**Project No. 3 (2)**
**"Pricing of a call option"**
**(Cox-Ross-Rubintein Formula)**

---

# Pricing of Derivatives

## Multiple-Time step Binomial Tree

---

December 1, 2022

Submitted by Liteshwar Rao
MA Mathematics, Yeshiva University
Mathematics of Finance, Fall 2022, MAT-5640-1

# Contents

# 1 Abstract

This project is in continuation of **project 3.2** on finding the price of a call option by using binomial model. In this project we derive the price of a call option directly by using **Cox-Ross-Rubintein** formula instead of working backwards from last time period time $= T$ to time $= 0$.

## 2 Introduction

We use two different versions of the **Cox-Ross-Rubintein** formula. In first version we have a constant $X$ for up or down movement in stock prices from each previous time step $t = i$ to each successive time step $t = i + 1$. For instance at time $t = i$ the price of stock is $C$, then at time $t = i + 1$ the price is $C + X$ or $C - X$. However, in second version we don't have a constant movement in stock prices but a fixed rate $U$ for upward movement or $D$ for downward movement by which stock prices either go up or down respectively. For instance at time $t = i$ the price of stock is $C$, then at time $t = i + 1$ the price is $C.(1 + U)$ or $C.(1 - D)$.

- Following is the list of variables we use in this project for version 1 of the formula:
  - $V$ = price of option at time $t = 0$.
  - $S$ = price of stock at time $t = 0$.
  - $K$ = the strike price of option for the underlying security.
  - $X$ = upward or downward movements in stock price.
  - $S_u(T)$ = price of stock when its price goes up by $X$.
  - $S_d(T)$ = price of stock when its price goes down by $X$.
  - $N$ = number of time steps from 0 to $T$.
  - $J$ = number of up moves in the binomial tree.
  - $P = 1/2$ risk neutral probability.
- Following is the list of variables we use in this project for version 2 of the formula:
  - $V$ = price of option at time $t = 0$.
  - $S$ = price of stock at time $t = 0$.
  - $K$ = the strike price of option for the underlying security.
  - $U$ = percentage increase in stock price.
  - $D$ = percentage decrease in stock price.
  - $S_u(T)$ = price of stock when its price goes up by $X$.
  - $S_d(T)$ = price of stock when its price goes down by $X$.
  - $N$ = number of time steps from 0 to $T$.
  - $J$ = number of up moves in the binomial tree.
  - $P$ = risk neutral probability to be calculated as mentioned in methodology section below.
  - $R$ = risk free interest rate.

## 3 Methodology

- Version 1.

  For version 1 of the formula, we use following equation. This formula works as follows. Using the figure 1, under scenario B where stock price is 106, we calculate out of total time steps = N = 5, how many up steps we need to take from 100 to reach 106. It is 4 up and 1 down step. So $\binom{N}{J} = \binom{5}{4}$. Similarly, for scenario C we have $\binom{N}{J} = \binom{5}{3}$. We do this for all scenario. For each scenario we need to calculate the maximum of (S + (2J - N).X - K, 0) because option will be exercised only if stock price is higher than the strike price.

$$\boxed{V = \sum_{J=0}^{N} \binom{N}{J}.P^J.(1-P)^{N-J}.\max(S + (2J-N).X - K, 0)} \tag{1}$$

- Version 2.

  Under version 2 we calculate risk neutral probability and formula to calculate option price are shown below.

$$P = \frac{R+D}{U+D} \tag{2}$$

$$V = \frac{1}{(1+R)^N} \sum_{J=m}^{N} \binom{N}{J}.P^J.(1-P)^{N-J}.[S(1+U)^J.(1-D)^{N-J} - K] \tag{3}$$

where $m$ is the smallest integer for which $S(1+U)^m.(1-D)^{N-m} > K$. We use the same approach as we did in version 1 but with one change. Instead of counting our up steps from stock price at $t = 0$ to one of the scenario at $t = T$, we ignore a particular scenario where option is not exercised. In other words, we don't consider time step movements where stock price at $t = T$ under different scenario is either equal to less than the strike price $K = 100$. Using the figure 2, we say that we don't consider step movement for scenario E,F and G. We intend to simplify equation 3 by considering the following equation.

$$Q = \frac{P.(1+U)}{(1+R)} \tag{4}$$

Then equation 3 can be written as:

$$\begin{aligned} V = \frac{1}{(1+R)^N} \sum_{J=m}^{N} \binom{N}{J}.P^J.(1-P)^{N-J}.S(1+U)^J.(1-D)^{N-J} \\ - \frac{1}{(1+R)^N} \sum_{J=m}^{N} \binom{N}{J}.P^J.(1-P)^{N-J}.K \end{aligned} \tag{5}$$

After using $Q$ in equation 5 and collecting some terms we arrive at the following equation.

$$\boxed{V = S.\sum_{J=m}^{N}.\binom{N}{J}.Q^J.(1-Q)^{N-J} - \frac{K}{(1+R)^N}.\sum_{J=m}^{N}.\binom{N}{J}.P^J.(1-P)^{N-J}} \tag{6}$$

# 4 Simulation

- Version 1

  We simulate option pricing with the following parameters:

- time steps to maturity $N = 5$.
- stock price at time $t = 0$ is $S = 100$.
- strike price is $K = 100$.
- upward and downward movements at each time step is $X = 2$.
- Risk neutral probability is $P = 1/2$.

The binomial tree with these parameters and stock price at each node are shown below in Fig 1. Our objective is to find the value of call option at time $\boldsymbol{t = 0}$ by using equation 1. Using the strategy mentioned in methodology section we calculate option price at time $\boldsymbol{t = 0}$ is $\boldsymbol{V = 1.875}$ as we obtained in project 3(1).

- Version 2

  We simulate option pricing with the following parameters:

  - time steps to maturity $N = 6$.
  - stock price at time $t = 0$ is $S = 100$.
  - strike price is $K = 100$.
  - increase in stock price $U = 5\%$.
  - decrease in stock price $D = 4\%$.
  - risk free interest rate $R = 3\%$.
  - Risk neutral probability after calculation as per equation 2 is $P = 0.78$.
  - Q after calculation as per equation 4 is $Q = 0.79$.
  - Up step movement starts from $m = 3$.

  The binomial tree with these parameters and stock price at each node are shown below in Fig 2. Our objective is to find the value of call option at time $\boldsymbol{t = 0}$ by using equation 6. Using the strategy mentioned in methodology section we calculate option price at time $\boldsymbol{t = 0}$ is $\boldsymbol{V = 16.4}$.

**Fig 1: Stock prices under five time steps binomial model**

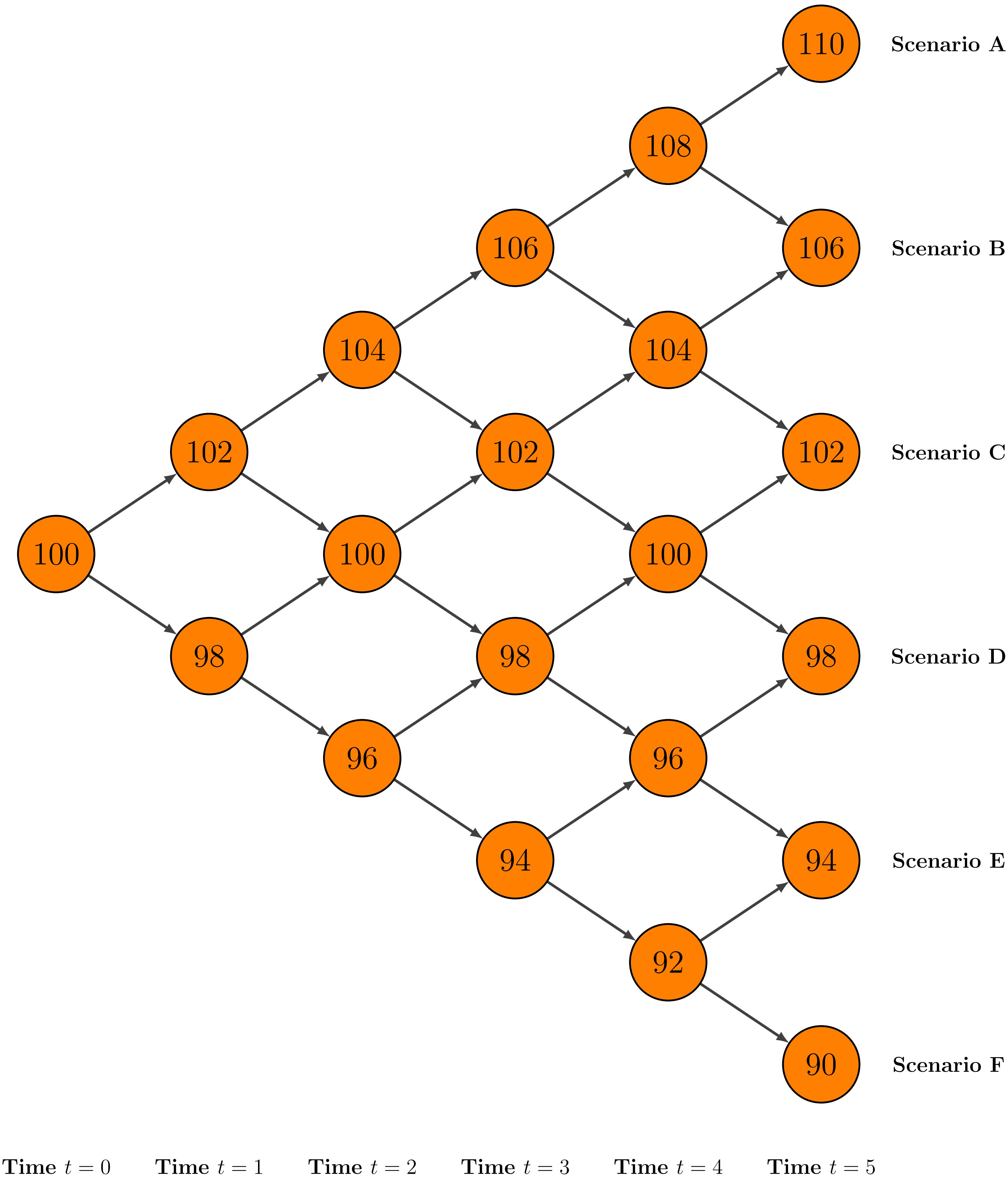
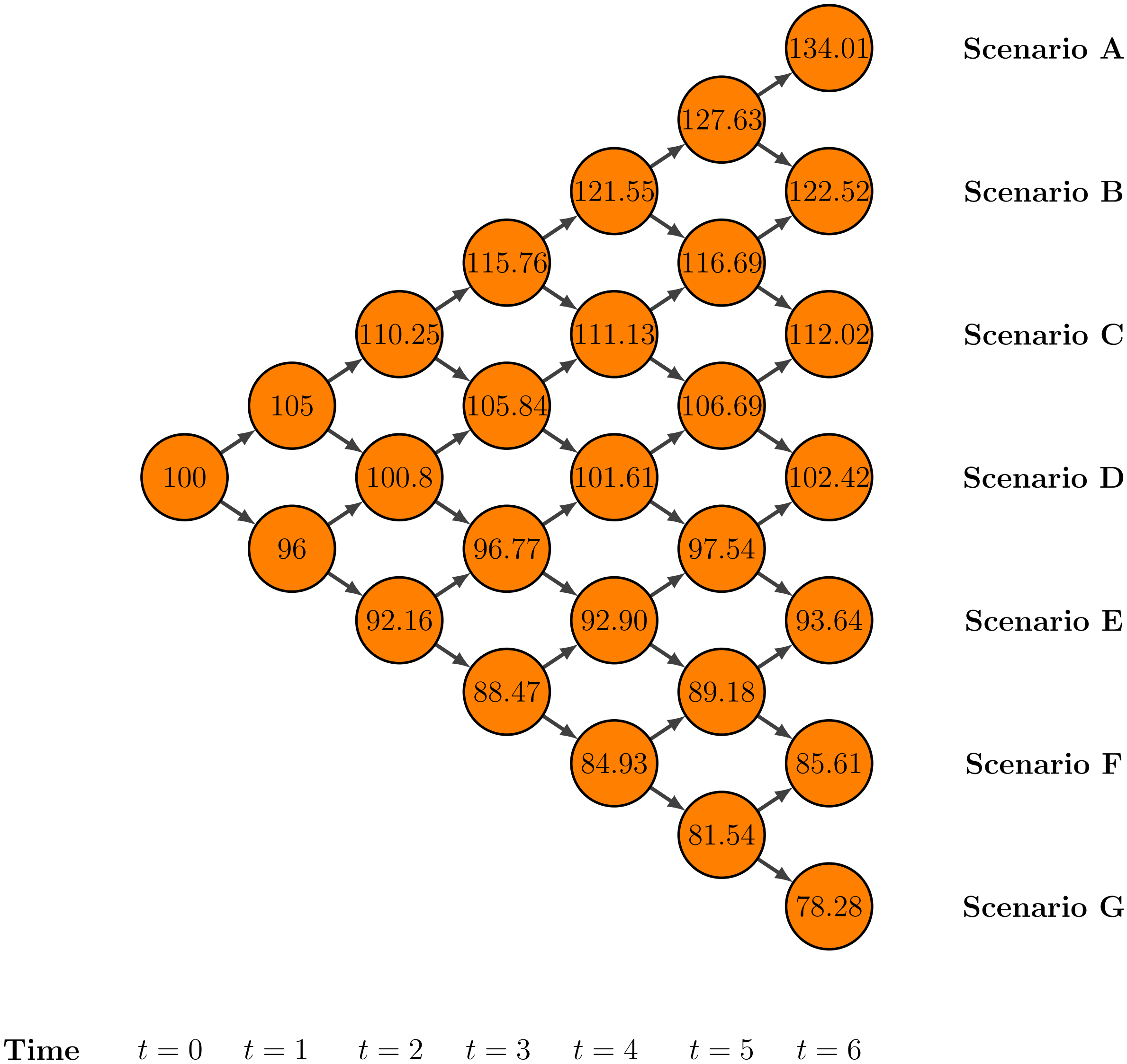


**Fig 2: Stock prices under six time steps binomial model**



# Appendix

In [1]:
```
from scipy.stats import binom
import numpy as np
from simple_colors import *
```

## Exercise 1 (Different method applied on project 3(1))

```
In [2]: # Calculation of option price using 'Cox-Ross-Rubintein formula' for the same exercise in project 3(1)

print(yellow('Call option pricing by Cox-Ross-Rubintein formula', ['bold', 'reverse']))

s0 = 100 #initial stock price at t = 0
f = 2 # add and sub factor at each step
n = 5 # no. of time steps to maturity
x = 100 # strike price
s = np.zeros((n+1,n+1))
s[0,0] = s0
for i in range(1,n+1): # setting first row
    s[0,i] = s[0,0] - f*i  #setting first row
for i in range(n,0,-1): #setting columns now starting from last colmn
    for j in range(1,i+1):
        s[j,i] = s[j-1,i] + 4

print()
print('__________________________________________________')
print(magenta('Binomial tree of Stock price at each node:',['bold', 'bright']))
print('__________________________________________________')
print(s)
print()

s0 = 100
st = np.array([90,94,98,102,106,110])
z = 0
n = 5 # number of time steps to maturity
p = 0.5 # risk neutral probability is always half
for i in range(0,n+1):
    dist = binom.pmf(i,n,p)
    p1 = binom.pmf(i,n,p)* max(s[i,5]-s0,0)
    print(magenta('Calculation for J=',['bold', 'bright']), i)
    print(round(p1,2))
    print()
    z = z + p1

print()
print('__________________________________________________')
print(magenta('Option Price (Total of Calculations for all J)',['bold', 'bright']))
print('Option price=', round(z,2))
print('__________________________________________________')
```

```
Call option pricing by Cox-Ross-Rubintein formula
```

---

**Binomial tree of Stock price at each node:**

---

```
[[100.  98.  96.  94.  92.  90.]
 [  0. 102. 100.  98.  96.  94.]
 [  0.   0. 104. 102. 100.  98.]
 [  0.   0.   0. 106. 104. 102.]
 [  0.   0.   0.   0. 108. 106.]
 [  0.   0.   0.   0.   0. 110.]]
```

**Calculation for J=** 0
0.0

**Calculation for J=** 1
0.0

**Calculation for J=** 2
0.0

**Calculation for J=** 3
0.62

**Calculation for J=** 4
0.94

**Calculation for J=** 5
0.31

---

**Option Price (Total of Calculations for all J)**
Option price= 1.87

---

## Exercise 2 (Up and down movements in stock price are not constant but their rate is)

In [3]:

```
# Setting up the known parameters
n = 6
s0 = 100
k = 100
u = 0.05
d = 0.04
r = 0.03

# Setting up the binomial tree of stock prices
s = np.zeros((n+1,n+1))
s[0,0] = s0
for i in range(1,7):
    for j in range(0,i+1):
        s[j,i] = round(s[0,0]* (1+u)**j * (1-d)**(i-j),2)

# Printing the matrix of stock prices

print('__________________________________________________')
print(magenta('Binomial tree of Stock price at each node:',['bold', 'bright'
]))

for i in range(0,7):
    print()
    print(s[i,:])

print('__________________________________________________')

# Calculating and printing risk neutral probability and Q
print()
print()
print('__________________________________________________')
print(magenta('Risk-neutral probability and Q',['bold', 'bright']))

print()
p = (r + d)/(u + d)
print('p =',round(p,2))
print()
q1 = p*(1+u)
q2 = (1+r)
q = q1/q2
print('q =',round(q,2))
print('__________________________________________________')


# Setting up binomial PMF function to calculate option price

z1 = 0
z2 = 0
t = 6
for i in range(3,t+1):
    dist1 = binom.pmf(i,t,q)
    dist2 = binom.pmf(i,t,p)
    z1 = z1 + dist1
    z2 = z2 + dist2

# Printing the sum of two binomials
```

```
print()
print('__________________________________________________')
print(magenta('Sum of two binomial functions',['bold', 'bright']))

print()
print('z1=',round(z1,2))
print()
print('z2=',round(z2,2))
print('__________________________________________________')
print()

# Calculating the option price
a1 = z1*s0
disc1 = (1+r)**t
disc2 = k/disc1
a2 = z2*disc2
price = a1-a2
print('__________________________________________________')
print(magenta('Option Price',['bold', 'bright']))

print()
print('Option price=', round(price,2))
print('__________________________________________________')
```

**Binomial tree of Stock price at each node:**

```
[100.    96.    92.16  88.47  84.93  81.54  78.28]
[  0.   105.   100.8   96.77  92.9   89.18  85.61]
[  0.     0.   110.25 105.84 101.61  97.54  93.64]
[  0.     0.     0.   115.76 111.13 106.69 102.42]
[  0.     0.     0.     0.   121.55 116.69 112.02]
[  0.     0.     0.     0.     0.   127.63 122.52]
[  0.     0.     0.     0.     0.     0.   134.01]
```

**Risk-neutral probability and Q**

p = 0.78

q = 0.79

**Sum of two binomial functions**

z1= 0.98

z2= 0.98

**Option Price**

Option price= 16.4

In [ ]: