UNIVERSITÉ CARTHAGE
ECOLE SUPÉRIEURE DE LA STATISTIQUE
ET DE L'ANALYSE DE L'INFORMATION

Calcul d'option par le modèle de Black Scholes

Elaboré par : Madani SARRA

Année Universitaire :2016/2017

Introduction :

Le modèle Black-Scholes est un modèle mathématique du marché pour une action, dans lequel le prix de l'action est un processus stochastique en temps continu .

La formule de Black-Scholes permet de calculer la valeur théorique d'une option européenne à partir des cinq données suivantes :

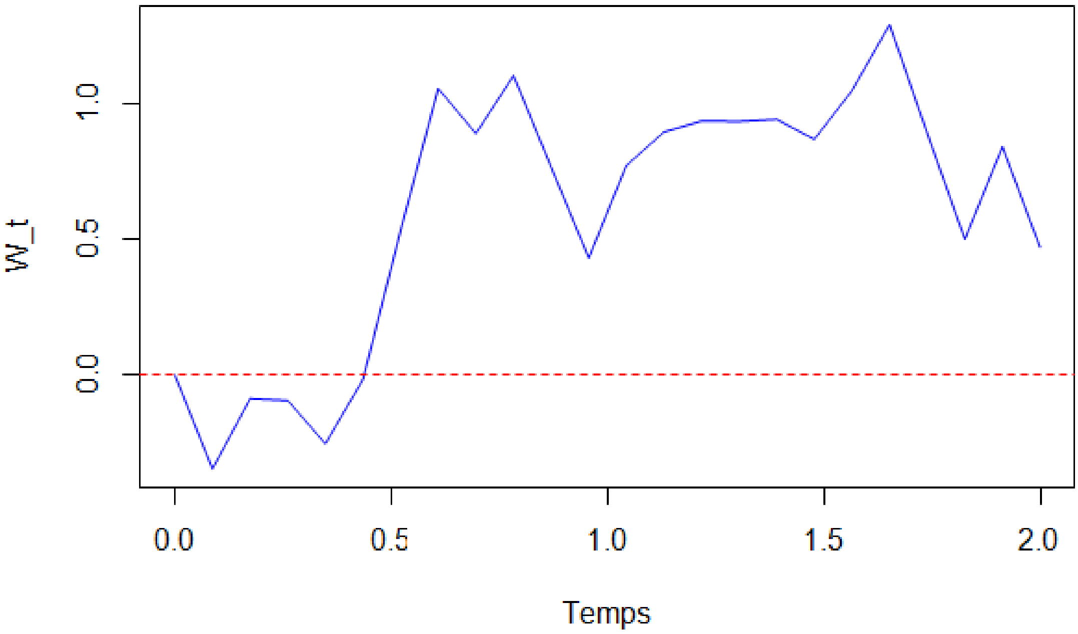
- La valeur actuelle de l'action sous-jacente,
- Le temps qu'il reste à l'option avant son échéance (exprimé en années),
- Le prix d'exercice fixé par l'option,
- Le taux d'intérêt sans risque,
- La volatilité du prix de l'action.

Pour commencer le travail, on fixe tous les paramètres :

```
sigma =0.1
r =0.03
K =110
S_0 =100
T =2
mu =0.07
n=24     #nombre  de  subdivisions
```

1. Généralisation du mouvement brownien :

```
brownienStandard =function(n, T, plot){
  Delta =T/n
  W_accroissement =rnorm(n-1,sd =          sqrt(Delta))      #W_t - W_s

  W = c(0, cumsum(W_accroissement))


  if(plot){
    t = seq(0,T,length.out =n)
    plot(t,W,type ="l",xlab ="Temps",ylab ="W_t",col ="blue")
    abline(h =0,col ="red",lwd=1,lty=4)
  }
  return(W)

}
```




```
## [1]  0.00000000 -0.34662497 -0.08684256 -0.09354013 -0.25304999
## [6] -0.01630959  0.53269377  1.05706039  0.89049303  1.10466695
## [11]  0.76735503  0.43158443  0.77215207  0.89589185  0.93634167
## [16]  0.93504132  0.94279390  0.86912639  1.04884654  1.29274085
## [21]  0.89219592  0.50057747  0.84192367  0.47090368
```

2. Option Européenne (Call) :

```
simul_S_T =function (mu, sigma, T,n){
  S_0=100
  deltaT =T/n
  W = brownienStandard  (n, T,plot  =FALSE)

  S T =S 0 *  exp((mu -(sigma^2)/2)*T +sigma*(W[n] -W[n-1]))
  return(S T)
}
```

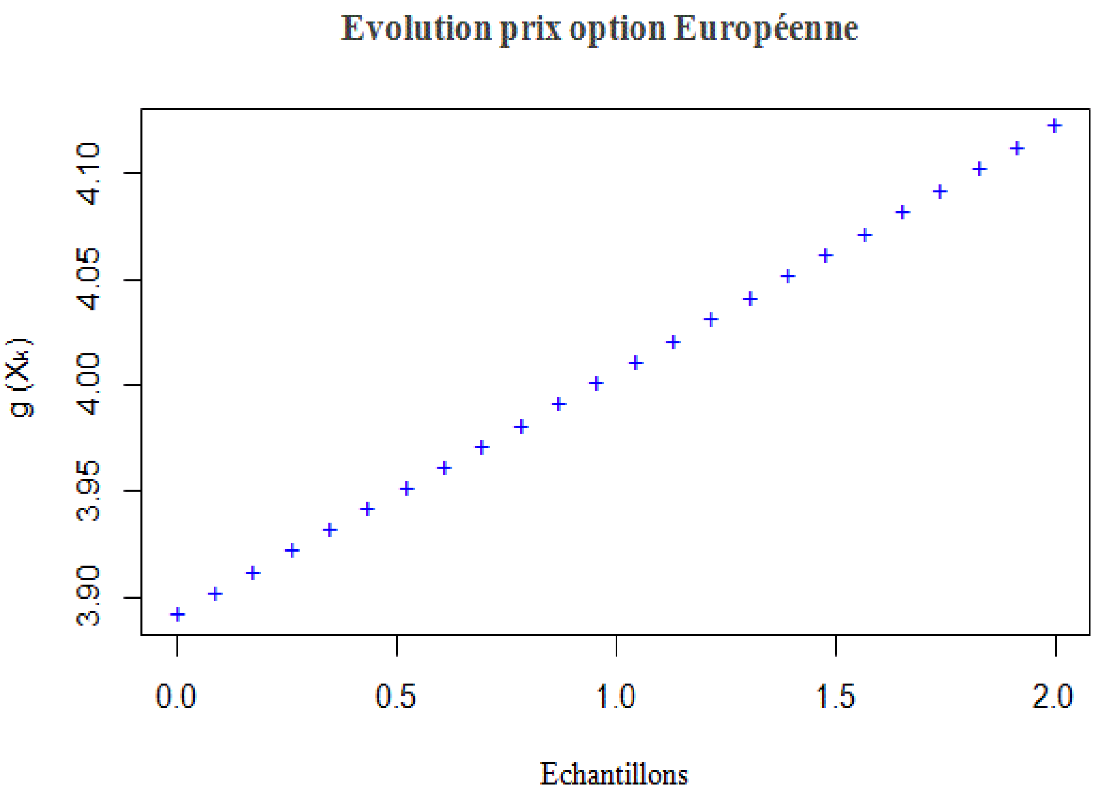
Exemple:

```
simul_S_T(mu, sigma, T, n)
```

```
## [1] 117.8615
```

```r
simul_europ_option=function(taille,r,T,n,K,plot){
  deltaT =T/n
  C = numeric(n)
  g = numeric(taille)

  for(i in1:taille) g[i] = max(simul_S_T(mu, sigma, T, n) -K,0)
  print(mean(g))

  for(i in0:n) C[i] = exp(-r*(T -i*deltaT))* mean(g)

  if(plot){
    plot(seq(0,T,length.out =n), C,pch ="+",col ="#0000ff",xlab ="Echantillons",
         ylab="g(X_k)",main="Evolution prix option europÃ ©en")
  }
  return(C)
}

simul_europ_option(1000, r,T, n, K,plot =TRUE)
```

```
## [1] 4.121136
```



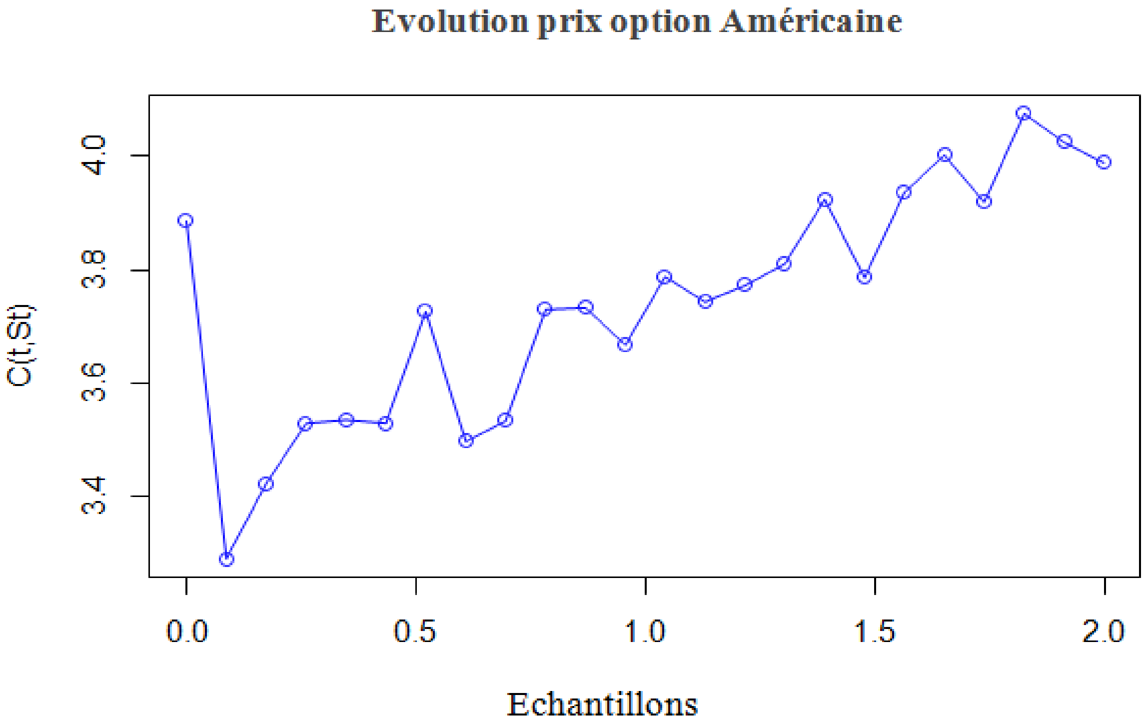

```
##  [1] 3.890855 3.900594 3.910358 3.920146 3.929959 3.939796 3.949658
##  [8] 3.959544 3.969455 3.979391 3.989352 3.999338 4.009349 4.019385
## [15] 4.029446 4.039532 4.049644 4.059780 4.069943 4.080130 4.090343
## [22] 4.100582 4.110846 4.121136
```

3. Option Américaine (Call):

```
simul_americ_option =function(taille, r, T, n, K, plot){
  iteration = seq(0,T,length.out =n)

  C=numeric(n) j=1

  for(t in iteration){ sequence =
    seq(t,T,by=T/n)
    v =numeric(length(sequence))

    i =1
    for(to in sequence){
      v[i]=exp(-r*(to-t))*              mean(replicate(taille,max(simul_S_T(mu,sigma,to,n)-K,0 i=i+1

    }
    print(v)


    C[j] = max(v) j
    =j+1
  }

  if(plot){
    plot(seq(0,T,length.out =n), C,type ="o",col ="#0000ff",xlab ="Echantillons", ylab
        ="C(t,S_t)",main ="Evolution prix option americain")
  }

  return(C)

}
```

```
##  [1] 0.000000000  0.000000000 0.000000000 0.000000000 0.000000000
##  [6] 0.000000000  0.000000000 0.000000000 0.000508857 0.001554559
## [11] 0.004191180  0.019811178 0.066739540 0.141066805 0.275302581
## [16] 0.394515021  0.616884453 0.841777384 1.217898631 1.566977429
## [21] 2.089454121  2.308749916 2.640611312 3.288504965 3.885786196
##  [1] 0.0000000000  0.0000000000 0.0000000000 0.0000000000 0.0000000000
##  [6] 0.0000000000  0.0000000000 0.0001070239 0.0005758199 0.0123823760
## [11] 0.0308553299  0.0564425081 0.1318941822 0.2587174826 0.4121036880
## [16] 0.6212585868  0.8302694339 1.1883412506 1.6096127000 1.9953791570
## [21] 2.3972817501  2.8434987074  3.2891688609
##  [1] 0.000000000  0.000000000 0.000000000 0.000000000 0.000000000
##  [6] 0.000000000  0.000261339 0.004053879 0.018608363 0.024167846
## [11] 0.069573258  0.152628472 0.271170410 0.381813030 0.674278717
## [16] 0.811675718  1.261319085 1.580985397 2.135930373 2.415359867
## [21] 2.829162590  3.422221720
##  [1] 0.0000000000  0.0000000000  0.0000000000  0.0000000000  0.0000000000
##  [6] 0.0004337946  0.0041533876  0.0071233210  0.0260021326  0.0764007000
## [11] 0.1612620947  0.2425825041  0.3760640516  0.6189317419  0.9076332373
## [16] 1.1343389834  1.6544166456  1.9403370782  2.5416038747  2.9488870712
## [21] 3.5285361091
##  [1] 0.000000000  0.000000000 0.000000000 0.000000000 0.001083300
##  [6] 0.003477703  0.010258053 0.042548424 0.073574641 0.133388303
```

```
## [11]  0.230175248 0.436509572 0.649113402 0.931487372 1.192713327
## [16]  1.603442540 1.989088464 2.664683295 3.017277984 3.534409552
##  [1] 0.000000000 0.000000000 0.000000000 0.005583026 0.010339383
##  [6] 0.027396265 0.069229800 0.171407900 0.287588449 0.460167304
## [11] 0.728386356 0.925068063 1.262077472 1.681311581 2.148443532
## [16] 2.491401729 3.045897680 3.725968293
##  [1] 0.000000000 0.002286983 0.006580670 0.012139576 0.050213990
##  [6] 0.096436183 0.163529179 0.299908268 0.520324851 0.773789015
## [11] 1.003488994 1.296405103 1.688622739 2.105612423 2.613342233
## [16] 3.127977230  3.496231767
##  [1] 0.0008573376 0.0058907157 0.0170330746 0.0398969153 0.0954744505
##  [6] 0.1841523003 0.2859410399 0.5085100864 0.7529449290 1.0007617942
## [11] 1.3690553171 1.6582043628 2.2428770145 2.5570888160 3.0892318567
## [16] 3.5326791334
##  [1] 0.005870876 0.017648557 0.038649558 0.075041334 0.190552266
##  [6] 0.283499750 0.466945994 0.685288952 1.055604418 1.337358399
## [11] 1.728081953 2.074103301 2.646608842 3.081757904 3.729475720
##  [1] 0.01570275  0.05449687  0.09292932  0.19177092  0.30759063  0.52747822
##  [7] 0.69839637  0.96820200  1.37799785  1.74992366  2.23853294  2.62090274
## [13] 3.14390894 3.73287242
##  [1] 0.04726102  0.08540454  0.20435100  0.34470676  0.52967177  0.74071316
##  [7] 1.06922293  1.37718246  1.93162039  2.25272132  2.60258206  3.17555734
## [13] 3.66681087
##  [1] 0.09156828  0.17030518  0.31668253  0.51943412  0.76634463  1.07340826
##  [7] 1.31240997  1.85647085  2.21642338  2.81270842  3.11529553  3.78749525
##  [1] 0.2010655 0.3436661 0.5702370 0.8071291 1.0460524 1.4483139 1.8300195
##  [8] 2.2629006  2.7760013  3.2936185  3.7430372
##  [1] 0.3450713 0.4901372 0.8159343 1.1059841 1.4293131 1.7853484 2.2338709
##  [8] 2.8450982 3.3479961 3.7719097
## [1] 0.4952912 0.8458460 1.0515825 1.4023009 1.8955240 2.4512481 2.9090294
## [8] 3.3565197 3.8099862
## [1] 0.8206117 1.1679084 1.5520856 1.9272565 2.4049828 2.8236499 3.2221274
## [8] 3.9238090
## [1] 1.224029 1.502047 1.718767 2.612502 2.898217  3.382663  3.785837
## [1] 1.506537  2.011794 2.415767 2.975524 3.343433 3.936345
## [1] 1.918411  2.357686 2.919941 3.404642 4.002486
## [1] 2.500905 2.978854 3.503347 3.919195
## [1] 2.978848 3.491626 4.075582
## [1] 3.353117  4.024951
## [1] 3.988425
```




```
##  [1]  0.00000000 -0.34662497 -0.08684256 -0.09354013  -0.25304999
##  [6] -0.01630959  0.53269377  1.05706039  0.89049303  1.10466695
## [11]  0.76735503  0.43158443  0.77215207  0.89589185  0.93634167
## [16]  0.93504132  0.94279390  0.86912639  1.04884654  1.29274085
## [21]  0.89219592  0.50057747  0.84192367  0.47090368
```