In [1]:
```
import numpy as np
import pandas as pd
```

In [7]:
```
#more than one stock
#prameters setting
Rf = 80/70
init_A = 70
init_B = 70

state_A = [100,50]
state_B = [80,65]

strike_price = 100

x1 = np.array([1/Rf, init_A, init_B])
x2 = pd.DataFrame([[1]*len(state_A), state_A, state_B]).T
#################################################################

x2_inv = np.linalg.inv(x2)
#vector of state prices
x3 = x1.dot(x2_inv)
#vector of risk-neutral probabilities
P_risk_neutral = x3*Rf
#payoff vector
#call
payoff_vector = np.array(np.maximum(np.sum(x2.iloc[:,1:], axis=1) - strike_
call_price = payoff_vector.dot(x3)
#put
payoff_vector = np.array(np.maximum(strike_price - np.sum(x2.iloc[:,1:], ax
put_price = payoff_vector.dot(x3)
```

```
---------------------------------------------------------------------------
--
LinAlgError                               Traceback (most recent call las
t)
Cell In[7], line 16
     13 x2 = pd.DataFrame([[1]*len(state_A), state_A, state_B]).T
     14 #################################################################
---> 16 x2_inv = np.linalg.inv(x2)
     17 #vector of state prices
     18 x3 = x1.dot(x2_inv)

File <__array_function__ internals>:200, in inv(*args, **kwargs)

File E:\software\anaconda3\Lib\site-packages\numpy\linalg\linalg.py:533,
in inv(a)
    531 a, wrap = _makearray(a)
    532 _assert_stacked_2d(a)
--> 533 _assert_stacked_square(a)
    534 t, result_t = _commonType(a)
    536 signature = 'D->D' if isComplexType(t) else 'd->d'

File E:\software\anaconda3\Lib\site-packages\numpy\linalg\linalg.py:190,
in _assert_stacked_square(*arrays)
    188 m, n = a.shape[-2:]
    189 if m != n:
--> 190     raise LinAlgError('Last 2 dimensions of the array must be squ
are')

LinAlgError: Last 2 dimensions of the array must be square
```

In [ ]: