TP 6 d'AD

Hugo Happy et Yu Zigong

13/04/2020

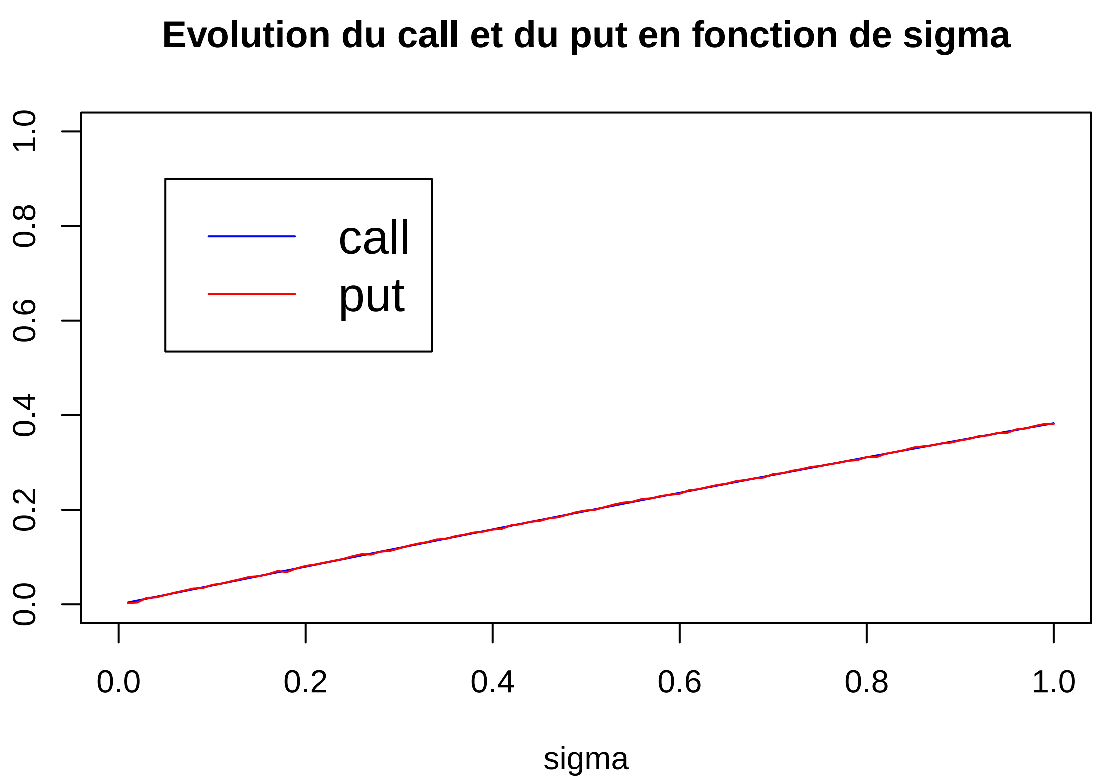
<u>Exercice 2</u>: Calcul du put et du call

1) Ecriture des fonctions call et put

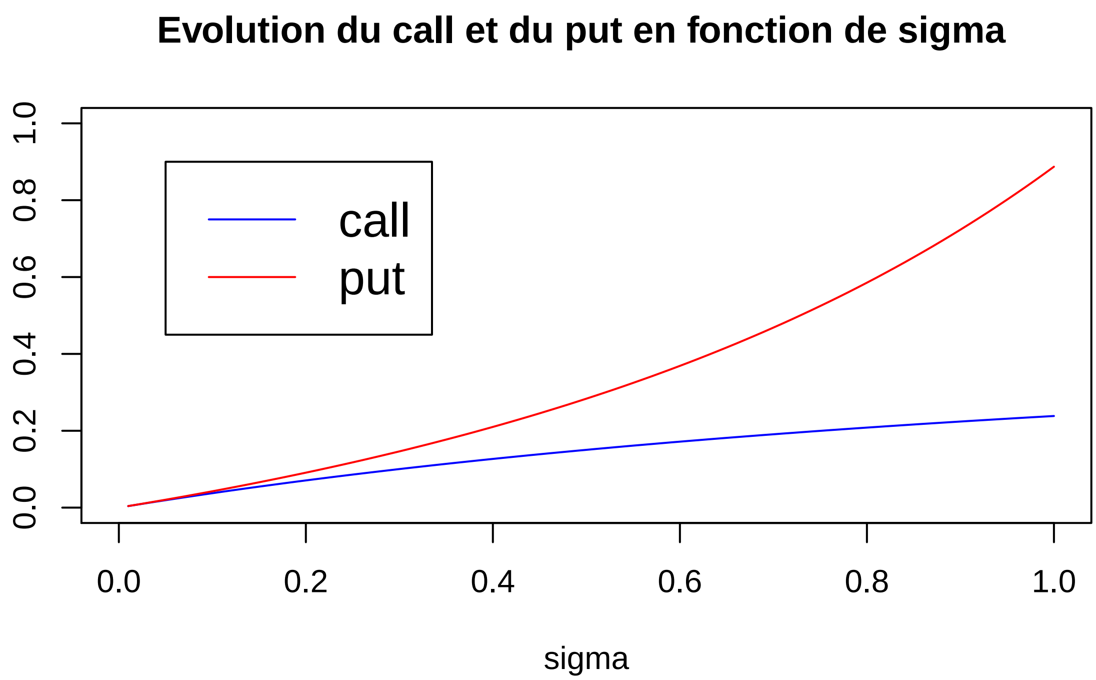
```
S=1
r=0

call <- function(S,r,sigma,K)
{
  return(S*exp(r)*(1-pnorm(log(K/S)/sigma-r/sigma-sigma/2,0,1))-
           K*(1-pnorm(log(K/S)/sigma-r/sigma+sigma/2,0,1)))
}

put <- function(S,r,sigma,K)
{
  return(K*pnorm(log(K/S)/sigma-r/sigma+sigma/2,0,1)-
           S*exp(r)*pnorm(log(K/S)/sigma-r/sigma-sigma/2,0,1))
}
```

2) Evolution du call et du put en fonction du taux d'intérêt instantanné

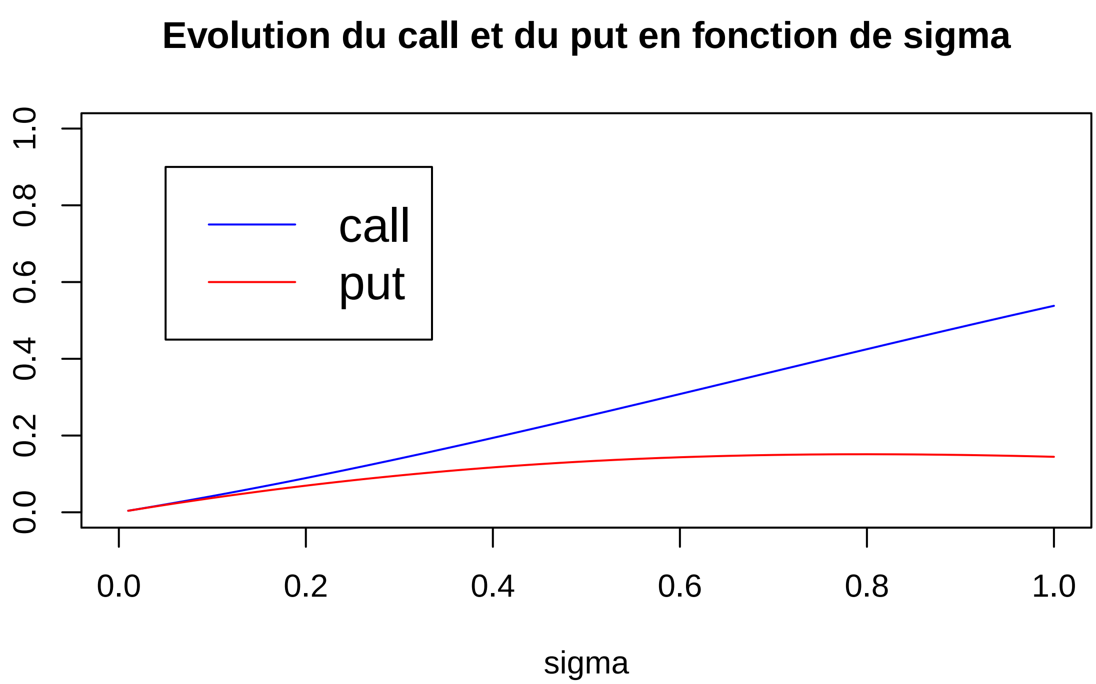
Pour K = 1 :



Pour $K = e^{\sigma^2/2}$:



Pour $K = e^{-\sigma^2/2}$:



On remarque que quand K=1, call=put, et quand K>1 (dans le cas $K = e^{\sigma^2/2}$ et $\sigma > 0$), call<put, et quand K<1(dans le cas $K = e^{-\sigma^2/2}$ et $\sigma > 0$) call>put.

<u>Exercice 3</u>: Loi des grands nombres

1) Ecriture de la fonction rcallput

```
sigma=1
K=1
rcallput <- function(n,sigma,K)
{
  Xi = rnorm(n,0,1)
  Yi = exp(sigma*Xi-(sigma^2)/2)-K
  Zi = K-exp(sigma*Xi-(sigma^2)/2)
  resultat = matrix(nrow=n,ncol=2)
  resultat[,1]=Yi
  resultat[,2]=Zi
  resultat[resultat<0]=0
  return(resultat)
}
```

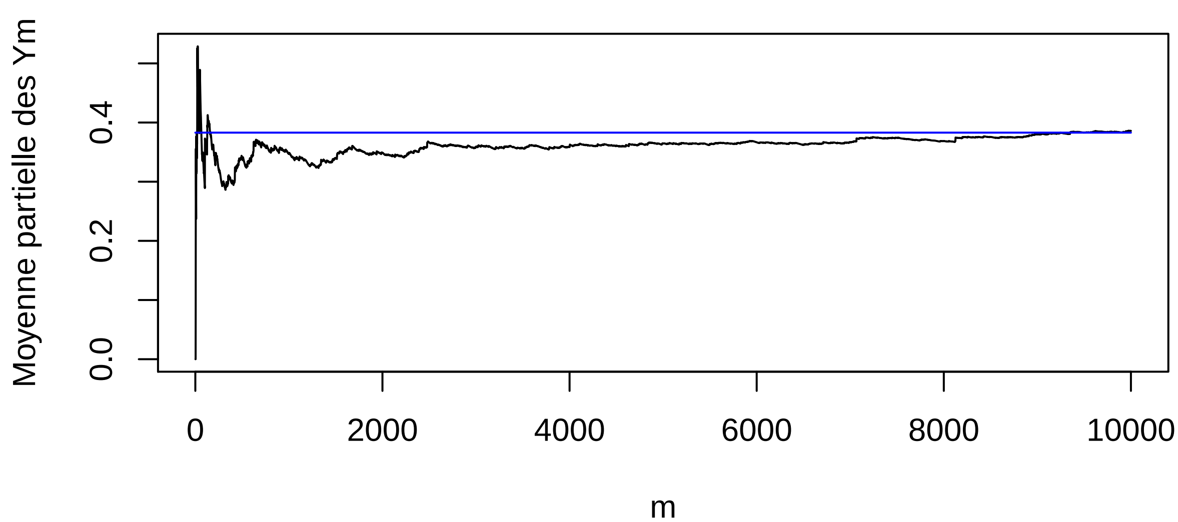
2) Simulation de 10000 réalisations pour $\sigma = 1$ et $K = 1$

```
       Yi                 Zi
 Min.   : 0.0000   Min.   :0.0000
 1st Qu.: 0.0000   1st Qu.:0.0000
 Median : 0.0000   Median :0.4106
 Mean   : 0.3859   Mean   :0.3875
 3rd Qu.: 0.1937   3rd Qu.:0.6951
 Max.   :49.4350   Max.   :0.9795
```

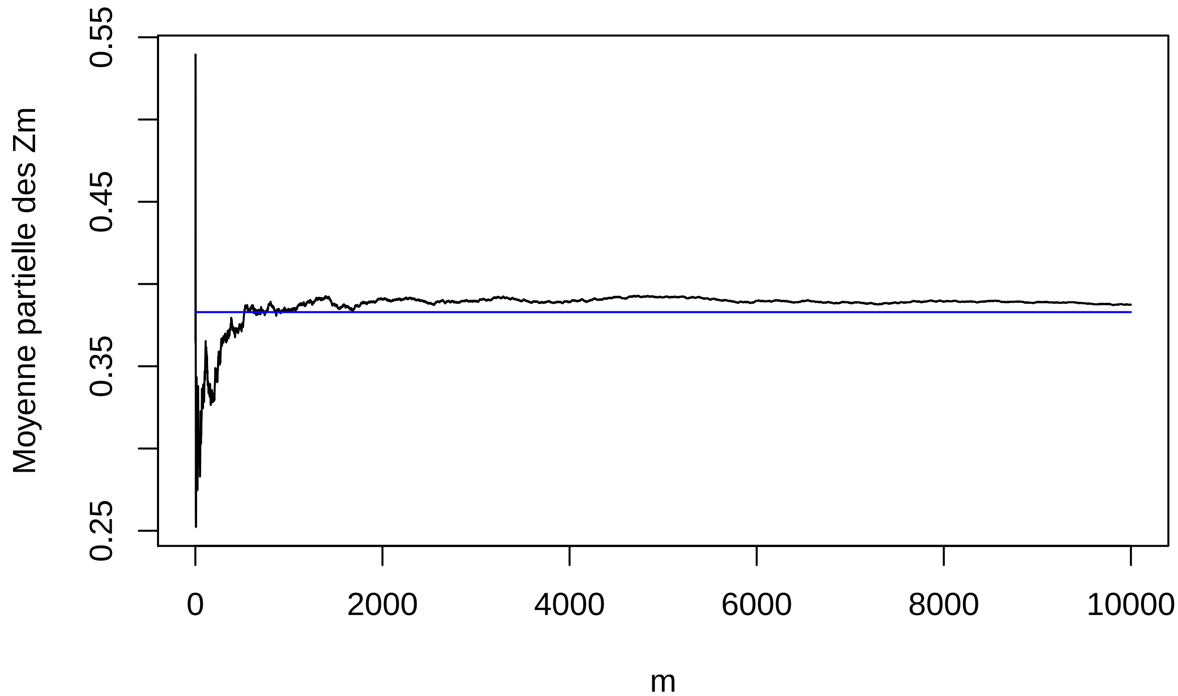
3) Calcul des moyennes partielles $\overline{Y_m}$

```
vecteur_entier = seq(1,10000,length.out = 10000)
moyenne_partielle_Y = cumsum(simulations[,1])/vecteur_entier
```

4) 5) Evolution de $\overline{Y}_m$ en fonction de m et superposition de la valeur calculée du call



En faisant plusieurs fois l'expérience on remarque que les graphes des moyennes partielles sont différents pour des m petits mais qu'ils convergent bien tous vers le Call pour un m assez grand. Ce qui confirme la théorie car, $\overline{Y}_m$ tend vers m d'après la loi des grands nombres. **6) Evolution de $\overline{Z_m}$ en fonction de m et superposition du put P**

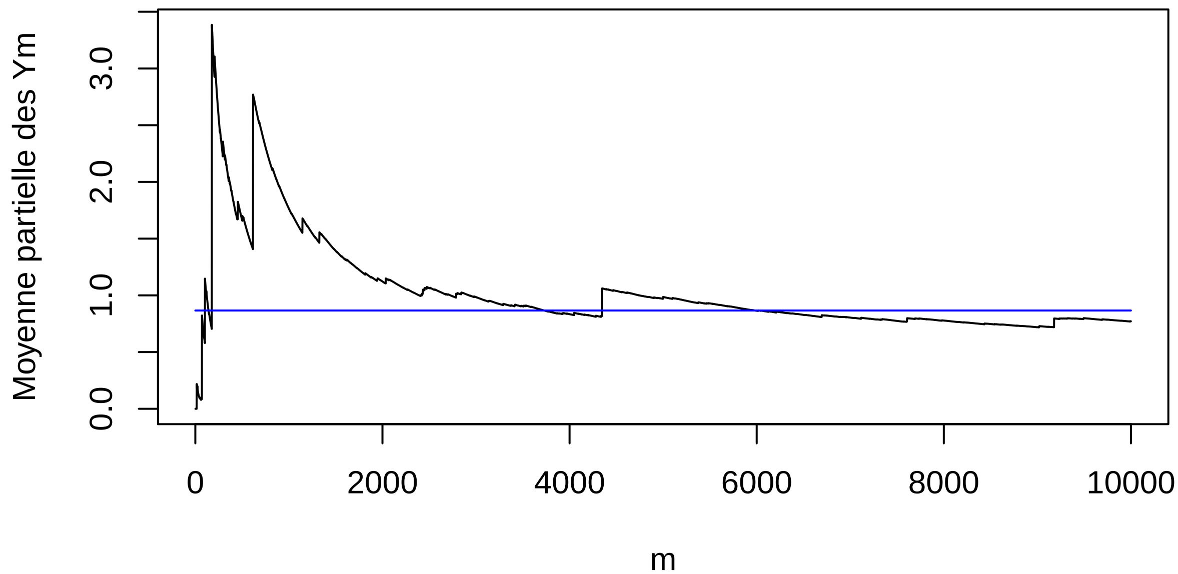


7) 8)
Comme $C - P = S_0e^r - K$, si S0=1 et r=0, On a C=P+1-K, donc il suffit d'estimer l'un d'entre eux.

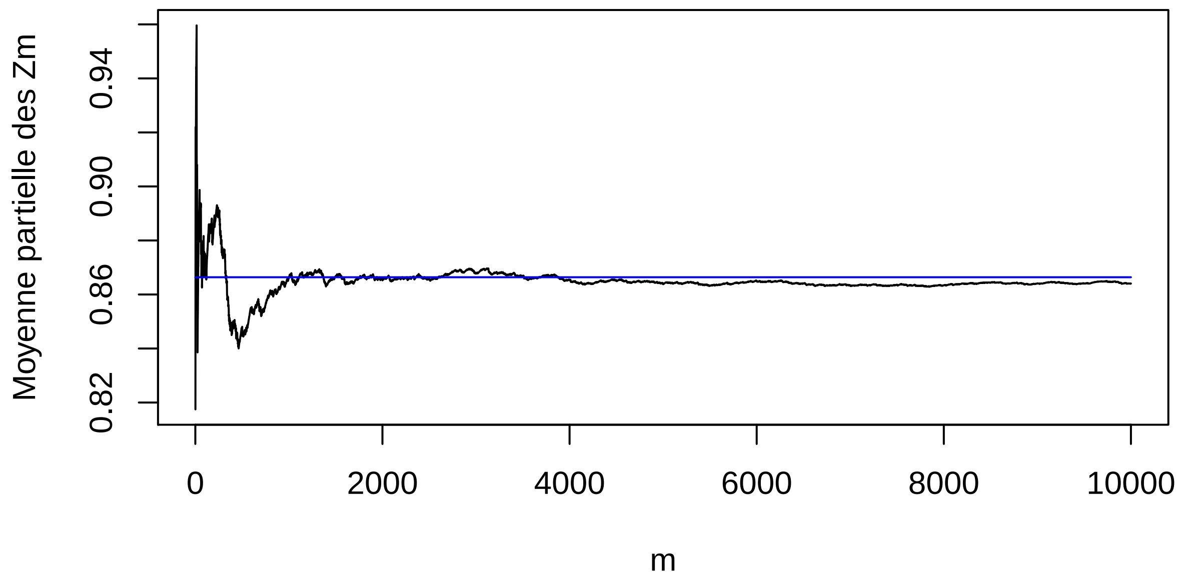
Voici ci-dessous un exemple du code utilisé pour tracer tous les graphiques de cette question :

$\sigma = 3$, $K = 1$

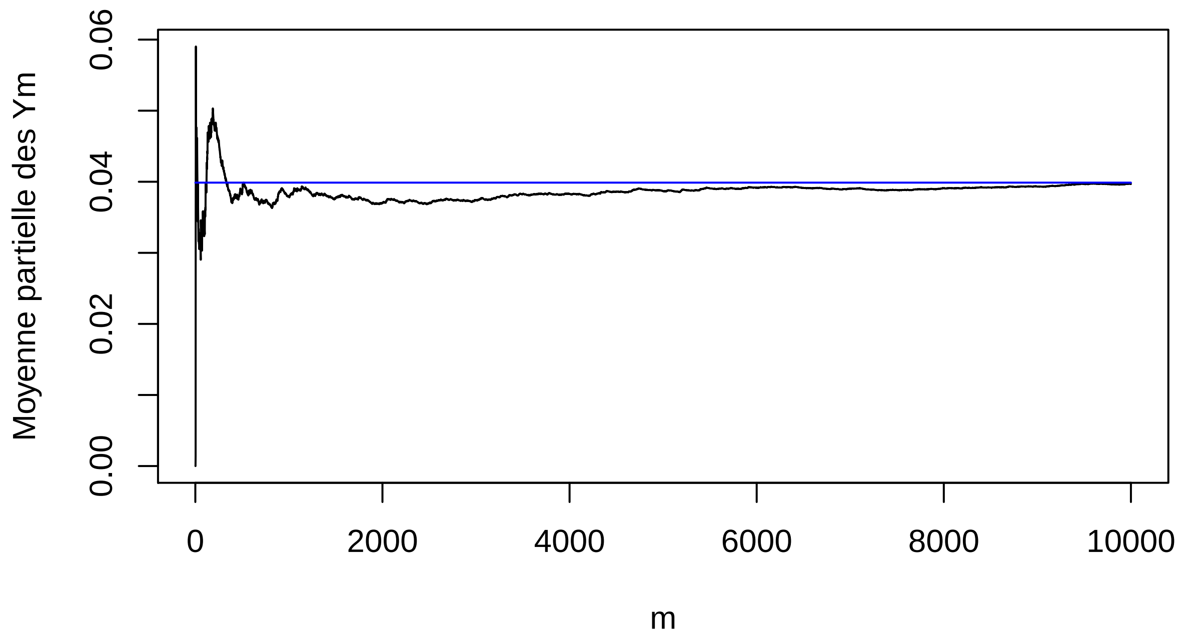
```
sigma = 3 ; K = 1
simulations = rcallput(10000,sigma,K)
moyenne_partielle_Y = cumsum(simulations[,1])/vecteur_entier
plot(vecteur_entier,moyenne_partielle_Y,xlab="m",ylab="Moyenne partielle des Ym",type="l")
points(c(0,10000),c(call(1,0,sigma,K),call(1,0,sigma,K)),type="l",col="blue")
```

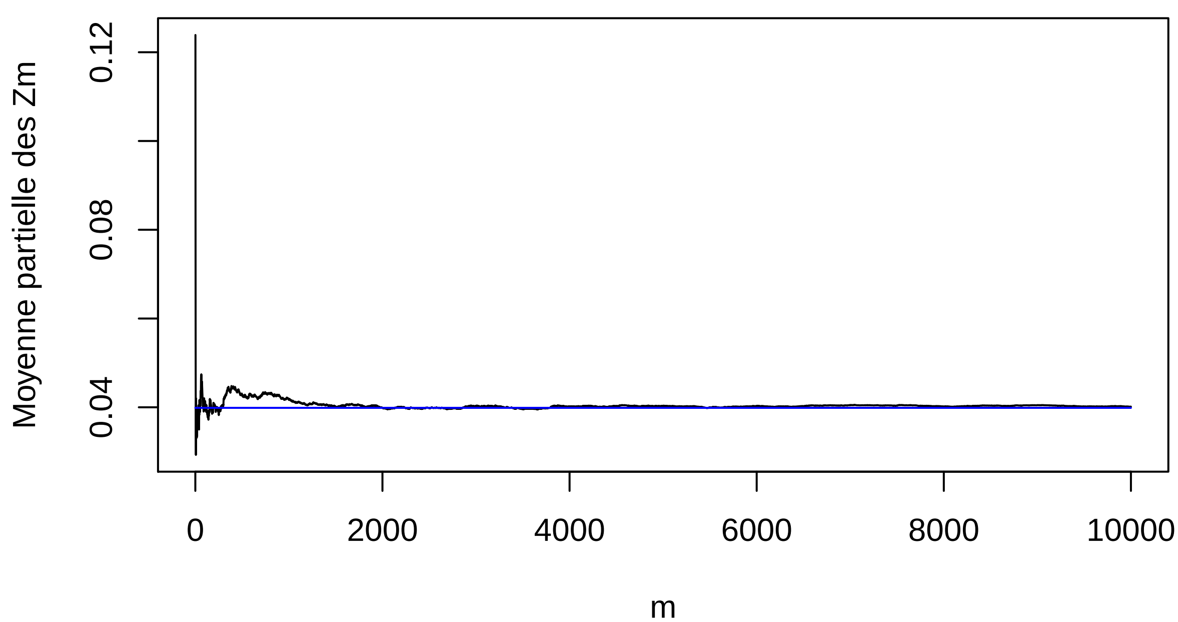



```
moyenne_partielle_Z = cumsum(simulations[,2])/vecteur_entier
plot(vecteur_entier,moyenne_partielle_Z,xlab="m",ylab="Moyenne partielle des Zm",type="l")
points(c(0,10000),c(put(1,0,sigma,K),put(1,0,sigma,K)),type="l",col="blue")
```

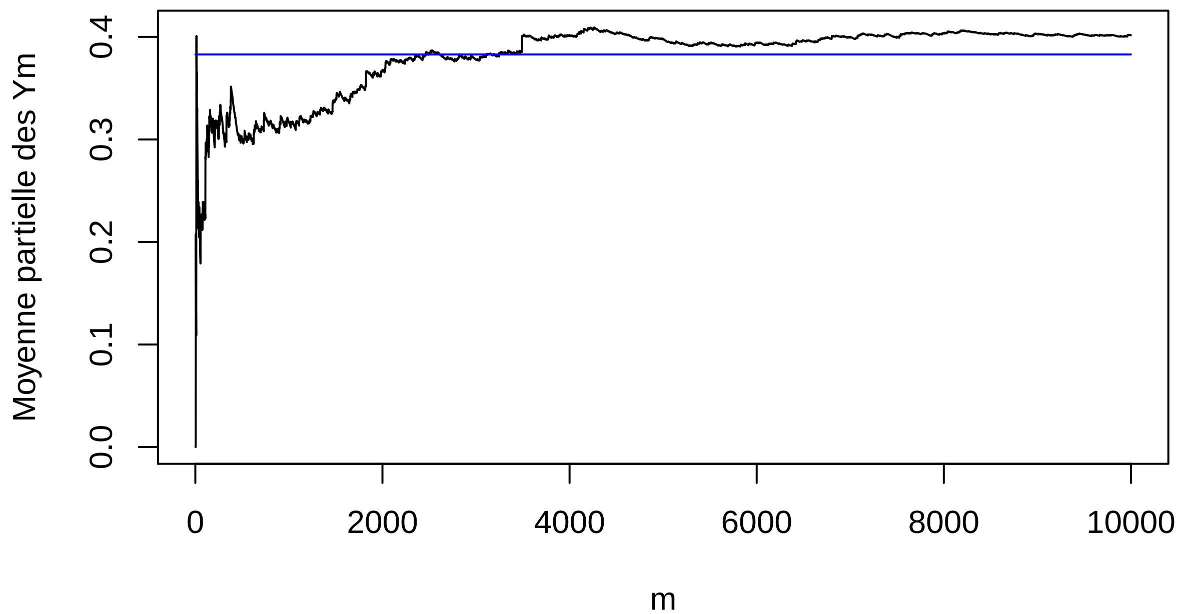


$\sigma = 0.1$, $K = 1$

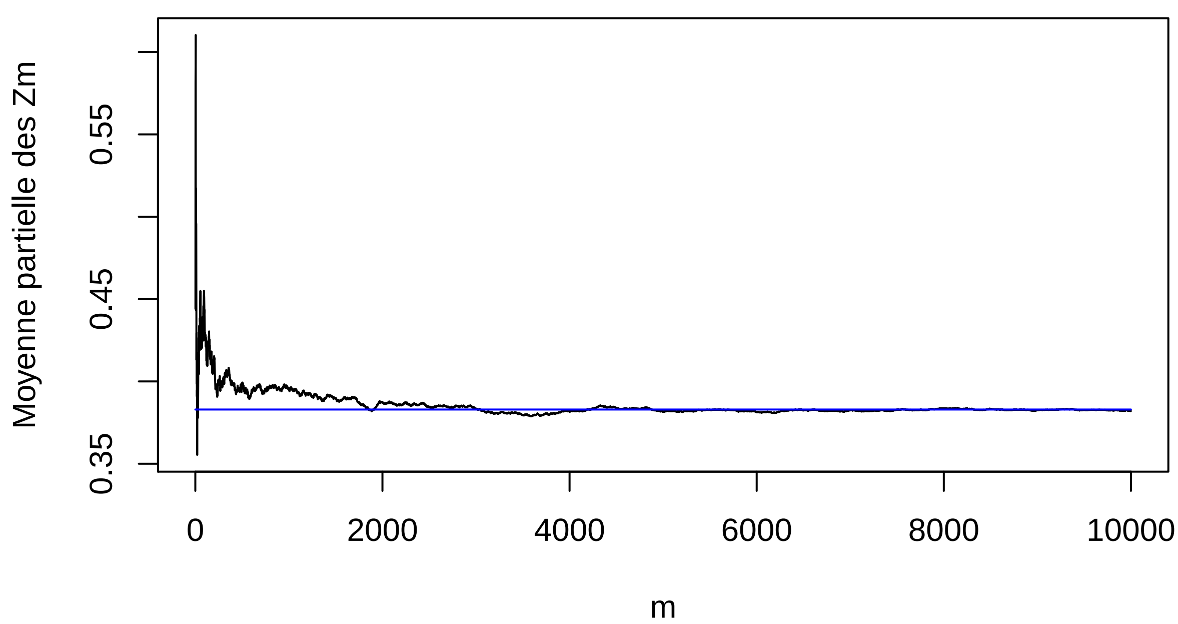

Moyenne partielle des Ym
0.00
0.02
0.04
0.06
0
2000
4000
6000
8000
10000
m


Moyenne partielle des Zm
0.04
0.08
0.12
0
2000
4000
6000
8000
10000
m

$\sigma = 1$, $K = 1$

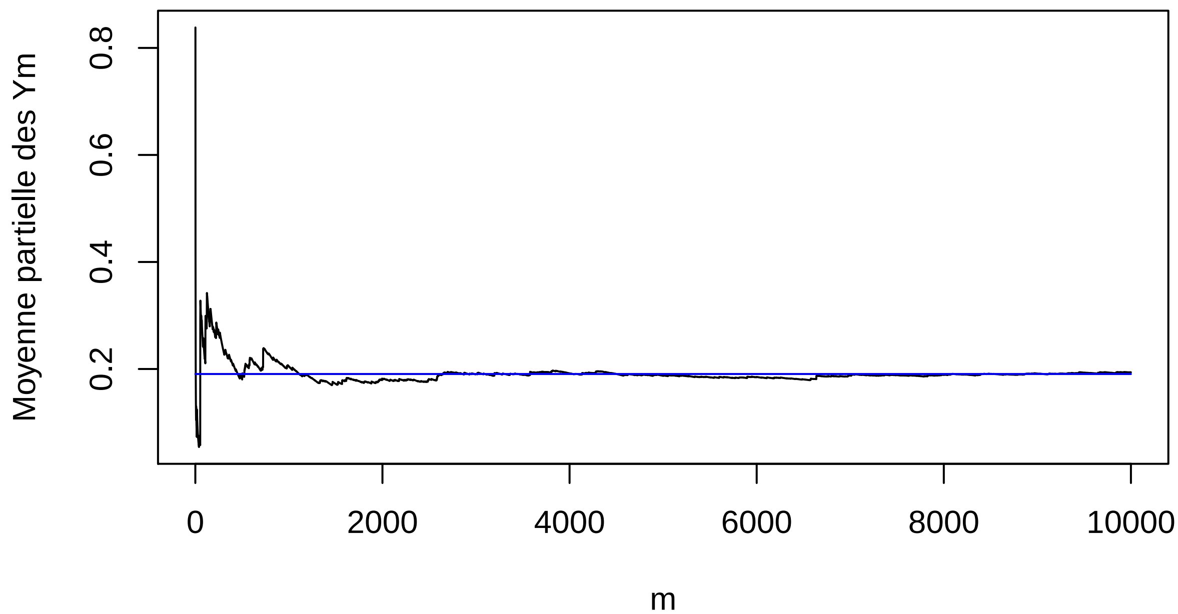


Moyenne partielle des Ym
0.0
0.1
0.2
0.3
0.4
0
2000
4000
6000
8000
10000
m

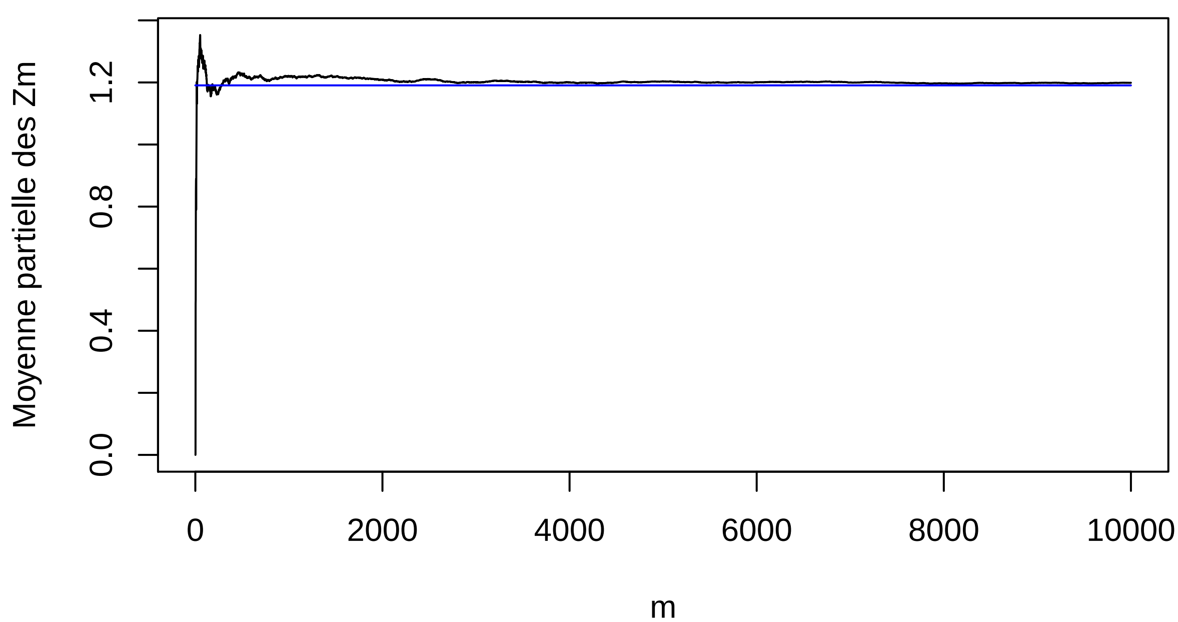


Moyenne partielle des Zm
0.35
0.45
0.55
0
2000
4000
6000
8000
10000
m

$\sigma = 1,\ K = 2$

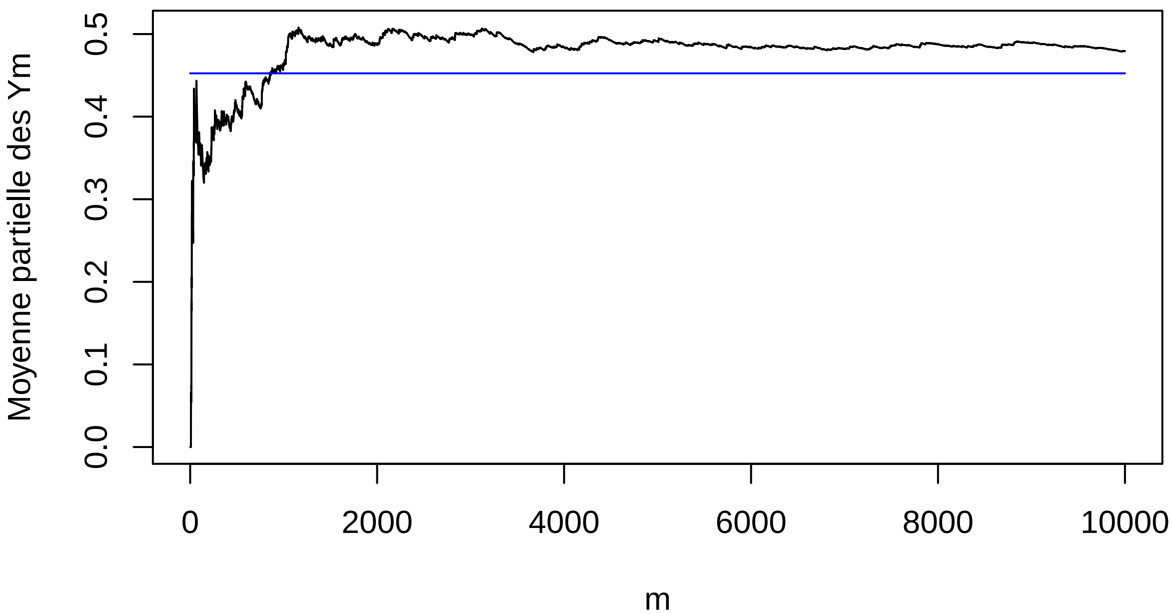


Moyenne partielle des Ym
0.2
0.4
0.6
0.8
0
2000
4000
6000
8000
10000
m

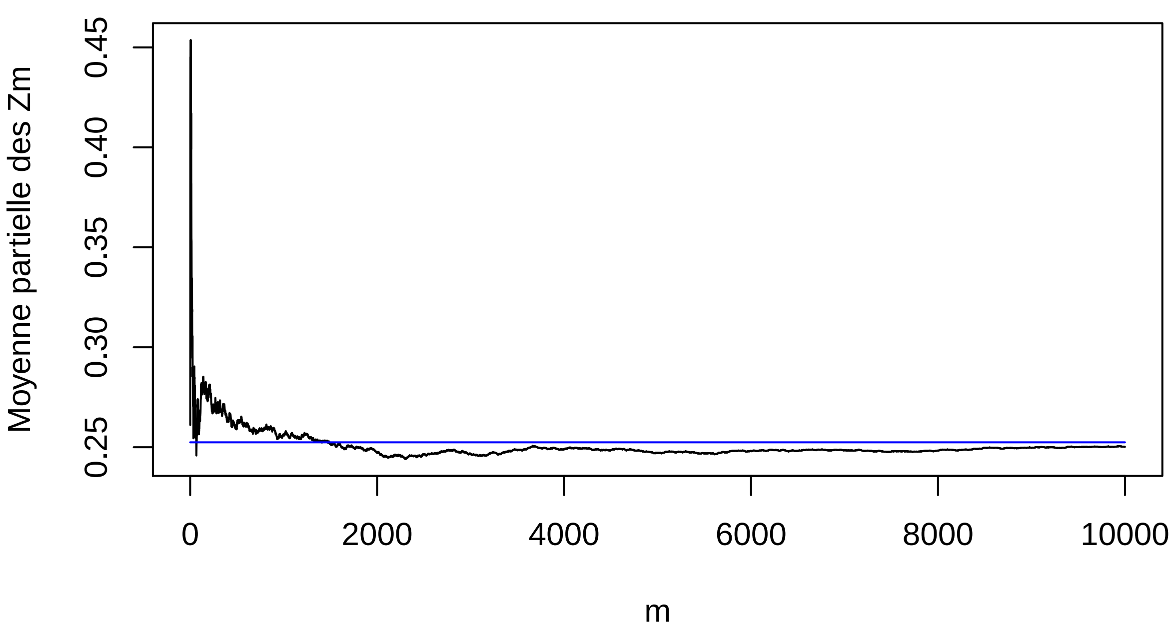


Moyenne partielle des Zm
0.0
0.4
0.8
1.2
0
2000
4000
6000
8000
10000
m

$\sigma = 1$, $K = 0.8$



Moyenne partielle des Ym
0.0
0.1
0.2
0.3
0.4
0.5
0
2000
4000
6000
8000
10000
m



Moyenne partielle des Zm
0.25
0.30
0.35
0.40
0.45
0
2000
4000
6000
8000
10000
m

Le calcule des variances corrigées donne la matrice suivante :

```
     sigma   K        Var_Yi     Var_Zi
[1,]   3.0 1.0 278.86148901 0.07689156
[2,]   1.0 0.8   1.60524986 0.06521430
[3,]   1.0 1.0   1.54731572 0.10880949
[4,]   1.0 2.0   1.02937631 0.37420079
[5,]   0.1 1.0   0.00395001 0.00299150
```

On constate que la variance de Y est supérieure à celle de Z pour chacun des cas.

9)
Dans le cas k=1,$\sigma = 1$,

$$Y = [e^{X-0.5} - 1]_+ = 0 \Rightarrow e^{X-0.5} - 1 \leq 0 \Rightarrow X \leq 0.5$$

$$Z = [1 - e^{X-0.5}]_+ = 0 \Rightarrow 1 - e^{X-0.5} \leq 0 \Rightarrow X \geq 0.5$$

on a $\mathbb{E}[Y] = C$ et $\mathbb{E}[Z] = P$, de plus $C - P = S_0 e^r - K = 0$, donc $\mathbb{E}[Z] = \mathbb{E}[Z] = C$, donc on a:

$$\mathbb{V}ar[Y] = \mathbb{E}[Y^2] - \mathbb{E}^2[Y] = \mathbb{E}[(e^{X-0.5} - 1)^2 \mathbb{I}_{X>=0.5}] - C^2$$

et

$$\mathbb{V}ar[Z] = \mathbb{E}[Z^2] - \mathbb{E}^2[Z] = \mathbb{E}[(1 - e^{X-0.5})^2 \mathbb{I}_{X<=0.5}] - C^2$$

On étudie la valeur de la fonction $(1-e^{X-0.5})^2$, si $X \geq 0.5$, alors $(e^{X-0.5}-1)^2$ prend les valeurs dans $[0, +\infty[$, mais par contre si $X \leq 0.5$, $(1 - e^{X-0.5})^2$ prend les valeurs dans [0,1]. Donc, C'est logique que

$$\mathbb{E}[(1 - e^{X-0.5})^2 \mathbb{I}_{X<=0.5}] \leq \mathbb{E}[(e^{X-0.5} - 1)^2 \mathbb{I}_{X>=0.5}]$$

qui implique que

$$\mathbb{V}ar[Z] \leq \mathbb{V}ar[Y].$$

Exercice 4: Convergence en loi

1) Simulation de m x n vecteurs Y_i

```
m = 1000  ;n = 100 ; sigma = 1 ; r = 0 ; K = 1
donnee=rcallput(m*n,sigma,K)
Y=matrix(donnee[,1],nrow=m,ncol=n)
```

2) Construction du vecteur y des moyennes des lignes de Y

```
y=rowMeans(Y)
```

3) Simulation de Z et construction du vecteur z de ses moyennes par lignes

```
Z=matrix(donnee[,2],nrow=m,ncol=n)
z=rowMeans(Z)
```

4) Représentation des diagrammes quantiles/quantiles

Diagramme de Y

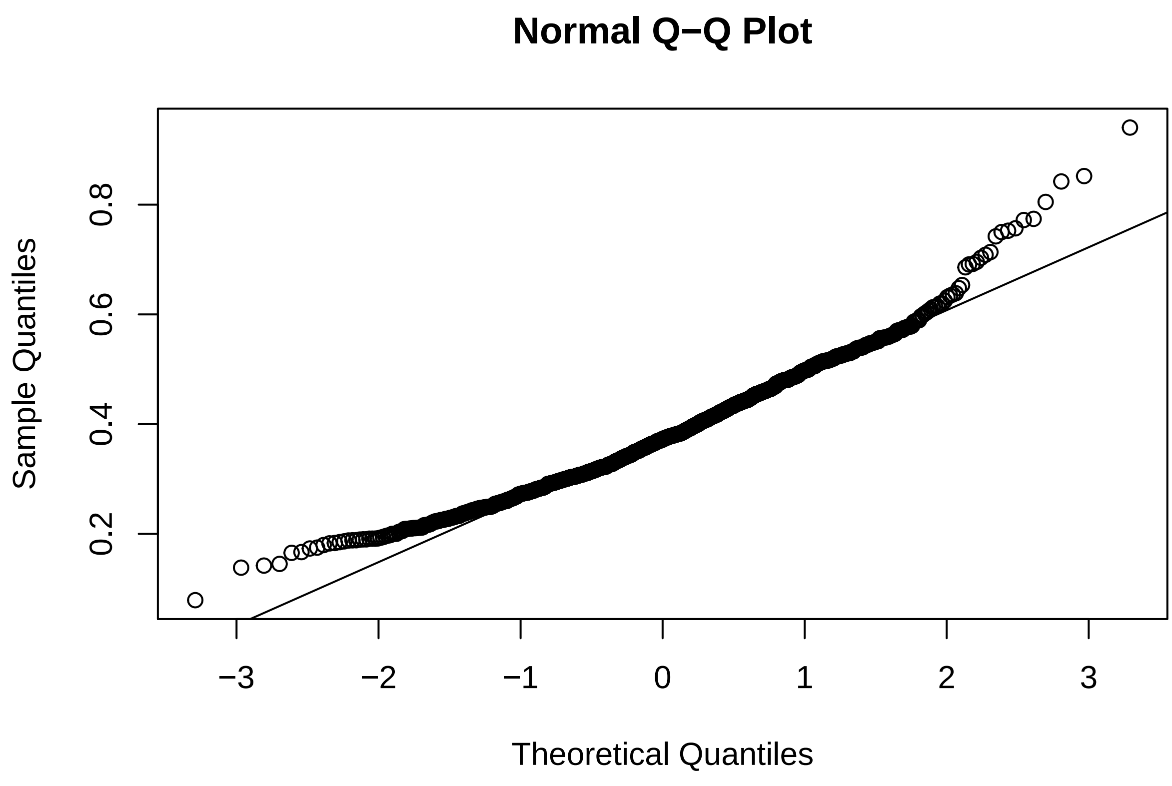
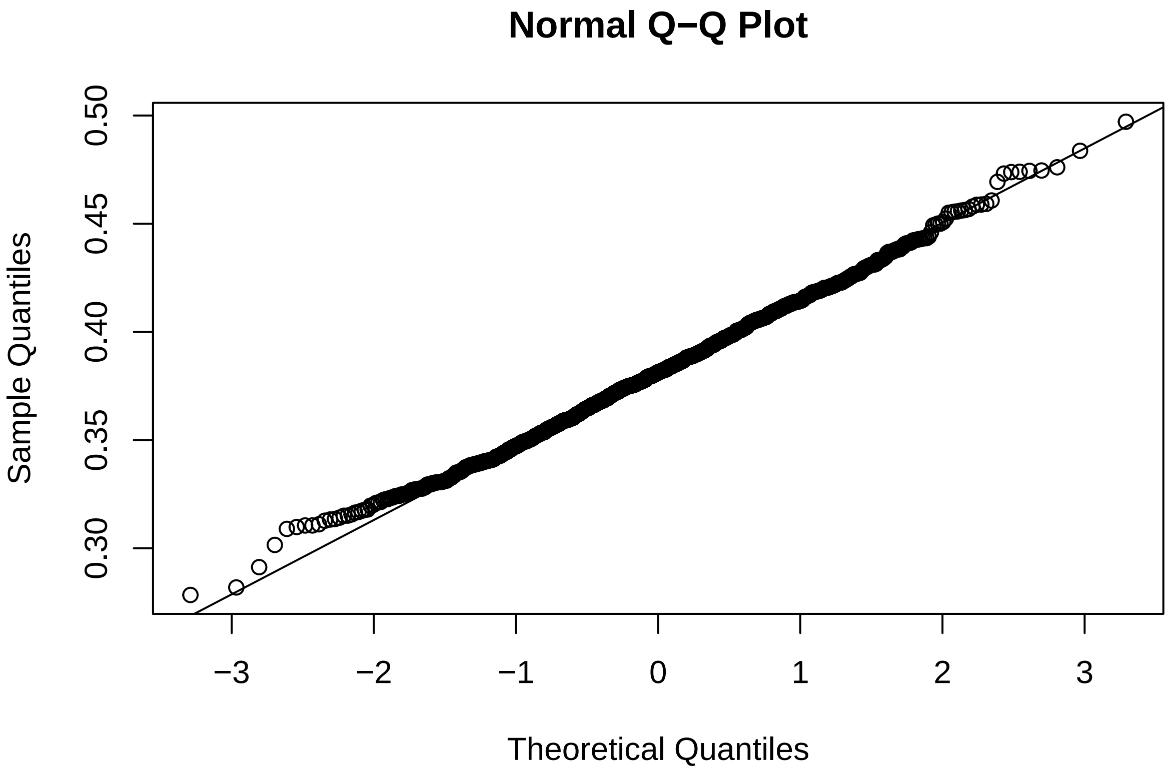
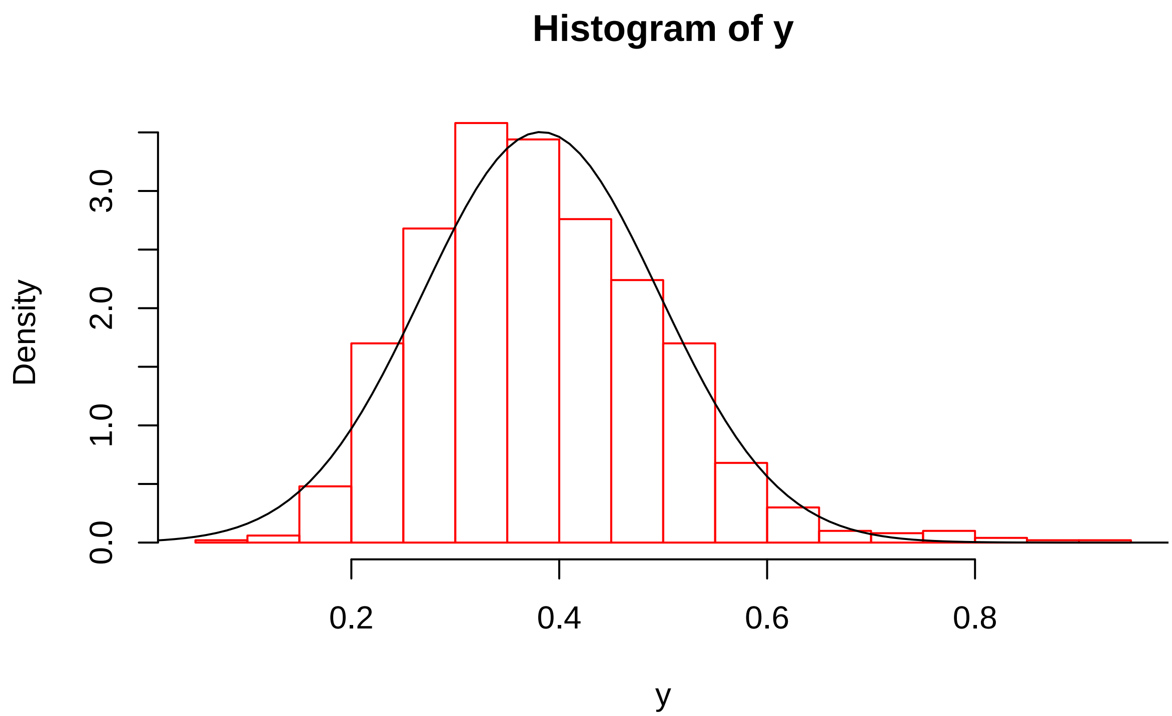


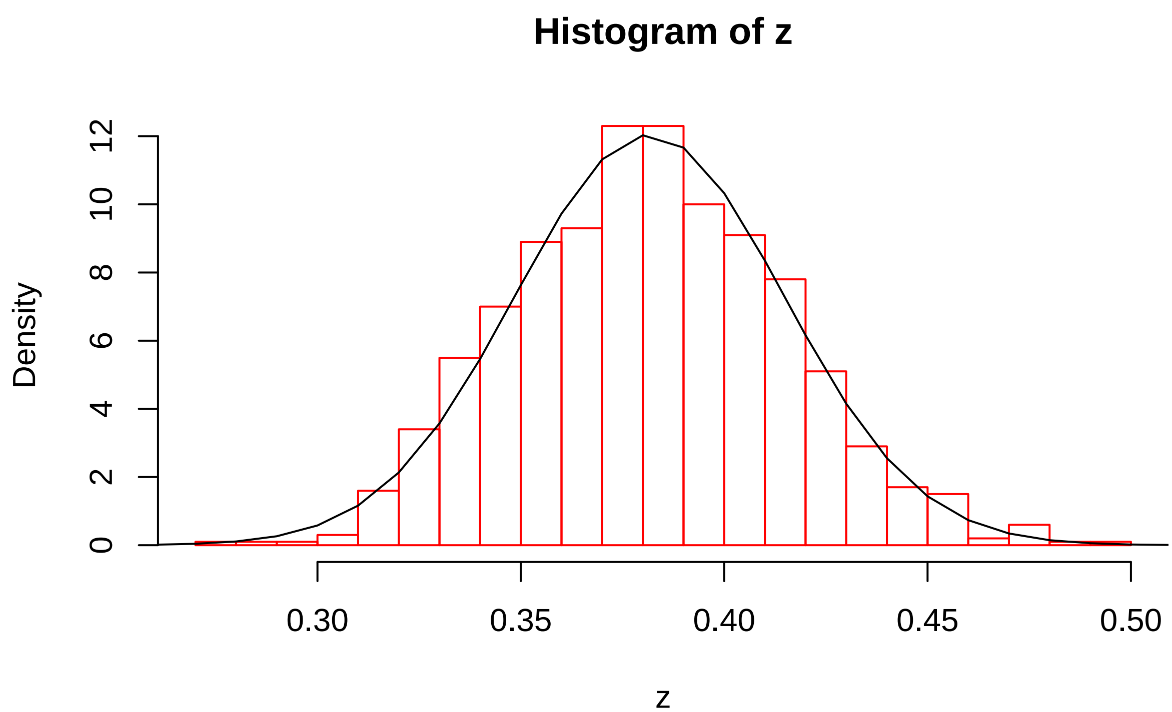
Diagramme de Z



5) 6) Histogramme de y et densité de la loi adaptée.



7) Histogramme de z



8) Intervalle de confiance approché à 95%
On se donne m=10 et n=10 pour IC.

```
m=10;n=10;sigma=1;K=1
donnee=rcallput(m*n,sigma,K)
Y=matrix(donnee[,1],nrow=m,ncol=n)
y=rowMeans(Y)
var_y=sum(diag(var(Y))/10)/10 # var_y est la somme de la variance intra
IC_Y=c(mean(y)-qnorm(0.975)*sqrt(var_y)/sqrt(100),
       mean(y)+qnorm(0.975)*sqrt(var_y)/sqrt(100))
IC_Y
```

```
[1] 0.236037 0.304204
```

```
Z=matrix(donnee[,2],nrow=m,ncol=n)
var_z=sum(diag(var(Z))/10)/10
IC_Z=c(mean(z)-qnorm(0.975)*sqrt(var_z)/sqrt(100),
       mean(z)+qnorm(0.975)*sqrt(var_z)/sqrt(100))
IC_Z
```

```
[1] 0.3610604 0.4022378
```

On sait bien que dans le cas K=1 et $\sigma = 1$, call=put=0.3829249, donc en comparant la longueur de l'IC et la précis de IC, On trouve que l'IC de P est meilleur que l'IC de C, qui vérifie bien le résultat de l'exo 3!

Exercice 5: **Méthode de Monte-Carlo et réduction de la variance**

1)

```
I <- function(x){
  n=length(x)
  IC1=mean(x)-qnorm(0.975)*sqrt(var(x))/sqrt(n)
  IC2=mean(x)+qnorm(0.975)*sqrt(var(x))/sqrt(n)
  return(c(IC1,IC2))
}
```

2)

```
r = 0 ; S = 1 ;K = 1 ;sigma = 1
n = 1000
echantillon = rcallput(n,sigma,K)[,1]
I(echantillon)
```

```
[1] 0.3107948 0.4316297
```

3)

```
echantillon_controle=1-K+rcallput(n,sigma,K)[,2]
I(echantillon_controle)
```

```
[1] 0.3540130 0.3947727
```

4)

```
echantillon_prefer<- function(m,sigma,K){
  V=rexp(n,0.5)
  s1=K-exp(sigma*sqrt(V)-sigma^2/2)
  s2=K-exp(-sigma*sqrt(V)-sigma^2/2)
  P=(s1*(s1>=0)+s2*(s2>=0))/sqrt(2*pi*V)
  return(P)
}
I(echantillon_prefer(1000,1,1))
```

```
[1] 0.3193813 0.5011908
```

On constate que l'intervalle de confiance n'est pas meilleure que celui trouvé dans la question 2), donc on va rechercher l'autre méthode pour réduire la variance.

5) Etude de la monotonie de $f(x) = [k - e^{\sigma x - \sigma^2/2}]_+$

La fonction $f(x) = [k - e^{\sigma x - \sigma^2/2}]_+$ est décroissante.
6)
On pose T(X)=-X et si X~$\mathcal{N}(0,1)$, alors T(X)=-X~$\mathcal{N}(0,1)$.
7)
Comme f(x) est une fonction monotone et T est une fonction décroissante tq T(X) suit mêmê loi que X, on repose un estimateur

$$e_2 = \frac{1}{2n}\sum_{i=1}^{n}(f(X_i) + f(T(X_i)))$$

il est meilleur que $e_1 = \frac{1}{2n}\sum_{i=1}^{2n} f(X_i)$. Car la variance de e_2 est plus petite que celle de e_1.
8)

```
echantillon_anti <-function(n,sigma,K){
  X=rnorm(n,0,1)
  Y_brut=exp(sigma*X-sigma^2/2)-K #On calcule exp(sigma*X-sigma^2/2)-K
  Y_brut2=exp(-sigma*X-sigma^2/2)-K #On calcule exp(-sigma*X-sigma^2/2)-K
  Y1=-Y_brut*(Y_brut<=0)  #On calccule Y_brut+
  Y2=-Y_brut2*(Y_brut2<=0) #On calccule Y_brut2+
  Y=c(Y1,Y2)
  return(c(Y,Y))
}
I(echantillon_anti(1000,1,1))
```

```
[1] 0.3733073 0.3936102
```

On trouve cet IC est meilleur que l'acien parce que sa longueur est réduite.

Rem: en utilisant la methode de stratification, on peut obtenir un meilleur IC pour P!

```
echantillon_strati<- function(n,m,sigma,K){
  X=rnorm(n*m,0,1)
  Y_brut=exp(sigma*X-sigma^2/2)-K #On calcule exp(sigma*X-sigma^2/2)-K
  Y_brut2=exp(-sigma*X-sigma^2/2)-K #On calcule exp(-sigma*X-sigma^2/2)-K
  Y1=-Y_brut*(Y_brut<=0)
  Y2=-Y_brut2*(Y_brut2<=0)
  Y_n=c(Y1,Y2)
  Y=matrix(Y_n,nrow=m)  #On range les donnee dans matrice Y
  return(Y)
}
IC_strati<- function(Y){
  m=dim(Y)[2]
  n=dim(Y)[1]
  y=rowMeans(Y)
  var_y=sum(diag(var(Y))/m)/n  #on calcule la variance intra
  IC_Y=c(mean(y)-qnorm(0.975)*sqrt(var_y)/sqrt(m*n),
       mean(y)+qnorm(0.975)*sqrt(var_y)/sqrt(m*n))
  return(IC_Y)
}
Y=echantillon_strati(100,10,1,1)
IC_strati(Y)
```

```
[1] 0.3793241 0.3884985
```

Remarque:Résumé

On va faire un tableau de résumé pour comparer la qualité de notre résultat.
On estime C par 1000 échantillons dans le cas $\sigma = 1$ et $K = 1$. On pose $X_i \sim \mathcal{N}(0,1)$ et $V_i \sim \mathcal{E}xp(0.5)$. On rappele les estimateurs:
l'estimateur original:

$$e_1 = \frac{1}{1000}\sum_{i=1}^{1000}[e^{X_i-0.5}-1]_+$$

la méthode de la variance contrôle:

$$e_2 = \frac{1}{1000}\sum_{i=1}^{1000}[1-e^{X_i-0.5}]_+$$

la méthode de la variance contrôle améliorée par l'échantillonage préférentiel:

$$e_3 = \frac{1}{1000}\sum_{i=1}^{1000}\frac{[1-e^{\sqrt{V_i}-0.5}]_+ + [1-e^{-\sqrt{V_i}-0.5}]_+}{\sqrt{2\pi V_i}}$$

la méthode de la variance contrôle améliorée par variable antithétique:

$$e_4 = \frac{1}{2000}\sum_{i=1}^{1000}([1-e^{X_i-0.5}]_+ + [1-e^{-X_i-0.5}]_+)$$

la méthode de la variance contrôle améliorée par variable antithétique et la méthode de stratification:

$$e_5 = \sum_{i=1}^{10}\frac{1}{10}\left(\frac{1}{200}\sum_{j=1}^{100}([1-e^{X_i-0.5}]_+ + [1-e^{-X_i-0.5}]_+)\right)$$

```
table_resume<- matrix(nrow=5,ncol=3)
e_1=rcallput(1000,1,1)[,1];table_resume[1,1:2]=I(e_1)
e_2=rcallput(1000,1,1)[,2];table_resume[2,1:2]=I(e_2)
e_3=echantillon_prefer(1000,1,1);table_resume[3,1:2]=I(e_3)
e_4=echantillon_anti(1000,1,1);table_resume[4,1:2]=I(e_4)
e_5=echantillon_strati(100,10,1,1);table_resume[5,1:2]=IC_strati(e_5)
table_resume[,3]=table_resume[,2]-table_resume[,1]
colnames(table_resume)=c("IC-","IC+","length of IC")
rownames(table_resume)=c("e_1","e_2","e_3","e_4","e_5")
table_resume
```

```
          IC-       IC+ length of IC
e_1 0.3291076 0.4999454   0.17083774
e_2 0.3651861 0.4057380   0.04055183
e_3 0.3433738 0.3880193   0.04464554
e_4 0.3716257 0.3919889   0.02036316
e_5 0.3787214 0.3878388   0.00911735
```

On voit que la troisieme colonne est la longueur de l'intervalle confiance qui implique la qualité de notre estimateur. On a vu que la méthode de la variance contrôle améliorée par variable antithétique et la méthode de stratification nous donne un meilleur estimateur.