CISC 5352 Implied Volatility Analytics (I): Financial Data Analytics Quiz (5)[1]

[1] This is a group project. Distinguish yourselves!

Implied Volatility Analytics (I) (100 points)

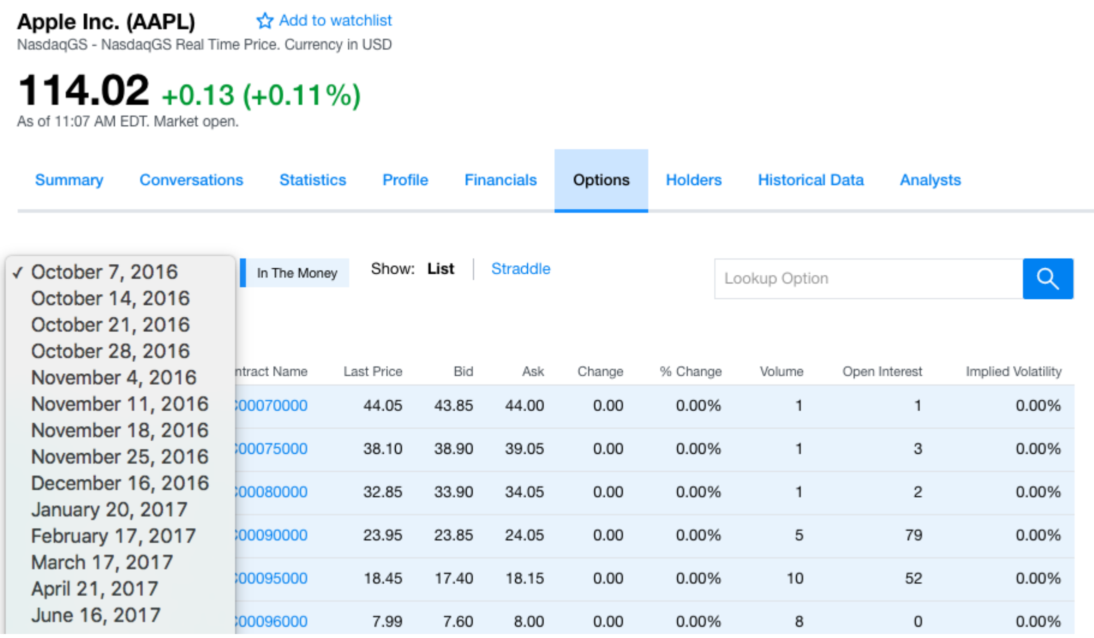
1. Retrieve option data from Yahoo Finance

- Yahoo finance has changed their option data format in their database. It means all pandas option retrieval methods can not work (e.g. get_all_option(). Similarly, all R packages/functions to retrieve option data from Yahoo Finance can't work (e.g. fromJSON in jsonlite package). However, a business analyst should get used to such an issue for open source programming languages like Python /R
- Go to `https://finance.yahoo.com/options/`to Download all call/put options for the following companies. Choose the expiration time by using Dec 2016, Jan 2017 and Feb 2017 options[2].
- These data should be stored in .xlsx or csv format.
- Visualize each option data set by generating their implied volatility surface.

1. `GOOGLE(GOOG),`
2. `YAHOO (YHOO),`
3. `APPLE (AAPL),`
4. `Microsoft (MSFT),`
5. `Amazon (AMZN),`
6. `JPMorgan Chase & Co.  (JPM),`
7. `Bank of America (BAC)`
8. `HSBC USA Inc (New)`
9. `CIT Group Inc (CIT)`

[2] Extra credits given to those who can write their own web parsing program to retrieve data

    10. Goldman Sachs Group Inc (GS)



ntract Name	Last Price	Bid	Ask	Change	% Change	Volume	Open Interest	Implied Volatility
00070000	44.05	43.85	44.00	0.00	0.00%	1	1	0.00%
00075000	38.10	38.90	39.05	0.00	0.00%	1	3	0.00%
00080000	32.85	33.90	34.05	0.00	0.00%	1	2	0.00%
00090000	23.95	23.85	24.05	0.00	0.00%	5	79	0.00%
00095000	18.45	17.40	18.15	0.00	0.00%	10	52	0.00%
00096000	7.99	7.60	8.00	0.00	0.00%	8	0	0.00%

AAPL option data on Oct 7, 2016

2. Finish the following analytics work

1. Apply the following methods to estimate the implied volatility for the put and call option data
   (a) the classic Bisection method
   (b) Muller-Bisection method
   (c) Newton method
   (d) Halley's irrational formula (Note: you can only pick plus sign in your implementation)

$$x_{n+1} = x_n + \frac{-f'(x_n) \pm \sqrt{[f'(x_n)]^2 - 2f(x_n)f''(x_n)}}{f''(x_n)}$$

   You need to use

$$vomma = \frac{\partial^2 f}{\partial \sigma^2} = vega \times \frac{d_1 d_2}{\sigma}$$

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Note: σ is actually unknown, which is just the x_n in your iteration scheme.

2. Draw your conclusion about accuracy and time for the four methods in implied volatility prediction.

What should you turn in?

- 1. A folder that contains
  - A ppt to show details of your analytics (at least 30 pages)
  - your data
  - source files
  - corresponding related output.
- 2. Please name your folder last_name1_last-name2_CISC5352_Quiz_5. For example, Brown_Smith_CISC5352_quiz_5 if your group members with last names: Brown and Smith.
- 3. Send the zipped file (.zip instead of ,rar) of your folder to Blackboard before 11:59 pm Oct 14, 2016

Appendix: Muller-Bisection Method

Muller-Bisection method is a combination of Bisection method and Muller's method. It can increase the convergence speed as well as keep the robust characteristic.

The algorithm of Muller-Bisection method is described below.

1. Set two initial values a and b, satisfying f(a) and f(b) have opposite signs.

2. Find the midpoint between a and b, which is c=(a+b)/2.

3. Calculate f(c). If f(a)*f(c)<0, then set [a,c] as next subinterval$[a_2, b_2]$; if f(b)*f(c)<0, then set [c,b] as the next subinterval $[a_2, b_2]$.

4. Calculate the next approximation c_2 based on (a,f(a)), (b,f(b)) and (c,f(c)) using Muller's method. The basic idea of Muller's method is constructing a parabola passing through three initial points $(x_0, f(x_0))$, $(x_1, f(x_1))$, $(x_2, f(x_2))$, and the next approximation is one root of the quadratic function. If the quadratic polynomial is $P(x) = A(x - x_2)^2 + B(x - x_2) + C$, then there is

$$\begin{cases} f(x_0) = A(x_0 - x_2)^2 + B(x_0 - x_2) + C \\ f(x_1) = A(x_1 - x_2)^2 + B(x_1 - x_2) + C \\ f(x_2) = C \end{cases}$$

   , and we can calculate the next approximation

$$x_3 = x_2 - \frac{2C}{B + sign(B)\sqrt{B^2 - 4AC}}$$

   .

5. Check whether c_2 locates within the subinterval. If c_2 is in the subinterval, then c_2 is our next approximation; if c_2 is outside the subinterval, then we set the midpoint as our next approximation, which means the value of c_2 has to be changed into $\frac{(a_2 + b_2)}{2}$.

6. Repeat step 3-5, until $f(c_n)$=0 or $f(c_n)$<tolerant error.