Anàlisi de Dades Financeres

Universitat Autònoma de Barcelona

Entrega 4

Informe

Abril Pérez Martí - 1600601

Arnau Perich Iglesias - 1603567

Eric Jiménez Barril - 1599092

Joan González Martínez - 1597201

Laia Escursell Rof - 1600578

3 de novembre del 2023

Exercici 1

Programarem un algorisme d'arbre binomial per calcular els preus de les opcions. De manera excepcional inclourem part del codi en l'informe, doncs en aquest cas no és un mitjà per donar resposta a un exercici, sinó el que es demana.

(a) Construcció d'un arbre

Les variables necessàries per construir un arbre són el preu de les accions S, el moviment alcista permès u i el moviment descendent permès v. Recordeu que $0 < v < 1 < u$. Nosaltres també necessitem el nombre de passos de temps que cal utilitzar N. Escriu una funció amb el següent prototip que dóna sortida a un arbre binomial:

En primer lloc visualitzem el codi de la funció:

```
build_stock_tree = function(S,u,v,N){
  tree <- matrix(data=0,nrow=N+1, ncol=N+1)
  for (i in 1:(N+1)){
    for (j in 1:i){
      tree[i,j]=S*u^(j-1) * v^{i-j}
    }
  }
  return (tree);
}
```

Per l'exemple donat

```
build_stock_tree(S=10, u=1.1, v=0.9, N=2)
#       [,1] [,2] [,3]
# [1,] 10.0  0.0  0.0
# [2,]  9.0 11.0  0.0
# [3,]  8.1  9.9 12.1
```

obtenim el resultat que volíem.

En general, aquest codi retorna una matriu $(N+1)\times(N+1)$ que segueix la idea que es dona a l'exemple, on a cada fila de la matriu es troba el possible preu del *stock* en els diferents intervals de temps; des de $t = 0$, que es el preu inicial S (situat a la posició (1,1) de la matriu), fins a temps N (que representa l'enèsim moviment). Els preus a l'alça (amb un augment d'un u per 1) estan representats a la diagonal. Per altra banda, a la primera columna trobem els preus a la baixa (amb una caiguda d'un v per 1). Per últim, donat qualsevol preu a temps $t-1$, a sota es representa el preu si el moviment a temps t ha estat a la baixa (amb una caiguda d'un v per 1) i a sota a la dreta, el preu si el moviment ha estat a l'alça (amb un augment d'un u per 1).

(b) Probabilitat de risc neutral

Calculem ara la probabilitat de risc neutral d'un moment a l'alça q. Per a això necessitem els valors u, v i els tipus d'interès r amb el pas de temps dt (recordem que la probabilitat d'un moviment a la baixa serà (1-q)). Recordem des de la teoria, que per un argument de no-arbitratge tenim:

$$S(1 + rdt) = quS + (1 - q)vS \tag{1}$$

Escriu una funció que retorni q.

En primer lloc visualitzem el codi de la funció que segueixi el prototip proposat:

```
q_prob = function(r,u,v,dt){
  q <- (1+r*dt-v)/(u-v)
  return (q)
}
```

Notem que per l'exemple donat:

```
q_prob(0.1, 1.1, 0.9, 1/256)
# [1] 0.5019531
}
```

Simplement retornem el valor de q aïllant-lo de l'Equació 1.

$$S(1-rdt) = quS + (1-q)vS \iff q = \frac{1+rdt-u}{u-v}$$

Notem que no depèn del preu de l'actiu subjacent.

(C) Valoració recursiva

Escrivim ara una funció que calculi el valor d'una opció de forma recursiva. La primera tasca de l'opció és crear un arbre buit que omplirem amb el valor de l'opció recursivament. Ompliu els últims nodes de l'arbre amb la funció de pagament (la funció hauria de rebre una *Call* o una *Put-Option*). Ara utilitzeu un bucle per omplir els nodes un pas enrere amb la fórmula:

$$V(t) = \frac{V(t+dt)^{+}q + V(t+dt)^{-}(1-q)}{(1+rdt)} \tag{2}$$

on $V(t+dt)^{+}$ significa el valor de l'opció al moment *t+dt* en el node ascendent i similarment $V(t+dt)^{-}$.

En primer lloc visualitzem el codi de la funció que segueixi el prototip proposat:

```
value_binomial_option= function(tree, u,v,r,dt,K,type){
  option_tree = matrix(0, nrow=nrow(tree), ncol=ncol(tree))
  q <- q_prob(r,u,v,dt)
  N <- nrow(tree)

    # Omplim els  ltims  nodes de l'arbre amb la funci  de pagament
  if (type=='call'){
    # La funci  de pagament d'un K-Call es max(S_T-K,0)
    option_tree[N, 1:N] =  pmax(tree[N, 1:N]-K,0)
  }
  if (type=='put'){
    # La funci  de pagament d'un K-Put es max(K-S_T,0)
    option_tree[N,1:N] = pmax(K-tree[N,1:N],0)
  }
    # Omplim amb la f rmula
  for (i in (N-1):1){
    option_tree[i, 1:i]=(q*option_tree[i+1,2:(i+1)]+(1-q)*option_tree[i+1,1:i])/(1+r*dt)
  }
  return (option_tree)
}
```

En aquesta funció introduïm per valor l'arbre del preu de les accions que pot calcular-se amb la funció construïda en l'apartat (a), els moviments permesos alcista u i descendent v, el valor del tipus d'interès r, el pas de temps dt, el preu de l'opció K i el tipus d'opció (*Call* o *Put*). No cal pasar el preu de l'actiu subjacent ja que com hem vist a l'anterior apartat, la probabiltiat de risc neutral q no depèn d'aquest.

En primer lloc construïm un arbre de dimensió adequada per poder omplir-ho amb els valores desitjats.

Posteriorment calculem la probabilitat de risc-neutral aprofitant la funció de l'apartat anterior.

Continuem emplenant els últims nodes de l'arbre (que es corresponen amb els nodes de l'última fila de la matriu) amb el valor de la funció de pagament de una *K-Call Option* de prima 0:

$$\max(K - S(t), 0)$$

i el d'una *K-Put Option*, també de prima 0:

$$\max(S(t) - K, 0)$$

on $S(t)$ representa el valor de l'actiu en cadascun dels passos (en aquest cas els passos finals).

Per últim, omplim els nodes restats aplicant la fórmula 2 per tal d'obtenir l'arbre que volem i poder obtenir el valor de la opció desitjada.

Notem que per l'exemple donat l'output de la funció és:

```
S=10
N=5
u=1.01
v=0.99
r=0.1
dt=1/256
K=10
tree=build_stock_tree(S,u,v,N)
value_binomial_option(tree, u,v,r,dt,K, type='call')
#           [1,]       [2,]       [3,]       [4,]       [5,]       [6,]
#[1,] 0.10367391 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
#[2,] 0.04833025 0.15493435 0.00000000 0.00000000 0.00000000 0.00000000
#[3,] 0.01372347 0.08037135 0.22400759 0.00000000 0.00000000 0.00000000
#[4,] 0.00000000 0.02642543 0.13032163 0.3108179  0.00000000 0.00000000
#[5,] 0.00000000 0.00000000 0.05088385 0.2038846  0.4099448  0.00000000
#[6,] 0.00000000 0.00000000 0.00000000 0.0979801  0.3019797  0.5101005
}
```

(d) Preu Final

Ajunteu totes les funcions anteriors per crear una funció que doni el preu de l'opció.

En primer lloc visualitzem el codi de la funció que segueixi el prototip proposat:

```
binomial_option = function(type, u, v, dt, r, K,S,N){
  tree       <- build_stock_tree(S,u,v,N)
  value_tree <- value_binomial_option(tree, u,v,r,dt,K, type)
  price      <- value_tree[1,1]
  return (price)
}
```

En aquesta funció simplement unim les tres primeres. No cal passar per valor de la funció l'arbre (ja que es calcula al primer pas d'aquesta utilitzant la funció de l'exercici 1). Passem doncs els valors restants que hem passat a la funció de l'anterior apartat i a més el preu de l'actiu i el número de passos per construir l'arbre inicial.

Per últim notem que el preu que busquem se situa a la posició $(1, 1)$, per tant retornem aquest valor.

Notem que per l'exemple donat:

```
binomial_option(type='call', u=1.01, v=0.99, dt=1/256, r=0.1, K=10, S=10, N=5)
# [1] 0.1036739
```

Exercici 2

Usa la funció que construïda, per calcular el preu d'una *Call Option* amb strike ATM[1], amb actiu subyacent que actualmente cotitza a USD 100 i té venciment en un any. Utilitzeu un pas de temps mensual i un tipus d'interès anual del 10%. Les accions es poden moure a l'alça o a la baixa un 1% cada mes.

Per tal de calcular el preu d'una *Call Option* amb *strike ATM*, actiu subjacent que cotitza a \$100 i té venciment en un any, utilitzarem la funció '`binomial_option`' de l'apartat (d) de l'exercici anterior. Tindrem en compte que utilitzem un pas de temps mensual, és a dir, $dt = \frac{1}{N}$ on $N = 12$. El tipus d'interès anual serà del 10%, *i.e.*, $r = 0.1$. Per altre banda, les accions es poden moure a l'alça o la baixa un 1% cada mes, per tant, per cada dt, $u = 1.01$ i $v = 0.99$. Finalment, tenint en compte que la *Call Option* té *strike ATM*, és a dir, el preu de l'actiu subjacent en aquell moment, tenim que $K = S = 100$.

Un cop coneixem els valors de totes les variables que necessita la funció, l'executem i obtenim:

```
binomial_option(type='call', u=1.01, v=0.99, dt=1/12, r=0.1, K=100, S=100, N=12)
# [1] 9.478797
```

És a dir, el preu de la *Call option* amb les característiques donades serà de \$9.48.

[1] *ATM*: "At the money", en economia s'utilitza per descriure un contracte d'opcions on el preu de l'*strike* es idèntic al preu de l'actiu subjacent en aquell moment