

Instructivo para la valuación de derivados

28/10/2020

Integrantes del equipo :

Hernández García Yesenia Inés

Munguía Landín Luis

Ramírez Guízar Brenda Jazmín

Reyes Martínez Samuel Joshua

Observaciones:

Estas son las notaciones que se seguirán a lo largo del código:

S1 es el precio del subyacente a tiempo 0.

r es la tasa libre de riesgo, tiene que ponerse en decimales.

u factor por el que sube el precio.

d factor por el que baja el precio.

k es el precio strike.

r es la tasa libre de riesgo, tiene que ponerse en decimales.

N es el número de pasos del árbol.

TTT es el vencimiento de la opción en años.

valor digital (caso de opciones digitales).

type: son los derivados que se pueden valorar con el código ('put europea plain vanilla', 'call europea plain vanilla', 'forward', 'call digital', 'put digital').

Pasos para Valuar un derivado

1.- Abrir el archivo R que se adjuntó o simplemente copiar el código que aparece en este archivo en un nuevo proyecto de R studio.

2.- Ir a la línea 179 "valuacion del derivado...".

3.- Siguiendo la notación especificada anteriormente, se deberá escoger el tipo de derivado a evaluar y asignar los valores deseados. En caso de que no sea una opción digital, se le deberá asignar el valor 0 al pago digital.

4.- Ejecutar todo el código.

5.- Los árboles aparecerán en la parte inferior derecha de r-studio, y con las flechas que aparecen ahí se podrán observar los árboles generados.

6.- En caso de querer ver los árboles de una manera más detallada, en la parte inferior derecha de r-studio , aparece la opción de “zoom” y eso proporcionará una imagen más amplia de los árboles. De igual manera, a un lado de la opción “zoom”, podemos seleccionar la opción “export” y a través de esta se pueden descargar los árboles como imágenes o como pdf.

```
#summary(cars)
#rm(List=Ls())
```

#Código para valuar derivados europeos y digitales

#'Acomodar matriz'

```
mover <- function(A){
  B= A
  n=sqrt(length(A))
  B[1,1]=A[1,1]
  for (i in 2:n){
    for (j in 1:i){
      B[i,j]= A[i,(i+1-j)]
    }
  }
  return(B)
}
```

#'Arbol precio'

```
construyearbolpreciossb <- function(S1,u,d,N) {
  tree = matrix(0, nrow=N+1, ncol=N+1)
  for (i in 1:(N+1)) {
    for (j in 1:i) {
      tree[i, j] = S1 * u^(j-1) * d^((i-1)-(j-1))
    }
  }
  return(tree)
}
```

#'Proba q'

```
q_prob <- function(r,u, d,N,TTT) {
  delta_t=TTT/N
  return((exp(r*delta_t) - d)/(u-d))
}
```

```

#'Arbol derivado'
valuanderivado <- function(tree,S1,r,u,d,N,TTT, K, type,valordigital) {
  q = q_prob(r,u,d,N,TTT)
  delta_t=TTT/N
  option_tree = matrix(0, nrow=nrow(tree), ncol=ncol(tree))
  if(type == 'put europea plain vanilla') {
    option_tree[nrow(option_tree),] = pmax(K - tree[nrow(tree),], 0)
  } else if (type=='call europea plain vanilla'){
    option_tree[nrow(option_tree),] = pmax(tree[nrow(tree),] - K, 0)
  } else if(type=='forward'){
    option_tree[nrow(option_tree),] = tree[nrow(tree),] - K
  }else if(type=='call digital'){
    for(k in 1:nrow(tree)){
      if(tree[nrow(tree),k]< valordigital){
        option_tree[nrow(tree),k]<- valordigital
      }else{
        option_tree[nrow(tree),k]<-0
      }
    }
  }

  } else if(type=='put digital'){
    for(k in 1:nrow(tree)){
      if(tree[nrow(tree),k]>valordigital){
        option_tree[nrow(tree),k]<- valordigital
      }else{
        option_tree[nrow(tree),k]<-0
      }
    }
  }

  } else {
    print("Ingresa un tipo valido de derivado")
  }
  for (i in (nrow(tree)-1):1) {
    for(j in 1:i) {
      option_tree[i,j]=((1-q)*option_tree[i+1,j] +
q*option_tree[i+1,j+1])/exp(r*delta_t)
    }
  }
  return(option_tree)
}

#'Arbol alpha'
subyacente <-function(arbolito,payoff){
  subyacente_tree = matrix(0, nrow=nrow(arbolito)-1, ncol=ncol(arbolito)-
1)
  for (i in (nrow(arbolito)-1):1) {
    for(j in 1:i) {
      subyacente_tree[i,j]=(payoff[i+1,j+1]-
payoff[i+1,j])/(arbolito[i+1,j+1]-arbolito[i+1,j])
    }
  }
}

```

```

    }
    return(subyacente_tree)
}

#'Arbol beta'
cuentademercado <-function(arbolito,payoff,r,N,TTT){
  delta_t=TTT/N
  cuentademercado_tree = matrix(0, nrow=nrow(arbolito)-1,
ncol=ncol(arbolito)-1)
  for (i in (nrow(arbolito)-1):1) {
    for(j in 1:i) {
      cuentademercado_tree[i,j]=(payoff[i+1,j+1]-((payoff[i+1,j+1]-
payoff[i+1,j]))/(arbolito[i+1,j+1]-
arbolito[i+1,j]))*arbolito[i+1,j+1])/exp(r*delta_t)
    }
  }
  return(cuentademercado_tree)
}

#'Código para graficar el arbol'
my_BinomialTreePlot<-function (BinomialTreeValues,xlab,ylab,dx = -
0.00025, dy = 0.004, cex = 1,
                                digits = 2, ...)
{
  #Redondeo de Los valores
  Tree = round(BinomialTreeValues, digits = digits)
  #Tree=BinomialTreeValues
  depth = ncol(Tree)
  plot(x = c(0, depth-1), y = c(-depth + 1, depth -
1),axes=FALSE,xlab=xlab,ylab=ylab,type = "n",
       col = 0, ...)
  #Coloque el primer valor
  text(0 + dx, 0 + dy, deparse(Tree[1, 1]), cex = cex)
  for (i in 1:(depth - 1)) {
    y = seq(from = -i, by = 2, length = i + 1)
    x = rep(i, times = length(y)) + 0
    for (j in 1:length(x)) text(x[j] + dx, y[j] + dy,
deparse(Tree[length(x) +
j, i + 1]), cex = cex)
    #Dibujar Las líneas
    y = (-i):i
    x = rep(c(i, i-1), times = 2 * i)[1:length(y)]
    lines(x, y, col = 2)
  }
  invisible()
}

#'Imprimir'
arboldeprecio <- function(x) {

```

```
arboldepreciosubyacente<-my_BinomialTreePlot(t(mover(x)),ylab="Precio  
del subyacente",xlab="Paso")
```

```
  return(arboldepreciosubyacente)  
}
```

```
arboldeelder <- function(y) {  
  arboldeelder <-my_BinomialTreePlot(t(mover(y)),ylab="Precio del  
derivado",xlab="Paso")
```

```
  return(arboldeelder)  
}
```

```
arbolalpha <- function(z) {  
  arbolalpha <- my_BinomialTreePlot(t(mover(z)),ylab="Cantidad del bien  
subyacente",xlab="Paso")
```

```
  return(arbolalpha)  
}
```

```
arbolbeta <- function(v) {  
  arbolbeta <- my_BinomialTreePlot(t(mover(v)),ylab="Cantidad en la  
cuenta de mercado de dinero",xlab="Paso")  
  return(arbolbeta)
```

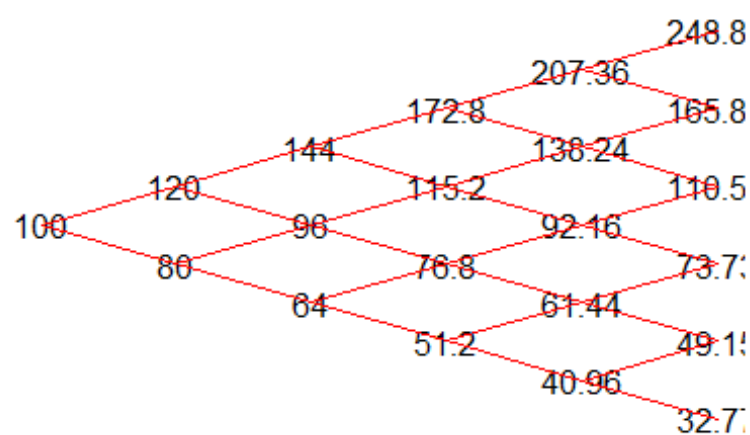
```
}
```

```
#'Función que devuelve Los precios y Los arboles'
```

```
valuacionderivado <- function(type,S1,r,u,d,N,TTT,K,valordigital=0) {  
  q <- q_prob(r,u, d,N,TTT)  
  x <- construyearbolpreciossub(S1,u,d,N)  
  y <- valuaderivado(x,S1,r,u,d,N,TTT, K, type,valordigital)  
  z <- subyacente(x,y)  
  v <- cuentademercado(x,y,r,N,TTT)  
  return(list(q=q, arboldeprecio(x),  
arboldeelder(y),arbolalpha(z),arbolbeta(v), prima=y[1,1]))  
}
```

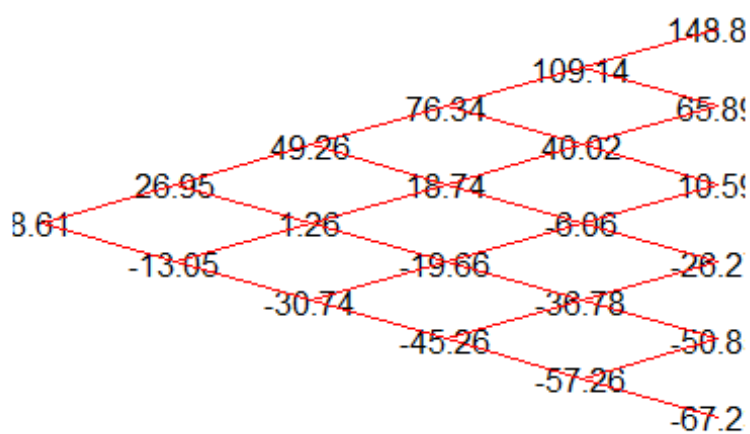
```
#valuacionderivado(type='call europea plain  
vanilla',S1=100,r=0.045,u=1.2,d=.8,N=5,TTT=2,K=100,valordigital = 0)  
#valuacionderivado(type='call  
digital',S1=100,r=0.045,u=1.2,d=.8,N=5,TTT=2,K=100,valordigital=100)  
valuacionderivado(type='forward',S1=100,r=0.045,u=1.2,d=.8,N=5,TTT=2,K=10  
0,valordigital=0)
```

Precio del subyacente



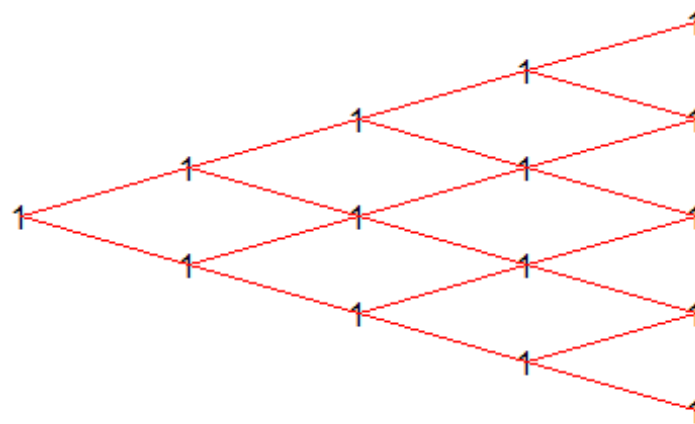
Paso

Precio del derivado



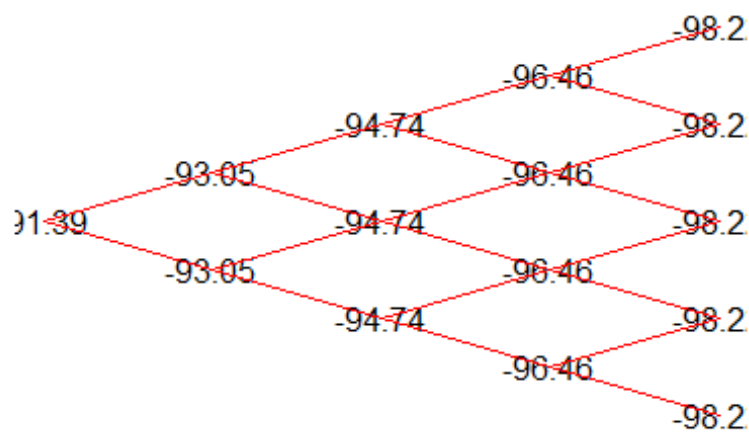
Paso

Cantidad del bien subyacente



Paso

Cantidad en la cuenta de mercado de dinero



Paso

```
## $q
## [1] 0.5454074
##
## [[2]]
```

```
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## NULL
##
## $prima
## [1] 8.606881

#'Los árboles son ejemplo de una opción forward'

'''
```