

# TAREA 5

Eduardo Tomás Leyva Díaz

## ÁRBOL BINOMIAL

```
library(quantmod)
```

La acción elegida de la Bolsa Mexicana de Valores para el análisis es **CEMEX**, así que primero se necesitan los datos del 01 de marzo del 2022 al 01 de marzo del 2024.

```
# PRECIOS
datos<-c("CEMEXCPO.MX")
getSymbols(datos,src="yahoo",from="2022-03-01",to="2024-03-02")
```

```
## [1] "CEMEXCPO.MX"
```

Se genera un data frame pero sólo con los precios de cierre de este activo y con los rendimientos de manera discreta ya que la acción cotiza en un mercado que tiene hora de apertura y de cierre, no es de 24 horas.

```
accion<-CEMEXCPO.MX$CEMEXCPO.MX.Close
accion<-as.data.frame(accion)
accion$Rendimientos<-NA
colnames(accion)<-c("Precio","Rendimientos")
for(k in 2:nrow(accion)){
  accion$Rendimientos[k]<-(accion$Precio[k]/accion$Precio[k-1]) - 1 # rendimientos discretos
}
```

Ahora se definen los datos de entrada del modelo de simulación (precio spot, precio strike, volatilidad, diferencial de tiempo, tasa libre de riesgo)

```
# INPUTS
s0<-accion[nrow(accion),1] # Último Precio Conocido
k<-15
step<-1/12
sigma<-sd(accion$Rendimientos,na.rm=TRUE) * sqrt(252)

r<-0.1025

# Variables del Modelo
u<-exp(+sigma*sqrt(step))
d<-exp(-sigma*sqrt(step))
p<-(exp(+r*step)-d)/(u-d)
q<-1-p
```

## Modelo Determinista

Hay que llenar los vectores de precios al final del periodo y después obtener las ganancias potenciales, esto se hará para la opción call y para la opción put.

```
# PRECIOS
periodos<-10
s<-c()
for(i in 1:(periodos+1)){
  s[i]<-s0 * (u^(periodos+1-i)) * (d^(i-1))
}
s
```

```
## [1] 38.545238 31.179765 25.221735 20.402203 16.503618 13.350000 10.798996
## [8] 8.735454 7.066227 5.715967 4.623723
```

```
# GANANCIAS POTENCIALES
gcall<-c()
for(i in 1:(periodos+1)){
  gcall[i]<-max(s[i]-k,0)
}
gcall
```

```
## [1] 23.545238 16.179765 10.221735 5.402203 1.503618 0.000000 0.000000
## [8] 0.000000 0.000000 0.000000 0.000000
```

```
gput<-c()
for(i in 1:(periodos+1)){
  gput[i]<-max(k-s[i],0)
}
gput
```

```
## [1] 0.000000 0.000000 0.000000 0.000000 0.000000 1.650000 4.201004
## [8] 6.264546 7.933773 9.284033 10.376277
```

Para hallar el valor de la prima, hay que obtener la esperanza al final del periodo, y esto se lleva a cabo con las probabilidades de una distribución binomial, es decir:

```
# PROBABILIDADES
# Va de subir 10 veces a subir 0 veces
proba<-c()
for(k in 1:(periodos+1)){
  proba[k]<-dbinom(periodos+1-k,periodos,p)
}
proba
```

```
## [1] 0.0012844687 0.0121501999 0.0517196798 0.1304621256 0.2159645104
## [6] 0.2451452741 0.1932422874 0.1044537440 0.0370522998 0.0077886564
## [11] 0.0007367539
```

Finalmente, sólo queda multiplicar las ganancias potenciales por las probabilidades de que ocurran y traerlas a valor presente

```
# PRIMA CALL
primacall<-sum(gcall*proba) * exp(-r*step*periodos)
primacall
```

```
## [1] 1.638867
```

```
# PRIMA PUT
primaput<-sum(gput*proba) * exp(-r*step*periodos)
primaput
```

```
## [1] 2.060812
```

## Simulación Montecarlo

Para hacer la simulación primero se define la función que contará el número de veces que el precio sube

```
sube<-function(periodos,proba){
  contador = 0
  for(j in 1:periodos){
    u<-runif(1,0,1)

    if(u<proba){
      # subió
      contador = contador +1
    }else{
      #bajó
      contador = contador + 0
    }
  }
  return (contador)
}
```

Con esto ya se puede realizar el procedimiento para un número fijo de simulaciones, el cual consiste en asignar la ganancia potencial dependiendo las veces que el precio subió o bajo. Después se trae a valor presente cada ganancia y al final se saca un promedio para obtener el valor final de la prima.

## 10 SIMULACIONES

```
# PRIMA CALL
n<-10 # simulaciones
primacall<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){
    primacall[k]<-gcall[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primacall[k]<-gcall[10] * exp(-r*step*periodos)
  }
}
```

```

}else if (resultado==2){
  primacall[k]<-gcall[9] * exp(-r*step*periodos)
}else if (resultado==3){
  primacall[k]<-gcall[8] * exp(-r*step*periodos)
}else if (resultado==4){
  primacall[k]<-gcall[7] * exp(-r*step*periodos)
}else if (resultado==5){
  primacall[k]<-gcall[6] * exp(-r*step*periodos)
}else if (resultado==6){
  primacall[k]<-gcall[5] * exp(-r*step*periodos)
}else if (resultado==7){
  primacall[k]<-gcall[4] * exp(-r*step*periodos)
}else if (resultado==8){
  primacall[k]<-gcall[3] * exp(-r*step*periodos)
}else if (resultado==9){
  primacall[k]<-gcall[2]
}else{
  primacall[k]<-gcall[1]
}
}
mean(primacall)

```

```
## [1] 2.122021
```

```

# PRIMA PUT
n<-10 # simulaciones
primaput<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){
    primaput[k]<-gput[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primaput[k]<-gput[10] * exp(-r*step*periodos)
  }else if (resultado==2){
    primaput[k]<-gput[9] * exp(-r*step*periodos)
  }else if (resultado==3){
    primaput[k]<-gput[8] * exp(-r*step*periodos)
  }else if (resultado==4){
    primaput[k]<-gput[7] * exp(-r*step*periodos)
  }else if (resultado==5){
    primaput[k]<-gput[6] * exp(-r*step*periodos)
  }else if (resultado==6){
    primaput[k]<-gput[5] * exp(-r*step*periodos)
  }else if (resultado==7){
    primaput[k]<-gput[4] * exp(-r*step*periodos)
  }else if (resultado==8){
    primaput[k]<-gput[3] * exp(-r*step*periodos)
  }else if (resultado==9){
    primaput[k]<-gput[2]
  }else{
    primaput[k]<-gput[1]
  }
}

```

```

    }
  }
  mean(primaput)

```

```
## [1] 1.954311
```

## 100 SIMULACIONES

```

# PRIMA CALL
n<-100 # simulaciones
primacall<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){
    primacall[k]<-gcall[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primacall[k]<-gcall[10] * exp(-r*step*periodos)
  }else if (resultado==2){
    primacall[k]<-gcall[9] * exp(-r*step*periodos)
  }else if (resultado==3){
    primacall[k]<-gcall[8] * exp(-r*step*periodos)
  }else if (resultado==4){
    primacall[k]<-gcall[7] * exp(-r*step*periodos)
  }else if (resultado==5){
    primacall[k]<-gcall[6] * exp(-r*step*periodos)
  }else if (resultado==6){
    primacall[k]<-gcall[5] * exp(-r*step*periodos)
  }else if (resultado==7){
    primacall[k]<-gcall[4] * exp(-r*step*periodos)
  }else if (resultado==8){
    primacall[k]<-gcall[3] * exp(-r*step*periodos)
  }else if (resultado==9){
    primacall[k]<-gcall[2]
  }else{
    primacall[k]<-gcall[1]
  }
}
mean(primacall)

```

```
## [1] 1.328715
```

```

# PRIMA PUT
n<-100 # simulaciones
primaput<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){

```

```

    primaput[k]<-gput[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primaput[k]<-gput[10] * exp(-r*step*periodos)
  }else if (resultado==2){
    primaput[k]<-gput[9] * exp(-r*step*periodos)
  }else if (resultado==3){
    primaput[k]<-gput[8] * exp(-r*step*periodos)
  }else if (resultado==4){
    primaput[k]<-gput[7] * exp(-r*step*periodos)
  }else if (resultado==5){
    primaput[k]<-gput[6] * exp(-r*step*periodos)
  }else if (resultado==6){
    primaput[k]<-gput[5] * exp(-r*step*periodos)
  }else if (resultado==7){
    primaput[k]<-gput[4] * exp(-r*step*periodos)
  }else if (resultado==8){
    primaput[k]<-gput[3] * exp(-r*step*periodos)
  }else if (resultado==9){
    primaput[k]<-gput[2]
  }else{
    primaput[k]<-gput[1]
  }
}
mean(primaput)

```

```
## [1] 1.984127
```

## 1000 SIMULACIONES

```

# PRIMA CALL
n<-1000 # simulaciones
primacall<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){
    primacall[k]<-gcall[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primacall[k]<-gcall[10] * exp(-r*step*periodos)
  }else if (resultado==2){
    primacall[k]<-gcall[9] * exp(-r*step*periodos)
  }else if (resultado==3){
    primacall[k]<-gcall[8] * exp(-r*step*periodos)
  }else if (resultado==4){
    primacall[k]<-gcall[7] * exp(-r*step*periodos)
  }else if (resultado==5){
    primacall[k]<-gcall[6] * exp(-r*step*periodos)
  }else if (resultado==6){
    primacall[k]<-gcall[5] * exp(-r*step*periodos)
  }else if (resultado==7){

```

```

    primacall[k]<-gcall[4] * exp(-r*step*periodos)
  }else if (resultado==8){
    primacall[k]<-gcall[3] * exp(-r*step*periodos)
  }else if (resultado==9){
    primacall[k]<-gcall[2]
  }else{
    primacall[k]<-gcall[1]
  }
}
mean(primacall)

```

```
## [1] 1.785899
```

```

# PRIMA PUT
n<-1000 # simulaciones
primaput<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){
    primaput[k]<-gput[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primaput[k]<-gput[10] * exp(-r*step*periodos)
  }else if (resultado==2){
    primaput[k]<-gput[9] * exp(-r*step*periodos)
  }else if (resultado==3){
    primaput[k]<-gput[8] * exp(-r*step*periodos)
  }else if (resultado==4){
    primaput[k]<-gput[7] * exp(-r*step*periodos)
  }else if (resultado==5){
    primaput[k]<-gput[6] * exp(-r*step*periodos)
  }else if (resultado==6){
    primaput[k]<-gput[5] * exp(-r*step*periodos)
  }else if (resultado==7){
    primaput[k]<-gput[4] * exp(-r*step*periodos)
  }else if (resultado==8){
    primaput[k]<-gput[3] * exp(-r*step*periodos)
  }else if (resultado==9){
    primaput[k]<-gput[2]
  }else{
    primaput[k]<-gput[1]
  }
}
mean(primaput)

```

```
## [1] 2.073427
```

10000 SIMULACIONES

```

# PRIMA CALL
n<-10000 # simulaciones
primacall<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){
    primacall[k]<-gcall[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primacall[k]<-gcall[10] * exp(-r*step*periodos)
  }else if (resultado==2){
    primacall[k]<-gcall[9] * exp(-r*step*periodos)
  }else if (resultado==3){
    primacall[k]<-gcall[8] * exp(-r*step*periodos)
  }else if (resultado==4){
    primacall[k]<-gcall[7] * exp(-r*step*periodos)
  }else if (resultado==5){
    primacall[k]<-gcall[6] * exp(-r*step*periodos)
  }else if (resultado==6){
    primacall[k]<-gcall[5] * exp(-r*step*periodos)
  }else if (resultado==7){
    primacall[k]<-gcall[4] * exp(-r*step*periodos)
  }else if (resultado==8){
    primacall[k]<-gcall[3] * exp(-r*step*periodos)
  }else if (resultado==9){
    primacall[k]<-gcall[2]
  }else{
    primacall[k]<-gcall[1]
  }
}
mean(primacall)

```

```
## [1] 1.686497
```

```

# PRIMA PUT
n<-10000 # simulaciones
primaput<-c()

for(k in 1:n){
  resultado = sube(periodos,p)

  if(resultado==0){
    primaput[k]<-gput[11] * exp(-r*step*periodos)
  }else if (resultado==1){
    primaput[k]<-gput[10] * exp(-r*step*periodos)
  }else if (resultado==2){
    primaput[k]<-gput[9] * exp(-r*step*periodos)
  }else if (resultado==3){
    primaput[k]<-gput[8] * exp(-r*step*periodos)
  }else if (resultado==4){
    primaput[k]<-gput[7] * exp(-r*step*periodos)
  }else if (resultado==5){

```



```

    primaput[k]<-gput[6] * exp(-r*step*periodos)
  }else if (resultado==6){
    primaput[k]<-gput[5] * exp(-r*step*periodos)
  }else if (resultado==7){
    primaput[k]<-gput[4] * exp(-r*step*periodos)
  }else if (resultado==8){
    primaput[k]<-gput[3] * exp(-r*step*periodos)
  }else if (resultado==9){
    primaput[k]<-gput[2]
  }else{
    primaput[k]<-gput[1]
  }
}
mean(primaput)

```

```
## [1] 2.042242
```