

MTH9893 BDIF Assignment B

Date: Mar.13 2016

Prepared by: Yue Wang

Summary: This document outlines the implementation of Assignment B that designs to test hands-on big data skills in synthetic data generation and in-memory database with KDB+/Q
Methodology: Q is used in this assignment from implementation of synthetic data generation to VWAP calculation. The main q library being used are: .qcs.rng, .qcs.mc, .qcs.sample.

Assumption:

- a) The time-stamps between each other is assumed to be uniformly distributed between 0 to 3 seconds (random incremental throughout the day).
- b) Stock price is assumed to follow a geometric Brownian motion with some input parameters in terms of spot price, drift(mean), volatility and time steps. In turn spot price is randomly distributed by centering around 80, and volatility is also assumed to be randomly between 0 and 0.05
- c) The trading volume of each stock is assumed to randomly distributed between 0 to 10000
- d) There are 25 stocks and 1000 trades per day – as being the Q setup

1. To simulate the synthetic data:

- I implemented the Box-Muller method that design to generate normal distribution by using provided uniform distribution generator from q.

There are three main functions associated with synthetic data as defined below:

- I. [simulateStock](#): is the base function that used to generate stock price for each stock given number of time stamps
- II. [sumulateStockByDate](#): elaborate simulateStock function to generate a given day stock price simulation with timestamp added
- III. [createSampleStock](#): create the data by using above two function by defining the number of days to simulate

2. To calculate the vwap:

- I created a function called [getVwap](#) by assuming the given table has column date, symbol, price and volume using a SQL like statement to store the result as vwap;

3. To change the parametres:

- number of trades per stock per day: change the line 81 on num_trade
- number of stocks: change line 90 on sym
- number of days: change the testing code 30 followed by [createSampleStock\[30\]](#)

Result:

To test the code, using vwap.q on the Github, the commented line after 125th rows are the testing code. Firstly using [.qcs.sample.createSampleStock\[30\]](#); the stock price simulation which will be saved under [.qcs.sample.container](#). To test the vwap function, using [vwap_table:.qcs.sample.getVwap\[\]](#); it will save the vwap price in vwap_table. After that, the last two line on github will output the results to a csv files [stockTimeSeries.csv](#) and [stockVWAP.csv](#)