

## BdiF HW2

### 1. Data generator:

Data generator function is implemented with C++ (DataGenerator.cpp).

Adjust the number of dates, which is the parameter N in the cpp file in the first line, we can adjust the data size we want to generate. Default number is 50, and we will get a data.csv with about 54 Mb. This csv file was used to test my q code to calculate daily vwap.

The data contains information in "Date", "Time", "Symbol", "Price", and "Volume".

There are 30 stocks in the generated data file with names from "stock1" to "stock30". The trading time is set to from 2018/01/01, 9:00:00, with time tick of 1 second. The stock price is generated from uniform distribution [10, 200], and volume from [100, 5000]. The stock trading tick is 1 cent. The traded stock symbol is randomly generated from the 30 stocks.

### 2. Calculate vwap

The code is implemented with q (hw2.q)

In the q code, there are three functions.

.taq.logline( ) gives summary output in the screen;

.taq.import\_file ( ) loads csv data into kdb to create a table "trade\_data";

.taq.cal\_vwap\_file ( ) loads csv data into kdb (through calling function

.taq.import\_file ( ) ), calculate the daily vwap, then write into a csv file with name result.csv.

Steps in q with test data.csv file:

1. load q code into kdb:

```
q)\l /Users/xiaobohe/q/BDiF/hw2.q
```

we can check the loading result with this command:

```
q)key `
```

```
`q`Q`h`j`o`taq
```

2. Run this cmd:

```
q).taq.cal_vwap_file["/Users/xiaobohe/q/BDiF/data.csv";"/Users/xiaobohe/q/BDiF/result.csv"]
```

```
"2016.03.13T16:12:11.219 taq | file loaded:
```

```
/Users/xiaobohe/q/BDiF/data.csv"
```

```
"2016.03.13T16:12:11.219 taq | records: 1468800"
```

Then we will get the output csv file with name "result.csv".