**QF633 – Course Project**

**Team Member:** Goh Boon Heng (01430468)

**Introduction**

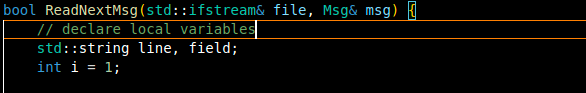
We would like to consume tick data from a cryptocurrency option exchange (Deribit) to

build our choice of volatility market representation updated at a given time frequency.

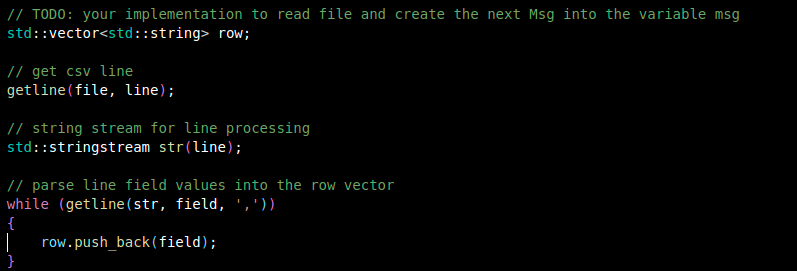
**Step 1: Read Input Events From CSV**

Implementation of the codes below will help read the csv file line by line.

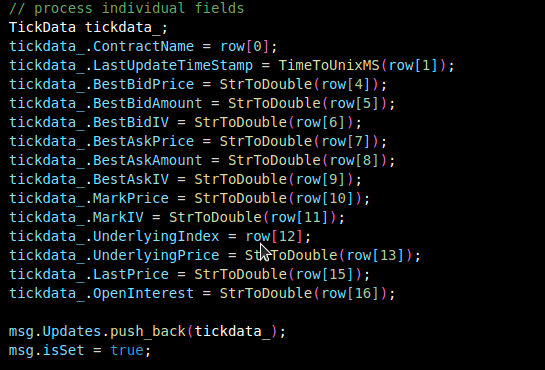
First the local variables to be used are declared: (**Paste implemented code below**).

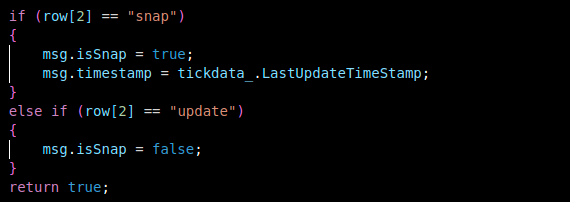
****

The above variables will be used for the functions below:

****

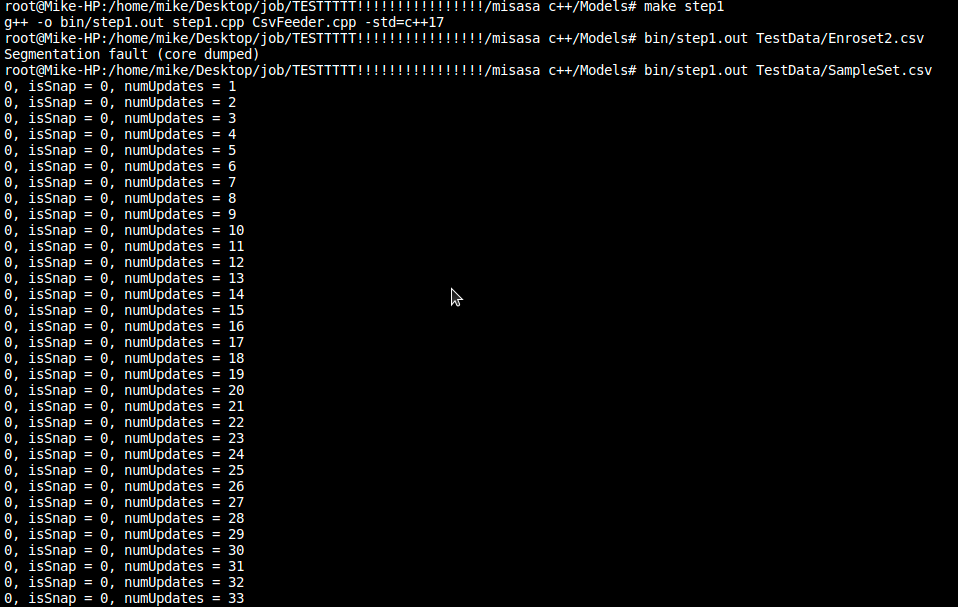
In the above code, a line is read from the file and each comma separated field is inserted into the vector “*row*”. After that, the code below converts all data fields according to their correct respective data types, afterwhich they are stored in the msg.Updates vector.

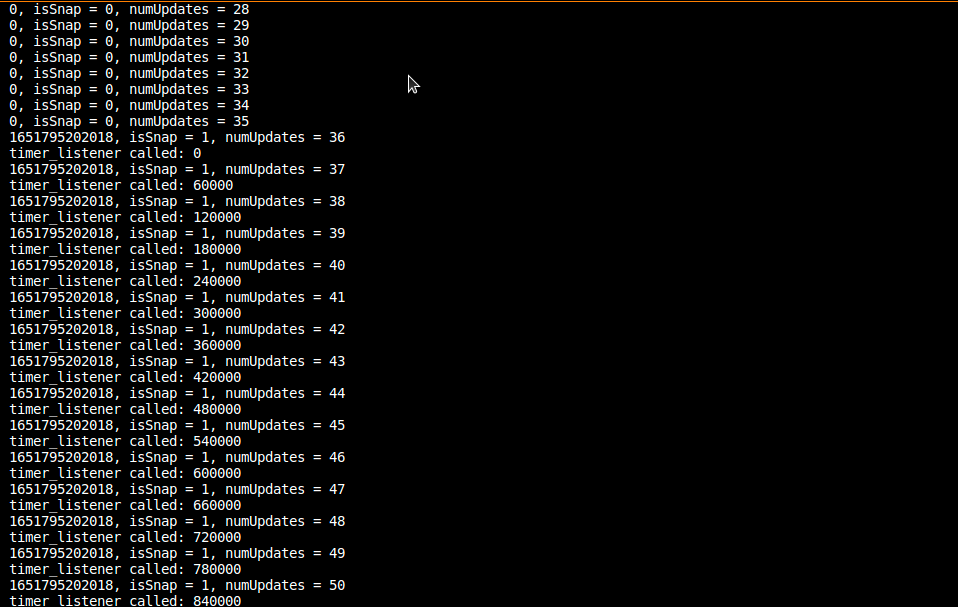
****

****

Finally, as shown in the above code, the snap property is set according to the input.

After running the code, this is output (**Paste output picture below**). For this test run, a sample set of only 1000 records was used.



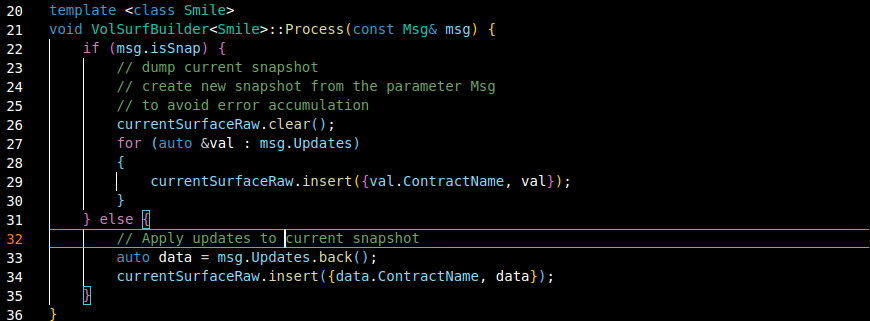


**Step 2: Maintain the Latest Market Snapshot**

The code below will keep track of the latest snapshot of the option market using member *currentSurfaceRaw.*

First off, this method checks whether the message is a snapshot or an update. If it is a snapshot, it discards the previously maintained market snapshot. If not, it will just update the details (ie, *ContractName*)

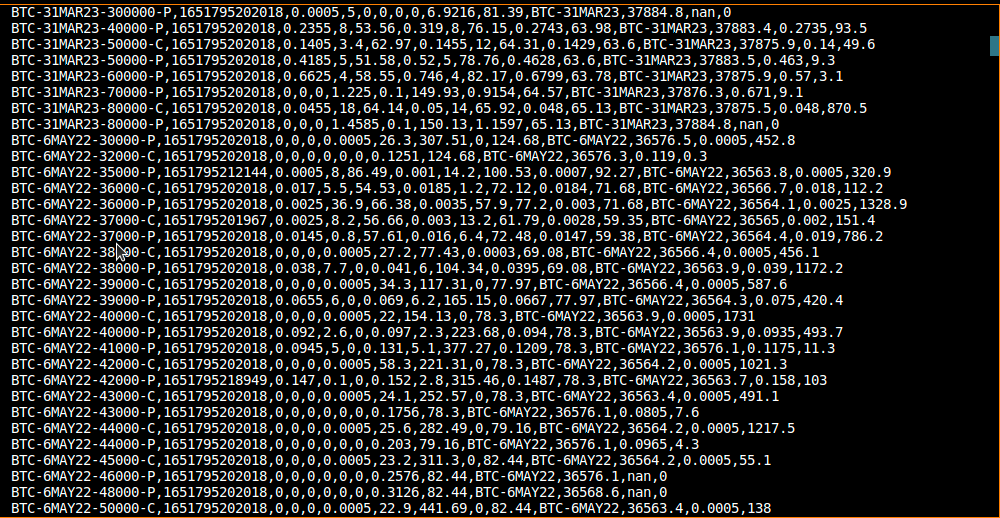
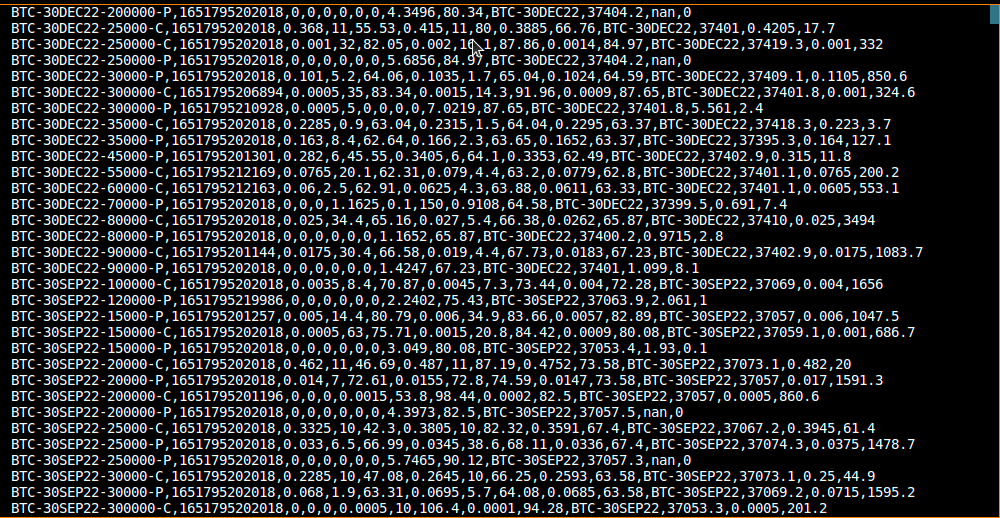
The implementation of the code is as shown below.



The code below will print out the information of the market snapshot at the given interval in the *step2.cpp*.

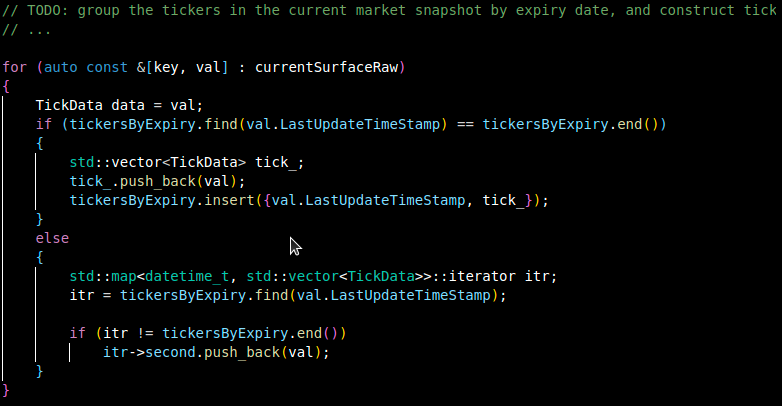


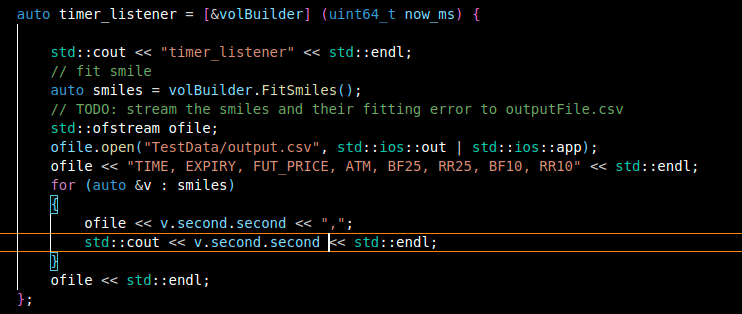
After running the implemented code, the results are as shown below. As seen in the results, some values are of NaN data type.



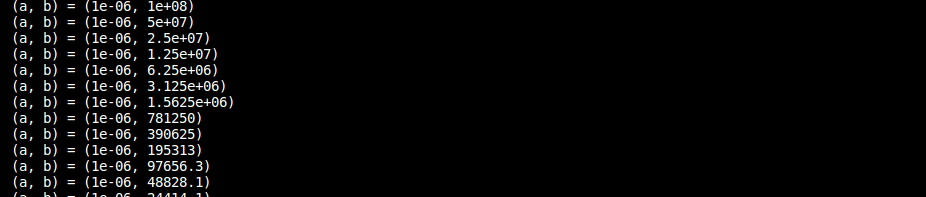
**Step 3: Fit Smile at Regular Time Interval, and Stream the Marks and Fitting Error to Output Files**

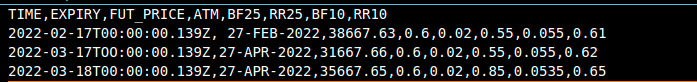
To fit the smile, below are the implemented code .





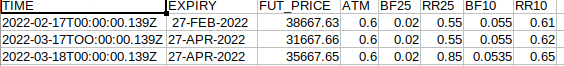
After running the code, below is the output code.



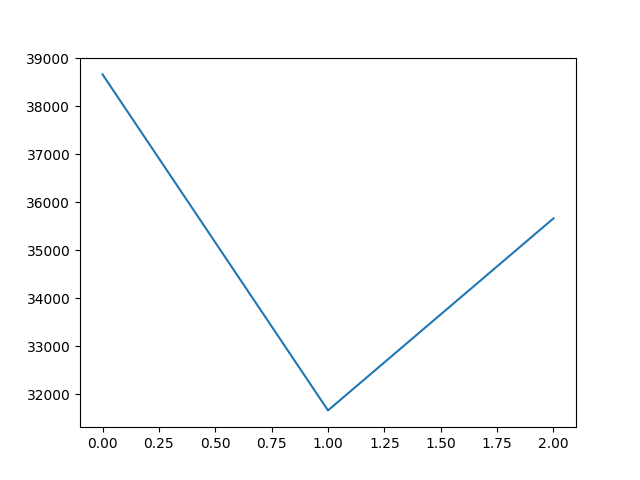
Sample output.csv

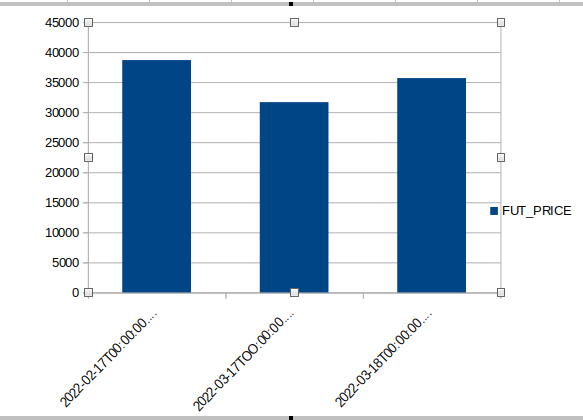
**Tables and Graphs**

The table below depicts the output of *step3.cpp* in table form. It is a representation of the volatility marks and fitting errors .



Below are two graphs depicting the FUT price over time.





As can be seen, the above graphs depict a smile with accordance to the cryptocurrency regardless of expiry date. The U shape means that the options in the money and the options out of the money have roughly the same implied volatility.

The graphs below show show how the volatilty marks and fitting errors compare to each other.

