TP 4 : Méthode de Box-Muller, risque quadratique

Ce TP n°4 aborde la méthode de Box-Muller vu en TD (exercice 1) et la notion de risque quadratique associé à un estimateur (exercice 2).

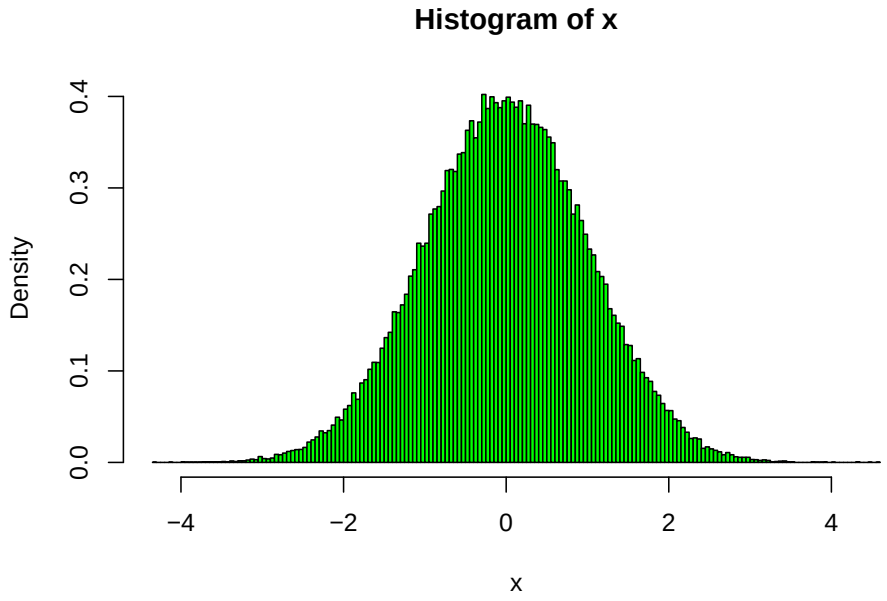
Exercice 1 : Méthode de Box-Muller en pratique, et digression.

1. Rappelons que d'après la méthode de la fonction de répartition, si $U \sim \mathcal{U}([0,1])$, alors $-2\log(U) \sim \mathcal{E}(1/2)$ et $2\pi U \sim \mathcal{U}([0,2\pi])$.

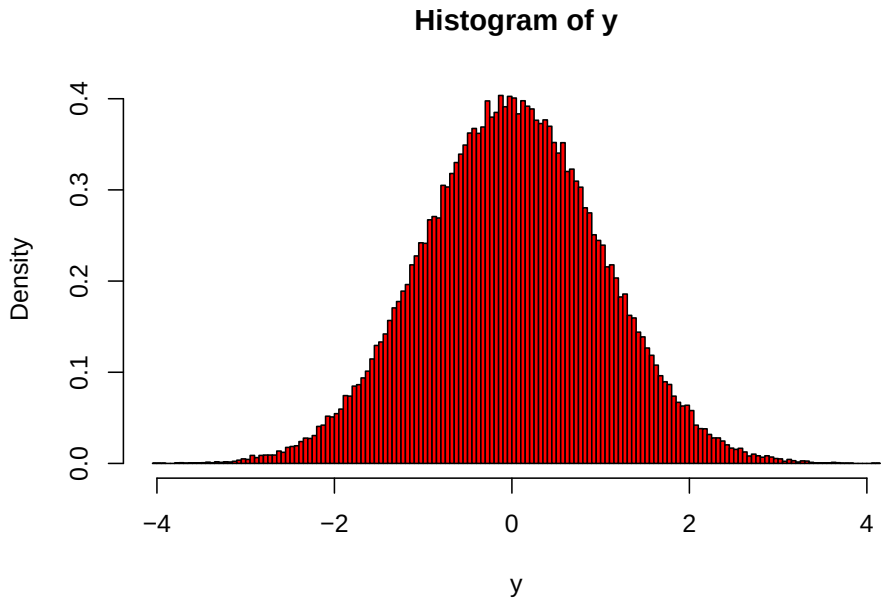
```
n = 100000
u = runif(n)
r2 = -2*log(1-u) # méthode d'inversion de la fonction de répartition
r = sqrt(r2) # on prend la racine carré pour avoir r

## plus directement r2 = rexp(n,rate=1/2)

u = runif(n)
theta = 2*pi*u

# Attention, il faut prendre garde à simuler theta et r avec des "u" DIFFERENTS
# (c'est a dire des variables indépendantes), sinon les deux variables sont dépendantes et
# la méthode ne fonctionne pas.

x = r*cos(theta)
y = r*sin(theta)
```

2. Voici le code pour comparer les histogrammes :

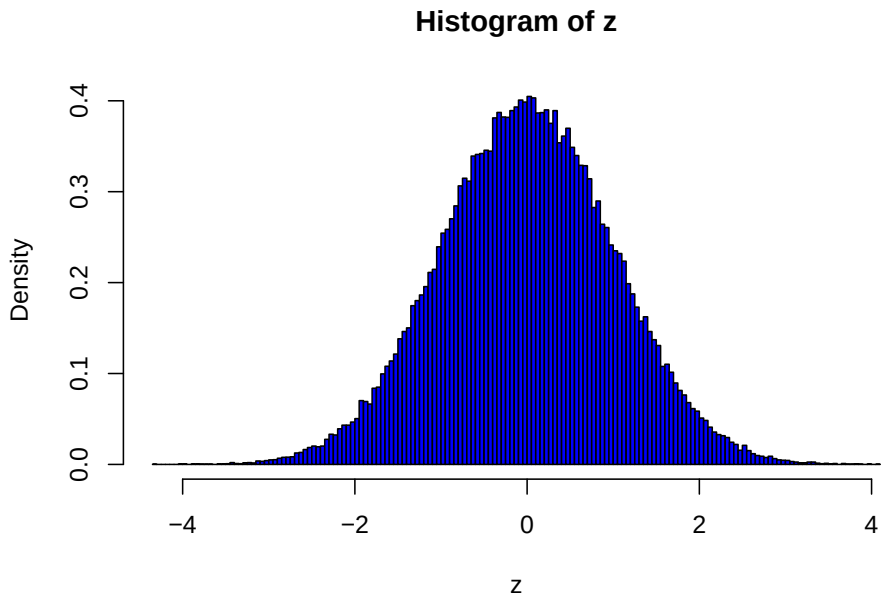
```
z = rnorm(n)

hist(x,freq=F,breaks=150,col="green") #on va vérifier que c'est un histogramme gaussien
```




```r
hist(y,freq=F,breaks=150,col="red") #on va vérifier aussi c'est un histogramme gaussien
```




```r
hist(z,freq=F,breaks=150,col="blue") #on l'utilise comme référence
```



On observe bien sûr que ce sont (quasiment) les mêmes, ce qui montre que la méthode de Box-Muller marche bien.

3. Le code donné dans l'énoncé simule et représente des vecteurs de dimension 3 dont chaque coordonnée est une gaussienne indépendantes, le tout renormalisé de sorte que le vecteur soit de norme 1. On observe que les points sont sur la sphère unité de $\mathbb{R}^3$ (logique), et qu'ils semblent se répartir uniformément sur celle-ci. La loi du vecteur en question semble donc être uniforme sur la sphère de $\mathbb{R}^3$ de rayon 1 et centre 0.

Exercice 2 : Estimation et risque quadratique

1. Comme on sait (on ou sait retrouver rapidement) que $\mathbb{E}[X] = a/2$ et $\mathrm{Var}(X) = a^2/12$, on propose logiquement les estimateurs

$$\hat{a}_1 := 2\bar{X}_n \quad \text{et } \hat{a}_2 = \sqrt{12\tilde{S}_n^2}.$$

2. Voici le code pour renvoyer le biais, variance et risque estimé du premier estimateur.

```
R1 <- function(a,n,N){
  estims = replicate(N,0) # on initialise le vecteur des a1
  for (i in 1:N){
    X = runif(n,min = 0, max = a) # on simule un échantillon de taille n de loi U([0,a])
    estims[i] = 2*mean(X) # on calcule l'estimateur de a, ici a1, pour l'échantillon numéro i
  }
  biais = mean(estims-a) # le biais, c'est la moyenne de l'écart
  variance = var(estims) # le variance, c'est la variance de a1
  risque = mean((estims-a)^2) # le risque, c'est la moyenne de l'écart QUADRATIQUE
  c(biais, variance, risque) # on retourne le vecteur des trois valeurs
}
```

```
R1(a=2,n=2*10^3, N=5*10^3)
```

```
## [1] 0.0002492450 0.0006699996 0.0006699277
```

Voici le code pour renvoyer aussi le biais, variance et risque du deuxième estimateur.

```
R12 <- function(a,n,N){
  estims1 = replicate(N,0) # on initialise
```

```
  estims2 = replicate(N,0) # on initialise
  for (i in 1:N){
    X = runif(n,min = 0, max = a) # on simule un échantillon de taille n de loi U([0,a])
    estims1[i] = 2*mean(X) # on calcule hat(a)_1
    estims2[i] = sqrt(12*var(X)) # on calcule hat(a)_2
  }
  biais1 = mean(estims1-a)
  variance1 = var(estims1)
  risque1 = mean((estims1-a)^2)
  resultat1 = c(biais1, variance1, risque1)

  biais2 = mean(estims2-a)
  variance2 = var(estims2)
  risque2 = mean((estims2-a)^2)
  resultat2 = c(biais2, variance2, risque2)

  rbind(resultat1,resultat2) # on combine les deux vecteurs
}
```

```
R12(a=4,n=2*10^3, N=5*10^3)
```

```
##                    [,1]        [,2]        [,3]
## resultat1 -0.0001514105 0.002663382 0.002662873
## resultat2 -0.0004464980 0.001553678 0.001553566
```

L'estimateur 2 semble être meilleur (risque plus faible).

3. On sait d'après le cours que "risque = biais^2 + variance".

4. Voici le code adapté pour les trois estimateurs.

```
R123 <- function(a,n,N){
  estims1 = replicate(N,0) # on initialise
  estims2 = replicate(N,0) # on initialise
  estims3 = replicate(N,0) # on initialise
  for (i in 1:N){
    X = runif(n,min = 0, max = a) # on simule un échantillon de taille n de loi U([0,a])
    estims1[i] = 2*mean(X) # on calcule hat(a)_1
    estims2[i] = sqrt(12*var(X)) # on calcule hat(a)_2
    estims3[i] = max(X) # on calcule hat(a)_3
  }
  biais1 = mean(estims1-a)
  variance1 = var(estims1)
  risque1 = mean((estims1-a)^2)
  v1 = c(biais1, variance1, risque1)
  biais2 = mean(estims2-a)
  variance2 = var(estims2)
  risque2 = mean((estims2-a)^2)
  v2 = c(biais2, variance2, risque2)
  biais3 = mean(estims3-a)
  variance3 = var(estims3)
  risque3 = mean((estims3-a)^2)
  v3 = c(biais3, variance3, risque3)
```

```
  rbind(v1,v2,v3) # on combine les trois vecteurs
}
```

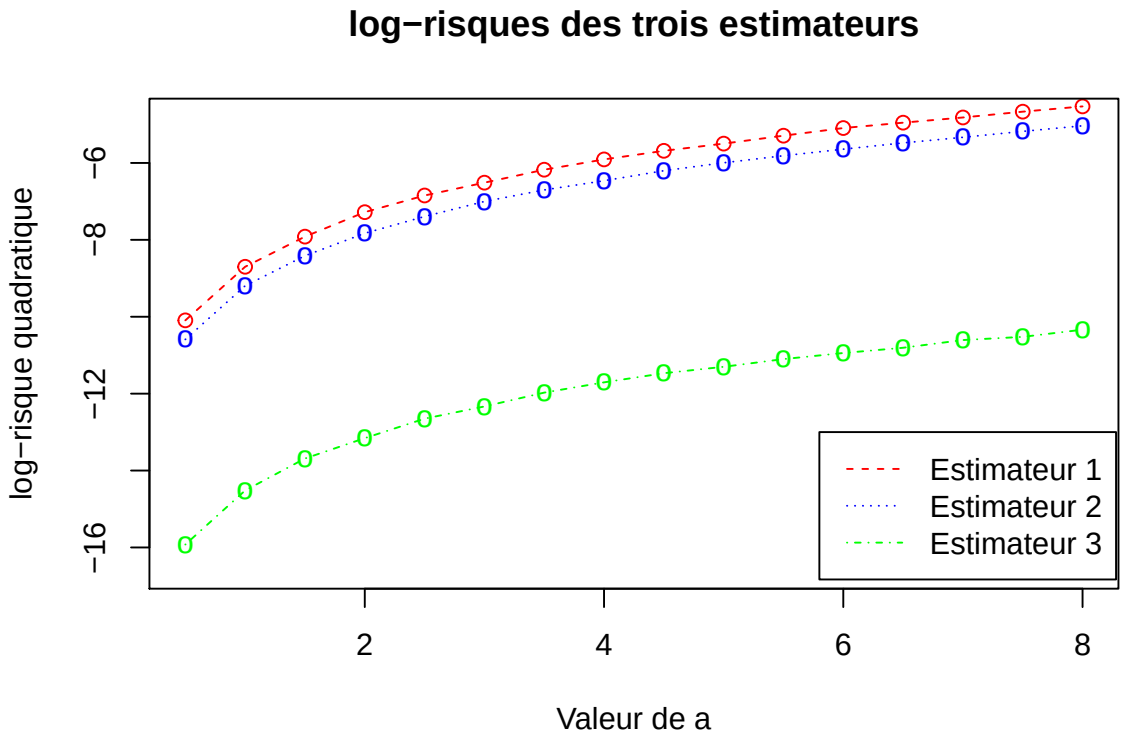
```
R123(a=4,n=2*10^3, N=5*10^3)
```

```
##             [,1]         [,2]         [,3]
## v1  4.694859e-05 2.615264e-03 2.614743e-03
## v2 -1.078458e-03 1.631441e-03 1.632278e-03
## v3 -2.014479e-03 4.053464e-06 8.110778e-06
```

C'est le troisième estimateur qui semble être le meilleur, avec peut-être un biais un peu plus important, mais une variance bien plus faible ! C'est ce qui explique que le risque quadratique est bien meilleur (cf relation de la question 3).

5. Voici le code complété :

```
x = seq(0.5,8,by=0.5) # on définit les abscisses valeurs de a
taille = length(x)
risques1 = replicate(taille,0) # on initialise
risques2 = replicate(taille,0) # on initialise
risques3 = replicate(taille,0) # on initialise

for (i in 1:taille){
  res = R123(a=x[i],n=2*10^3, N=5*10^3) # on fait appel à R123
  #on stocke les variables qui nous intéressent :
  risques1[i] = res[1,3]
  risques2[i] = res[2,3]
  risques3[i] = res[3,3]
}

# puis, on réalise le plot
plot(x,log(risques1),type="o",col="red",xlab = "Valeur de a",
     ylab = "log-risque quadratique",main = "log-risques des trois estimateurs",
     lty = 2, ylim=c(-16.6,-4.8))
points(x, log(risques2), col="blue",pch="o")
lines(x, log(risques2), col="blue", lty = 3)
points(x, log(risques3), col="green",pch="o")
lines(x, log(risques3), col="green", lty = 4)
legend(5.8,-13,legend=c("Estimateur 1","Estimateur 2","Estimateur 3"),
       col=c("red", "blue","green"),lty=2:4)
```



Le code met un petit moment à tourner, mais on visualise bien que le meilleur estimateur est le numéro 3 (avec le maximum). Les courbes sont parrallèles en échelle logarithmique, ce qui indique que les trois risques sont reliés par un facteur constant (typiquement d'ordre $\exp(6) \sim 400$ entre l'estimateur 3 et les autres !). On a donc la preuve que l'estimateur du maximum est bien plus performant que les estimateurs construits avec les estimateurs des moments. Ce résultat est en général valable pour d'autres cas, comme nous le verrons peut-être dans la suite.

remarque : l'option `lty` *dans les plots permet d'utiliser différents types de lignes (pointillés, ligne brisée, etc).*