

TP7 Correction

Clement LAROCHE

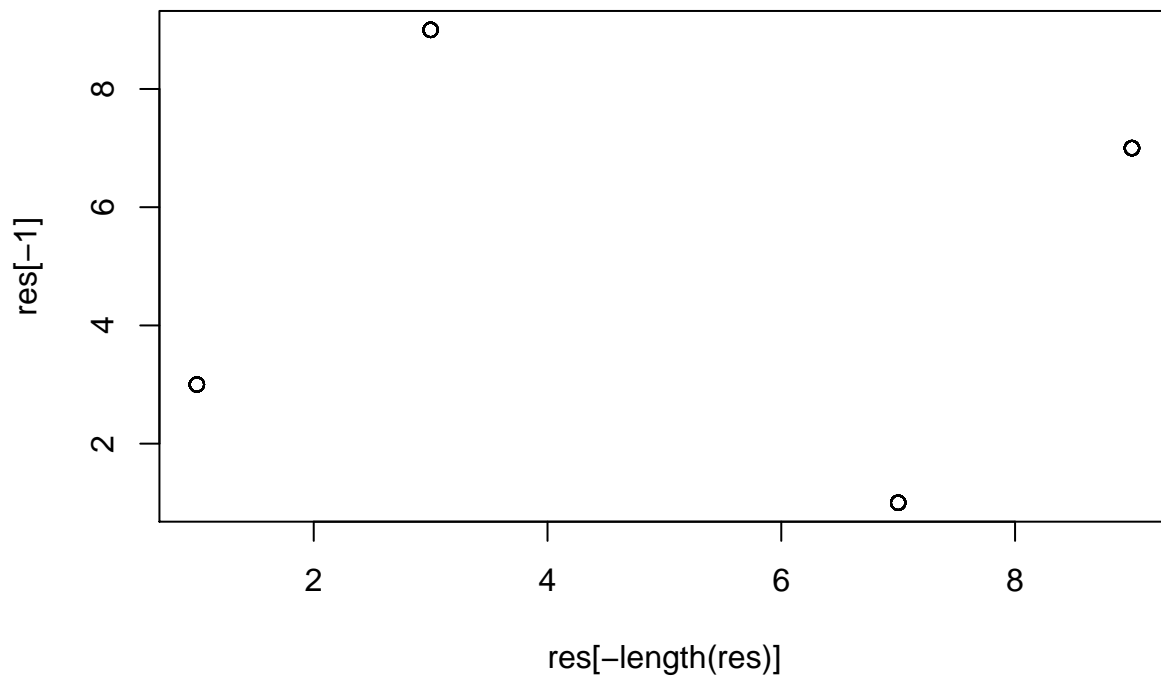
8 avril 2019

Génération de nombres pseudo-aléatoires

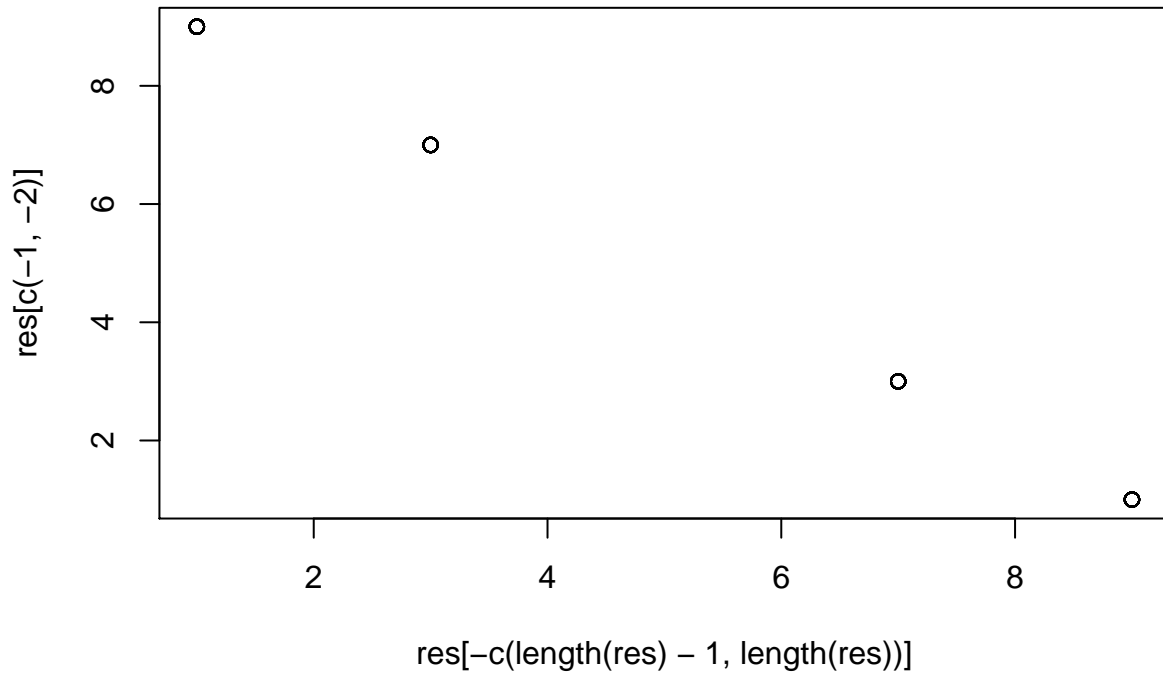
```
pseudo1 <- function(x)
{
  (3*x)%%10
}
res <- 1
for(i in 1:99)
{
  res <- c(res,pseudo1(res[length(res)]))
}
res

## [1] 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9
## [36] 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3
## [71] 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7 1 3 9 7

plot(res[-length(res)],res[-1])
```

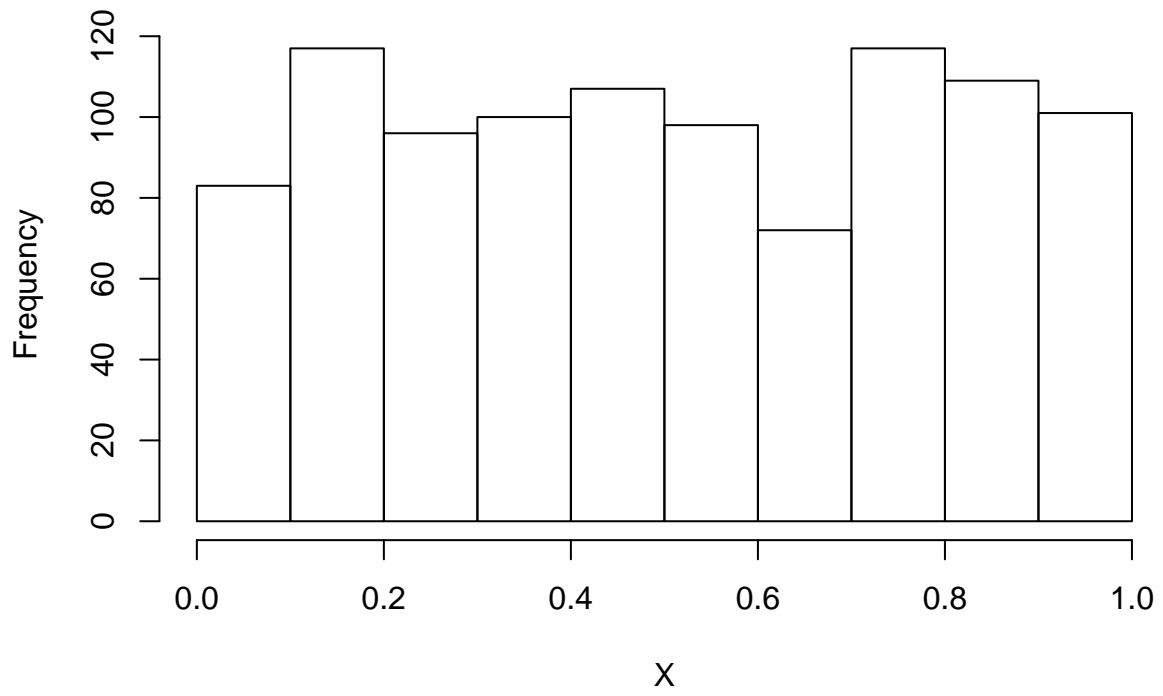


```
plot(res[-c(length(res)-1,length(res))],res[c(-1,-2)])
```



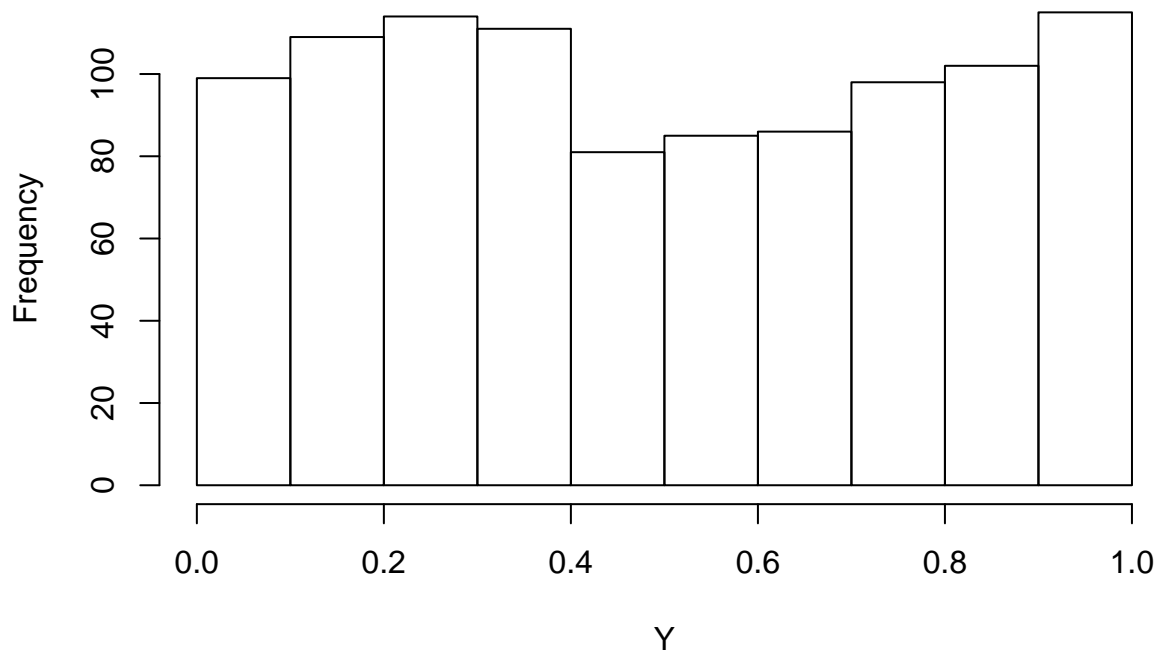
```
pseudo2 <- function(N,x0)
{
  a=7^7
  b=2^31-1
  x=1:N
  X=1:N
  x[1] <- x0
  X[1] <- (x0%%b)/b
  for (j in c(1:(N-1)))
  {
    x[j+1] <- (a*x[j]+1)%b
    X[j+1] <- x[j+1]/b
  }
  return(X)
}
X <- pseudo2(1000,1)
hist(X)
```

Histogram of X



```
Y = runif(1000)  
hist(Y)
```

Histogram of Y



```
#pseudo2(100,round(proc.time()[3]))
```

Simulation d'autres lois que la loi uniforme

```
# Bernoulli de paramètre 0.5
```

```
n <- 100  
X <- runif(n)  
X[X < 0.5] <- 0  
X[X > 0.5] <- 1  
X
```

```
## [1] 1 0 1 1 0 0 1 1 0 0 1 0 0 1 1 1 0 0 1 0 0 0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 0 0  
## [36] 0 1 0 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1 0 0 1 1 0 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 1  
## [71] 1 0 0 1 0 1 1 0 1 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1
```

```
sum(X)
```

```
## [1] 50
```

```
# Bernoulli de paramètre 0.25
```

```
n <- 100  
X <- runif(n)  
X[X < 0.75] <- 0  
X[X > 0.75] <- 1  
X
```

```
## [1] 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 1 0 0 0
```

```
## [36] 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1
## [71] 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 1 1 0 0
```

```
sum(X)
```

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

```
# simulation dé équilibré
```

```
n <- 100
x <- runif(n)
x[x < 1/6] <- 1
x[x > 1/6 & x < 1/3] <- 2
x[x > 1/3 & x < 1/2] <- 3
x[x > 1/2 & x < 2/3] <- 4
x[x > 2/3 & x < 5/6] <- 5
x[x > 5/6 & x < 1] <- 6
x
```

```
## [1] 5 1 6 4 4 4 5 6 3 5 1 4 4 6 2 5 1 3 3 2 1 5 6 1 6 5 1 1 4 6 2 2 3 5 6
## [36] 6 6 5 5 2 1 1 2 3 3 2 3 4 2 1 2 1 6 5 1 5 2 1 5 1 6 6 1 2 6 3 1 6 1 5
## [71] 1 1 2 5 3 4 6 4 4 3 3 3 4 1 2 2 3 5 1 6 2 4 4 5 2 5 2 2 6 6
```

```
sum(x==1)
```

```
## [1] 21
```

```
sum(x==2)
```

```
## [1] 18
```

```
sum(x==3)
```

```
## [1] 13
```

```
sum(x==4)
```

```
## [1] 13
```

```
sum(x==5)
```

```
## [1] 17
```

```
sum(x==6)
```

```
## [1] 18
```

Dans notre exemple, c'est-à-dire avec $I = \mathbb{R}$, la fonction F définit une bijection entre I et $]0, 1[$. donc on peut définir F^{-1} . On va maintenant s'intéresser à la manière de simuler des variables aléatoires à partir de l'inverse de leur fonction de répartition. L'idée vient du constat suivant, soit $X \sim U_{[0,1]}$, on a pour $t \in [0, 1]$:

$$\mathbb{P}(X \leq t) = t$$

Très bien, partons maintenant de la variable de l'énoncé $F_Z^{-1}(X)$. Pour connaître la loi de cette variable, on se souvient que si on arrive à trouver et identifier une formule explicite de la fonction de répartition de cette variable, cela nous donnera sa loi. On a que :

$$\begin{aligned} \mathbb{P}(F_Z^{-1}(X) \leq t) &= \mathbb{P}(X \leq F_Z(t)) \quad \text{car } F \text{ est une bijection dans notre cas de figure} \\ &= F_Z(t) \quad \text{car } X \text{ est une variable suivant une loi uniforme} \end{aligned}$$

On obtient donc que la fonction de répartition de la variable $F_Z^{-1}(X)$ est la même que celle de Z . Elles ont donc la même loi.

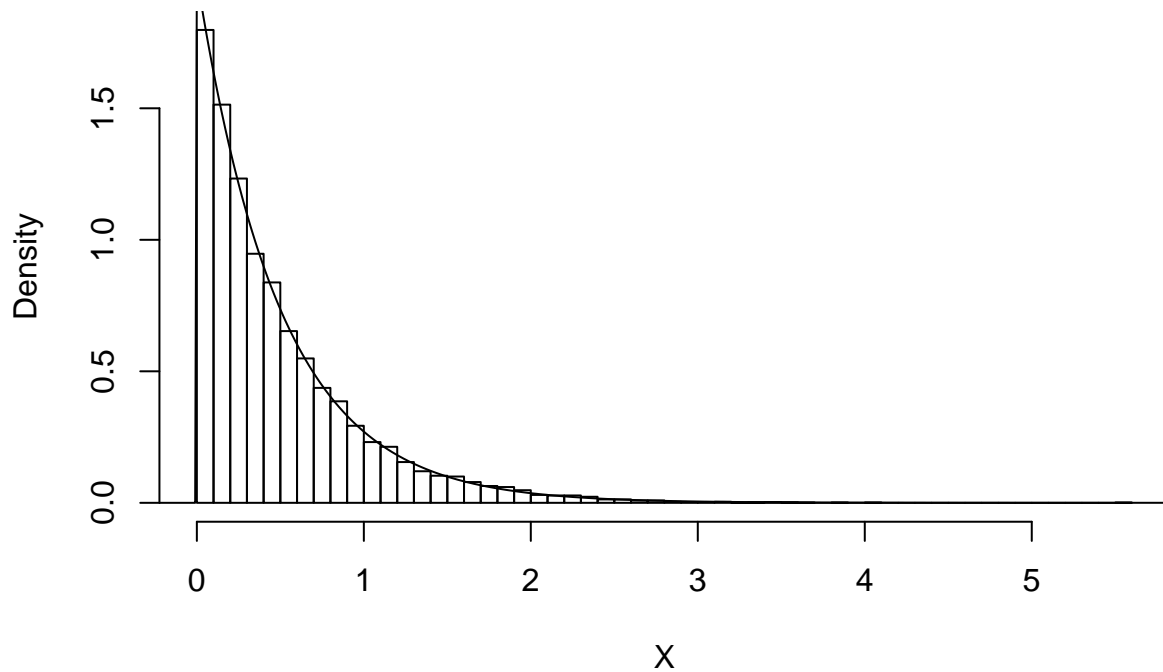
On passe au code :

```

n <- 10000
X <- runif(n)
X <- -1/2*log(1-X)
{
  hist(X,breaks = 50,freq = FALSE)
  lines(seq(from = -100,to = 100,by = 0.01),dexp(seq(from = -100,to = 100,by = 0.01),2))
}

```

Histogram of X



On calcule la fonction de répartition d'une variable aléatoire suivant une loi de Cauchy :

$$F_z(t) = \int_{-\infty}^t \frac{1}{\pi(1+x^2)} dx = \frac{1}{\pi} \arctan(t) + \frac{1}{2}$$

On inverse cette fonction de répartition, on obtient :

$$F_Z^{-1}(t) = \tan\left(\pi\left(t - \frac{1}{2}\right)\right)$$

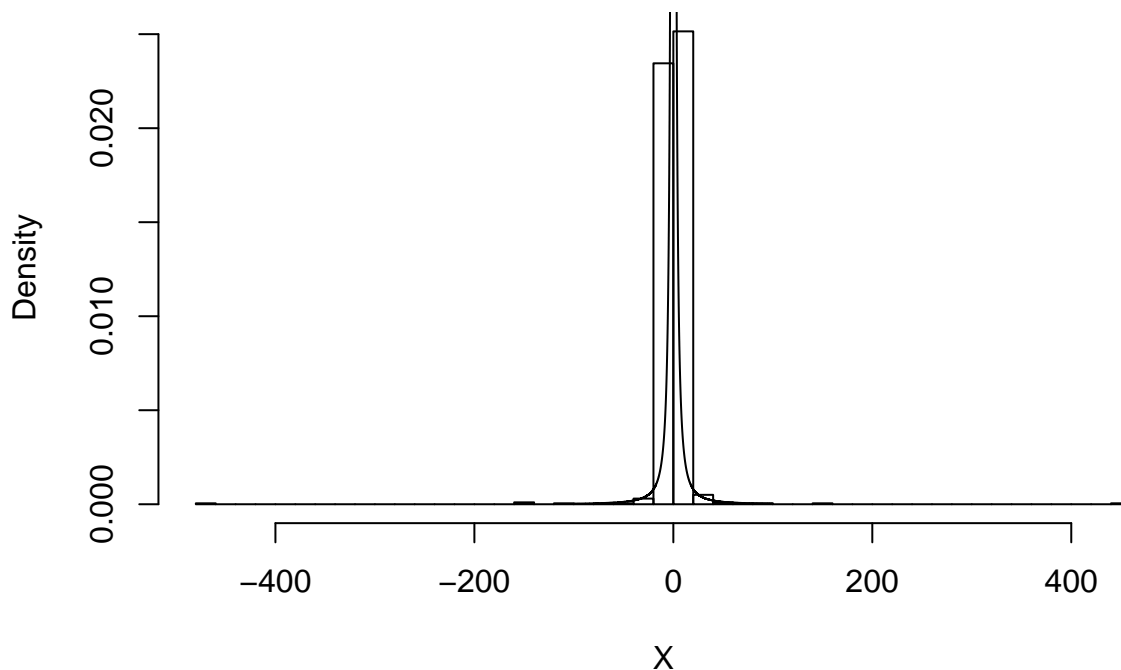
Reste plus que le code.

```

n <- 1000
X <- runif(n)
X <- tan(pi*(X-0.5))
{
  hist(X,breaks = 50,freq = FALSE)
  lines(seq(from = -100,to = 100,by = 0.01),dcauchy(seq(from = -100,to = 100,by = 0.01)))
}

```

Histogram of X



Quand on rentre dans le cas de la simulation de variables gaussiennes, on se rend compte que la méthode d'inversion est un peu problématique. en effet, je vous mets au défi de trouver une formule explicite de l'inverse de la fonction de répartition d'une loi normale. D'autres méthodes existent pour atteindre cet objectif.

Je vous précise ici la démarche de la méthode de **Box-Muller**. L'énoncé est le suivant.

Soient U_1 et U_2 deux variables aléatoires indépendantes suivant une loi uniforme sur $[0, 1]$. Soient :

- $Z_1 = \sqrt{-2\ln(U_1)} \cos(2\pi U_2)$
- $Z_2 = \sqrt{-2\ln(U_1)} \sin(2\pi U_2)$

Alors Z_1 et Z_2 suivent une loi normale centrée réduite. Pourquoi ?

Cela vient d'un habile changement de variables (passage de coordonnées cartésiennes en coordonnées polaires).

Si on écrit la loi jointe des deux variables Z_1 et Z_2 , on a alors :

$$\begin{aligned}
 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z_1^2} e^{-\frac{1}{2}z_2^2} dz_1 dz_2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{2\pi} e^{-\frac{1}{2}(z_1+z_2)^2} dz_1 dz_2 \\
 &= \int_{-\infty}^{\infty} \int_0^{2\pi} \frac{1}{2\pi} e^{\frac{1}{2}r^2} r d\theta dr \\
 &= \int_{-\infty}^{+\infty} r e^{-\frac{1}{2}r^2} dr \int_0^{2\pi} \frac{1}{2\pi} d\theta \\
 &= \int_0^{+\infty} \frac{1}{2} e^{-\frac{1}{2}s} ds \int_0^{2\pi} \frac{1}{2\pi} d\theta
 \end{aligned}$$

Deux changements de variables ont été effectués ici. Le premier consistait à écrire que $z_1 = r \cos(\theta)$ et $z_2 = r \sin(\theta)$. C'est le changement de variables qui permet le passage des coordonnées cartésiennes aux coordonnées polaires. Le deuxième changement de variables est $r^2 = s$. Celui là permet de retrouver la

densité de l'exponentielle (intégrale de gauche).

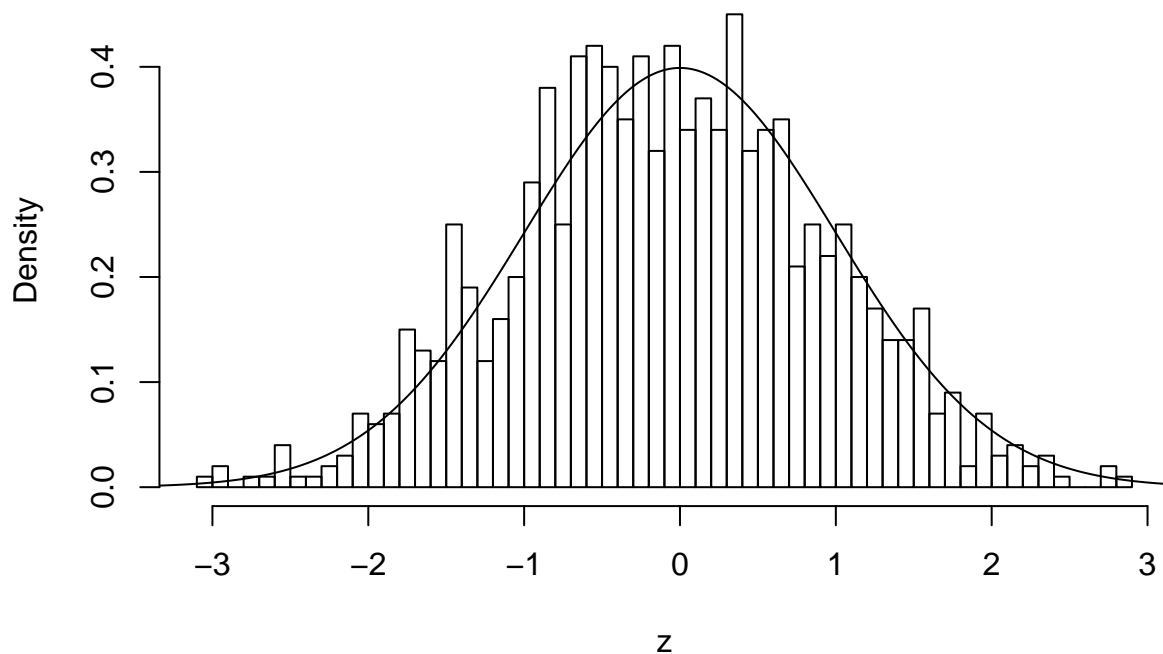
On reconnaît ici la densité d'une loi exponentielle de paramètre $\frac{1}{2}$ ainsi que celle d'une uniforme sur $[0, 2\pi]$.

On retrouve donc bien que Z_1 et Z_2 suivent une loi normale centrée réduite ! En effet, le terme $\sqrt{-2\ln(U_1)}$ (ou U_2) correspond bien à la simulation par méthode d'inversion d'une exponentielle !!!

On sait donc maintenant simuler des gaussiennes (centrées réduites) grâce à la méthode de Box-Muller !!!
Passons donc à la simulation.

```
u1 <- runif(1000) # simulation de 1000 réalisations d'une loi uniforme sur [0,1]
u2 <- runif(1000) # de même
z <- sqrt(-2*log(u1))*cos(2*pi*u2) # on applique la formule de Box Muller
# on vérifie qu'on obtient déjà une loi normale centrée réduite
X <- seq(from = -4,to = 4,by = 0.001)
{
  hist(z,breaks = 50,freq = FALSE)
  lines(X,dnorm(x = X))
}
```

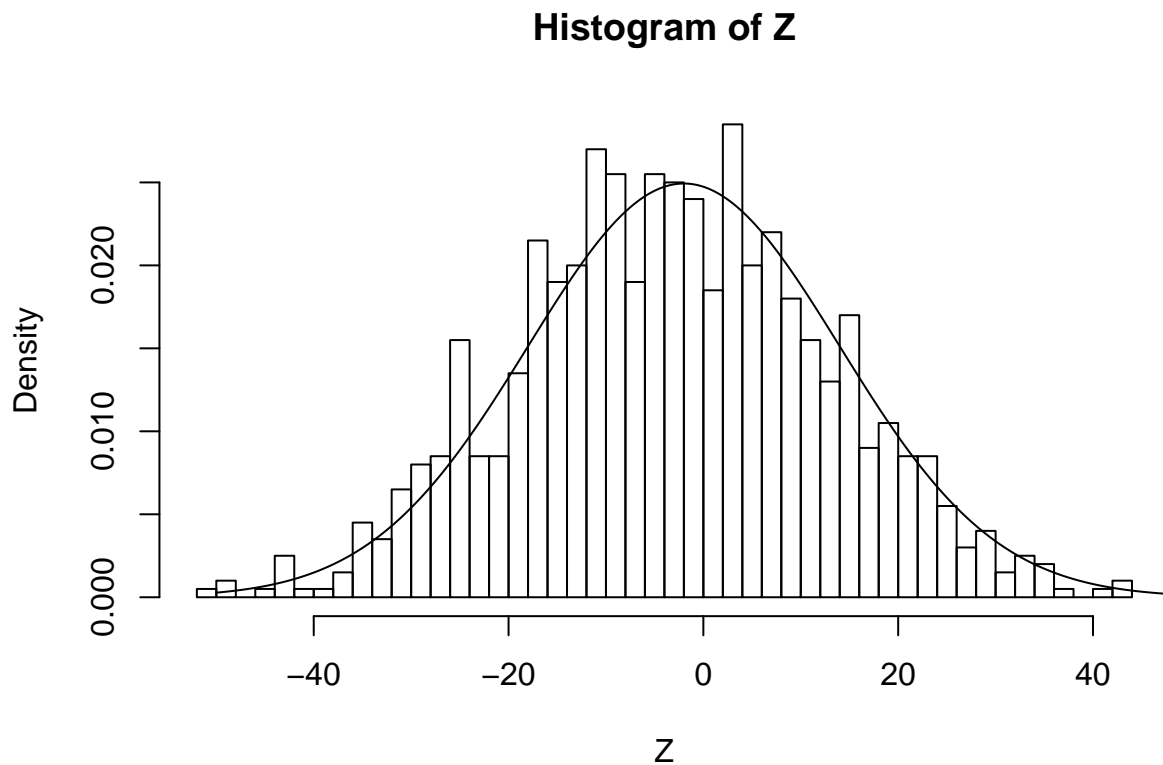
Histogram of z



```
# youpi !!!
```

Maintenant que l'on peut simuler une loi normale centrée réduite, pour simuler la loi $\mathcal{N}(-2, 4^2)$, la procédure est assez directe. Si $X \sim \mathcal{N}(0, 1)$, alors $\sigma X + \mu \sim \mathcal{N}(\mu, \sigma^2)$. Donc, passons au code :

```
Z <- 16*z-2
X <- seq(from = -50,to = 50,by = 0.01)
{
  hist(x = Z,breaks = 50,freq = FALSE)
  lines(x = X,y = dnorm(X,mean = -2,sd = 16))
}
```

Loi des grands nombres et théorème de la limite centrale

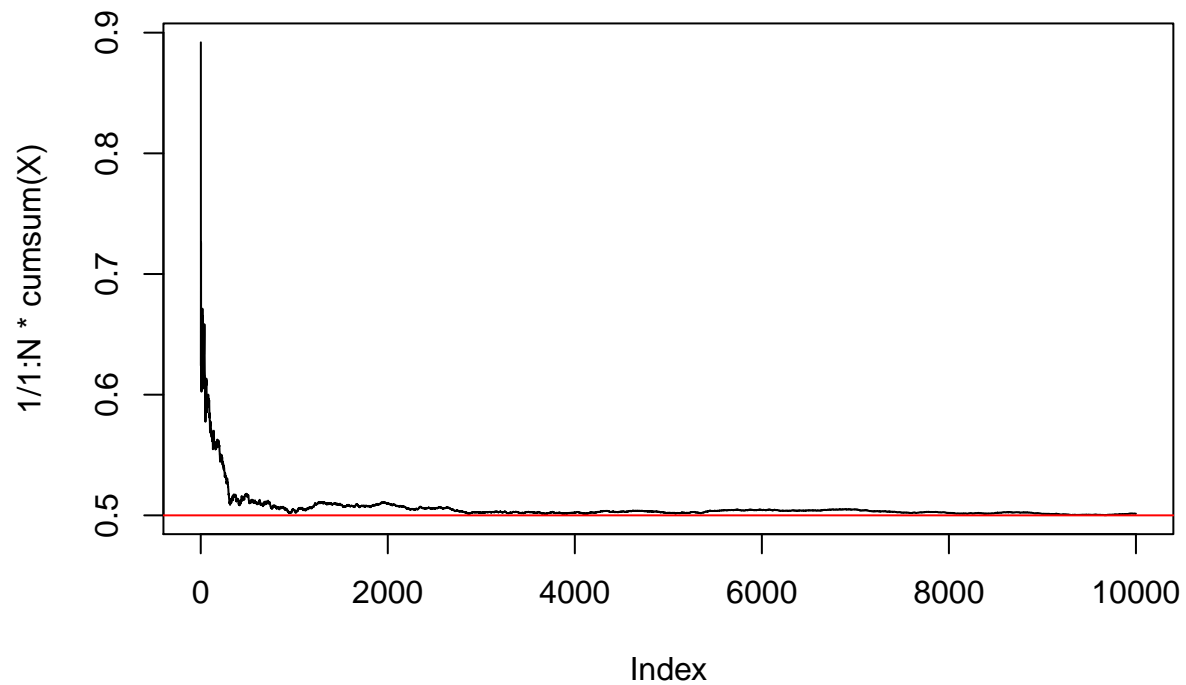
Lois précédemment définies :

- loi uniforme
- loi de Bernoulli
- loi exponentielle
- loi de Cauchy
- loi normale

```
N <- 10^4
# loi uniforme
X <- runif(N)
mean(X)

## [1] 0.5013895

{
  plot(1/1:N*cumsum(X),type = "l") # tracé de mes moyennes empiriques (les (n,X_n) de l'énoncé)
  abline(h = 0.5,col = "red") # tracé de la moyenne théorique
}
```



```

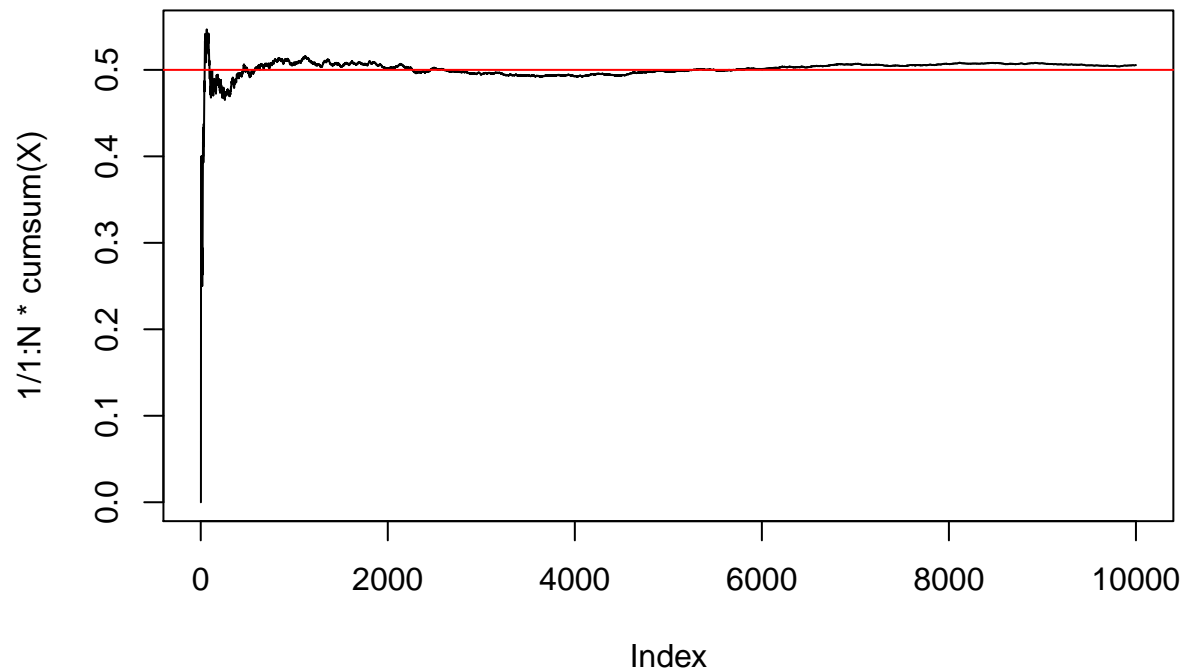
# convergence

# loi de Bernoulli
X <- rbinom(n = N,size = 1,prob = 0.5)
mean(X)

## [1] 0.5055

{
  plot(1/1:N*cumsum(X),type = "l") # tracé de mes moyennes empiriques (les (n,X_n) de l'énoncé)
  abline(h = 0.5,col = "red") # tracé de la moyenne théorique
}

```



```

# convergence

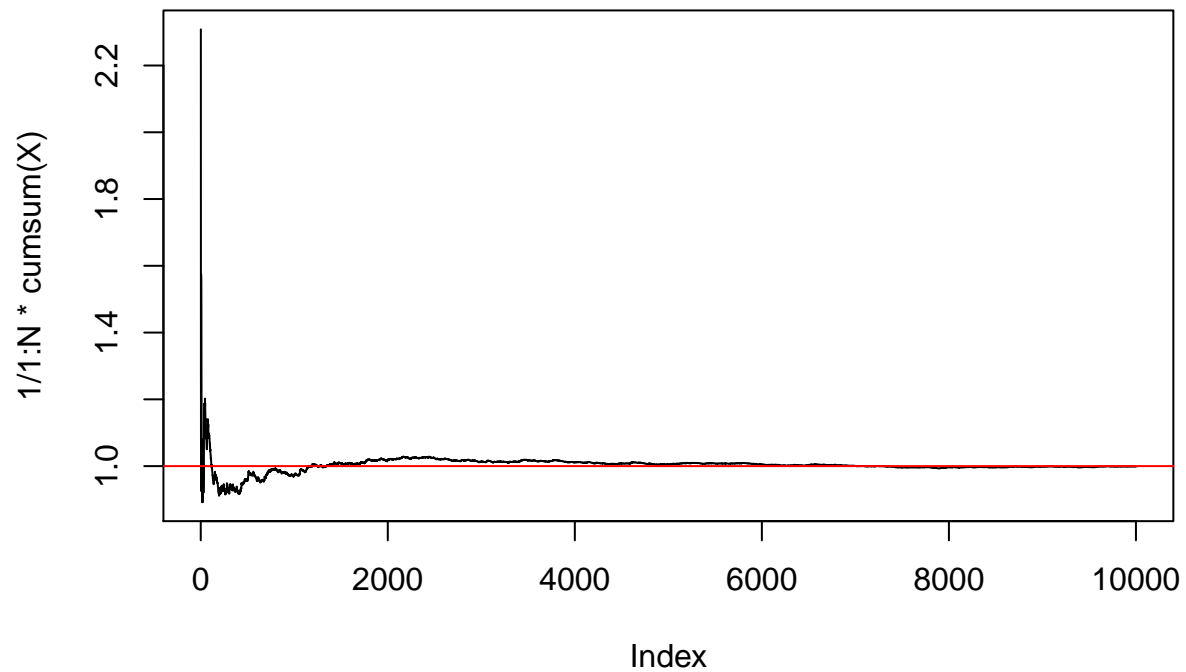
# loi exponentielle

X <- rexp(N)
mean(X)

## [1] 0.9981963

{
  plot(1/1:N*cumsum(X),type = "l") # tracé de mes moyennes empiriques (les (n,X_n) de l'énoncé)
  abline(h = 1,col = "red") # tracé de la moyenne théorique
}

```



```

# convergence

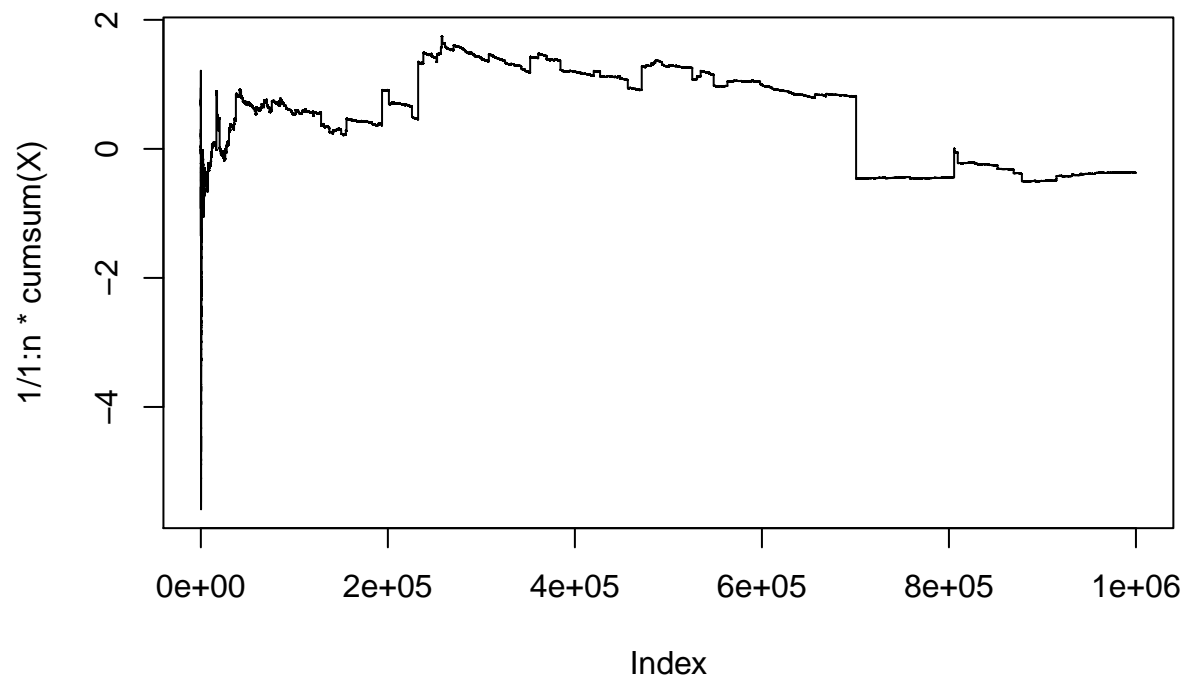
# loi de cauchy

n <- 10^6
X <- rcauchy(n)
mean(X)

## [1] -0.3710671

{
  plot(1/1:n*cumsum(X),type = "l") # tracé de mes moyennes empiriques (les (n,X_n) de l'énoncé)
}

```



```
# ne converge pas
```

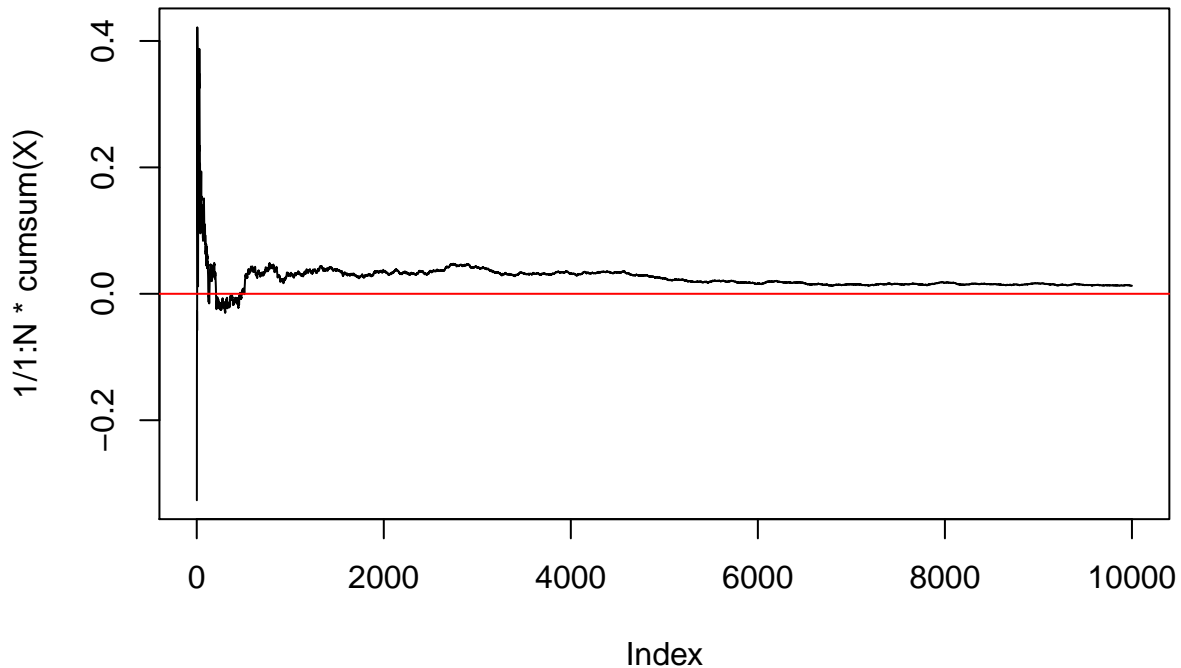
```
# loi normale
```

```
X <- rnorm(N)
```

```
mean(X)
```

```
## [1] 0.0128026
```

```
{
  plot(1/1:N*cumsum(X),type = "l") # tracé de mes moyennes empiriques (les (n,X_n) de l'énoncé)
  abline(h = 0,col = "red") # tracé de la moyenne théorique
}
```



convergence

On voit ici que toutes les lois convergent sauf la loi de Cauchy !!! Pourquoi la loi de Cauchy (au demeurant fort sympathique) ne converge pas ? Eh bien lançons nous dans le calcul de son espérance théorique. On a, pour $X \sim Cauchy(0, 1)$:

$$\begin{aligned} \mathbb{E}(X) &= \int_{-\infty}^{+\infty} x \frac{1}{\pi} \frac{1}{1+x^2} dx \\ &= [x \arctan(x)]_{-\infty}^{+\infty} - \int_{-\infty}^{+\infty} \arctan(x) dx \quad \text{par intégration par parties} \end{aligned}$$

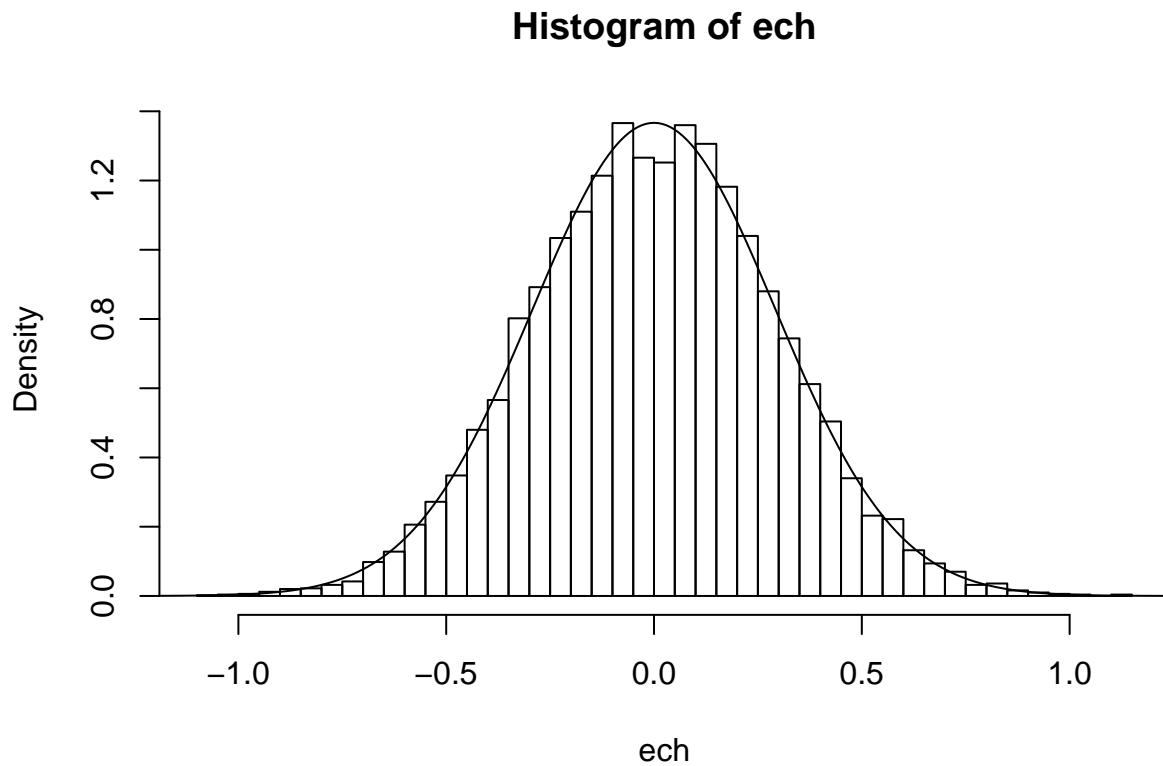
Or le terme de gauche diverge !!! La loi de Cauchy n'admet donc pas d'espérance théorique. Le paramètre de gauche rentré dans la loi de Cauchy n'est qu'un paramètre de position.

On recommence pour l'illustration du théorème de la limite centrale.

loi uniforme

```
N <- 10^4
M <- 10^4
ech <- c()
for(i in 1:M)
{
  X <- runif(N)
  ech <- c(ech, sqrt(N)*(mean(X)-0.5))
}
abs <- seq(from = -5, to = 5, by = 0.01)
{
  hist(x = ech, breaks = 50, freq = FALSE)
```

```
lines(x = abs,y = dnorm(x = abs,sd = sd(ech))) # juste pour bien se convaincre que ça a une allure de
}
```



```
# ceci marche pour toutes les lois sauf celle de Cauchy, observons
ech <- c()
for(i in 1:M)
{
  X <- rcauchy(n = N)
  ech <- c(ech,sqrt(N)*(mean(X)-0.5))
}
abs <- seq(from = -5,to = 5,by = 0.01)
{
  hist(x = ech,breaks = 50,freq = FALSE)
  lines(x = abs,y = dnorm(x = abs,sd = sd(ech))) # juste pour bien se convaincre que ça N'A PAS une allu
}
```

Histogram of ech

