Executive summary

Background

"Derivative" has been a really hard subject to learn for many students. While being a TA for the BusM411 Advanced Investment class, I found it sometimes hard as well to explain some derivatives concepts to students without visualizing them. I decided to do some researches on a couple of websites that helps generate a trading strategies involving options. The results on these websites were helpful but not really customized enough to be used by our students. These reasons really encouraged me to build a VBA model that would assist the learning process in options trading. Some similar models have been developed by many traders/programmers in many firms and are being sold for hundreds or even thousands of dollars. Even though this project would not be as complicated as real software's that are being used in the industry, it is intended to be expandable for further use by separating the codes into many standalone usable functions/units.

Overview

I built a VBA program that accepts users' input to generate graphs that represent the relationship between the stock's spot prices with their profit/loss position at option expiration, and relationship with different Greek values of Options trading. The project accepts users' inputs of Stock Ticker to grab stocks' information. A different method to get stock's information by using Yahoo Finance API has been implemented to improve the processing speed without going through the process of recording a macro. The Workbook then will be customized to offer various positions in popular trading strategies (Long Put/call, Bull Spread, or Butterfly Spread, etc.) Users can also input their own trading strategies into the model. The Option Trading Workbook uses Black Scholes Model to valuate stock options and generate the underlying data, which are different data vectors/arrays. These vectors will be used as inputs for the model's data series.

Conclusion

Working on the project has been really helpful for me in terms of:

- Understand more about options trading and the Black Scholes Model
- Learn more about working with VBA's dynamic arrays, data series for charts, user forms, and reading a downloaded file
- Learn more about the importance of designing a complete project
- Organizational skills to keep track and finish a whole project

Implementation

Underlying Functions

The focusing of this project is to translate quantitative formulas into a visualized and easy to understand graphs. In order to create vectors of data for my projects, the following formulas have been used:

1) Black-Scholes Option Pricing Function

Many different versions of the Black Scholes model have been invented, and for the purpose of this Project, the simplest form of Black Scholes model was used to price Call and Put Options. However, it is not complicated to bring new variables into the code as the functions are separated and easy to be improved.

$$c = S_o e^{-qT} N(d_1^*) - K e^{-rT} N(d_2^*)$$

$$d_1^* = \frac{\ln(S_o/K) + (r - q + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad d_2^* = d_1^* - \sigma\sqrt{T}$$

Figure 1: Calculate Call Options using Black Scholes Model

Underlying assumptions for the formulas used in the model

- Volatility remains unchanged throughout the life of options
- Continuously paying out dividend
- There is no arbitrage opportunity (i.e., there is no way to make a riskless profit).
- It is possible to borrow and lend cash at a known constant risk-free interest rate.
- It is possible to buy and sell any amount, even fractional, of stock (this includes short selling).
- The above transactions do not incur any fees or costs (i.e., frictionless market).
- The stock price follows a geometric Brownian motion with constant drift and volatility

2) Greek Letter formulas

delta	$e^{-q\tau}\Phi(d_1)$	$-e^{-q\tau}\Phi(-d_1)$
vega	$Se^{-q\tau}\phi(d_1)\sqrt{\tau} = Ke^{-r\tau}\phi(d_2)\sqrt{\tau}$	
theta	$-e^{-q\tau}\frac{S\phi(d_1)\sigma}{2\sqrt{\tau}} - rKe^{-r\tau}\Phi(d_2) + qSe^{-q\tau}\Phi(d_1)$	$-e^{-q\tau}\frac{S\phi(d_1)\sigma}{2\sqrt{\tau}} + rKe^{-r\tau}\Phi(-d_2) - qSe^{-q\tau}\Phi(-d_1)$
rho	$K\tau e^{-r\tau}\Phi(d_2)$	$-K\tau e^{-r\tau}\Phi(-d_2)$

Figure 2: Greek Letters Formulas - Source: Wikipedia

Users Input

Users have options to input their own information into the yellow box below, or using real data from Yahoo! Finance by clicking New Stock Button. For the model to work appropriately, the following information is required:

- The underlying stock price
- The Stock's Volatility
- Risk Free rate
- Deal Date
- Deal Expiration Days
- Dividend Yield

Deal Details		
Comp	Google Inc.	New Stock
Stock Price	790.39	
Volatility	30.00%	
Risk Free rate	7.00%	
Deal Date	4/11/2013	
Deal Expiration	6/19/2013	
Dividend Yield	0.00%	*Graph Increment*

Figure 3: Input Information by Clicking the Stock Button or Hard-Code into the yellow box

Alternatively, users can click the New Stock Button for inputting data. A ticker is required for the Ticker textbox. I used a Yahoo Finance API to do download a customized ".CSV" file from its server. This method has proved to be more efficient and a lot faster than the traditional method of using web-queries and recording a macro.

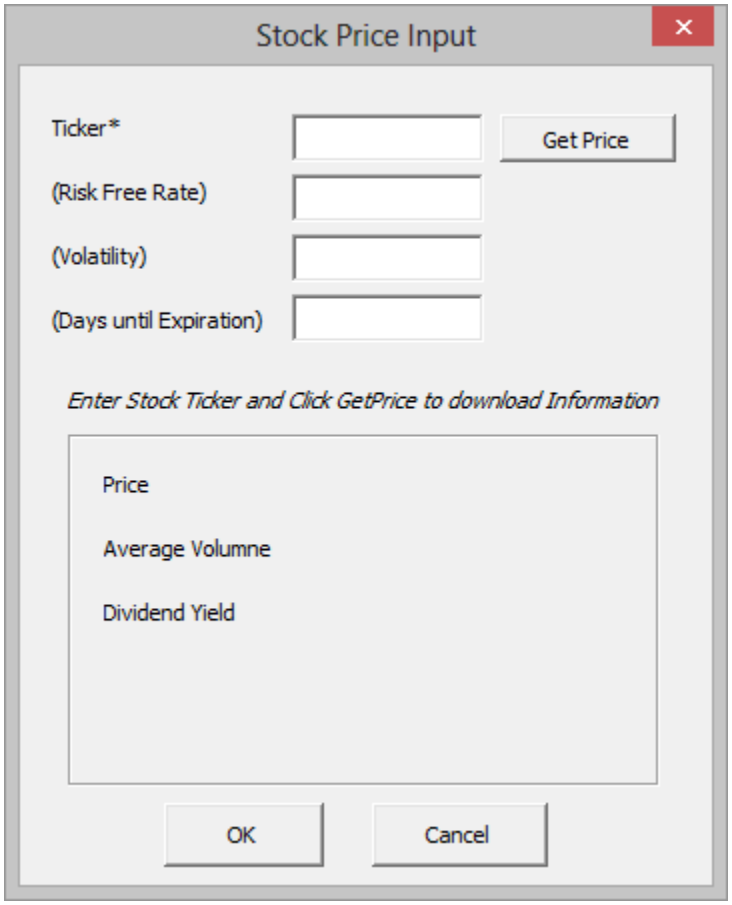

Figure 4: Userform for getting Online Stock Data

Graph Zoom Function

Different Stocks have different underlying prices; thus I decided to implement a Graph Increment function to help Zoom in and out the graph to make sure that the users have good views of graphs.

Figure 5: Graph Zooming Function

Strategies Input

Users can either choose from the list of Strategies Combo box to select a strategy. Alternatively, users can input their own information of contracts, type, and strike into the yellow area below. The picture below shows a combination of buying a Call Option at strike price of $12.6 and selling a Call Option at a strike price of $14.6

Strategies	5. Bull Call Spread		
	Strat1	Strat2	Strat3
Contracts	1	-1	
Type	Call	Call	
Strike	12.6	14.6	

Figure 6: Option Trading Strategies

Output

The underlying functions will produce couple of vectors of data to be inputted into the Series Collections. The processes to calculate the values of X-axes and Y-axes are summarized below:

Stock Information → Black Scholes Model Function to calculate Value of Options → Generate a Vector of underlying Prices around the Strike prices of Options → Calculate Profits at Expiration = f(Underlying Prices)

The Graph on the right represents the correlations between stock price and five different Greek Letters: Delta, Gamma, Theta, Vega, and Rho. The processes to calculate the values of X-axes and Y-axes are the same with the processes to calculate those of the Profit/Loss graph (Instead of using Black Scholes formula, here I use the formulas for each of five Greek letters).

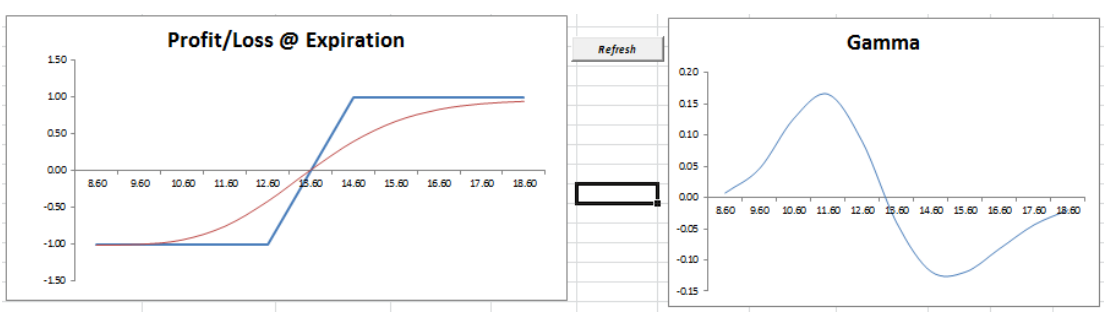

Figure 7: The Result after Clicking the Refresh Button

Difficulties:

1) Quantitative formulas: it is challenging to work on a quantitative model with many variables and vectors of variables. The most challenging part has been generating a payoff formula to represent profit/loss at the day of expiration. There are some examples in the Advanced Investment course textbook but I did not find any general formula to fit all the scenarios of Stock, Call Option, Put Option, In the Money, or Out the Money. The process to think about a general formula actually took good amount of time.

2) Working with Charts: Even though it is not hard to generate a chart using VBA, it is actually hard to come up with an underlying data vectors. Let's think about a user who generates a chart for Google, knowing that its stock price is around 700, and then change her mind to generate a chart for Pandora, (stock price is around 20). The chart have to understand the different scale between these two stocks and give the user an option to zoom in/out to have a right perspective about the data series. The solution to the problem was that I implemented a "graph_increment" variable, controlled by a scrollbar, so that users can adjust the increment of underlying data accordingly. The graphs below represent the before and after adjustment of "graph_Increment" variable. The first graph is not showing a good view for users. By adjusting the graph_increment from 4 to 37, users now can enjoy a better view of the information.

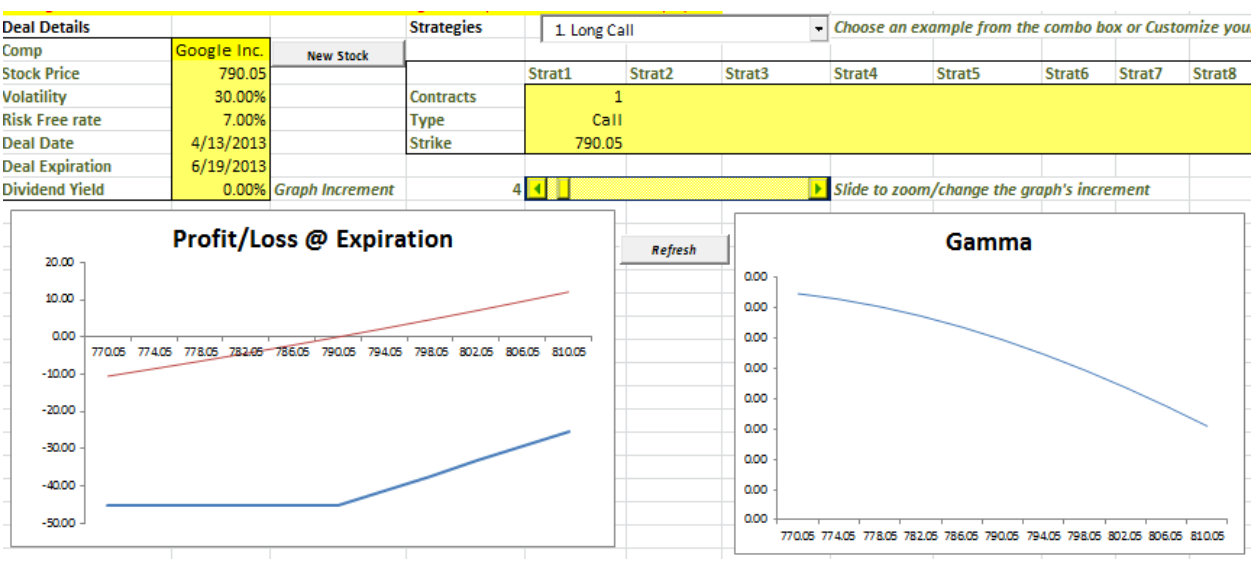

Figure 8: Output before Increment Adjustment

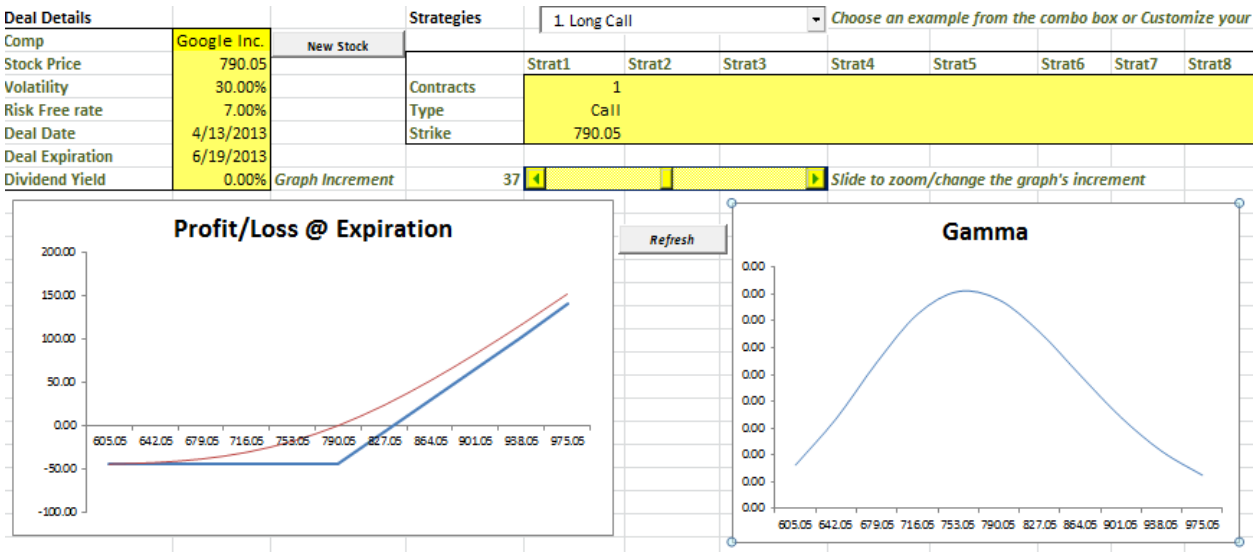

Figure 9: Output after Increment Adjustment

3) Design: This project was intended to be a useful tool that is easy to use. So I have been focusing on designing a workbook that is as simple as possible. This is challenging given the amount of data inputs and the required information to display.

What I learned:

1) Coding: There are many small details about VBA that without actually doing it, I would never get it. I have some experience in coding with Pascal, C++, and HTML, but still found it challenging and interesting with coding the "VBA way."

2) Options Trading: Obviously, I feel more comfortable now than before with option derivatives. I would never understand how Greek letters work without actually involving in this project.

3) Organizational skill: One of the most important skills that I feel improved was my organizational skill. Working with tons of data required me to be organized to be able to keep track of all the codes I have inputted in.

Assistance:

I did many researches on similar option trading software's, including option-price.com and the Option Strategy Evaluation Tool from Peter Hoadley – preview version. Also, I used my forums and blogs to help figure out specific code samples for the project. For example, I used a common technique in the industry to download a CSV file from Yahoo Finance database, and read the downloaded database. I highly recommend this way of downloading Yahoo data instead of recording a macro and get a lot of unnecessary data. I did not, however, receive any help from any individuals and no one helped me on any of the direct code used in my own Functions, Procedures and UserForms