G08-020844

Thomas BERATTO, Nermine BACCOUR, Guangyue CHEN

13 février 2019

Exercice 5.

3. En déduire un algorithme de simulation de |X| par la méthode de rejet

Algorithm 1: Rejection method for |X|

Simuler : U de $\mathcal{U}[0,1]$ et Y de $\mathcal{E}(\lambda = 1)$

Calculer : $h(Y) = f(Y)/cg(Y) = \frac{exp(\frac{-Y^2}{2}+Y)}{\sqrt{e}}$

while $U > h(Y)$ **do**

  ? *Simuler* : U *de* $\mathcal{U}[0,1]$ *et* Y *de* $\mathcal{E}(\lambda = 1)$

  ? *Calculer* : $h(Y) = f(Y)/cg(Y) = \frac{exp(\frac{-Y^2}{2}+Y)}{\sqrt{e}}$

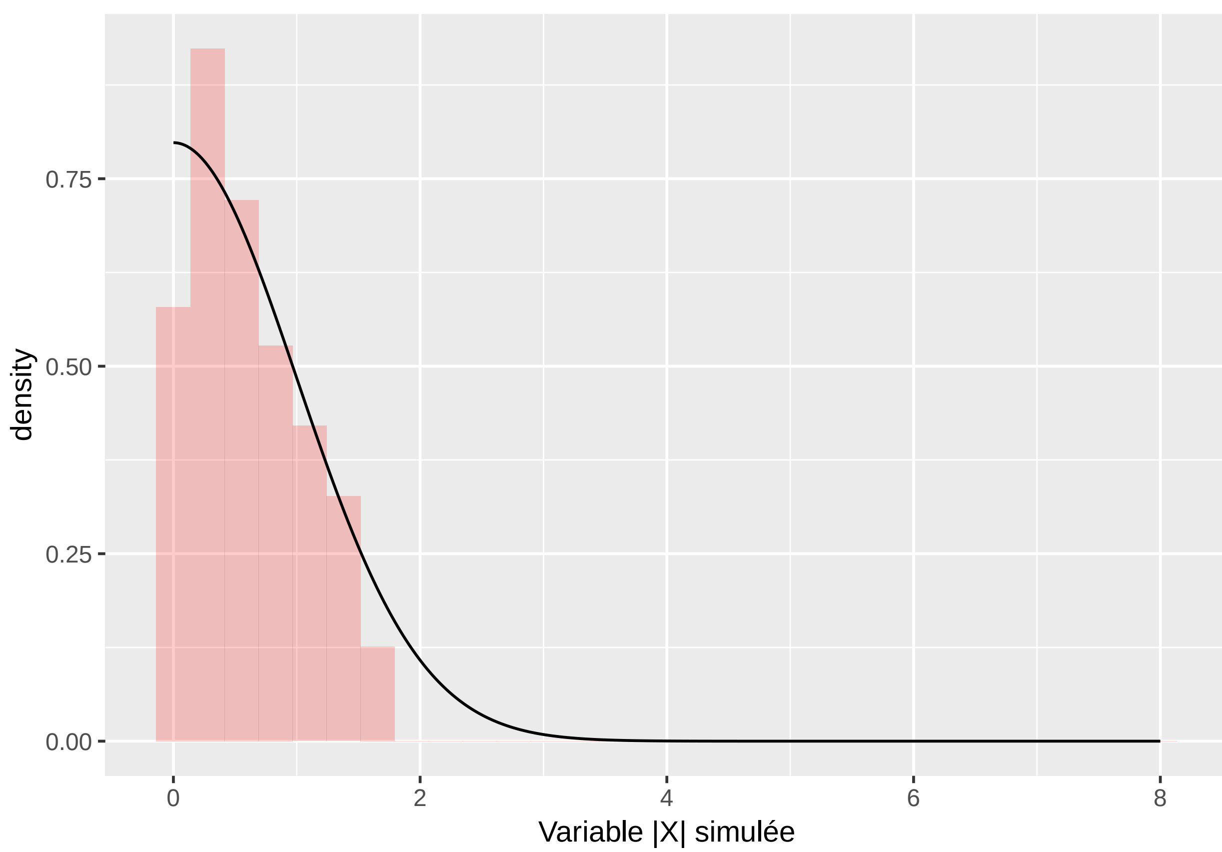
Retourner : Y

```
rejet1 <- function() {
  U <- runif(1,0,1)
  Y <- -log(1-U)
  h <- exp((-0.5*(Y^2))+Y)
  h <- h / (sqrt(exp(1)))
  while (U >= h) {
  U <- runif(1,0,1)
  Y <- -log(1-U)
  h <- exp(-0.5*(Y^2)+Y)
  h <- h / (sqrt(exp(1)))
  }
  return(Y)
}
```

4. Tracerl'histogramme des fréquences pour un échantillon de taille N assez grand (on prendra N=10000) et le comparer avec la densité théorique de |X|

```
X_abs <- rejet1()
for (i in 1:9999) {
  X_abs <- c(X_abs,rejet1())
}

density <- 0
abs <- 8/10000
density <- (2*exp(-0.5*(abs ^2)))/sqrt(2*3.14)
for (i in 2:10000) {
  x <- i * (8 / 10000)
  abs <- c(abs,x)
  density <- c(density,(2*exp(-0.5*(x^2)))/sqrt(2*3.14))
}

DF <- data.frame(X_abs,density)
plot1 <- ggplot(DF,aes(x=DF$X_abs)) +
  geom_histogram(aes(y=..density..),fill = "red", alpha = 0.2) +
  geom_line(aes(x=abs, y=DF$density)) + xlab("Variable |X| simulée")
plot1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



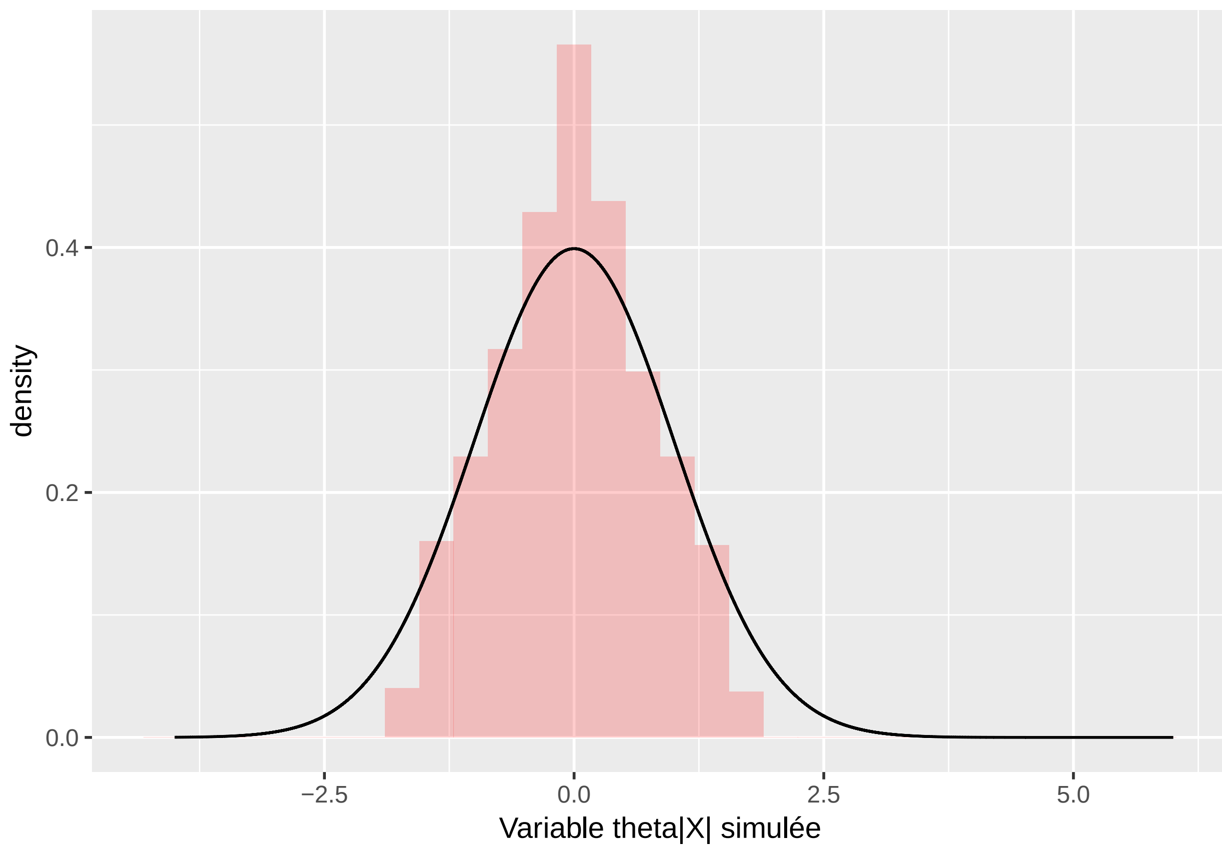
On constate bien que la densité empirique simulée correspond bien à la densité théorique.

6. Simuler un échantillon indépendant de taille N de loi normale en utilisant les questions précédentes et comparer les densités théoriques et empiriques

```
U <- runif(1,0,1)
if(U<0.5) {
  theta <- -1
} else{
  theta <- 1
}
for (i in 1:9999) {
  U <- runif(1,0,1)
  if(U<0.5) {
    theta <- c(theta,-1)
  } else{
    theta <- c(theta,1)
  }
}

theta_X_abs <- theta[1]*X_abs[1]
for (i in 2:10000){
  theta_X_abs <- c(theta_X_abs,theta[i]*X_abs[i])
}
```

```
ab <- -4
current = ab + 1/10000
for(i in 1:99999) {
  ab <- c(ab, current)
  current = current + 1/10000
}

y <- vector("numeric", 100000)
density <- function(x) {
  return((exp((-x*x)/2))/(sqrt(2*3.14)))
}
for (i in 1:100000) {
  y[i] = density(ab[i])
}

DF2 <- data.frame(theta_X_abs,y)
plot2 <- ggplot(DF2,aes(x=DF2$theta_X_abs)) +
  geom_histogram(aes(y=..density..),fill = "red", alpha = 0.2) +
  geom_line(aes(x=ab, y=DF2$y)) + xlab("Variable theta|X| simulée")
plot2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

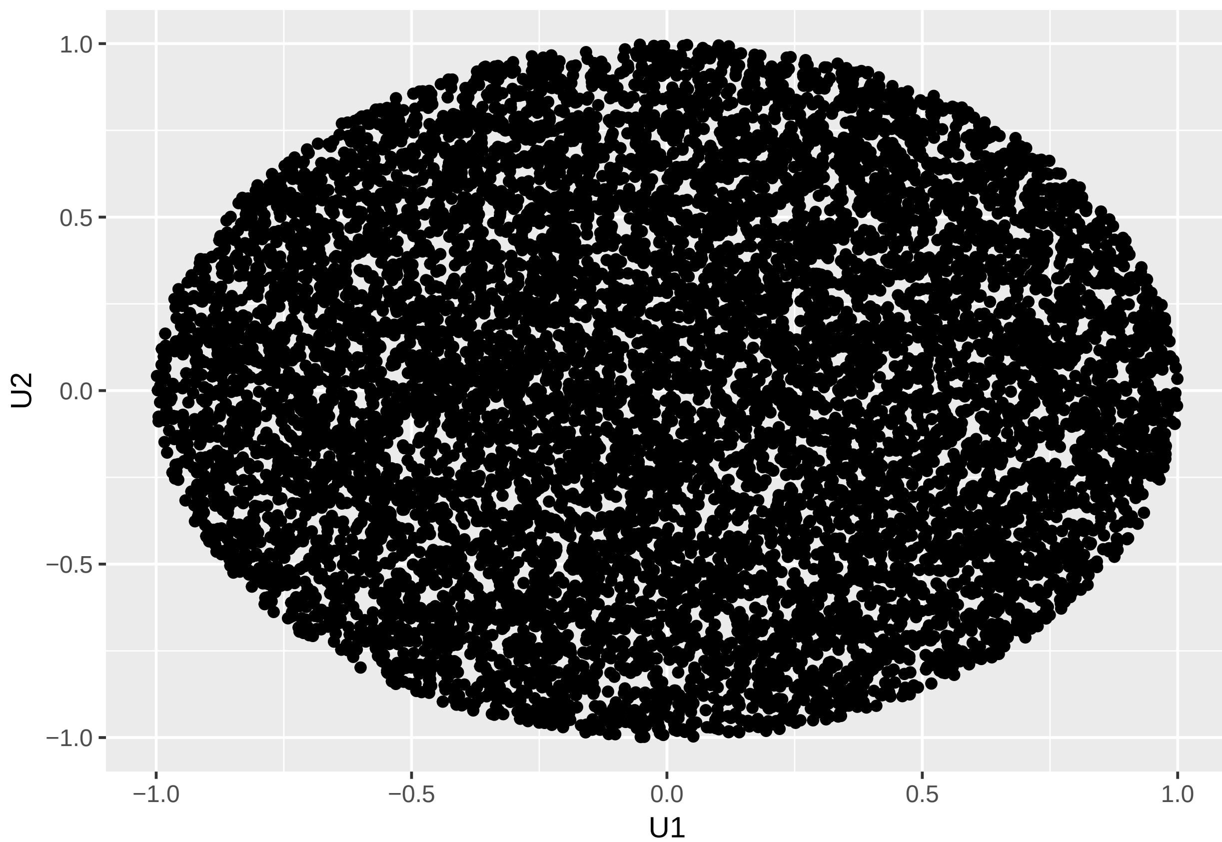


On constate bien que la densité empirique simulée correspond bien à la densité théorique d'une loi normale centrée réduite.

Exercice 6

Générer un échantillon indépendant de taille N = 10000 de la v.a. précédente et représenter les points simulés sur R2.

```
six <- function(N) {
  u1 <- 2*runif(1,0,1) -1
  u2 <- 2*runif(1,0,1) -1
  while(u1*u1 + u2*u2 > 1) {
    u1 <- 2*runif(1,0,1) -1
    u2 <- 2*runif(1,0,1) -1
  }
  U1 <- u1
  U2 <- u2
  for (i in 1:(N-1)) {
    u1 <- 2*runif(1,0,1) -1
    u2 <- 2*runif(1,0,1) -1
    while(u1*u1 + u2*u2 > 1) {
      u1 <- 2*runif(1,0,1) -1
      u2 <- 2*runif(1,0,1) -1
    }
    U1 <- c(U1,u1)
    U2 <- c(U2,u2)
  }
  return(data.frame(U1,U2))
}

exo6 <- six(10000)
plot3 <- ggplot(exo6,aes(x = exo6$U1,y=exo6$U2)) + geom_point() +
  xlab("U1") + ylab("U2")
plot3
```



1.0
0.5
0.0
−0.5
−1.0
U2
−1.0
−0.5
0.0
0.5
1.0
U1

Exercice 6. (Méthode de Box-Muller)

1.Montrer que X1 et X2 sont des v.a. gaussiennes indépendantes, centrées et réduites

Soit f une fonction borélienne bornée

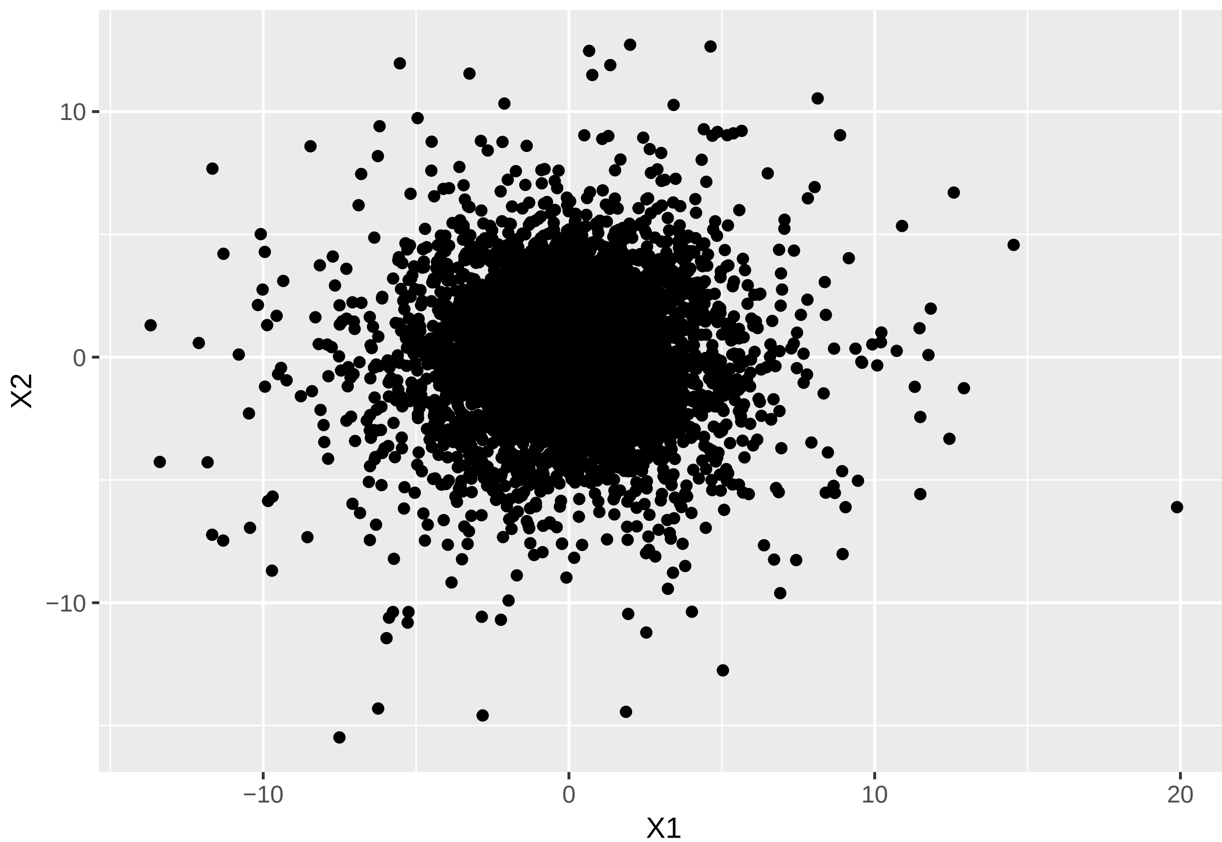
$\mathbb{E}[f(\sqrt{R}cos(\theta), \sqrt{R}sin(\theta))] = \int_{R+*[0,2\pi]} f(\sqrt{r}cos(\theta), \sqrt{r}sin(\theta))\frac{exp(-X/2)}{4\pi}dXd\theta$

Ainsi

$\mathbb{E}[f(x1, x2)] = \int_{R^2} f(x1, x2)\frac{exp(-x1^2/2)}{\sqrt{2\pi}} \frac{exp(-x2^2/2)}{\sqrt{2\pi}}$ Donc X1 et X2 sont deux v.a indépendantes et de loi normale centrée réduite

2. En déduire une simulation de N = 10000 réalisations indépendantes de la loi gaussienne Z = (X1,X2) et représenter les points générés sur R2

```r
R <- rexp(1,rate=0.5)
THETA <- runif(1,0,2*3.14)

for (i in 1:9999) {
  R <- c(R,rexp(1,rate=0.5))
  THETA <- c(THETA,runif(1,0,2*3.14))
}

X1 = vector("numeric",10000)
X2 = vector("numeric",10000)

for (i in 1:10000) {
  X1[i] = R[i] * cos(THETA[i])
  X2[i] = R[i] * sin(THETA[i])
}

DF4 <- data.frame(X1,X2)
plot4 <- ggplot(DF4,aes(x = DF4$X1,y=DF4$X2)) + geom_point() + xlab("X1") + ylab("X2")
plot4
```

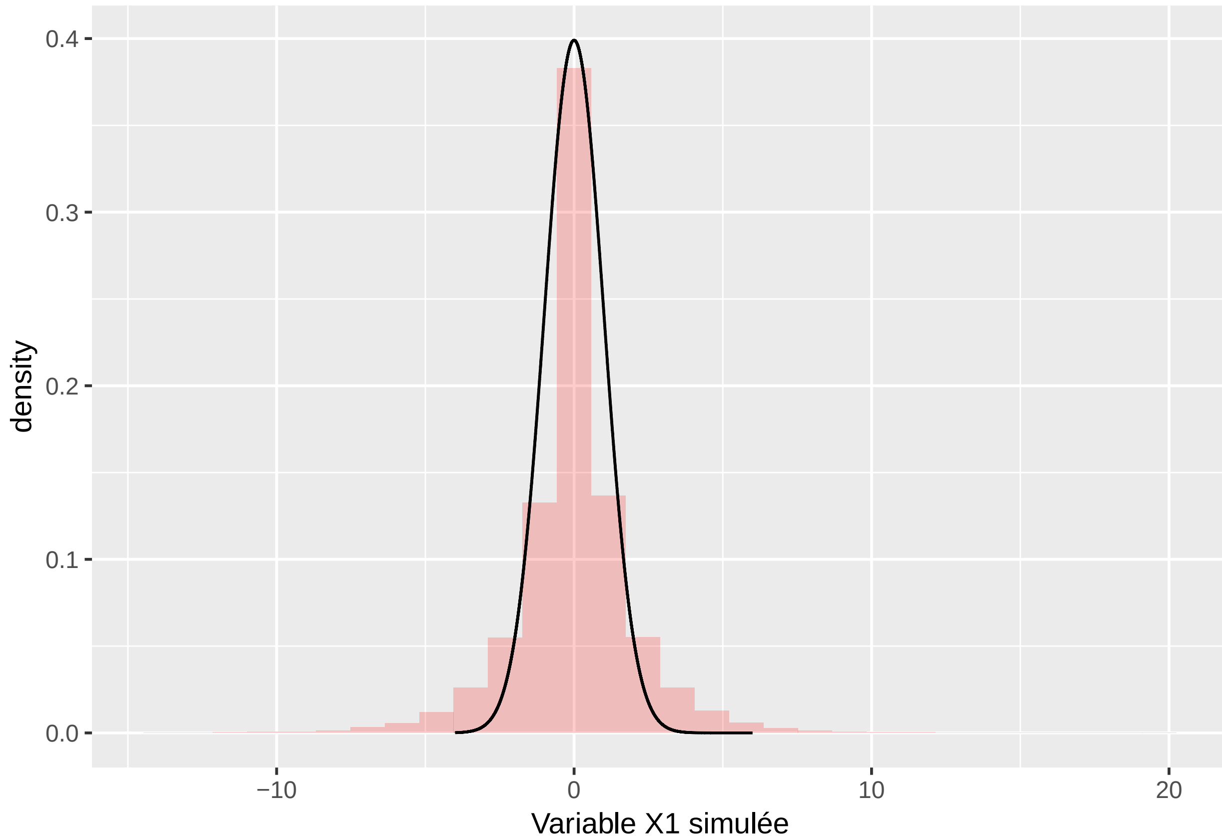


4. En utilisant la question 1., simuler un N-échantillon de X1 de taille N = 10000

5. Tracer la densité empirique de votre échantillon de la question précédente et le comparer avec la densité théorique

```
DF5 <- data.frame(X1,y)
plot5 <- ggplot(DF5,aes(x=DF5$X1)) + geom_histogram(aes(y=..density..),fill = "red", alpha = 0.2) +
  geom_line(aes(x=ab, y=DF5$y)) + xlab("Variable X1 simulée")
plot5
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

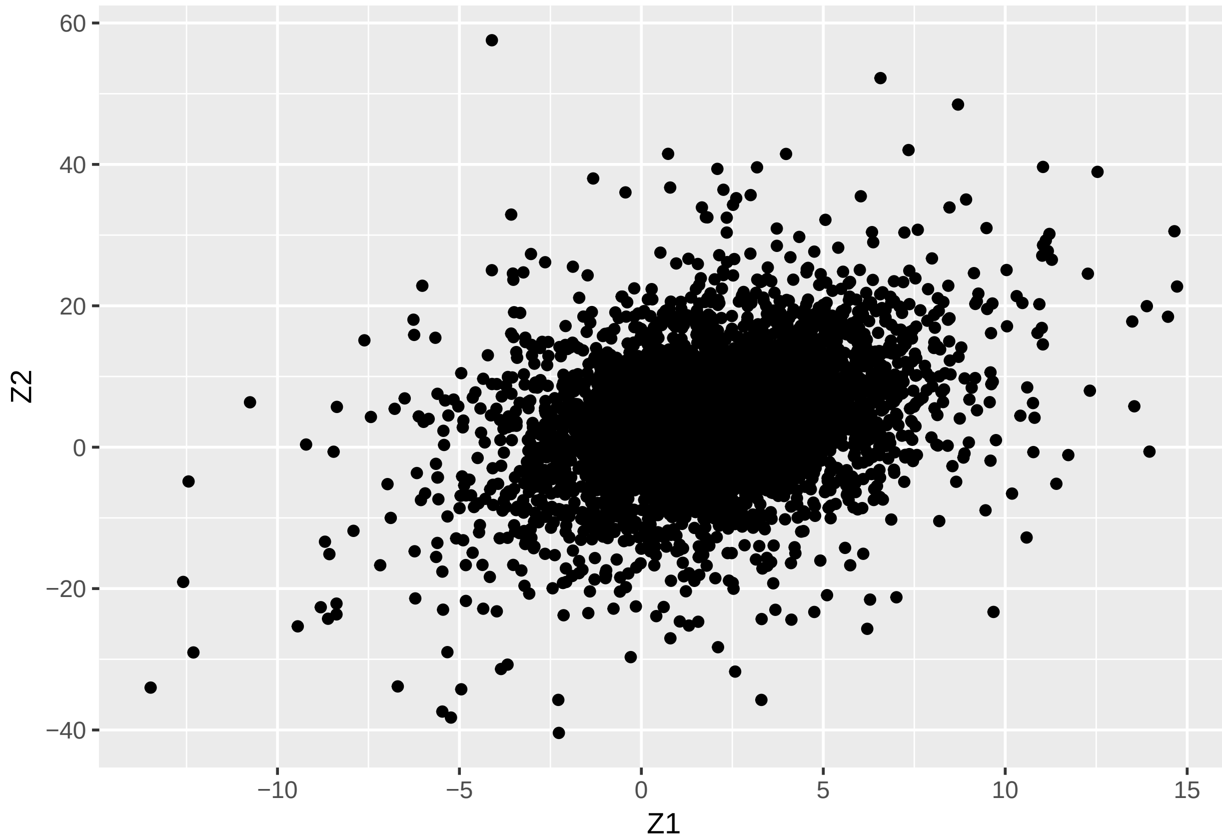


On constate bien que la densité empirique simulée correspond bien à la densité théorique.

6.

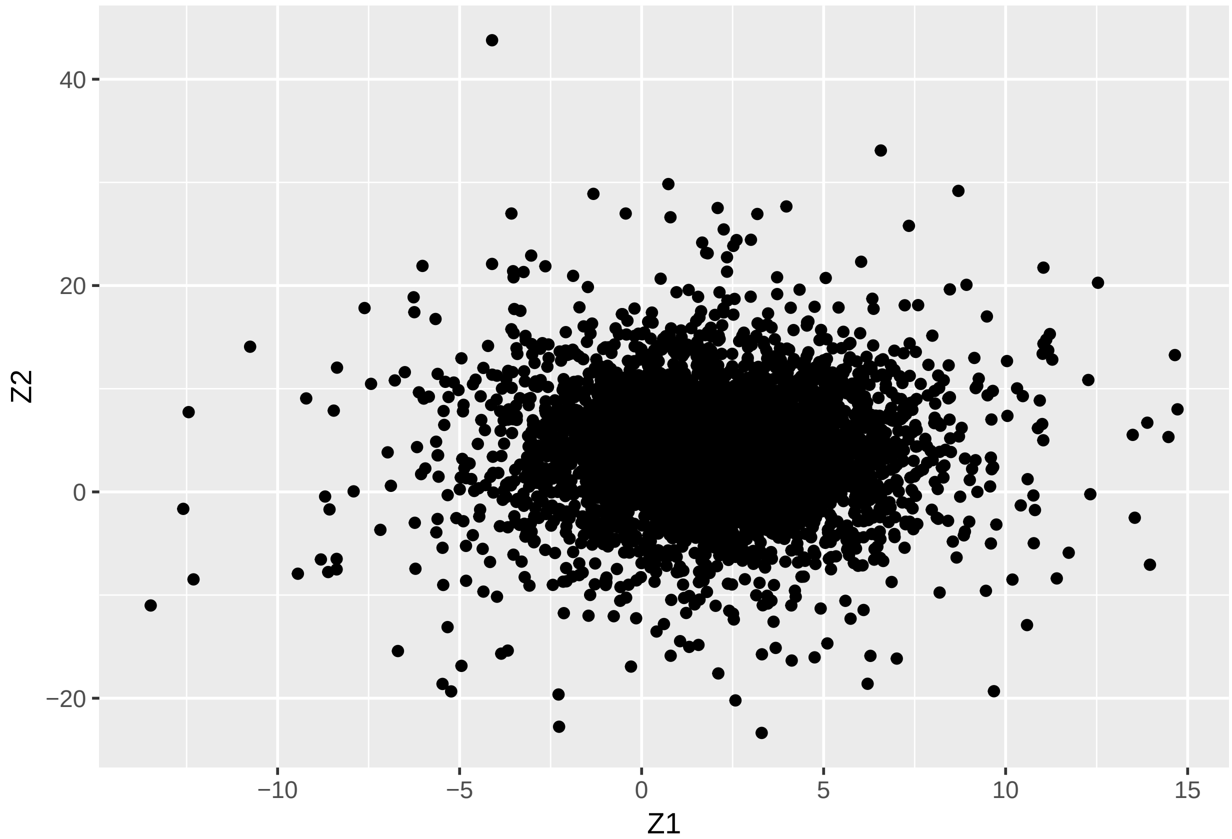
Pour p=0.1

```
sigma11 <- 0
sigma12 <- 1
sigma21 <- 3
sigma22 <- 1

sigma1 <- sqrt(sigma11*sigma11 + sigma12*sigma12)
sigma2 <- sqrt(sigma21*sigma21 + sigma22*sigma22)

Z1 <- 2 + sigma11*X1 + sigma12 * X2
Z2 <- 4 + sigma21 * X1 + sigma22 * X2

DF6 <- data.frame(Z1,Z2)
plot6 <- ggplot(DF6,aes(x = DF6$Z1,y=DF6$Z2)) + geom_point() + xlab("Z1") + ylab("Z2")
plot6
```



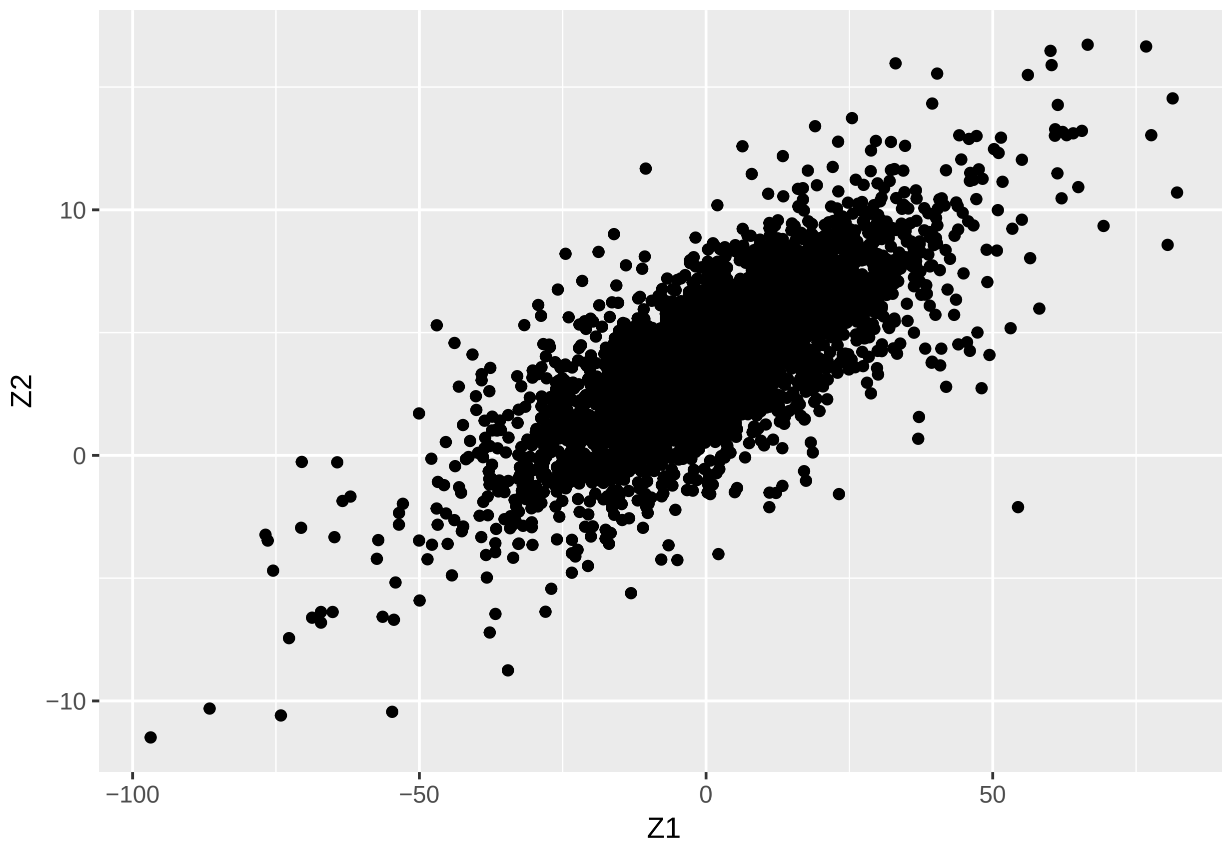
Pour p=0.5

```
sigma11 <- 0
sigma12 <- 1
sigma21 <- 2
sigma22 <- 0

sigma1 <- sqrt(sigma11*sigma11 + sigma12*sigma12)
sigma2 <- sqrt(sigma21*sigma21 + sigma22*sigma22)

Z1 <- 2 + sigma11*X1 + sigma12 * X2
Z2 <- 4 + sigma21 * X1 + sigma22 * X2

DF7 <- data.frame(Z1,Z2)
plot7 <- ggplot(DF7,aes(x = DF7$Z1,y=DF7$Z2)) + geom_point() + xlab("Z1") + ylab("Z2")
plot7
```



Pour p=0.9

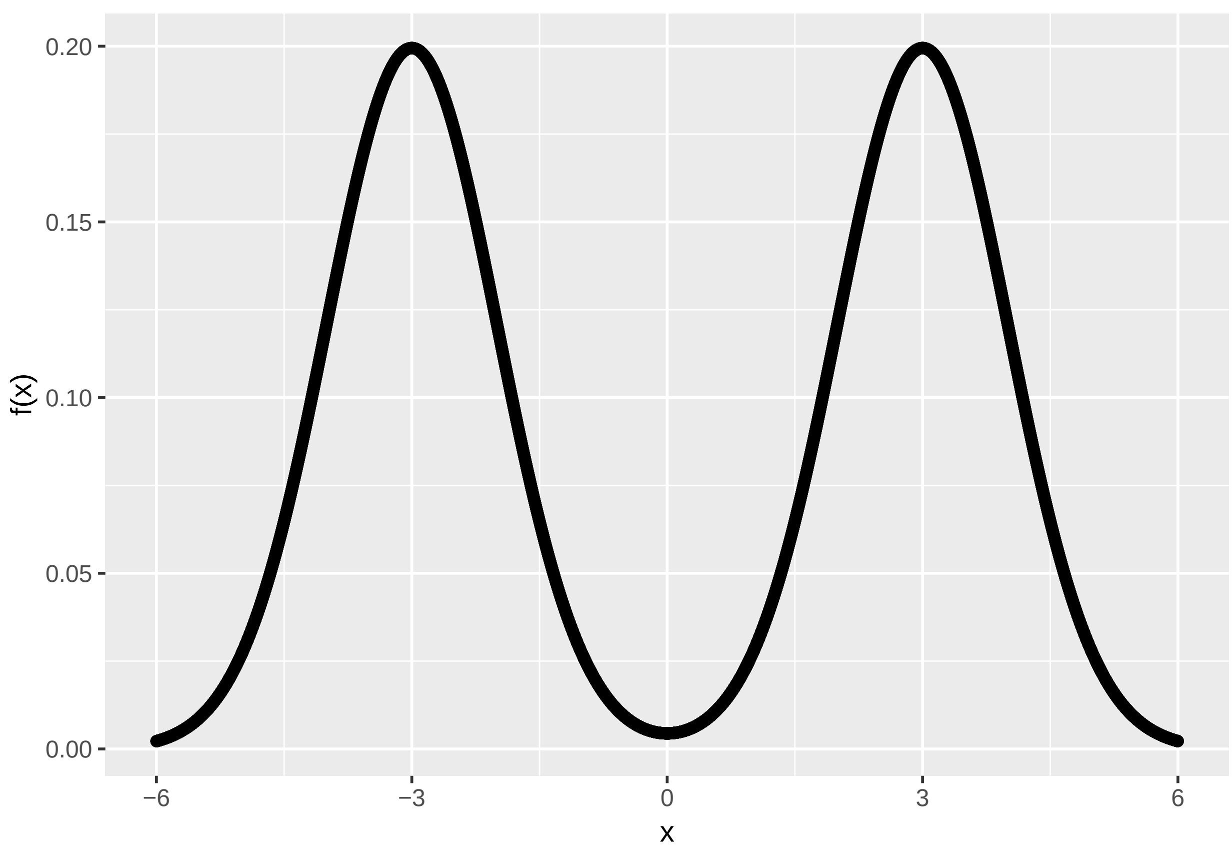
```
sigma11 <- 4
sigma12 <- 1/0.225
sigma21 <- 0
sigma22 <- 1

sigma1 <- sqrt(sigma11*sigma11 + sigma12*sigma12)
sigma2 <- sqrt(sigma21*sigma21 + sigma22*sigma22)

Z1 <- 2 + sigma11*X1 + sigma12 * X2
Z2 <- 4 + sigma21 * X1 + sigma22 * X2

DF8 <- data.frame(Z1,Z2)
plot8 <- ggplot(DF8,aes(x = DF8$Z1,y=DF8$Z2)) + geom_point() + xlab("Z1") + ylab("Z2")
plot8
```


Z2
10
0
−10
−100
−50
0
50
Z1

Exercice 7.

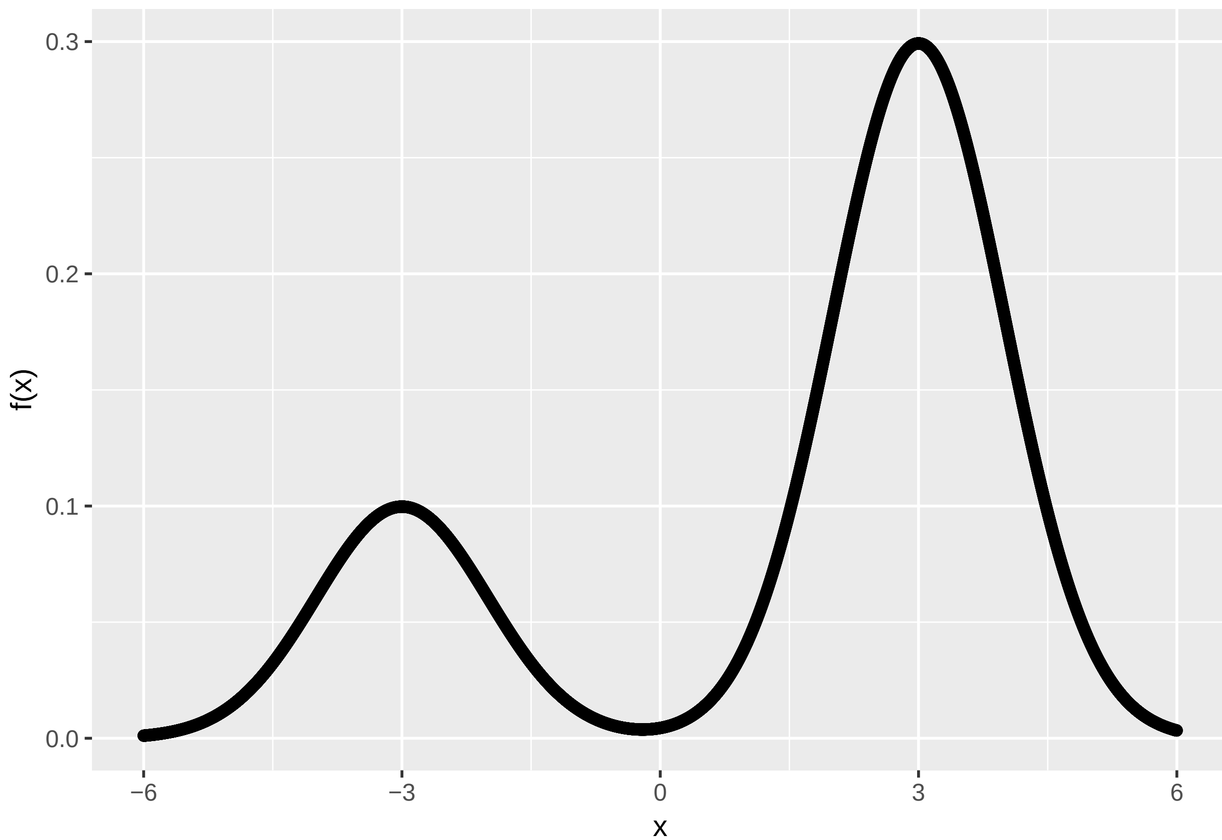
1. Représenter graphiquement f pour (p1,p2) = (1/2,1/2), (p1,p2) = (1/4,3/4), (p1,p2) = (3/4,1/4)

```
ab <- -6
current = ab + 1/1000
for(i in 1:9999) {
  ab <- c(ab, current)
  current = current + 1.2/1000
}

y1 <- vector("numeric", 10000)
for (i in 1:10000) {
  y1[i] = 0.5 * dnorm(ab[i],mean=-3,sd=1) + 0.5 * dnorm(ab[i],mean=3,sd=1)
}

DF9 <- data.frame(ab,y1)
plot9 <- ggplot(DF9,aes(x = DF9$ab,y=DF9$y1)) +
  geom_point() + xlab("x") + ylab("f(x)")
plot9
```



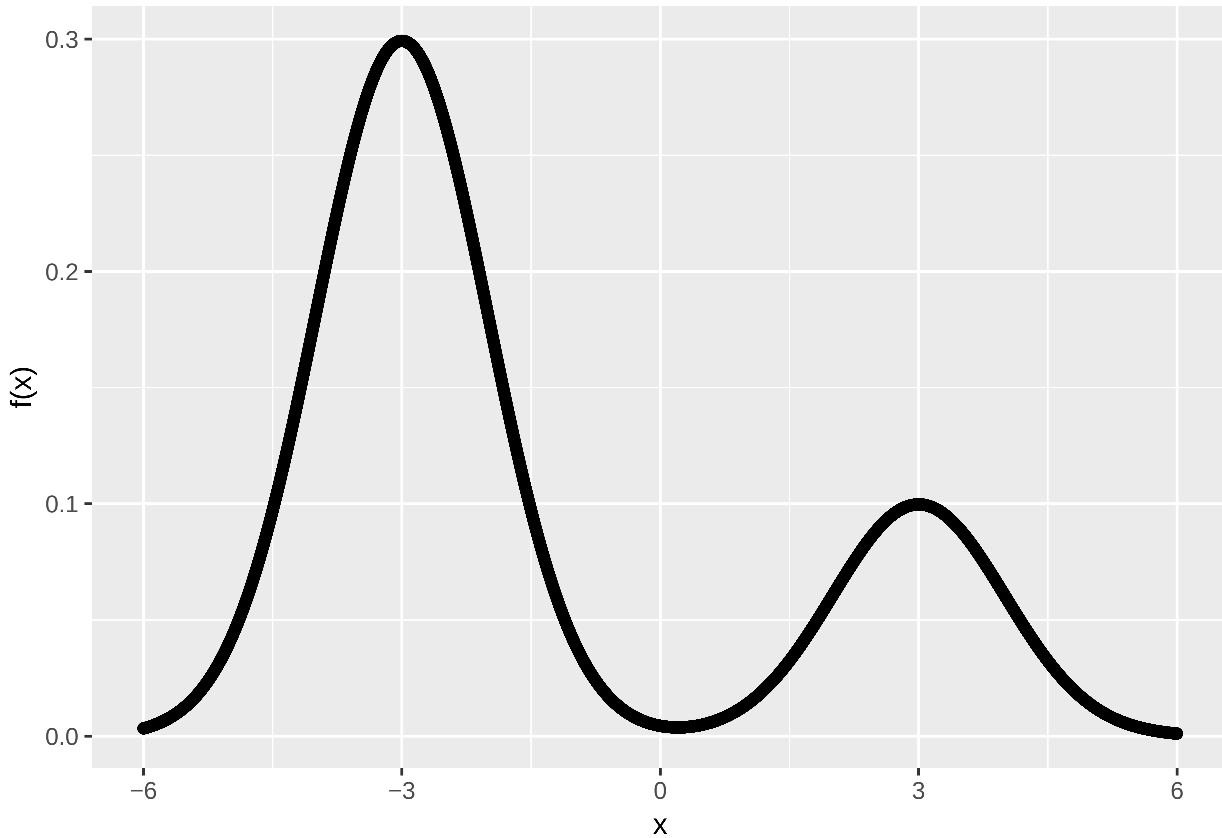

```
ab <- -6
current = ab + 1/1000
for(i in 1:9999) {
```

```
  ab <- c(ab, current)
  current = current + 1.2/1000
}

y2 <- vector("numeric", 10000)
for (i in 1:10000) {
  y2[i] = 0.25 * dnorm(ab[i],mean=-3,sd=1) + 0.75 * dnorm(ab[i],mean=3,sd=1)
}

DF10 <- data.frame(ab,y2)
plot10 <- ggplot(DF10,aes(x = DF10$ab,y=DF10$y2)) + geom_point() + xlab("x") + ylab("f(x)")
plot10
```




```
ab <- -6
current = ab + 1/1000
for(i in 1:9999) {
  ab <- c(ab, current)
  current = current + 1.2/1000
}

y3 <- vector("numeric", 10000)
for (i in 1:10000) {
  y3[i] = 0.75 * dnorm(ab[i],mean=-3,sd=1) + 0.25 * dnorm(ab[i],mean=3,sd=1)
}

DF11 <- data.frame(ab,y3)
```

```
plot11<- ggplot(DF11,aes(x = DF11$ab,y=DF11$y3)) + geom_point() + xlab("x") + ylab("f(x)")
plot11
```

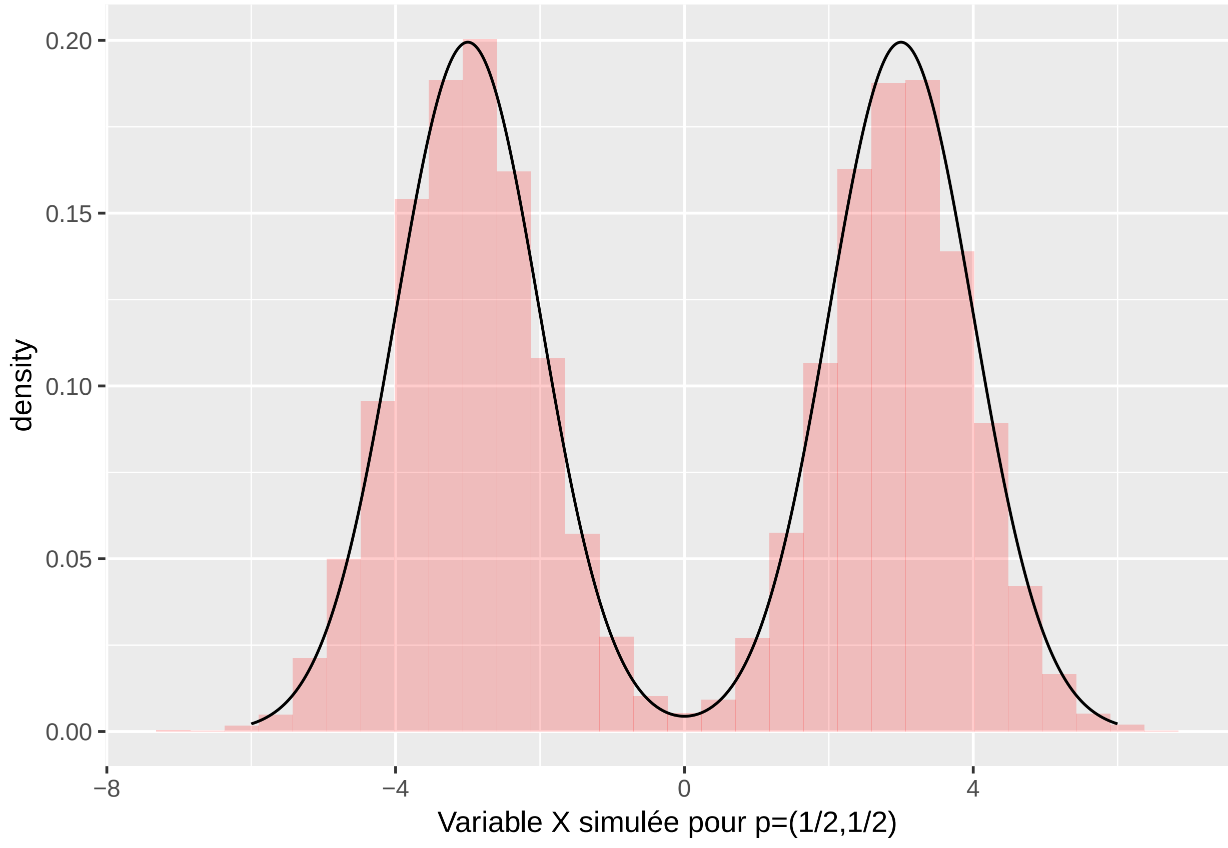


2. Générer un échantillon de taille N = 10000 de la loi de X pour chaque valeur de (p1,p2) et représenter les densités empiriques associées. Commenter les graphes obtenus

```
exo7 <- function(p1) {
  mu <- runif(1,0,1)
  if(mu < p1) {
    R <- rexp(1,rate=0.5)
    TETA <- runif(1,0,2*3.14)
    X <- -3 + sqrt(R) * cos(TETA)
  }
  else {
    R <- rexp(1,rate=0.5)
    TETA <- runif(1,0,2*3.14)
    X <- 3 + sqrt(R) * cos(TETA)
  }
  for (i in 1:9999) {
    mu <- runif(1,0,1)
    if(mu < p1) {
      R <- rexp(1,rate=0.5)
      TETA <- runif(1,0,2*3.14)
```

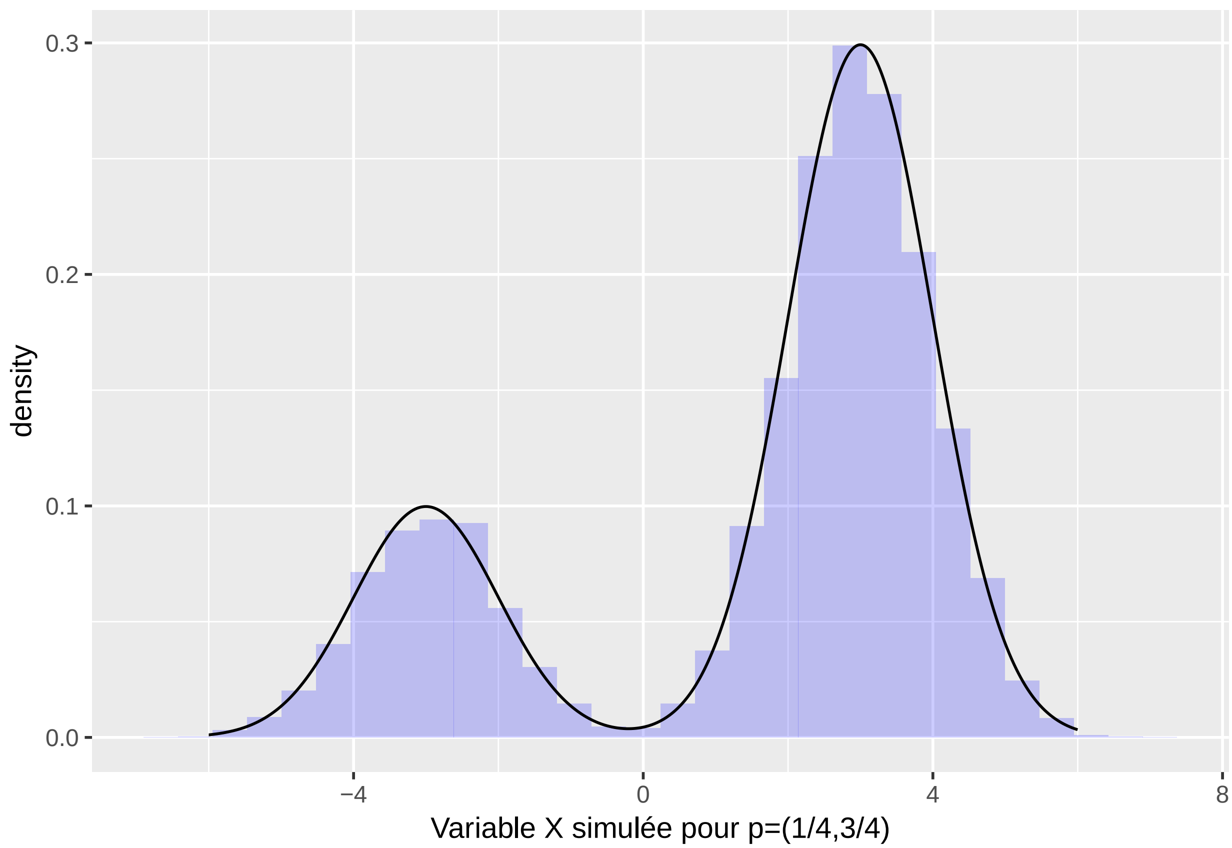
```
      X <- c(X,-3 + sqrt(R) * cos(TETA))
    }
    else {
      R <- rexp(1,rate=0.5)
      TETA <- runif(1,0,2*3.14)
      X <-c(X,3 + sqrt(R) * cos(TETA))
    }
  }
  return(X)
}

DF12 <- data.frame(exo7(0.5),y1)
plot12 <- ggplot(DF12,aes(x=DF12$exo7.0.5.)) + geom_histogram(aes(y=..density..),fill = "red", alpha = 0
  geom_line(aes(x=ab, y=DF12$y1)) + xlab("Variable X simulée pour p=(1/2,1/2)")
plot12
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



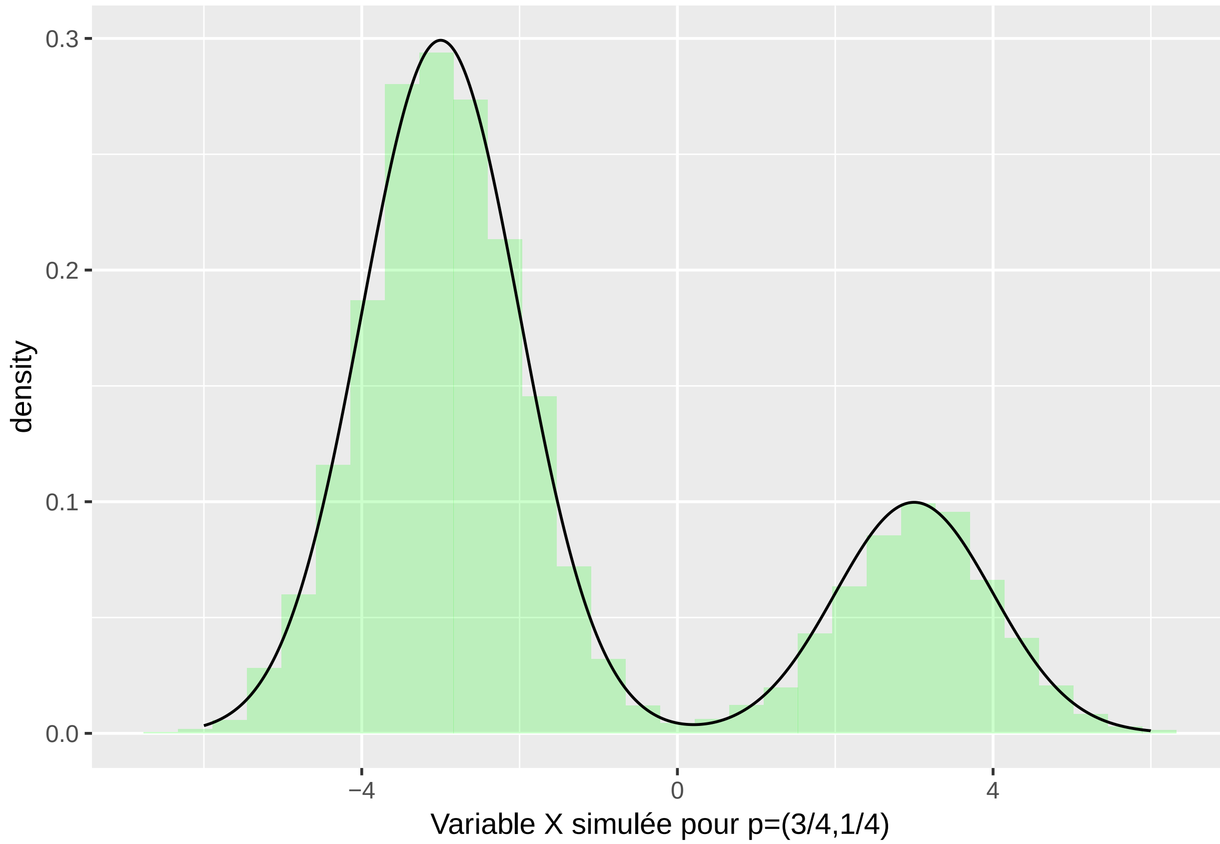

```
DF13 <- data.frame(exo7(0.25),y2)
plot13 <- ggplot(DF13,aes(x=DF13$exo7.0.25.)) + geom_histogram(aes(y=..density..),fill = "blue", alpha =
  geom_line(aes(x=ab, y=DF13$y2)) + xlab("Variable X simulée pour p=(1/4,3/4)")
plot13
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
DF14 <- data.frame(exo7(0.75),y3)
plot14 <- ggplot(DF14,aes(x=DF14$exo7.0.75.)) + geom_histogram(aes(y=..density..),fill = "green", alpha
  geom_line(aes(x=ab, y=DF14$y3)) + xlab("Variable X simulée pour p=(3/4,1/4)")
plot14
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



On constate bien que les densités empiriques simulées correspondent bien aux densités théoriques.