

To: Gove Allen
From: Brooks Benson
Date: April 15, 2014
Re: MBA 614- Final Project Write-up

The “Option Value Calculator”

1. Executive Summary

Every investor and every MBA student will eventually run into the problem of how to value a stock option. The former Director of the BYU MBA Program, Craig Merrill, says that every MBA student must learn how to value a stock as part of his or her MBA experience. More importantly, however, Professor Merrill says that every MBA student must learn how to value a stock using a binomial pricing model. But, there is more than one way to value an option, including the use of the Black-Scholes formula, which complicates the task.

I have taken several classes that required the students to build either a Black-Scholes or a binomial model to price options. These basic models have completed the task at hand, but were clunky and did not allow for variable inputs. For example, often the models would not allow the number of sub-periods before expiration to be altered. This is an important variable in any option calculation. Moreover, these models would only value a call or a put independently, but not both together for comparison. In the end, these models were hard to use and only spewed out limited data.

Most recently, Professor Thorley, of BYU’s Finance Department, asked the students of his Investments class to build a model that would value a European call option. As a student of that class, I decided to take the project much further and build a model using his template and VBA that would value both a put and call option for the same stock. By clicking on a single button, the “Option Value Calculator” will allow you to enter the necessary inputs into a user form, and by clicking “okay”, the model will automatically build out the binomial trees for both the call and put option. The model will also generate the put and call values using the Black-Scholes formula, allowing the user to compare values for each method.

As a result of the model, the complex task of pricing an option can be completed at the click of button.

no more, no less. This is a problem in that it does not allow flexibility in valuation. As more sub-periods are added or subtracted, the value of the option either increase or decreases. Therefore, in order to remedy this issue, I would need to write code that would build out binomial trees for as many periods as the user desires.

The third issue with this model is that it only calculates the value of a call option, but some users may want to know the price of a put option for the relevant stock, as well. As a result, I would also need to write code that would build out a binomial tree for a put option and that would calculate the value of a put option using the Black-Scholes formula.

The last issue with the model is functionality and appearance. The model forces the user to determine where and what inputs are necessary to calculate the price. I remedied this by creating a user-form that aggregates all of the required inputs. The model also was hard to interpret because of its lack of formatting, which I changed by adding color and texture.

b. Building the Option Pricing Calculator

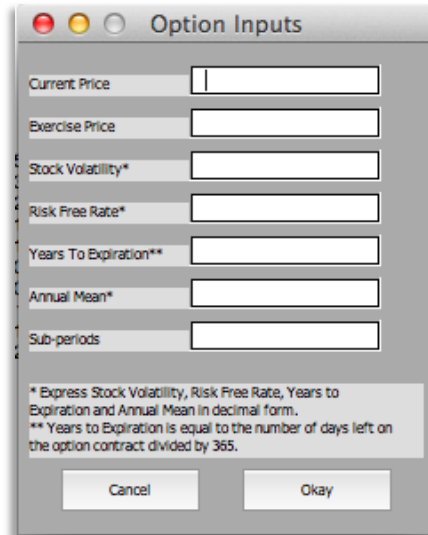
i. The Button

The first improvement to Professor Thorley's completed model was adding a button to run the sub-procedures. The button will allow the user to simply click the button and have the user-form popup and initiate the calculation process. The following is a snapshot of the button:



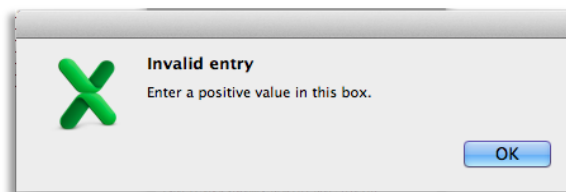
ii. The User-form

Once the "Option Price Calculator" button has been pushed, the following user-form will popup:



The image shows a dialog box titled "Option Inputs" with a standard macOS-style title bar (red, yellow, and green buttons). It contains seven input fields, each with a label and a text box: "Current Price", "Exercise Price", "Stock Volatility*", "Risk Free Rate*", "Years To Expiration**", "Annual Mean*", and "Sub-periods". Below the input fields is a small text box containing two footnotes: "* Express Stock Volatility, Risk Free Rate, Years to Expiration and Annual Mean in decimal form." and "** Years to Expiration is equal to the number of days left on the option contract divided by 365." At the bottom of the dialog are two buttons: "Cancel" and "Okay".

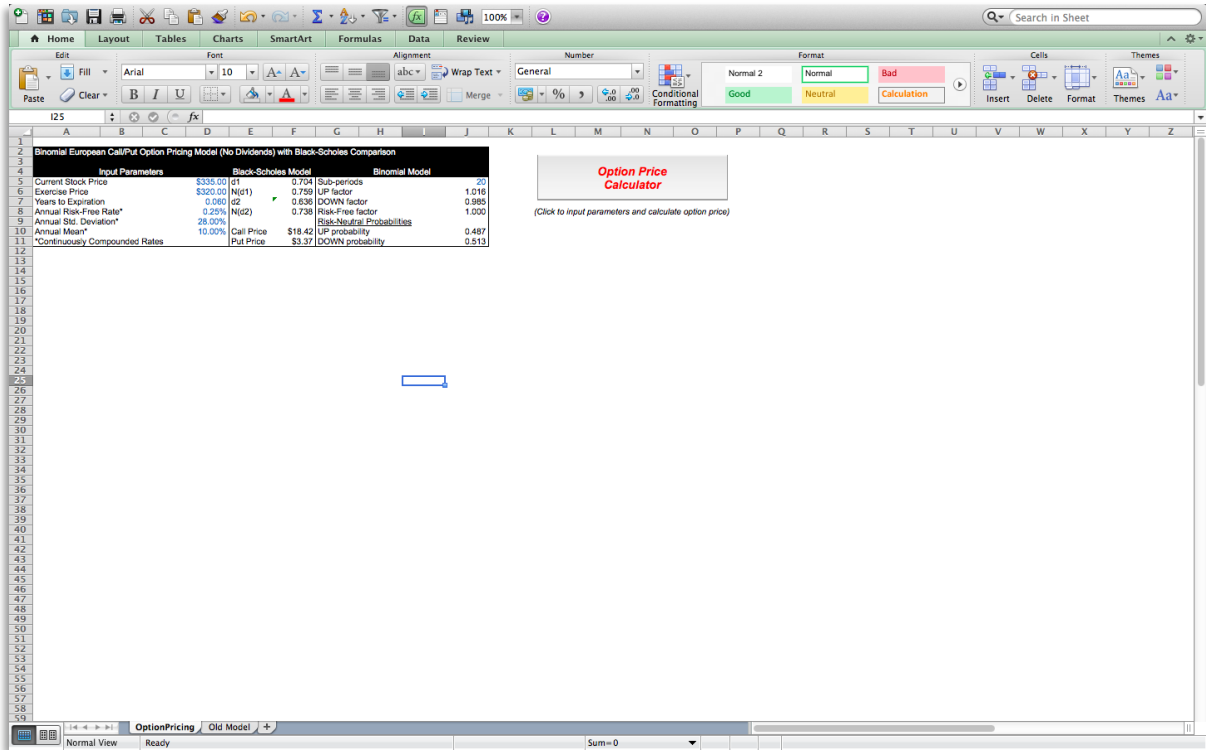
As you can see, the user-form allows the user to input the necessary variables, such as the current price and exercise price, to value the option. The user-form is coded such that a positive value must be entered for each input or a message box will show stating that a positive number is required.



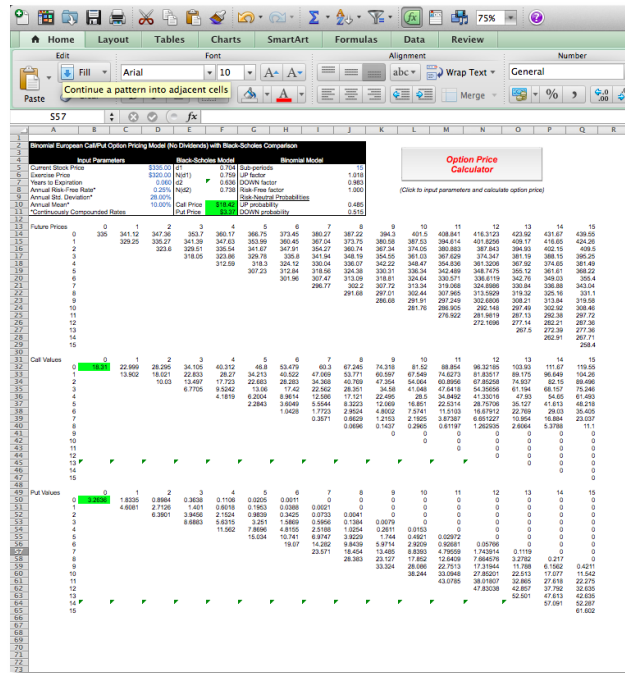
The user-form code is also written such that when the variables are inputted they will be populated into the appropriate cell on the model (i.e. the exercise price entered will be populated into the appropriate exercise price cell).

iii. Building out the binomial trees

Once the inputs have been entered into the user-form, the calculator is ready to run. The format of the model before the calculator runs is the following:



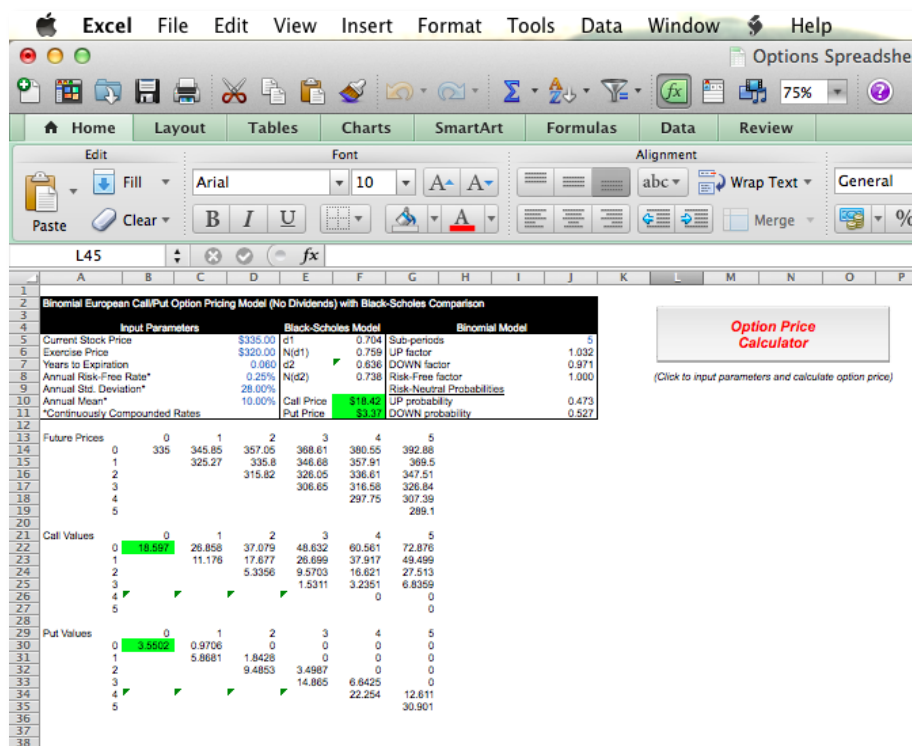
You will notice that a binomial tree does not exist. However, once the okay button is hit on the user-form, the binomial trees will populate, as such:



Building out the binomial trees was by far the most difficult part of the code. Not only did I have to code in the appropriate binomial formulas, I also had to use offset and if statement functions to build out the trees. This proved very difficult and was very time consuming, requiring a lot of trial and error on my part.

Three different trees would need to be populated: 1) the potential future prices of the stock; 2) the potential future value of a call option; and 3) the potential future values of a put option. Each tree required different formulas and different functions.

One of the main functionalities of the code is to allow any number of sub-periods. For example, the model illustrated above had 15 sub-periods, and as a result, the code built out binomial trees for 15 sub-periods. However, if the user desired to see the price of an option with only five sub-periods, the model will build out five period binomial trees:

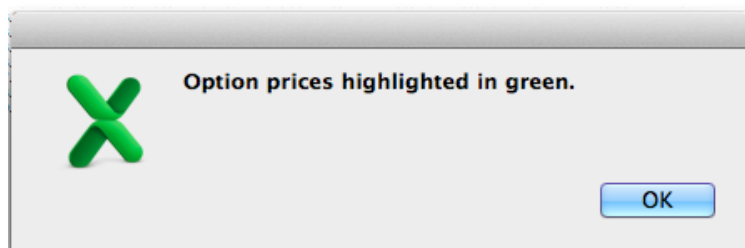


To change the number of periods, or any another variable for that matter, the user merely clicks on the “Option Price Calculator” button, enters the new inputs, and the code will clear the prior binomial trees and populate new trees and values based on the new inputs. Any number of proper variables and sub-periods is possible.

Not only does the calculator compute an option value for a European call, I added code to calculate the value of a European put, as well. As mentioned above, I created a binomial tree for a put option. This required adding code with the proper formulas for the put option binomial model.

iv. The Results

Once the okay button is pushed on the user-form, a message box will show stating that option prices are highlighted in green:



When the user clicks the okay button, he or she will see that there are four cells that are highlighted in green: 1) the value of a call, calculated using the Black-Scholes formula; 2) the value of a put option, calculated using the Black-Scholes formula; 3) the value of a call option using the binomial pricing method; and 4) the value of a put option using the binomial pricing method.

As discussed, the binomial trees for a call and put option are populated and the final value at time zero will be highlighted in green. Also, the user-form will input the required inputs for the Black-Scholes model and the user will see the values of a call and put under this method highlighted in green. The Black-Scholes put option calculation was a new addition to Professor Thorley's model.

In the end, the user will see the following displays in the "OptonPricing" spreadsheet: 1) inputs used in the calculation; 2) the value of a put and call option under both the Black-Scholes and binomial pricing methods; and 3) the binomial trees to project future outcomes.

3. Learning and Conceptual Difficulties Encountered

Where do I start? This was a very fun and difficult project for me. I struggled a lot, as

mentioned above, with the formulas for the binomial trees and with the offset functions and if statements to properly build out the binomial trees. In the end, however, I feel accomplished in that I was able get the code to work properly.

If time had permitted, I would have added more functionality to the model. Instead of manually entering the years to expiration, I could have added a calendar control to the user-form that would automatically calculate this for the user. I could have also distinguished between American and European style option prices by accounting for the ability to exercise early with an American style option.

4. Conclusion

In sum, I feel that the "Option Value Calculator" would be valuable tool for any MBA student or professional. I realize that there are many extant option-pricing models. However, I feel that the ability to have to both the Black-Scholes and binomial pricing methods, for both a call and put option, in one model presents a nuance that could be useful. Also, the ability to immediately change the number of sub-periods of an option helps in accuracy of valuation.

Note: Thanks, Gove, for a great class. I really appreciated the mentoring. Best of luck in the future.