Module Title: Computational methods for Finance

Module Code: 7FNCE041W

Table of Contents

[Answer 1 3](#_Toc156083085)

[Answer 1 a) 3](#_Toc156083086)

[Answer 1 b) 3](#_Toc156083087)

[Answer 1 c) 6](#_Toc156083088)

[Answer 2 10](#_Toc156083089)

[Answer 2a) 10](#_Toc156083090)

[Answer 2b) 10](#_Toc156083091)

[Answer 2c) 12](#_Toc156083092)

[Answer 2 d) 13](#_Toc156083093)

[Answer 2 e) 16](#_Toc156083094)

[Answer 3 17](#_Toc156083095)

[Answer 3a) 17](#_Toc156083096)

[Answer 3 b) 18](#_Toc156083097)

[Answer 3 c) 20](#_Toc156083098)

[Answer 3 d) 23](#_Toc156083099)

[References 24](#_Toc156083100)

# Answer 1

## Answer 1 a)

Python libraries

Library includes set of functions, packages or modules and comprises of prewritten codes which can be reused in other programs by anyone. It includes set of functions and methods which can be imported for performing tasks without writing that part of code again as it is already written in the library. Library is used in Python as it increases the functionality of Python (Lemenkova, 2019).

Libraries can be used easily by following this syntac

import name\_of\_library as any\_name

For example,   
import numpy as nump

Here, numpy is name of library and nump is a short name given to the library so that, if we have to use this library in later portion of the code, we can just write nump to access the library.

It is important to use appropriate library while solving or writing any code as without this, it can throw error like ‘ModuleNotFoundError` or ‘NameError’ (Gad, 2023).

For example,

Result= sum([1,2,3])

This will throw a Name Error as sum is not defined as a function but if we will use

import numpy as np

Result=np.sum([1,2,3])

Result

Then, it will give result as 6

## Answer 1 b)

1. NumPy:

**Definition**: Large complex array objects can store both different kinds of data and similar kinds using an open-source library for the Python programming language. On these arrays, it is used to compute scientific, mathematical, and numerical functions, operations, or procedures. These operations can be high-level functions or simple math operations. It lets you perform code vectorization (Johansson et al., 2019).

**How to make a call:**

The following codes can be used **to install** NumPy:

either pip numpy or conda numpy

The following code can be used to **call NumPy**:

**import** numpy as np

The numpy library can be called by using **"np"** once it has been defined as np. This can be done whenever an action is needed, such as performing a calculation during a program. The NumPy array object is known to as ndarray.

**Functions include:**

* Round operations (fix, floor, ceil);
* Trigonometric operations (sin, cos, tan);
* Exponents and logarithms (exp, log);
* Arithmetic operations: power, subtract, divide, add, subtract, positive, and negative
* These are just some of many functions of numpy.

**Importance or uses:**

* It is used for working with numerical data.
* It gives a fast and effective way for creating array, manipulating it, and running numerical (easy or complicated) functions in it.
* Its parts can be different or uniform.

(b) Pandas

Pandas is a very powerful library and is used commonly in open-source data manipulation in Python. It is very useful as it provides easy-to use data functions and structures which are required to work with structured data easily. The primary data structures in Pandas are Dataframe and Series (McKinney, 2019).

Functions:

read\_csv()- It is used in reading the data from various resources

head()- It is used for displaying first few rows of a Dataframe and provides a quick overview of the data

info()- It is used to provide a concise summary of a DataFrame like data types, memory usage and non-null values

describe()= It is used to generate descriptive statistics which comprises of dispersion, central tendency and distribution shape

Uses:

It is used in Data Preprocessing and Data Cleaning

Exploratory Data Analysis (EDA): Simplfies the process of analyzing and exploring data which comprises of generating creating visualizations, summary statistics and data distribution

Data Transformation and Manipulation

c)

Matplotlib

It is a comprehensive 2D plotting library in Python which is used for producing high-quality figures. This library is used widely for creating animated, static and interactive visualizations in Python. This library provides both objected-oriented interface and procedural interface for creating plots.

Function and Uses:

Plot()- Creating Line Plots

Scatter()- Creating Scatter Plots

Bar()- Creating Bar Charts

Imshow()- Displaying Images

Hist()- Creating Histograms

Applications

There are various applications like

Scientific Plots

Educational Purposes

Web Application Development

Data Visualization

Image Procesing

(d)

Yfinance

It is useful in fetching financial historical data from Yahoo Finance

It is commonly used by traders/investors as it is easy to use library for obtaining stock market data

Used in financial analysis and algorithmic trading

Supports historical data retrieval and real time quotes (Gollapudi and Gollapudi, 2019).

Similarities

All of these 4 libraries have extensive use in data science field and help Python in becoming a very Powerful language especially in data visualization and analysis

Differences

Numpy is used in numerical calculations

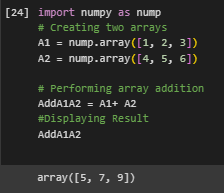
Pandas is used in data analysis and manipulation

Matplotlib is used for plotting

Yfinance is used in getting historical financial data

## Answer 1 c)

1. Numpy: Array Operations



We first imported the library and then defined two arrays which are A1= [1,2,3] and A2= [4,5,6]

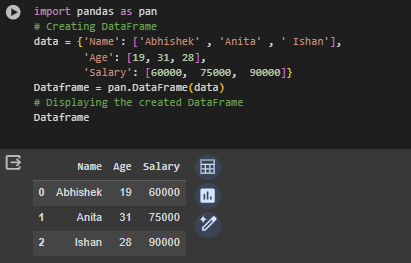
Then we added and displayed the result.

b)

Pandas

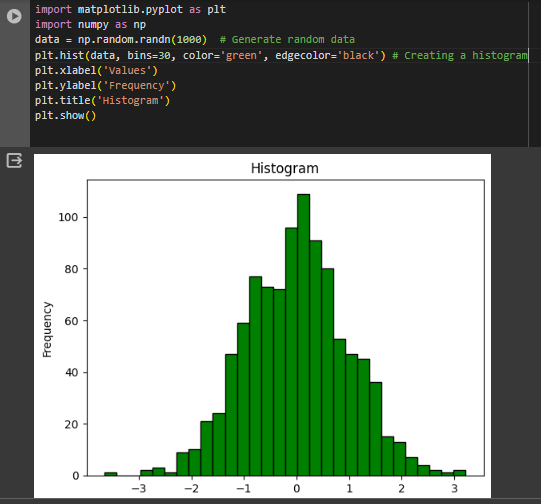
Steps:

Imported Library and then created a database



(c) MatPlotLib

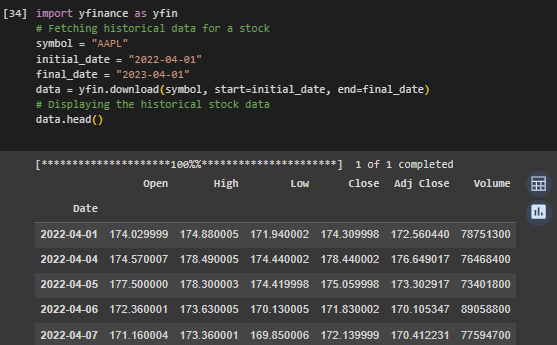
In this code, we imported library then Generate random data, then created a histogram by using .hist() command.

D)

Yfinance

Imported library and then defined the symbol AAPL which means APPLE

Defined initial and final date then displayed the loaded records



# Answer 2

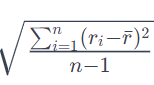
## Answer 2a) Historical Volatility vs. Implied Volatility

Historical Volatility

Historical Volatility is the measure of the price fluctuations of a financial instrument over specific period and it is done based on historic price data

It is measured by calculating the std. dev.of logarithmic returns over specific chosen historical period

Formula is



Where ri is logarithimic return at time I and r bar is average return and n is number of observations

Implied Volatility

It is a market-river estimation/calculation of future volatility and is calculated by financial option prices

It is derived from option prices by using mathematical models (Black-Scholes)

There is not a direct formula to calculate it but traders use option pricing models for solving and getting implied volatility based on the observed option prices

Both of these are measures of volatility and represents the degree of variation in the trading instruments price and are expressed in percentage

There are various differences , in their source, calculations, use and Volatility Smile.

## b)

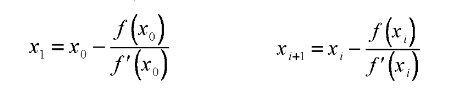
Newton- Raphson Iteration for Implied volatility (σIM)

The term "implied volatility" refers to a stock's implied volatility during the term of an option. When this volatility is factored into the option pricing model, the assumed value of the option will match the option's market price. It is employed to monitor how the market assesses the volatility of a stock. The Newton-Raphson method can be used to determine the implied volatility based on the option's market price. It makes use of the option price derivative with respect to volatility, or Vega.

The method of Newton-Raphson is used for a swift estimation of the functions that are real valued (Kambouroudis, et al., 2021).

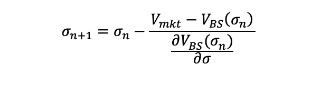
F(x) is equals to 0 is the starting value. F’(x) is known to be the first derivative of f(x).

x1 = value at 1; xi+1 = final value



Formula:

Newton-Raphson Iteration



Where,

Σn  is equals to the starting guess for volatility as n = 0

Vmkt can be called as the market price of the option.

VBS can be termed as the option price reached at the starting guess

Known value is the Black-Scholes formula where BBS( S0, t0; σ, r; K, t)

S0 is the price of the asset

t0 is the starting time

K is the strike price, r is the risk free rate, T is the time to maturity

Overall, everything in this equation is known except sigma σ

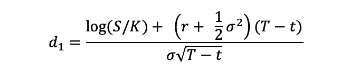
Vega at the starting guess is equals to ∂VBS/∂σ

Then, the updates implued volatility is equals to σn+1

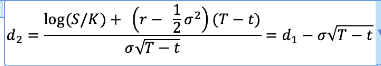
**Formula for non-divided European call option**



Where,



And



S is the present price of the asset

K is the option strike price

r is the rate that is risk free

T is the time until the expiration of the option (time to maturity), t is known to be the present time

σ is the asset’s return annualizaed volatility

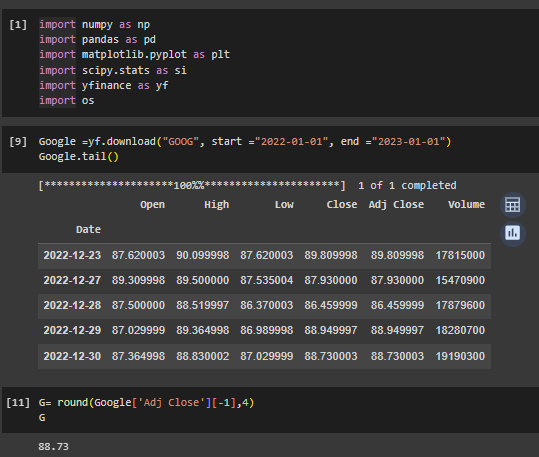
N(x) is the cumulative function of distribution for a standard normal distribution

(Yuan, et al., 2022

## Answer 2c)

Application:

Implied volatility is calculated by using Python and complete process is explained below

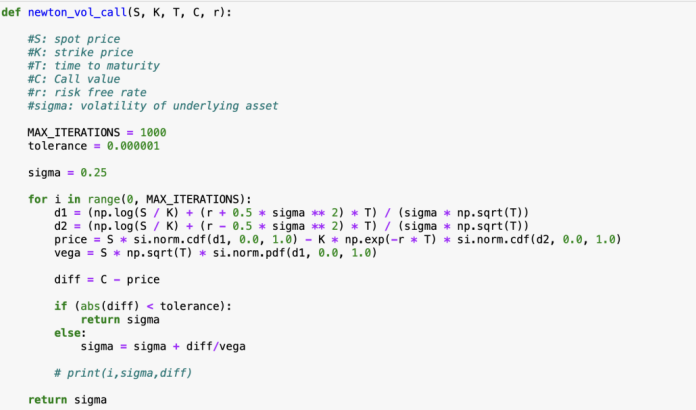


Step 1 (Line 1): Improted all the necessary libraries,   
yfinance for fetching the option prices and stock prices of the stock and in our case, We have selected Google as a chosen stock. Libraries like numpy, pandas, os, scipy.stats are also used as we had to calculate Black Scholes model for calculating Implied Volatility

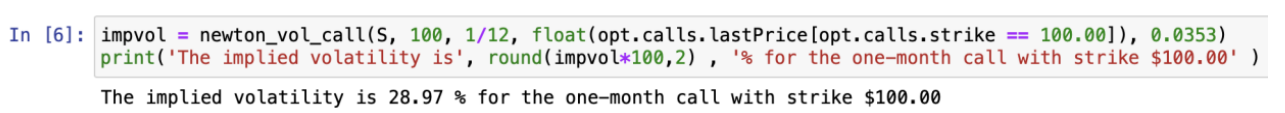
Step (2) (Line 9) We downloaded the Google historical data from Yahoo for finding the spot price and we took the data of period 2022-01-01 to 2023-01-01.

Used google.tail() as tail function helps in fetch the last few rows of the output.

Step 3(Line 11) Last traded price is extracted and Spot Price is $88.73



Step 4: New function is defined for calculating the implied volatility by famous methods Black Scholes. We have defined the value of max\_iterations is 1000 and tolerance is 0.0000001. Given values were, 25% for volatility (sigma) and all the formulas are used as discussed in section 2b



Step 6: We calculated the European Call option volatility. The implied volatility for one-month strike price $100 Is 28.97% with time maturity of 1 month and risk free rate of 3.53%

(Shao, et al., 2021)

## Answer 2 d)

Imported necessary Libraries

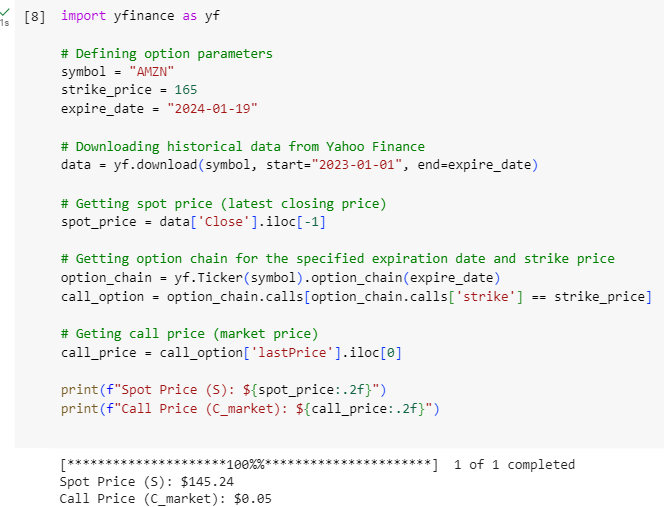
Downloaded data from Yahoo Finance

Fetched spot price

Fetched option chain for specific strike price and expiration date

Fetched call price

Printed obtained call price and spot price



Next step is to calculate Implied Volatility

We can use Newton-Raphson iteration method for estimating implied volatility as we have spot price and call price (Dai, et al., 2020).

Steps:

Imported necessary libraries (numpy and scipy.stats)

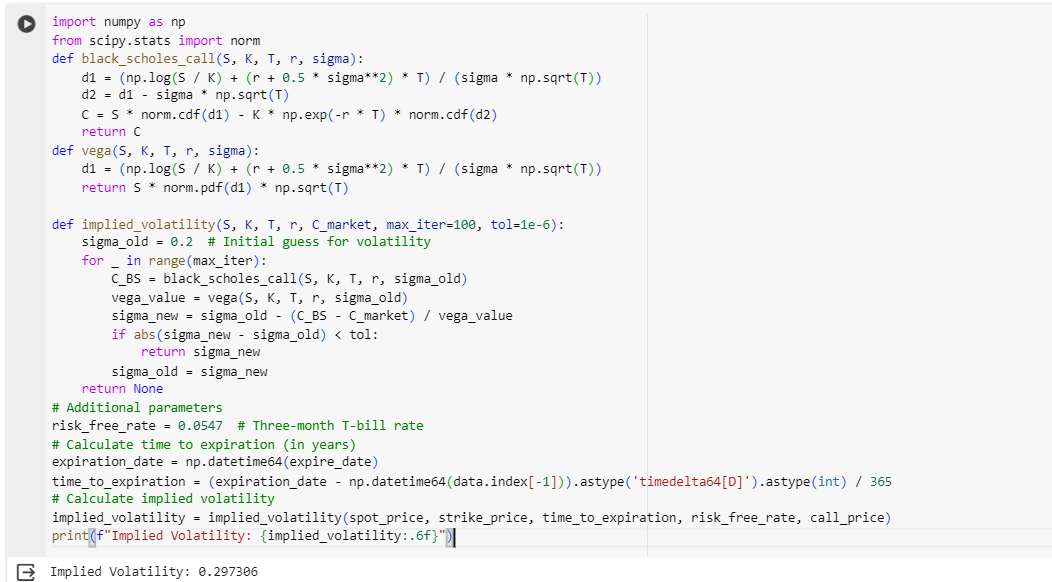
Then created a function black\_scholes\_call with arguments which are given to us like S, K, T, r and sigma

Then defined d1,dw and C

Created another function vega with again those 5 arguments

Created another function implied volatility with again 5 arguments and addition c\_market, max\_iteration=00 and tolerance=0.0000001

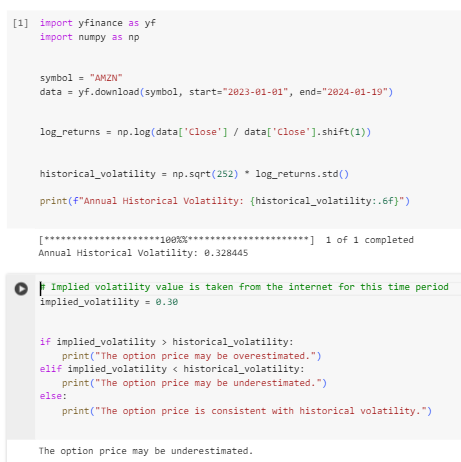
Then risk free rate is defined which was 0.547and Im plied Volatility is calculated



## Answer 2 e)

## 

We imported the necessary libraries, yfinance and numpy and downloaded the data of Amazon of time period 2023-01-01 and 2024-01-19, then calculated the Numerical volatility by formula and then if else statement is used for determining if option prices are underestimated or overestimated (Wang and Zhang, 2020).



# Answer 3

## Answer 3a)

Binomial Tree Option Pricing

Stocks is non-dividend paying and Spot Price (S) is $100

Risk Free Rate (r) is 5% p.a

Time to Maturity (T) is 1 Year with 4 three months period (n)

Volatility (σ) is 20%

(Muroi and Suda, 2022)

1. Formula for u = e^(σ√Dt)

dT= T/n and here T is 1

= ¼

=0.25

So,

u=exp^(0.20)\*√(0.25)

=1.1

d= 1/u

u-1.1

So,

d= 1/1.1

=0.9

p (Risk-Neutral Probability) is

(e^rdT-d)/(u-d)

q=1-p

Calculation:

P=(e^(5\*0.25)-0.9)/(1.1-0.9))

P=0.54

And,

Q=1-p

So q=0.46

Values of ‘u’= 1.1

Values of ‘d’= 0.9

Values of ‘p’= 0.54

(Parmikanti, et al., 2019)

## Answer 3 b)

European Call Option Value

Values of ‘u’= 1.1

Values of ‘d’= 0.9

Values of ‘p’= 0.54

Stock Price as Positions

A1=S0 equals $100

B1= S0\*u equals $110.5

B2= S0\*d equals $090.480

C1= S0\*u2 equals $122.14

C2= S0\*d\*u equals $100

C3= S0\*d2 equals $081.87

Formula for calculating European Call Option is

C =e^(-rT)[p fu +q fd]

Where,

T is known as Time to maturity

p is known as risk neutral probability

r is known as risk-free rate

Max Payoff is [Max(SI-k,0)]

fd is the payoff if stock moves down and it is S0\*d\*u -$95 equals $5 in this instance

fu is the payoff if stock moves up and it is S0\*u2 -$95 equals $27.14 in this instance

c = $016.70

Call option value at node C

fd is the payoff if stock moves down and it is S0-$95 equals $5 in this instance

fu is the payoff if stock moves up and it is S0\*d-$95 equals 0 in this instance

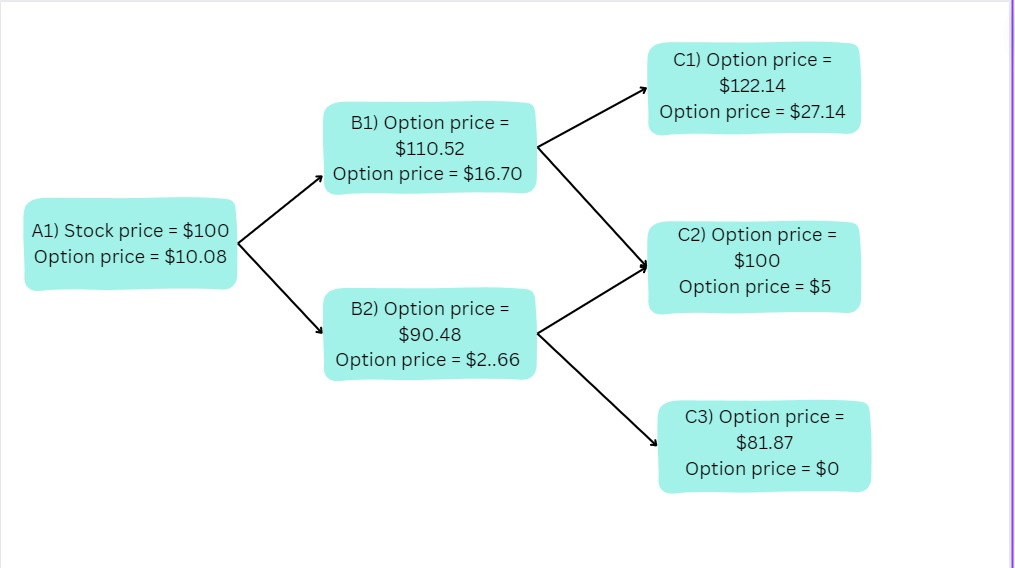
c = $02.66

Value for Up call option $016.70

Value for Down Put Option =$2.66

c = $010.08

So, the European Call Option Value is $010.08



(Tian, 2023)

## Answer 3 c)

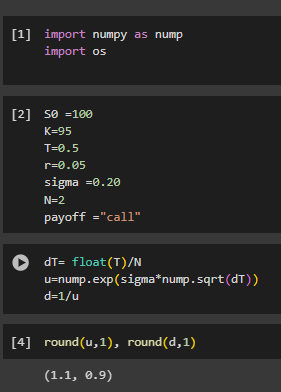
European Call Option Valuation via Python Code

Here, Step1 (Line 1)- I imported the necessary libraries, which were numpy and os as numpy will help in performing Mathematical Calculations like Multiplication, Addition, Square root and os library will help in fetching data respectively

Step 2 (Line 2): I defined all the given variables like Spot Price, Strike Price, risk free rate, volatility, Time to Maturity, Time step and Payoff

Step3 (Line 3): Define the formulas for dt and u. U is the factor by which prices of stock will go up and d is the factor by which prices of the stock will go down

Step4 (Line 4): I used the round function, and it will also display the result and it displayed. According to this, value for u and d is 1.1 and 0.9 respectively



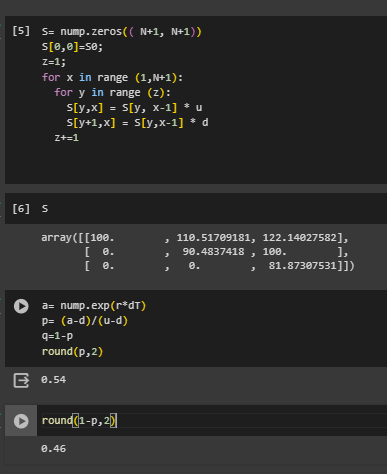
Step5 (Line 5): Then for loop is created for creating the binomial tree by using the u and d values (up and down movement) of stock price which is starting at $100.

Step 6 (Line 6): We, then print the array which includes up and down stock prices

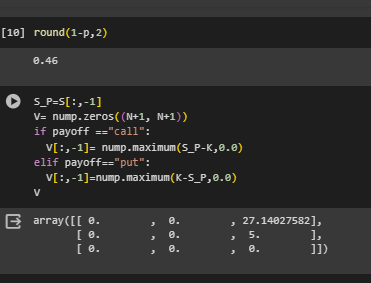
Step 7: Calculated the risk natural probability values for up and down movement of the stock

Up Value is p - 0.54

Down Value is q - 0.46

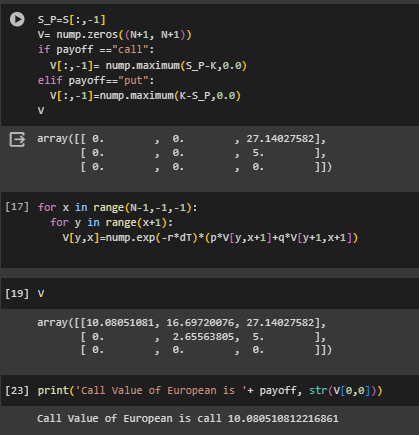


Step 8 (Line 11): If else condition is used for calculating the Option price at maturity



Step 9 (Line 16) Another for loop is created for calculating the option value at all the possible nodes. The option price is the value of beginning node and it is $10.080 for European Call option having spot price of $100 and strike price of $95 and this calculation has been done by keeping in mind the risk-free rate of 5% with the maturity of a year with four periouds of 3-months each

Then in Step10 (Line 23): Printed the European Call Option Value



## Answer 3 d)

Comparison of b) and c)

The European Call option value we got from both the methods is same which is $10.08

While doing manual calculation, u,d,q and p values are computed to find out the up and down stock price movement probability and formula for the option price at each node is calculated (Ghasemifard, et al., 2023).

In python, it was easy to calculate as we used for loop for the calculation process but in manual calculation, we had to find the value one by one for each processs/nodes.

If and else if condition is employed in the code for finding the option pay-offs and for calculating pay-off at all the nodes but is calculated manually for each node on the other hand (Hozman and Tichý, 2022).

# References

Lemenkova, P., 2019. Processing oceanographic data by Python libraries NumPy, SciPy and Pandas. Aquatic Research, 2(2), pp.73-91.

Gad, A.F., 2023. Pygad: An intuitive genetic algorithm python library. Multimedia Tools and Applications, pp.1-14.

Johansson, R., Johansson, R. and John, S., 2019. Numerical python (Vol. 1). New York: Apress.

McKinney, W., 2022. Python for data analysis. " O'Reilly Media, Inc.".

Sial, A.H., Rashdi, S.Y.S. and Khan, A.H., 2021. Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python. International Journal, 10(1).

Gollapudi, S. and Gollapudi, S., 2019. OpenCV with Python. Learn Computer Vision Using OpenCV: With Deep Learning CNNs and RNNs, pp.31-50.

Fassas, A.P. and Siriopoulos, C., 2021. Implied volatility indices–A review. The Quarterly Review of Economics and Finance, 79, pp.303-329.

Kambouroudis, D.S., McMillan, D.G. and Tsakou, K., 2021. Forecasting realized volatility: The role of implied volatility, leverage effect, overnight returns, and volatility of realized volatility. Journal of Futures Markets, 41(10), pp.1618-1639.

Dai, Z., Zhou, H., Wen, F. and He, S., 2020. Efficient predictability of stock return volatility: The role of stock market implied volatility. The North American Journal of Economics and Finance, 52, p.101174.

Wang, D. and Zhang, Z., 2020. High-efficiency nonlinear dynamic analysis for joint interfaces with Newton–Raphson iteration process. Nonlinear Dynamics, 100(1), pp.543-559.

Shao, Y., Nadkarni, S., Niu, K. and Chien, T.W., 2021. Understanding of the Newton-Raphson Iteration Method in Rasch model. J Bibliographical Anal Stat, 18, pp.71-6.

Yuan, R., Lazaric, A. and Gower, R.M., 2022. Sketched Newton--Raphson. SIAM Journal on Optimization, 32(3), pp.1555-1583.

Muroi, Y. and Suda, S., 2022. Binomial tree method for option pricing: Discrete cosine transform approach. Mathematics and Computers in Simulation, 198, pp.312-331.

Parmikanti, K., Irianingsih, I., Joebaedi, K. and Supian, S., 2019, July. Convergence of binomial tree methods to Black Scholes model on determining stock option prices. In IOP Conference Series: Materials Science and Engineering (Vol. 567, No. 1, p. 012013). IOP Publishing.

Tian, Y.S., 2023. The binomial option pricing model: The trouble with dividends. International Journal of Financial Engineering, p.2350039.

Ghasemifard, A., Banihashemi, S. and Babaei, A., 2023. Option valuation in markets with finite liquidity under fractional CEV assets. Journal of Mathematics and Modeling in Finance, 2(2), pp.167-180.

Hozman, J. and Tichý, T., 2022, September. Option pricing under the Bates model using the discontinuous Galerkin method. In AIP Conference Proceedings (Vol. 2505, No. 1). AIP Publishing.