676	23.99	234.72	235.90	79.94	2015-05-01 04:58:04.466	65619719
65597424	2015-05-01 00:23:05.230	28.36	234.54	236.05	86.75	2015-05-01 04:55:15.408	65619449

Market impacts

all impacts

lala.

```
impacts <- tradeImpacts(lob.data$trades)
impacts <- impacts[impacts$dir == "sell", ]
bps <- 10000 * (impacts$max.price - impacts$min.price) / impacts$max.price
types <- with(lob.data, events[match(impacts$id, events$id), ]$type)
impacts <- cbind(impacts, type=types, bps)
head(impacts[order(-impacts$bps), ], 10)
```

id	max.price	min.price	vwap	hits	vol	end.time	type	bps
65596324	235.09	234.20	234.33	8	37.13	2015-05-01 00:11:01.864	market	37.86
65619862	235.77	235.01	235.54	3	0.40	2015-05-01 05:00:08.424	market	32.23
65605893	236.96	236.20	236.27	5	8.61	2015-05-01 01:58:18.963	pacman	32.07
65619442	235.74	235.06	235.18	13	19.27	2015-05-01 04:55:13.891	market	28.85
65596235	235.22	234.55	234.85	2	0.25	2015-05-01 00:10:17.921	pacman	28.48
65608339	237.09	236.48	236.61	7	17.25	2015-05-01 02:26:54.539	market	25.73
65610618	236.27	235.75	235.77	6	31.21	2015-05-01 02:51:13.180	market-limit	22.01
65605081	237.23	236.81	237.06	2	0.05	2015-05-01 01:47:12.365	market	17.70
65596651	234.96	234.56	234.74	2	0.25	2015-05-01 00:15:10.098	market	17.02
65596775	234.57	234.19	234.35	5	29.53	2015-05-01 00:16:33.253	pacman	16.20

individual impact

lala.

```r
impact <- with(lob.data, trades[trades$taker == 65596324,
    c("timestamp", "price", "volume", "maker")])
makers <- with(lob.data, events[match(impact$maker, events$id), ])
makers <- makers[makers$action == "created",
    c("id", "timestamp", "aggressiveness.bps")]
impact <- cbind(impact, maker=makers[match(impact$maker, makers$id),
    c("timestamp", "aggressiveness.bps")])
age <- impact$timestamp - impact$maker.timestamp
impact <-  cbind(impact[!is.na(age), c("timestamp", "price", "volume",
    "maker.aggressiveness.bps")], age[!is.na(age)])
colnames(impact) <- c("timestamp", "price", "volume", "maker.agg", "age")
impact$volume <- impact$volume*10^-8
print(impact)
```

timestamp	price	volume	maker.agg	age
2015-05-01 00:11:01.533	235.09	0.21	16.62	0.891 secs
2015-05-01 00:11:01.563	234.70	1.00	5.54	3.331 secs
2015-05-01 00:11:01.592	234.70	1.03	0.00	1.094 secs
2015-05-01 00:11:01.657	234.42	3.81	-11.93	1.321 secs
2015-05-01 00:11:01.720	234.37	9.02	-13.64	6.730 secs
2015-05-01 00:11:01.786	234.35	3.68	-14.49	6.128 secs
2015-05-01 00:11:01.864	234.20	16.38	-20.88	5.552 secs