Mathematics of Finance, Courant Institute, Fall 2015

http://www.math.nyu.edu/faculty/goodman/teaching/mathFin/index.html

Always check the class message board before doing any work on the assignment.

Assignment 1, due September 16

Corrections: (`0.92` corrected to `0.83` in the `R` output)

1. Consider a forward contract on the S&P500 index. The contract date is 6 months from today. The risk free interest rate is 2.5%/year. The spot price of the index today is 2000. Find F, the forward price.

2. Use the one period two state model no arbitrage argument to find a theoretic price for a European style put. The spot price of the underlying asset today is $S_0 = 10$. The option strike price is $K = 11$. The risk free interest rate is 3%/year during that period. The option expires at $T = .5$ years. The possible asset prices at time T are

$$S(T) = S^u = u \cdot S_0 \ , \quad \text{or}$$
$$S(T) = S^d = d \cdot S_0 = \frac{1}{u} S_0 \ .$$

   What are the risk neutral probabilities q_u and q_d? What is the theoretical call price today?

3. Use T, S_0, r, and the put option from problem 2. Suppose the theoretical price of the put is .6, find u. If you get a nonlinear equation for u that you can't solve analytically, find a good approximate solution by trial and error.

4. Suppose there are three assets, whose prices are $A(t)$, $S_1(t)$, and $S_2(t)$. At time T, there are three possible states. This is a three state model instead of the two state model we have used up to now. The possible asset prices are

state	$A(T)$	$S_1(T)$	$S_2(T)$
1	1.1	.6	.7
2	1.1	1	.9
3	1.1	1.2	1.3

   Is there an arbitrage opportunity? If so, find an arbitrage portfolio. If not, find the risk neutral probabilities, q_1, q_2, and q_3. Use these to find the price of a put option on $S_1(T)$ struck at 1.

5. (*Introduction to* `R` *programming*) Download the file `BinaryModel.R` associated to this assignment. It is part of an `R` script that computed the theoretical price of a put option in the binary two state one period model. Modify the script so that

- it has the correct and complete formulas for the risk neutral probabilities q_u and q_d
- it reads u and d from the command window
- it prints all the parameters, not just the spot price and strike price
- all comments are correct, and the stuff lines you add are commented.
- the output numbers continue to line up

Hand in a printout of your code Here is a suggested sequence of steps:

(a) Create a directory (folder) for `R` scripts for this class and put `BinaryModel.R` there.

(b) Open the `R` command window, use `setwd("dir name")` to point it to the directory containing `BinaryModel.R` and type `source("BinaryModel.R")`. If you should get output like this (if you type in the same numbers):

```
Binary one period put option pricer
Enter the spot price of the stock
3.123456789
Enter the strike price of the put option
4
Enter the risk free interest rate, in percent per year
5
Enter the time to maturity, in years
6
Spot price is    S_0 =   3.12
Strike price is  K   =   4.00
The put price is f   =   0.83
```

(c) Read the comments in `BinaryModel.R` and fix the ones at the top to have your correct information.

(d) Replace the line `u = 1.2` with two lines that prompt the user and read `u_string`. Do the same for the the variable d. Add a comment that says you which lines you added.

(e) Add lines of the form `C_3 = sprintf(...)`, `C_4 = sprintf(...)`, etc., so that all the variables defining the put are printed out. As you do that, make sure the numbers continue to line up and that the put price comes last.

(f) (*the math*) Put in the correct formulas for q_u and q_d.

There will be programming assignments each week, each one using a new feature of `R` This one uses only assignments, arithmetic, and simple string manipulation. Next week we will do conditionals (tt if, then, else) and code blocks. The week after will by loops (`for, while`). After that will be vectors (arrays), reading and writing data to and from files, and so on. The programs will get longer so you get better at programming want to get a feel for more realistic computing software.