# Financial Calculator

Yuwen Jin | Minghao Kang

#### Motivation

Usually when we use financial calculators online, we'll get different results with the same input due to algorithms the author chose, and we don't know which formula they are using. In order to make calculations more clear to ourselves, we decided to create our own financial calculator. Moreover, with our own product we could check our homework answers for courses such as stochastic calculus more confidently, and get better understanding of some financial models.

### **Functionality**

Our program consists of six sub-functions, corresponding to stocks, options and portfolios.

#### Stock:

Simulate stock price by Geometric Brownian Motion with user defined starting price, risk free rate, volatility, time span and steps. Among these parameters, volatility can be either input manually, or calculated from historical data with given ticker.

#### Option:

Find out volatility with Black Scholes Model and Price an option using Binomial Tree model with user defined parameters such as strike and spot prices, output the result together with the stock price tree, option value tree at the same time.

#### Portfolio:

Allow user to build their own portfolio by inputting ticket names and corresponding weights. Then they could get the sharp ratio as well as the value change of this portfolio. Or user can use random generated weight to build their portfolio.

# **Usage (Input & output)**

### Stock

User could choose either to type in the volatility or point out a ticket name so that computer could calculate it with input file. For the second choice, there should be a csv file named with corresponding ticker in the "input" folder.

Aside from volatility, use should also determine initial price as starting point, as well as simulate time span, risk free rate and simulate steps.

Output here will be a number indicating the most likely price this stock would have after specific time period. Here are two samples:

#### With given volatility:

```
"E:\Stevens\2020-SPRING\FE-522\Final Project\Program\cmake-build-debug\Program.exe"

Do you want to analyze stock, option or portfolio? stock

You want to input data yourself or read from file? (A/B) A

What's the volatility: 0.2

What's the initial stock price: 30

How long you want to simulate (in year): 1

What's the risk free rate you want to use: 0.09

How many steps you want use in this simulation: 252

According to Monte Carlo simulation, the stock price is likely to be: 30.32690
```

#### With given ticker:

```
"E:\Stevens\2020-SPRING\FE-522\Final Project\Program\cmake-build-debug\Program.exe"

Do you want to analyze stock, option or portfolio? stock

You want to input data yourself or read from file? (A/B) B

What's the ticket symbol you want to analyze? BOA

What's the initial stock price: 30

How long you want to simulate (in year): 1

What's the risk free rate you want to use: 0.015

How many steps you want use in this simulation: 252

According to Monte Carlo simulation, the stock price is likely to be: 27.80510

Process finished with exit code 0
```

## Option Properties 1985

### Price an option:

Input requires information such as option type (call or put), spot price (of the underlying asset), strike price, interest rate, volatility (of the underlying asset) and time to maturity. And user could choose to see the binomial tree or not.

```
"E:\Stevens\2020-SPRING\FE-522\Final Project\Program\cmake-build-debug\Program.exe"

Do you want to analyze stock, option or portfolio? option

You want to price an option or calculate volatility? (A/B) A

Are you analyzing a call option or a put? put

What's the spot price: 100

What's the strike price: 95

What's the risk free rate you want to use: 0.05

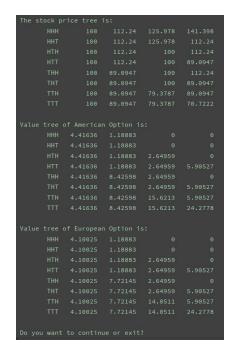
What's the volatility: 0.2

What's the time to mature in year: 1

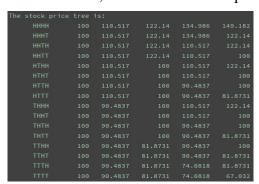
How many terms you want in this binomial tree model: 3

Do you want to see the binomial tree? yes
```

If user choose to see the process, they will get all these three trees:



Four or more teams can also be fulfilled, but will take more space to display



Value	tree	of American	Option	is:	

Value			

For better explanation, we also include the stock directions with all attributes. "H" stands for head while "T" stands for tail.

If user choose not to see the tree, output will be two sentences telling the expected price:

```
Do you want to continue or exit? continue

Do you want to analyze stock, option or portfolio? option

You want to price an option or calculate volatility? (A/B) A

Are you analyzing a call option or a put? put

What's the spot price: 100

What's the strike price: 95

What's the risk free rate you want to use: 0.05

What's the volatility: 0.2

What's the time to mature in year: 1

How many terms you want in this binomial tree model: 3

Do you want to see the binomial tree? no

Price of this American Option should be: 4.41636

Price of this European Option should be: 4.10025

Do you want to continue or exit? exit

Have a good day.

Process finished with exit code 1
```

#### Calculate volatility:

Input should be all other parameters and it price. Then user will get a sentence describing the implied volatility.

```
"E:\Stevens\2020-SPRING\FE-522\Final Project\Program\cmake-build-debug\Program.exe"

Do you want to analyze stock, option or portfolio? option

You want to price an option or calculate volatility? (A/B) B

Are you analyzing a call option or a put? call

What's the spot price: 100

What's the strike price: 105

What's the risk free rate you want to use: 0.09

What's the time to mature in days: 252

What's the option price? 2.56

The implied volatility is: 0.0553132
```

## <u>Portfolio</u>

We apply the template, linked node and sorted list concepts(insert nodes in the order of input), so it allows user to create portfolio containing whatever number of stocks they want.

```
class Portfolio{
public:
    SortedList<Stocks> Tickets;
    SortedList<double> Shares;
    SortedList<double> Weight;
    Portfolio()= default;
```

However, this function requires pre-stored data as csv files in the "input" folder. And all stocks files used in one portfolio should have the same time-span.

#### Set weights

User need to input the stocks tickers they want to take and corresponding weights. Output will be s short description of assets status and portfolio profitability

```
"E:\Stevens\2020-SPRING\FE-522\Final Project\Program\cmake-build-debug\Program.exe"

Do you want to analyze stock, option or portfolio? portfolio

You want to set shares, or use random weight? (A/B) A

How many stocks you want in this portfolio? 3

What's the ticket symbol you want to analyze? AAPL

How many shares you want to buy of this stock? 100

What's the ticket symbol you want to analyze? BOA

How many shares you want to buy of this stock? 120

What's the ticket symbol you want to analyze? FB

How many shares you want to buy of this stock? 200

At the beginning, value of this portfolio is: 62037.3

At the End, value of this portfolio is: 74156.2

Asset profit and loss: 12118.9

With every 100 dollars, you earned: 19.5348

Process finished with exit code 0
```

#### Use random weights

User need to input the tickers as well as the total capital they want to put into the market. Output here will be a short description indicating the optimal weight.

```
"E:\Stevens\2020-SPRING\FE-522\Final Project\Program\cmake-build-debug\Program.exe"

Do you want to analyze stock, option or portfolio? portfolio

You want to set shares, or use random weight? (A/B) B

How many stocks you want in this portfolio? 3

How much starting capital do you prefer? 10000000

What's the ticket symbol you want to analyze? AAPL

What's the ticket symbol you want to analyze? BOA

What's the ticket symbol you want to analyze? FB

The optimal weight with highest sharp ratio 0.0753028 is:
0.0593448 of AAPL, 0.632437 of BOA, 0.308219 of FB,

Buy 292.094 shares of AAPL

Buy 21555.4 shares of BOA

Buy 1587.45 shares of FB

Do you want to continue or exit?
```

2019-07-01 11.8519

Together with a csv file tracking the value change of this portfolio

LULD OF UL	2210025
2019-07-02	30.1194
2019-07-03	89.6056
2019-07-05	101.527
2019-07-08	119.627
2019-07-09	179.729
2019-07-10	191.749
2019-07-11	209.787
2019-07-12	270.568
2019-07-15	282.577
2019-07-16	300.746
2019-07-17	361.28
2019-07-18	373.042
2019-07-19	391.172
2019-07-22	451.509
2019-07-23	463.343
2019-07-24	481.567
2019-07-25	542.967
2019-07-26	554.918
2019-07-29	572.937
2019-07-30	635.422
2019-07-31	647.286
2019-08-01	665.516
2019-08-02	727.539
2019-08-05	739.494
2019-08-06	757.78
2019-08-07	820.925
2019-08-08	832.992
2019-08-09	851.135
2019-08-12	913.984

# Files contained in our program:

