**一、**

只有做Bonus1

**二、**

(1) Basic requirement :

**第一筆測資 : Option\_Rainboow\_1.py**

**第二筆測資 : Option\_Rainboow\_2.py**

**第三筆測資 : Option\_Rainboow\_3.py**

def cholesky\_decomposition(cov\_matrix):...

# main

K = 100

r = 0.1

T = 0.5

sims = 10000

reps = 20

n = 2

S10, S20 = 95, 95

q1, q2 = 0.05, 0.05

sigma1, sigma2 = 0.5, 0.5

sqDict = dict()

for i in range(n):

sqDict[f'S{i+1}, sigma{i+1}, q{i+1}'] = (locals()[f'S{i+1}0'], locals()[f'sigma{i+1}'], locals()[f'q{i+1}'])

print(sqDict)

print('==============================================================================================================')

######

cov\_matrix = [[sigma1\*\*2\*T, sigma1\*sigma2\*T],

[sigma2\*sigma1\*T, sigma2\*\*2\*T]]

######

直接調整 **#main** 下面的變數，將共變異數矩陣輸入 **cov\_matrix** (我已將所有參數輸入好)；再直接執行即可。

輸出看起來會是這樣 :

{'S1, sigma1, q1': (95, 0.5, 0.05), 'S2, sigma2, q2': (95, 0.5, 0.05), 'S3, sigma3, q3': (95, 0.5, 0.05), 'S4, sigma4, q4': (95, 0.5, 0.05), 'S5, sigma5, q5': (95, 0.5, 0.05)}

==============================================================================================================

Covariance Matrix:

--------------------

[0.125, 0.0625, 0.0625, 0.0625, 0.0625]

[0.0625, 0.125, 0.0625, 0.0625, 0.0625]

[0.0625, 0.0625, 0.125, 0.0625, 0.0625]

[0.0625, 0.0625, 0.0625, 0.125, 0.0625]

[0.0625, 0.0625, 0.0625, 0.0625, 0.125]

==============================================================================================================

Rainbow Option

--------------------

平均 : 30.418464

標準誤 : 0.39368

九十五趴信賴區間 : [29.631104, 31.205823]

(2) Bonus1 : **AntitheticVariate+MomentMatching.py**

同上，直接調整 **#main** 下面的變數，將共變異數矩陣輸入 **cov\_matrix** (我已將所有參數輸入好)；再直接執行即可。最後，如果模擬數量 **sims** 若輸入基數，我的程式碼會對其進行例外處理(我設計的程式碼無法處理 sims 為基數的情況)。