

TP 1 : Simulation, visualisation de données avec R

Ce TP n°1 a pour but la maîtrise des outils classiques de simulation probabiliste et de représentation de données.

Exercice 0 : Reprise en main de la console : calculs rapides

Exercice 1 : Loi normale, commandes classiques pour les lois continues

1. On simule l'échantillon demandé :

```
n=10000  
ech=rnorm(n,mean=-2,sd=3)
```

On calcule la moyenne et la variance empirique :

```
mean(ech)
```

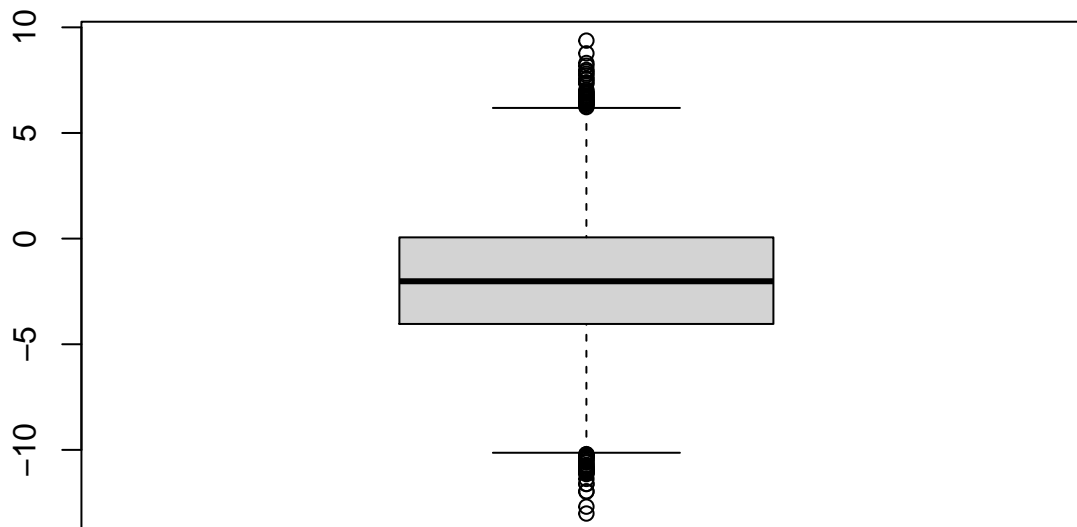
```
## [1] -1.993034
```

```
var(ech)
```

```
## [1] 9.185794
```

On affiche le boxplot :

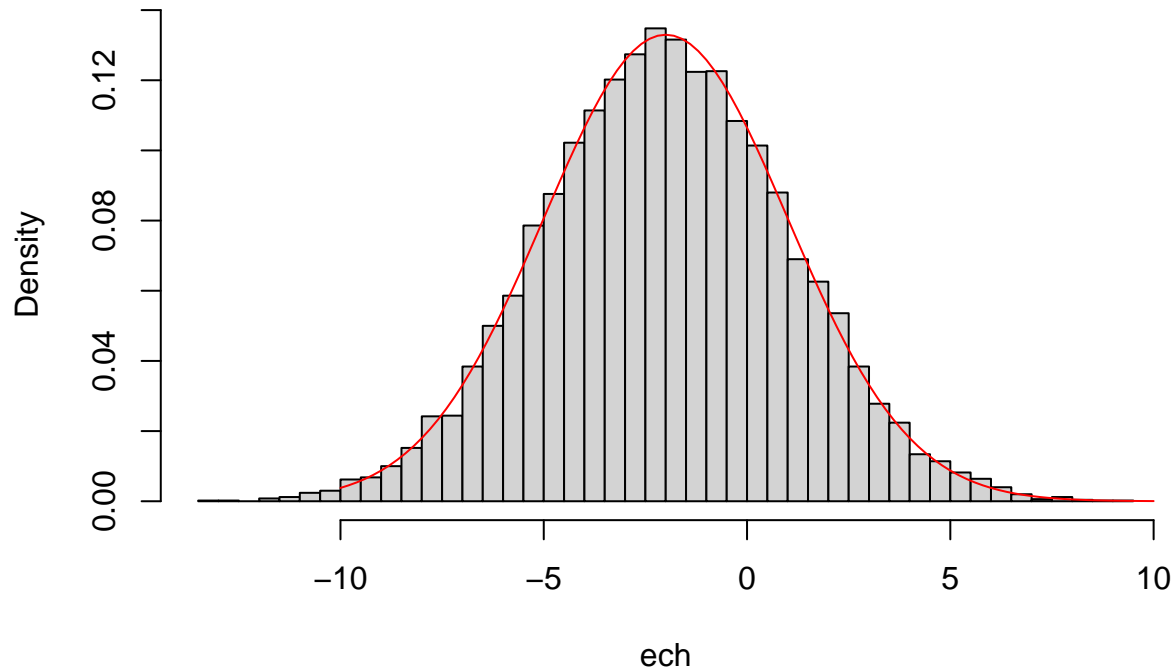
```
boxplot(ech)
```



2. On trace l'histogramme, on superpose la courbe :

```
hist(ech,freq=FALSE,breaks=50)  
curve(dnorm(x,mean = -2,sd = 3),add=TRUE,from=-10, to = 10, col="red")
```

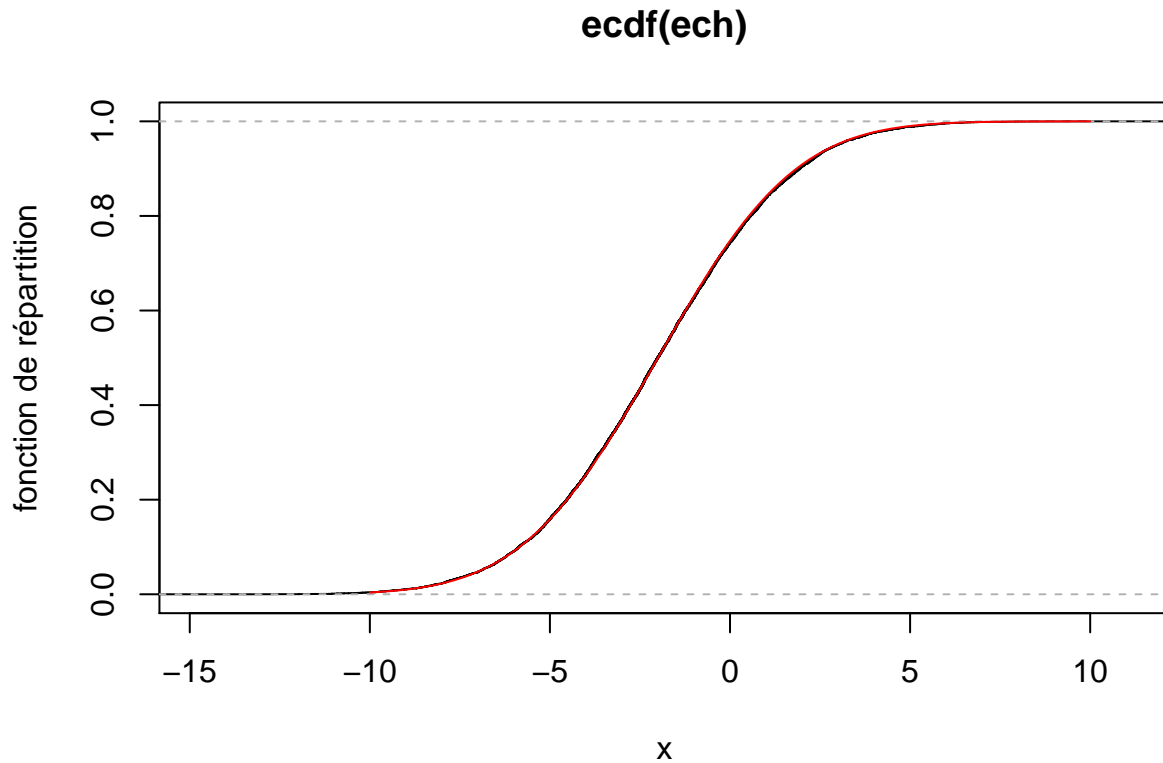
Histogram of ech



3. Réponse : L'histogramme a tendance à se confondre avec la courbe de la densité théorique.

4. Voici le code :

```
plot(ecdf(ech),ylab="fonction de répartition")  
curve(pnorm(x,mean = -2,sd = 3),from = -10,to = 10,add=T,col="red")
```

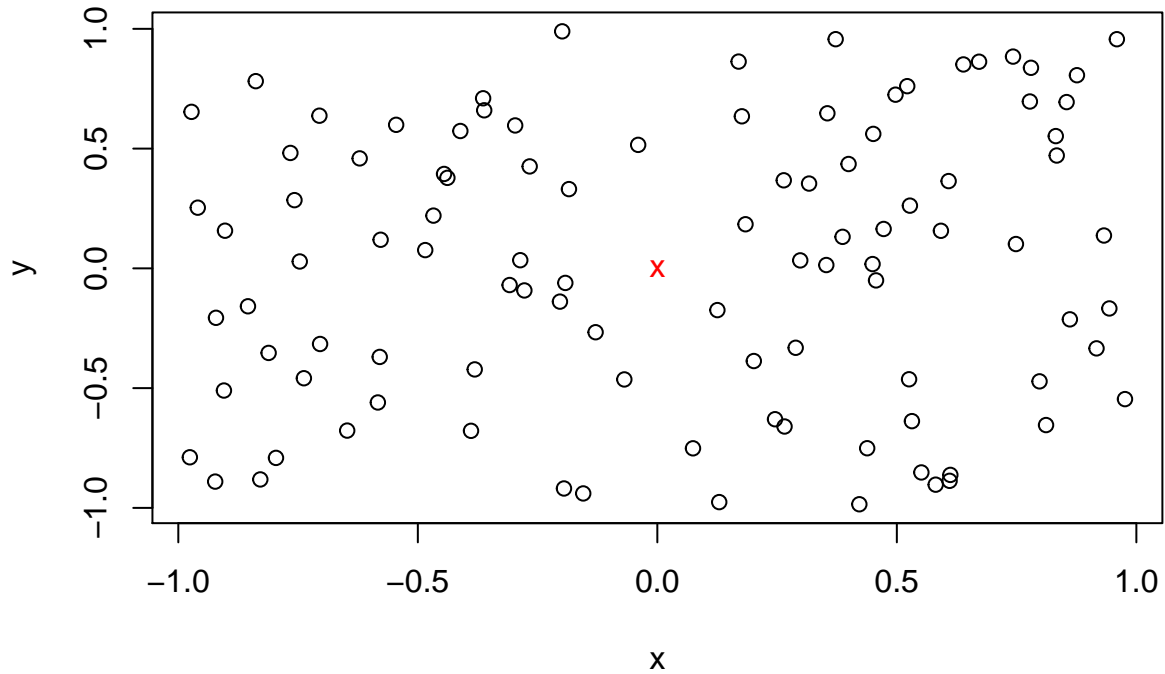


Réponse : La fonction de répartition empirique a tendance à se confondre avec la fonction de répartition théorique.

Exercice 2 : Lançons des fléchettes !

1. Ce n'était pas demandé, mais on peut également superposer un point enrouge qui représente le centre de la cible (avec la fonction `points`).

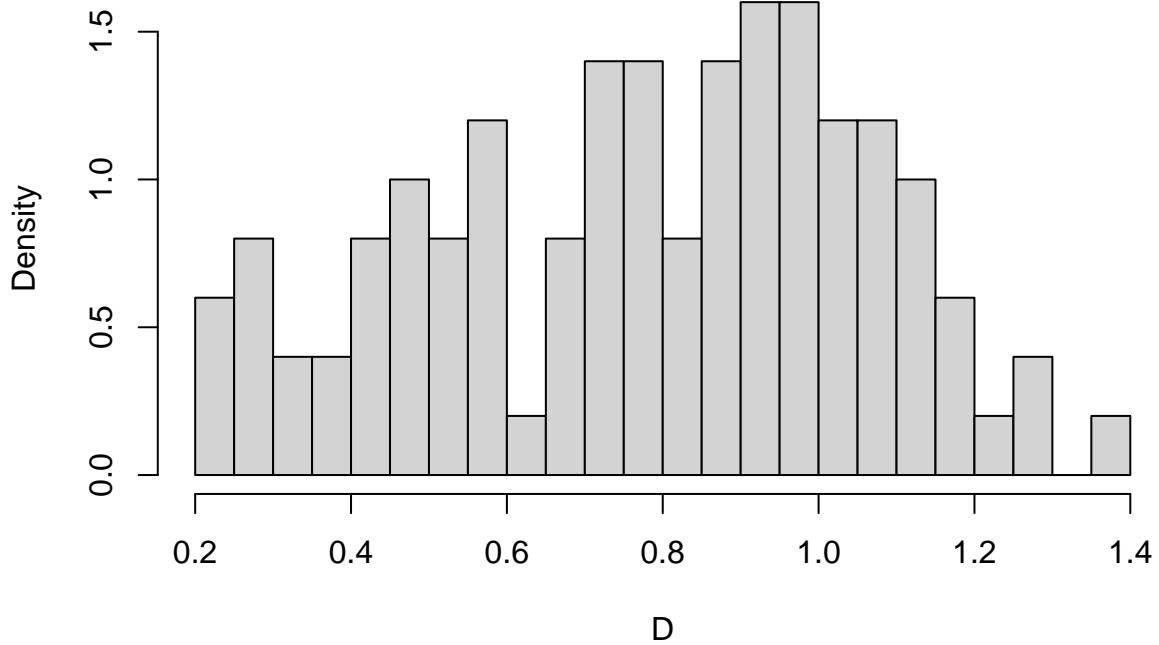
```
n=100
x = runif(n,-1,1)
y = runif(n,-1,1)
plot(x,y)
points(0,0,col='red',pch='x')
```



2. On trace l'histogramme des réalisations de D :

```
D = sqrt(x^2 + y^2) #et oui, on peut faire les calculs directement avec les vecteurs !!
hist(D, breaks=20, freq=F)
```

Histogram of D



affiche la moyenne de la réalisation :

```
mean(D)
```

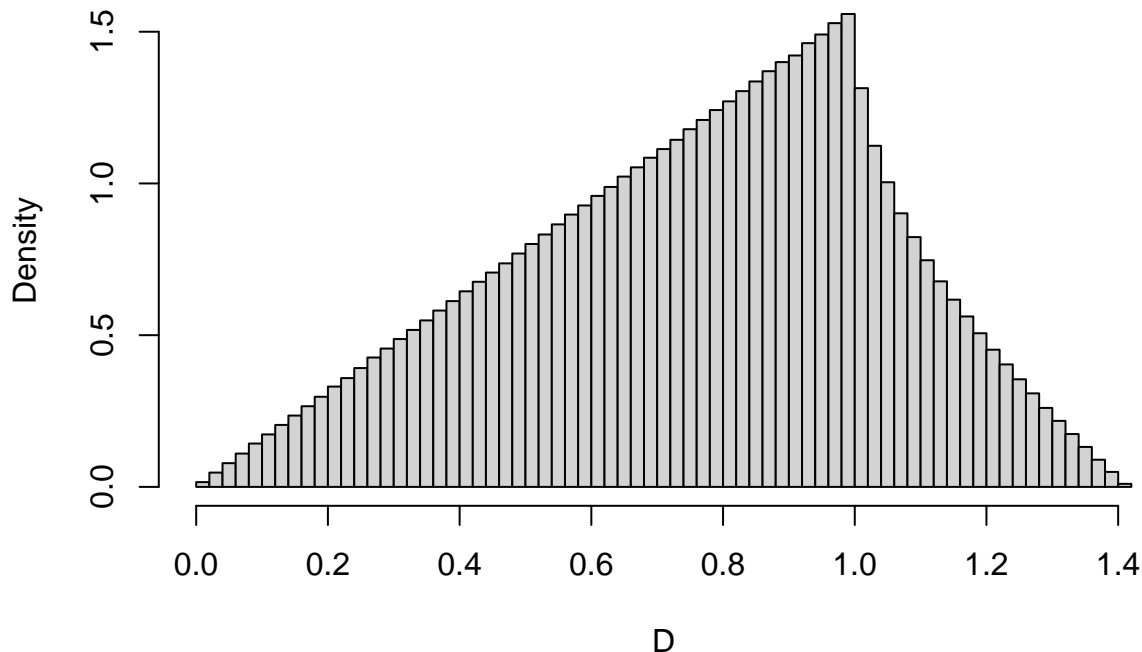
```
## [1] 0.7795755
```

On

3. Voici le nouvel histogramme de D pour $n = 10^7$

```
n=10^7
D= sqrt(runif(n,-1,1)^2 + runif(n,-1,1)^2)
hist(D, breaks=100,freq=F)
```

Histogram of D



Réponse : L'histogramme semble indiquer que D a une loi de densité linéaire avant la valeur 1, puis la densité s'effondre entre 1 et $\sqrt{2} \sim 1.4$. Cela peut s'interpréter en regardant le lieu géométrique des points vérifiant $D = x$: ce sont des cercles concentriques pour $x \leq 1$, puis ensuite ce sont des intersections de cercles plus grand et du carré.

4. On sélectionne les éléments qui vérifient $D < 1$ et on calcule la fréquence de tels éléments.

```
length(which(D<1))/n
```

```
## [1] 0.7854374
```

En utilisant les données simulées, on trouve environ 0.78...

Réponse : On calcule d'abord avec R la fraction des fléchettes vérifiant $D \leq 1$, cela nous donne une bonne estimation de la probabilité cherchée. Pour le calcul exact, il suffit de voir que l'on cherche la probabilité de tomber dans le cercle inscrit (de rayon 1) dans le carré de côté 2 qui sert de cible. La loi étant uniforme sur le carré, la probabilité cherchée n'est autre que le rapport des aires.

```
pi/4
```

```
## [1] 0.7853982
```

5. On calcule la moyenne et on représente l'histogramme de réalisations indépendantes de D sous le nouveau modèle.

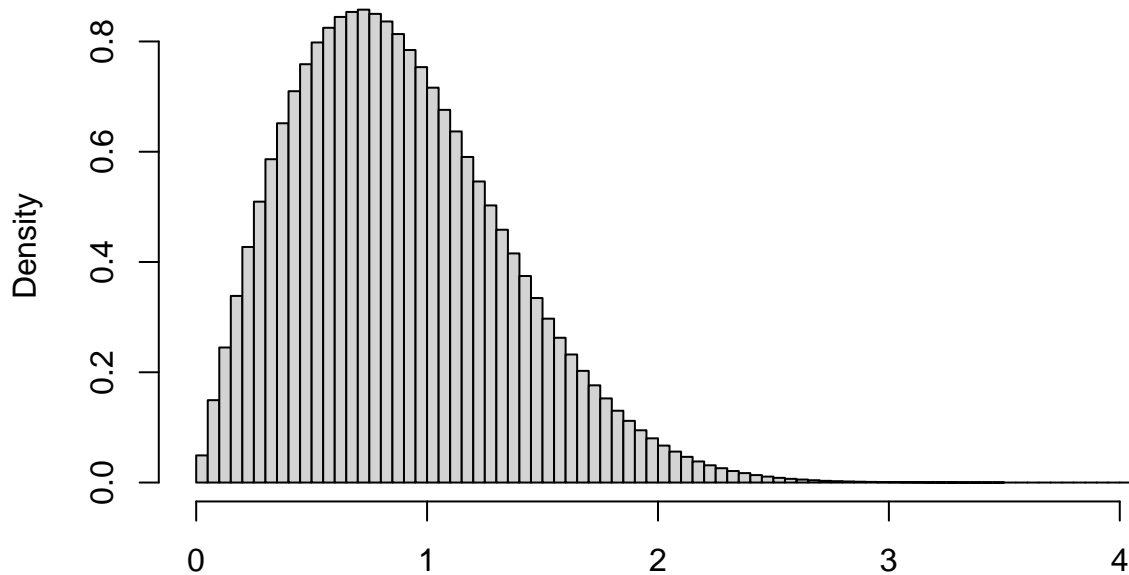
```
n=10^7
x = rnorm(n,0,1.0/sqrt(2))
```

```

y = rnorm(n,0,1.0/sqrt(2))
D2= sqrt(x^2 + y^2)
hist(D2, breaks=100,freq=F)

```

Histogram of D2



D2

Calculons

les statistiques demandées.

```
mean(D) #moyenne du premier joueur
```

```
## [1] 0.7652338
```

```
var(D) #variance du premier joueur
```

```
## [1] 0.08111191
```

```
mean(D2) #moyenne du deuxième joueur
```

```
## [1] 0.8862316
```

```
var(D2) #variance du deuxième joueur
```

```
## [1] 0.2145541
```

Réponse : Au vu des précédents résultats, le deuxième joueur à un D moyen plus grand que le premier joueur, avec une variance plus élevée. On peut donc dire qu'il est moins bon en moyenne que le premier joueur.

Exercice 3 : Promenons-nous sur $\mathbb{Z} \dots$

1. Voici un code optimisé pour la simulation d'une marche.

```
T=30 #le nombre de pas de la marche
```

```
p=0.65 # la proba d'aller à droite
```

```
Y = runif(T) #un échantillon de T variables uniformes dans [0,1]
```

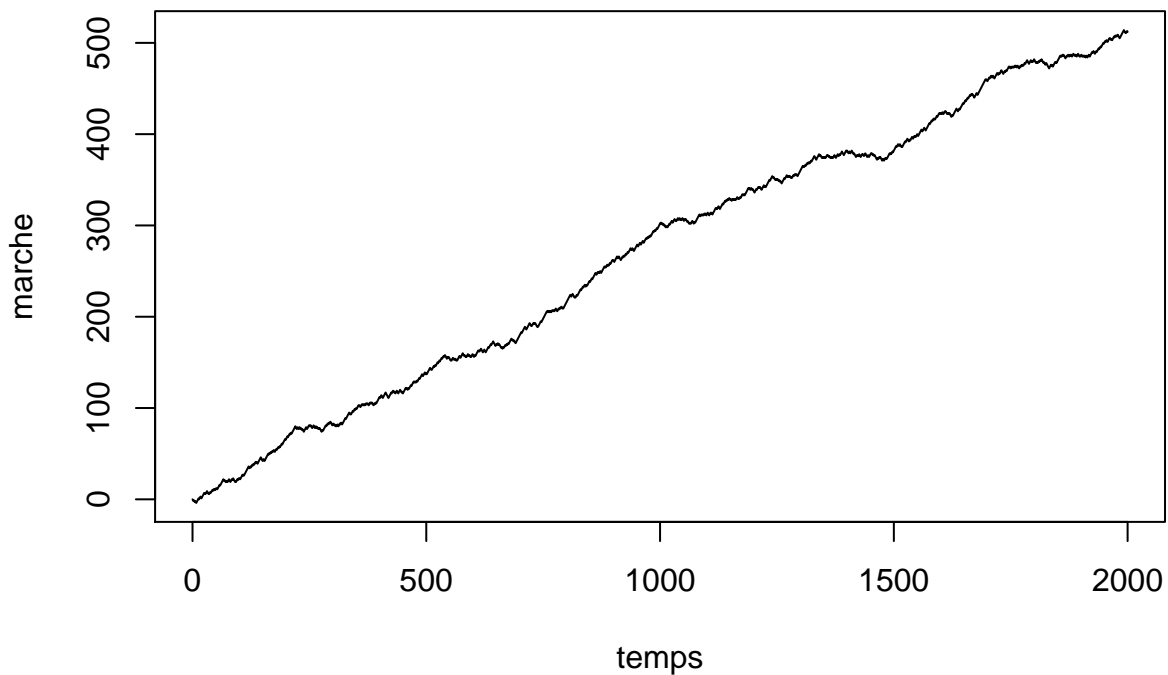
```
X = 2*(Y<p)-1 # si Y[i]<p, X[i] vaudra 2*1-1 = 1, et si Y[i]>p, X[i] vaudra 2*0-1 = -1
```

```
marche = c(0,cumsum(X)) # on fait une somme cumulée pour la marche, et on rajoute le point de départ qu
```

Voici un autre code, un peu moins direct mais plus pédagogique:

```
T=30 #le nombre de pas de la marche
p=0.65 # la proba d'aller à droite
Y = runif(T+1) #un échantillon de T variables uniformes dans [0,1]
X = c(1:(T+1))*0 #un vecteur avec T+1 zeros
for (i in 1:(T+1)) {
  if (Y[i]<p)
    X[i] = 1
  if (Y[i]>=p)
    X[i] = -1
}
marche=c(0,cumsum(X))
```

```
T=2000
p=0.65
Y = runif(T)
X = 2*(Y<p)-1
marche = c(0,cumsum(X))
temps = 0:T #vecteur des temps de 0 à T
plot(temps,marche,type="l")
```

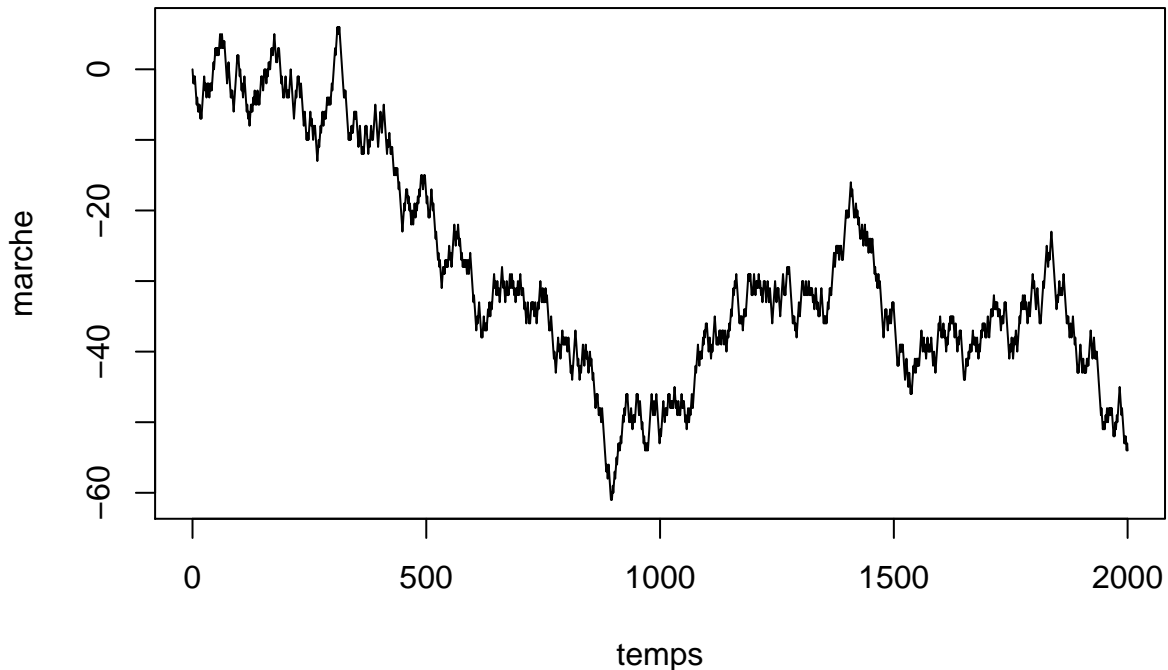


2.

Réponse : Dès que $p \neq 1/2$, la marche aléatoire est dirigée vers $\pm\infty$. La probabilité que le joueur rentre chez lui un jour (c'est à dire qu'il repasse par 0) est quasi-nulle.

```
T=2000
p= 0.5
Y = runif(T)
X = 2*(Y<p)-1
```

```
marche = c(0,cumsum(X))
temps = 0:T
plot(temps,marche,type="l")
```



3.

Réponse : Cette fois-ci pour $p = 1/2$, la marche aléatoire fait des sauts entre les négatifs et les positifs, et semble osciller indéfiniment. Non seulement il apparaît que l'on repasse par le point 0, mais en plus, on semble y revenir... une infinité de fois si on laissait une infinité de temps au promeneur.

4. Voici le code complété :

```
T0 <- function(){
t = 0
position = 0
while (t==0 | position != 0) {#sous windows, alt Gr+6 pour faire la barre verticale
t=t+1
position <- position + 2*(runif(1)<0.5)-1}
t
}
```

ou, sous une autre version :

```
T0bis <- function(){
position = 2*(runif(1)<0.5)-1 # j'ai fait un premier pas pour sortir de chez moi
t=1
while (position != 0) {
t=t+1
position <- position + 2*(runif(1)<0.5)-1}
t
}
```

On peut tester les fonctions:

```
T0()
```

```
## [1] 30
```



```
T0bis()
```

```
## [1] 38
```

```
m=500  
T0s = c(1:m)*0  
  
for (i in 1:m) {  
  T0s[i] = T0()  
}  
  
mean(T0s)
```

5.

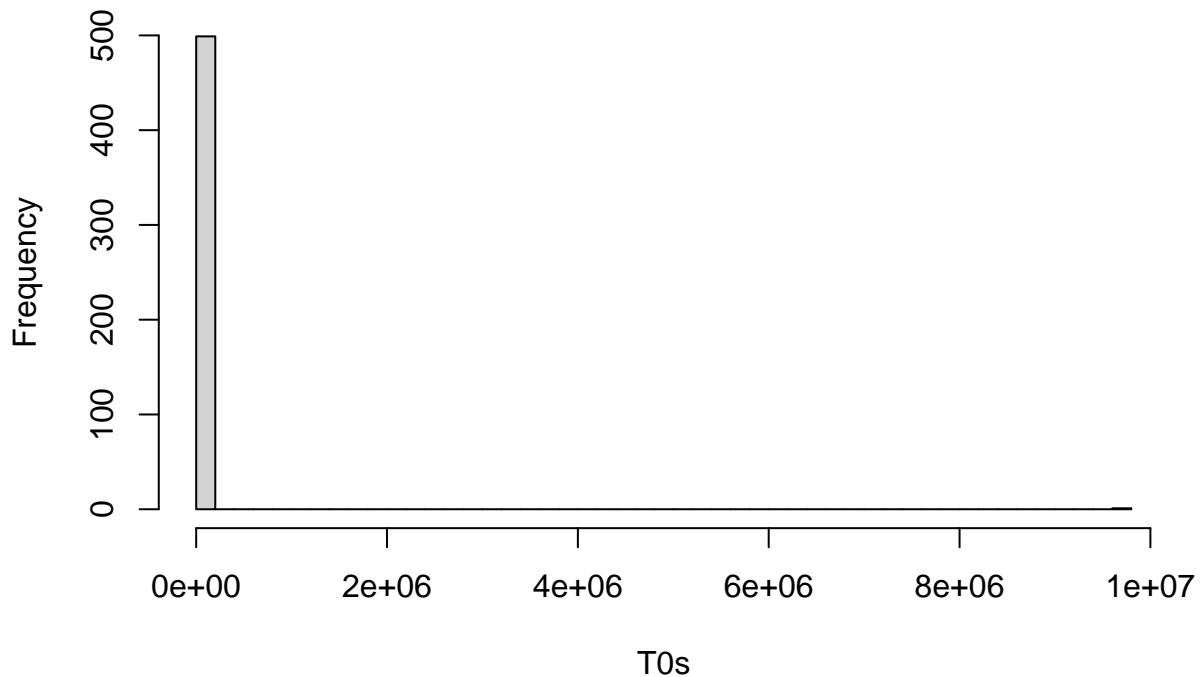
```
## [1] 19951.54
```

```
var(T0s)
```

```
## [1] 189641405719
```

```
hist(T0s, breaks = 50)
```

Histogram of T0s



Réponse : L'histogramme de T_0 n'est pas très beau : il y a des valeurs extrêmement grandes qui viennent faire varier énormément la moyenne des échantillons, y compris en augmentant m . Il semble qu'il n'y ait pas de convergence, et cela est le signe que l'espérance de T_0 semble être infinie. En fait, il se passe quelque chose d'assez singulier : le promeneur est sûr de revenir un jour chez lui, i.e. $\mathbb{P}(T_0 < \infty) = 1$, mais le temps moyen qu'il met pour rentrer est lui infini ($\mathbb{E}[T_0] = \infty$) !