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 réprésentation de données.

Exercice 0 : Reprise en main de la console : calculs rapides Recopier dans la console chacun de ces calculs, puis l'éxécuter (en appuyant sur Entrée). Assurez-vous de bien comprendre le résultat renvoyé par R.

```
3+1.2, 2-7.8, 2/5, 2*5.5, sqrt(100), log(2), exp(1), log(exp(3)), log10(10000), sin(pi/2), cos(pi)
```

La commande `?` devant une fonction permet d'obtenir de l'aide. Par exemple, pour demander de l'aide sur la fonction `exp` on écrira:

```
?exp
```

R permet aussi de définir des tableaux à l'aide de `c(...)`. Définir un tableau `tab` contenant 6 nombres réels de votre choix. Ensuite, que renvoient les commandes suivantes ?

```
tab[2], tab[-2], tab[2:5], tab[c(2,5)], tab+0.5, tab/2, tab*5, exp(tab)
```

Pour la suite du TP, y a quatre commandes simples à retenir avec R pour chaque loi usuelle : `rmaloi` permet de simuler des variables selon maloi, `dmaloi` calcule la densité de maloi, `pmaloi` pour la fonction de répartition, et `qmaloi` pour les quantiles de maloi.

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

1. Simulez une réalisation d'un échantillon de $n = 1000$ variables independantes et de même loi normale de moyenne $\mu = -2$ et de variance 9. On utilisera donc la fonction `rnorm` qui prend en paramètres n, `mean` (la moyenne) et `sd` (l'écart-type).

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

Calculer la moyenne et la variance de la réalisation.

```
mean(ech)
```

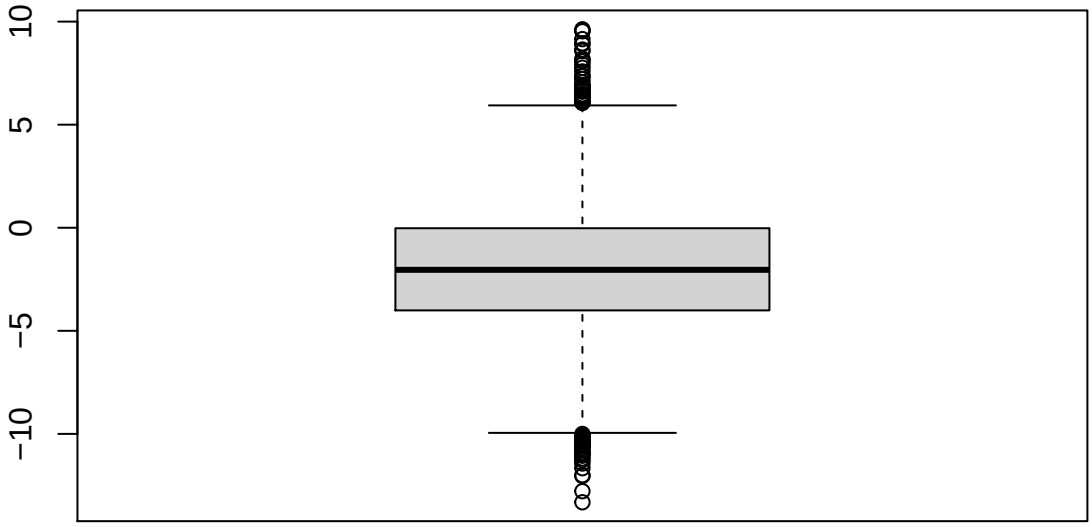
```
## [1] -2.012296
```

```
var(ech)
```

```
## [1] 8.913065
```

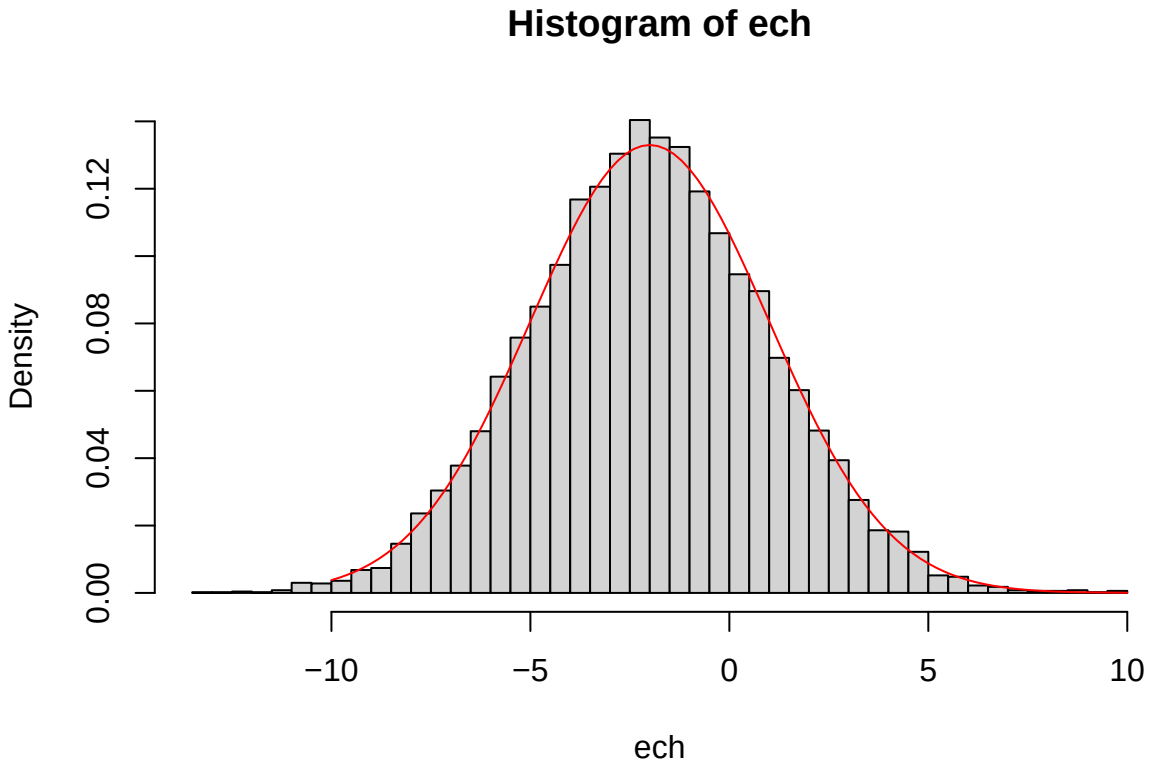
Afficher la représentation de la réalisation sous la forme d'une boîte à moustaches (fonction `boxplot`)

```
boxplot(ech)
```



2. Tracer l'histogramme du jeu de données ainsi obtenu (fonction `hist` avec option `freq=F`), et réfléchir à comment régler le paramètre `breaks`. Puis à l'aide de la fonction `curve` (avec option `add=T,col="red"`), superposer à l'histogramme la densité théorique de la loi en question.

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



3. Répéter les deux premières questions avec $n = 1000$ puis $n = 10000$. Qu'observe-t-on?

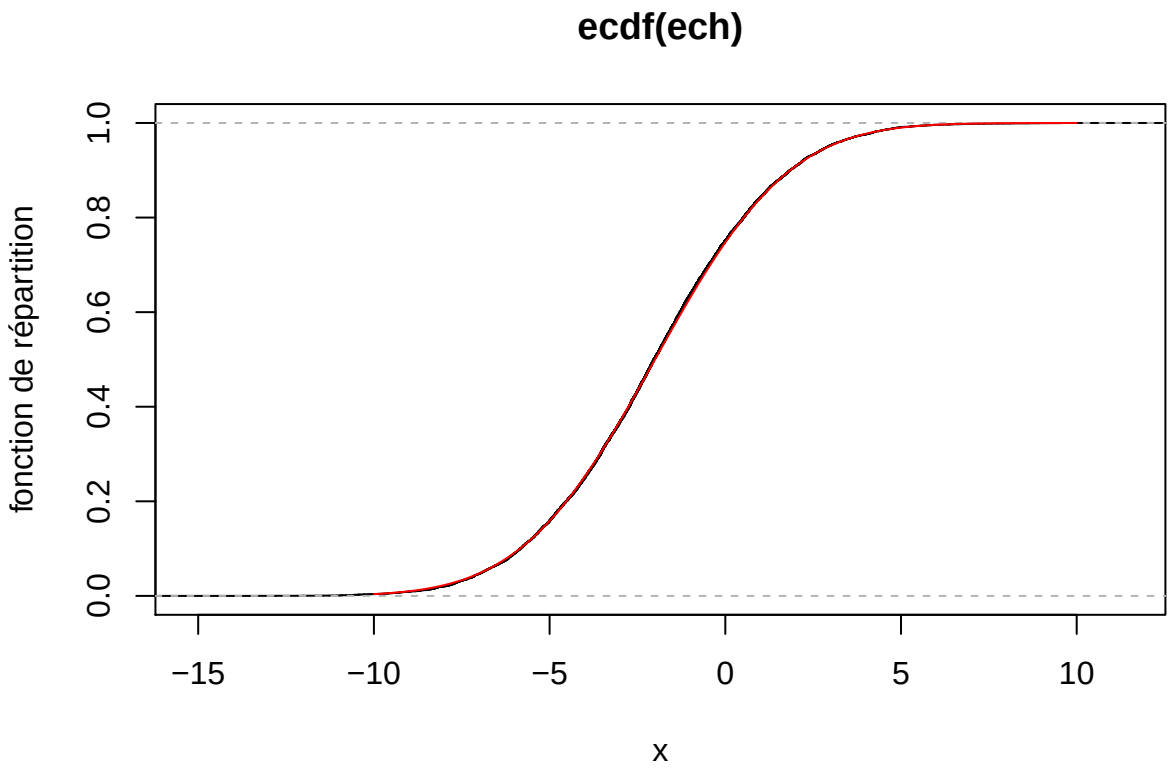
Réponse : L'histogramme a tendance à se confondre avec la courbe de la densité théorique.

4. Pour une réalisation $(x_1, \ldots, x_n)$ d'un échantillon, la fonction de répartition empirique F_{emp} est définie par

$$\forall y \in \mathbb{R},\ F_{emp}(y) = \frac{1}{n} \sharp \{1 \leq i \leq n,\ x_i \leq y\}\,.$$

A l'aide des fonctions `plot` et `ecdf`, représenter la fonction de répartition empirique de la réalisation. Superposer au graphique la fonction de répartition (théorique) de la loi normale étudiée. Qu'observe-t-on ?

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

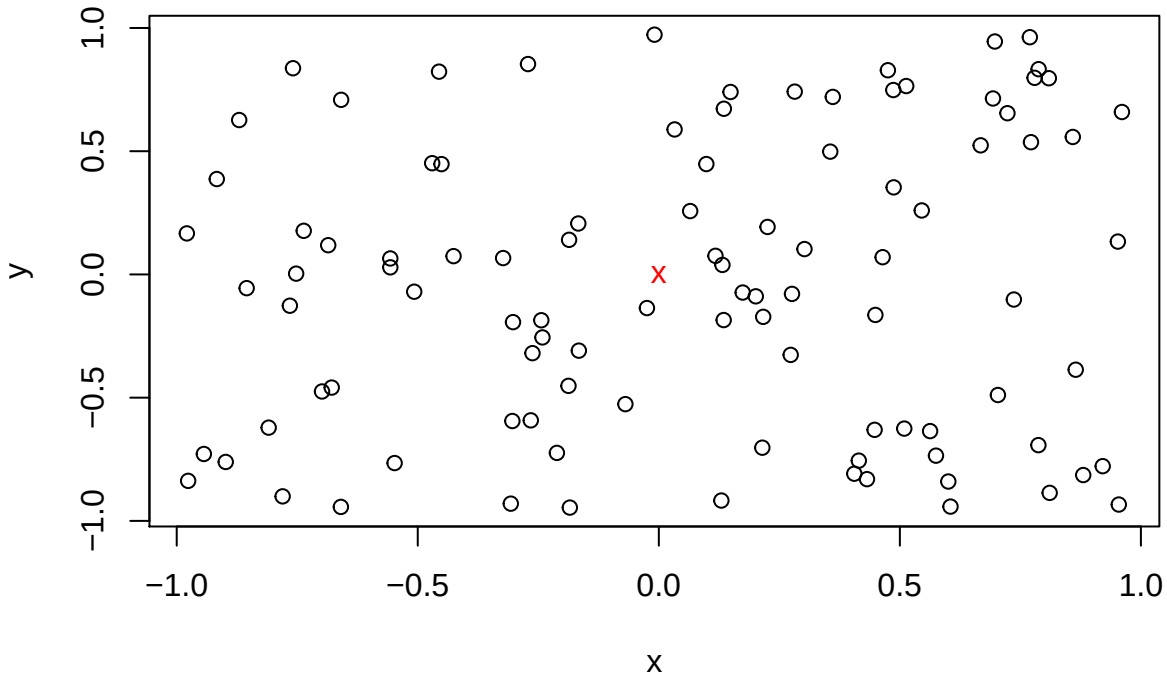


Réponse : La fonction de répartition empirique a tendance à se confondre avec la fonction de répartition théorique.

Exercice 2 : Lançons des fléchettes ! On dispose d'un carré de côté 2. On place l'origine O du repère au centre du carré $[-1,1] \times [-1,1]$. On considère tout d'abord l'expérience aléatoire suivante : on lance au hasard des fléchettes dans le carré, avec une abscisse et une ordonnée indépendantes et toutes deux de loi uniforme sur $[-1,1]$.

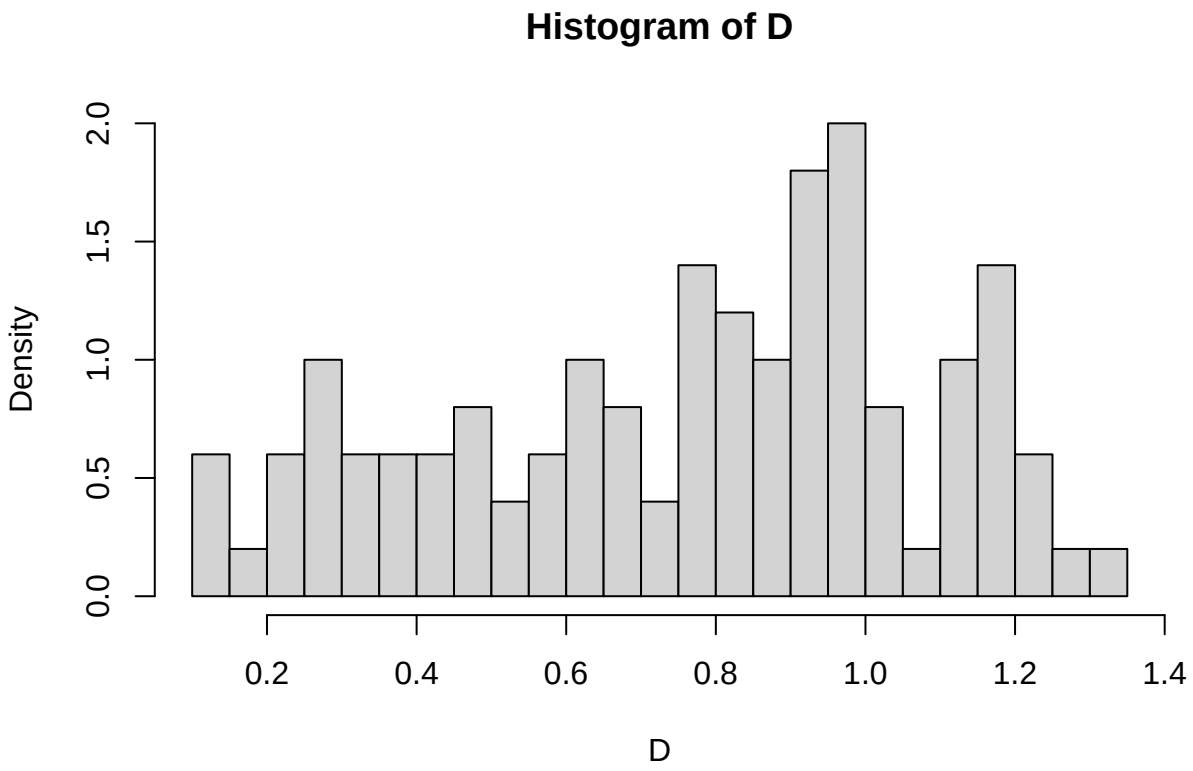
1. Simuler $n = 100$ lancers indépendants de fléchettes. Les représenter graphiquement sur le carré qui sert de cible.

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



2. On repère la qualité d'un lancer à la distance D de la flèche au centre de la cible. Tracer l'histogramme des réalisations de D.

```
D = sqrt(x^2 + y^2)
hist(D, breaks=20,freq=F)
```



Quelle est la moyenne de la réalisation ?

```
mean(D)
```

```
## [1] 0.7609691
```

3. Refaire l'histogramme de D pour $n = 10^7$. Interprétez. (Indication : là encore, attention à bien calibrer `breaks`).

```
n=10^6
D= sqrt(runif(n,-1,1)^2 + runif(n,-1,1)^2)
hist(D, breaks=100,freq=F)
```


```
mean(D)
```

```
## [1] 0.7654507
```

```
sqrt(2)/2
```

```
## [1] 0.7071068
```

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. Proposez deux méthodes pour estimer $\mathbb{P}(D \leq 1)$: l'une à l'aide de simulations, l'autre à l'aide d'un calcul.

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

```
## [1] 0.785743
```

```
pi/4
```

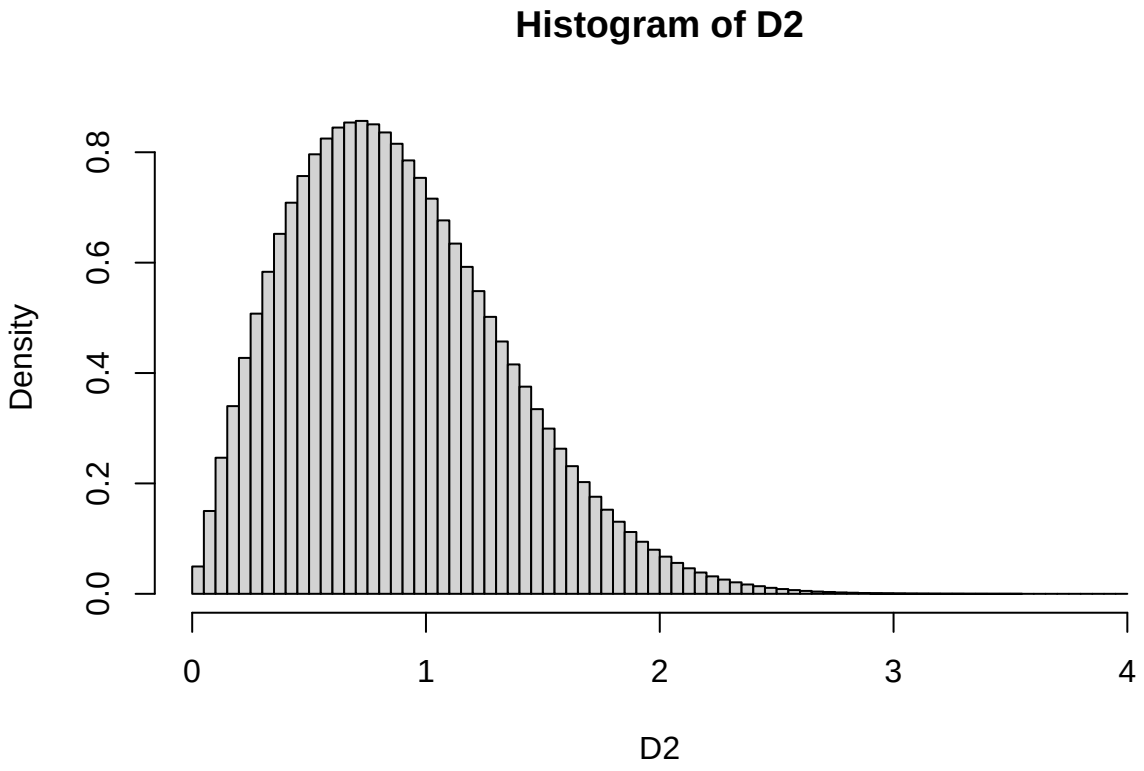
```
## [1] 0.7853982
```

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.

5. Maintenant arrive un autre lanceur de fléchettes qui se dit bien plus aguerri, et que l'abscisse x et l'ordonnée y du lancer sont deux variables indépendantes, gaussiennes centrées de variance $1/2$. Calculer la moyenne et représenter l'histogramme de réalisations indépendantes de D sous ce nouveau modèle. Commentez.

```
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)
```



Selon vous, ce joueur est-il vraiment meilleur ?

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

```
## [1] 0.7654507
```

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

```
## [1] 0.0809599
```

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

```
## [1] 0.8863443
```

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

```
## [1] 0.2146351
```

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 $\mathbb{Z}$... Un promeneur habite dans un monde à une seule dimension. Il décide d'aller se balader pour prendre l'air entre deux confinements. On suppose que ces déplacements ne peuvent se faire que sur l'axe des entiers relatifs ($\mathbb{Z}$) et on les modélise comme suit: Au temps $t = 0$, il est chez lui, au point 0, et à chaque pas de temps, il se déplace indépendemment vers la droite (de $+1$ donc) avec une probabilité p, ou bien vers la gauche (de -1) avec une probabilité $q = 1 - p$. La donnée successive de sa position en fonction du temps est appelée une marche aléatoire.

Une façon simple de simuler une telle marche aléatoire du temps $t = 0$ au temps $t = T$ est la suivante : il suffit de simuler un vecteur de taille T de variables $X_1, \dots, X_T$ indépendantes et de même loi, satisfaisant

$$\mathbb{P}(X_1 = -1) = 1 - p, \quad \mathbb{P}(X_1 = +1) = p.$$

Ensuite, il suffit de réaliser que la position de notre promeneur au temps t, notée S_t, est donnée par

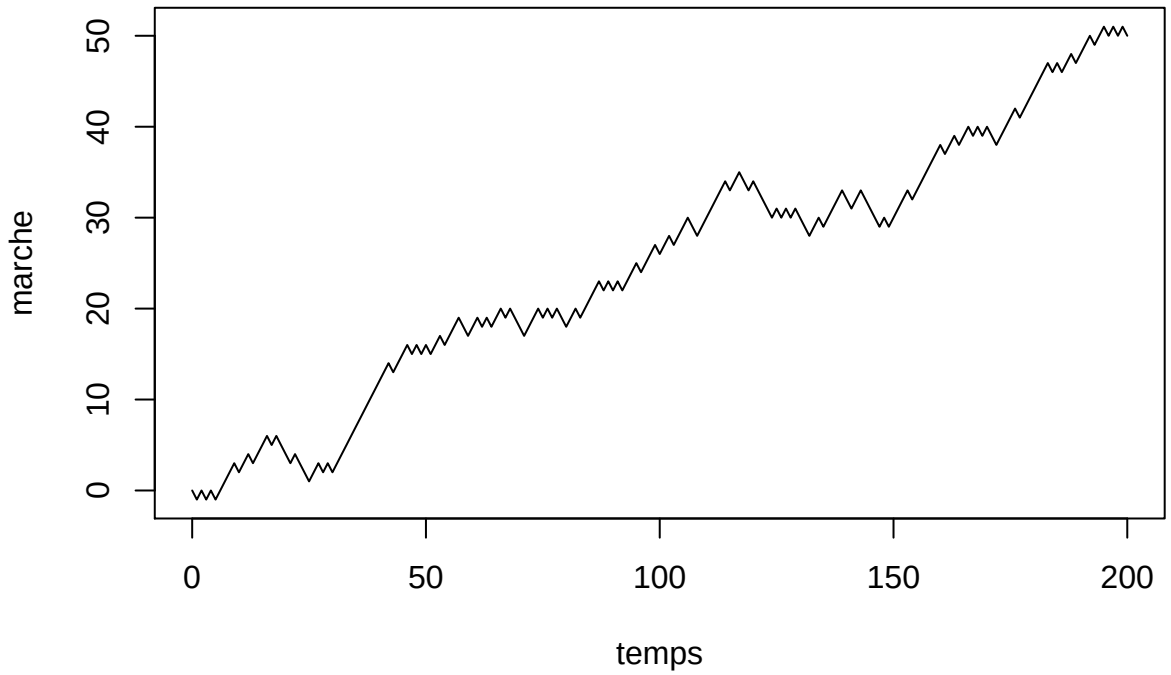
$$S_t := \sum_{s=1}^{t} X_s.$$

1. Simuler à l'aide de R une marche aléatoire de taille $T = 30$, avec $p = 0.65$. Stocker cette marche dans un vecteur *marche*. (Indication : pour simuler les X_i on pourra utiliser des variables i.i.d. de loi uniformes sur [0,1], puis regarder si elles sont inférieures à p...)

```
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
```

2. Refaire une simulation pour $T = 200$, et représenter graphiquement la marche aléatoire S_t en fonction du temps t: faites cela plusieurs fois. Le promeneur rentrera-t-il chez lui un jour ? Interprétez.

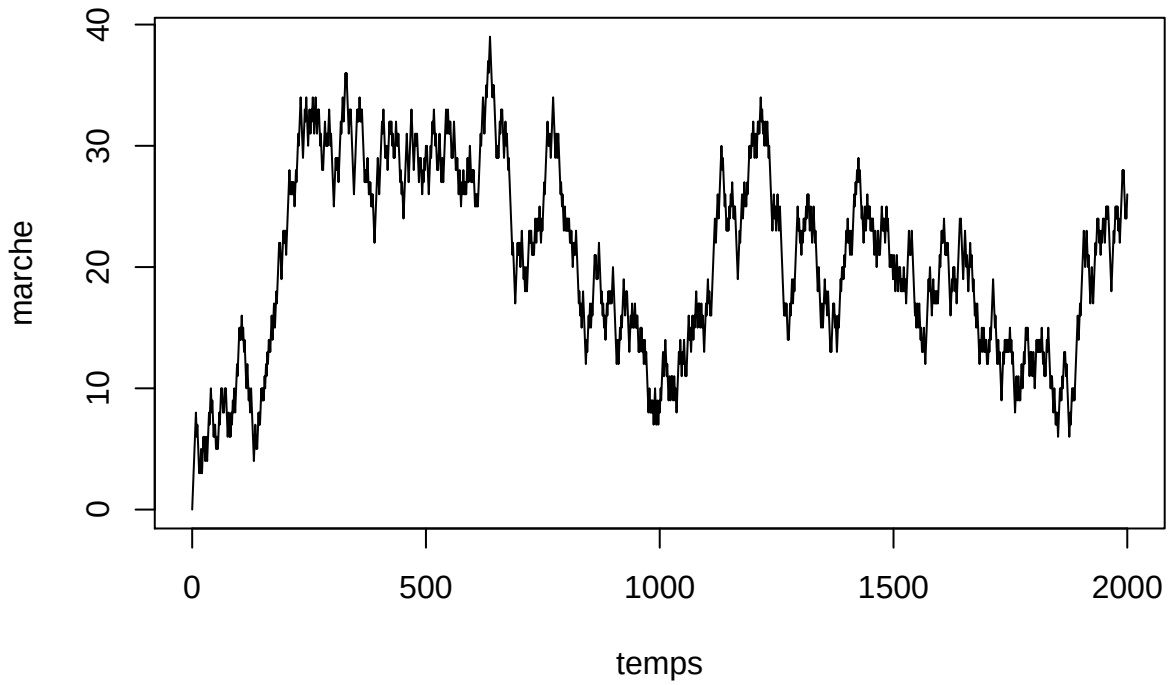
```
T=200
p=0.65
Y = runif(T)
X = 2*(Y<p)-1
marche = c(0,cumsum(X))
temps = 0:T
plot(temps,marche,type="l")
```



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.

3. Quelle valeur de p semble être la plus judicieuse pour espérer rentrer que le promeneur rentre un jour chez lui ? Refaire plusieurs simulations avec la nouvelle valeur de p, pour $T = 2000$, et représenter graphiquement la marche aléatoire S_t en fonction du temps t. Qu'observe-t-on cette fois-ci ? Interprétez.

```
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")
```



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.

Dans toute la suite on prend $p = 1/2$. On s'intéresse au premier temps T_0 de retour en 0, c'est-à-dire au premier instant où le promeneur revient chez lui. Mathématiquement,

$$T_0 := \inf \{n \in \mathbb{N}^*,\ S_n = 0\},$$

avec la convention $T_0 = \infty$ si $\forall n \in \mathbb{N}^*, S_n \neq 0$. On peut en fait montrer que presque sûrement $T_0 < \infty$.

4. Compléter le code ci-dessous, qui définit la fonction `T0()` (sans argument) qui simule et renvoie le temps de retour en 0 pour une marche du promeneur de paramètre $p = 1/2$.

```
T0 <- function(){
t = 0
position = 0
while (t==0 | position != 0) {
t=t+1
position <- position + 2*(runif(1)<0.5)-1}
t
}
```

```
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
}
```

```
T0bis()
```

```
## [1] 40
```

5. A l'aide de la fonction `T0`, simuler $m = 100$ valeurs de T_0. Calculer son espérance (empirique), sa variance (empirique). Représenter son histogramme.

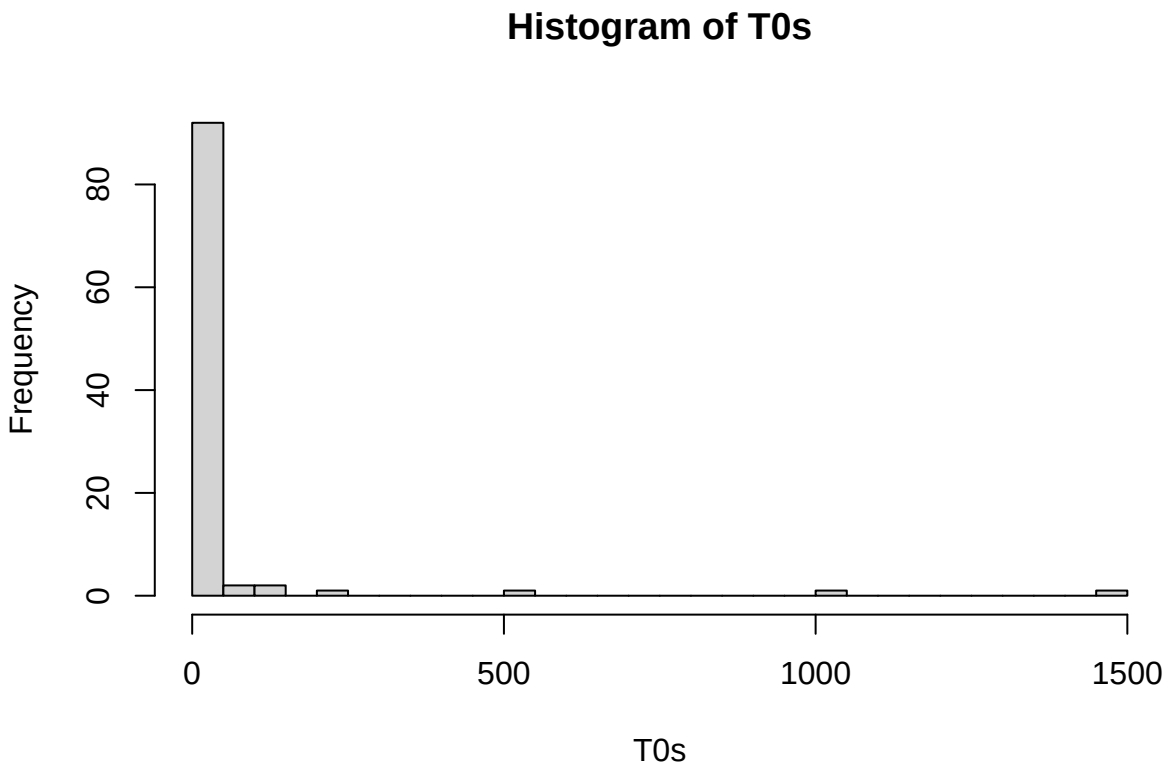
```
m=100
T0s = c(1:m)*0

for (i in 1:m) {
T0s[i] = T0()
}

mean(T0s)
```

```
## [1] 39.92
```

```
var(T0s)
```

```
## [1] 34141.33
```

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



A votre avis, que vaut l'espérance (théorique) de T_0 ?

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 ($\mathbb{E}[T_0] = \infty$) !