ZeyuCHEN_MSIM

Zeyu CHEN

2019/3/20

Zeyu CHEN numbre:10

Avant de faire tous les question ,on génere d'abord les réalisations de la loi normal par la méthode de rejet et la loi exponentielle qu'on a vu pendant le tp

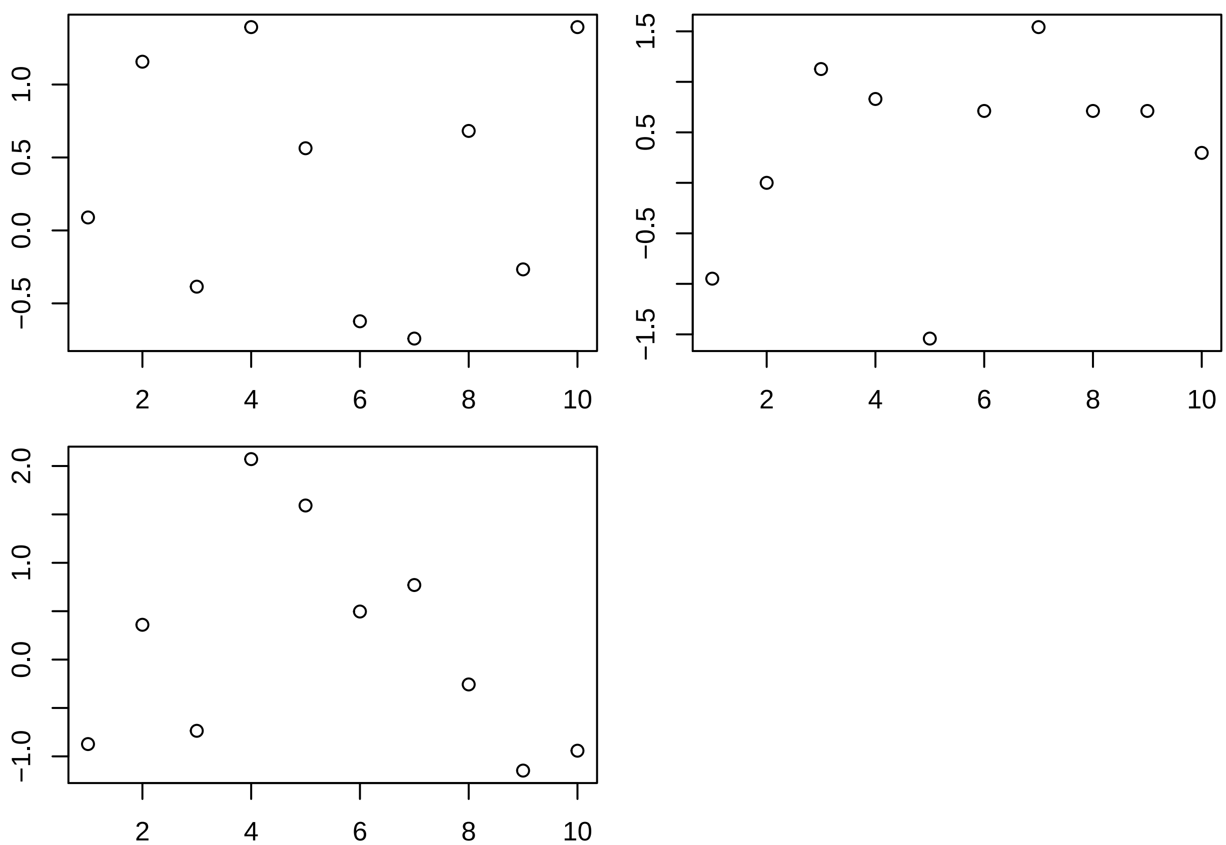
```
pi <- 3.1415926
rexpoentielle <- function(N,lambda)
{
  X <- rep(0,N)
  for(i in 1:N)
  {
  u <- runif(1)
  X[i] <- -log(u)/lambda
  }
  return(X)
}

r_abs_normal <- function(N)
{
  pi <- 3.1415926
  X <- rep(0,N)
  i <- 1
  for(i in 1:N)
  {
    u <- runif(1)
    Y <- rexpoentielle(1,1)
    c <- sqrt(2*exp(1)/pi)
    fY <- 2/sqrt(2*pi)*exp((-1/2)*Y^2)
    gY <- exp(-Y)
    hY <- fY/(c*gY)
    while(u>hY)
    {
      u <- runif(1)
      Y <- rexpoentielle(1,1)
      c <- sqrt(2*exp(1)/pi)
      fY <- 2/sqrt(2*pi)*exp((-1/2)*Y^2)
      gY <- exp(-Y)
      hY <- fY/(c*gY)
    }
    X[i]=Y
  }
  return(X)
}
r_normal <- function(N)
{
  X <- rep(0,N)
  for(i in 1:N)
```

```
  {
    u <- runif(1)
    if(u<1/2)
    {
      X[i] <-  r_abs_normal(1)
    }
    else{
      X[i] <- -r_abs_normal(1)
    }
  }
  return (X)
}
```

EX1.1

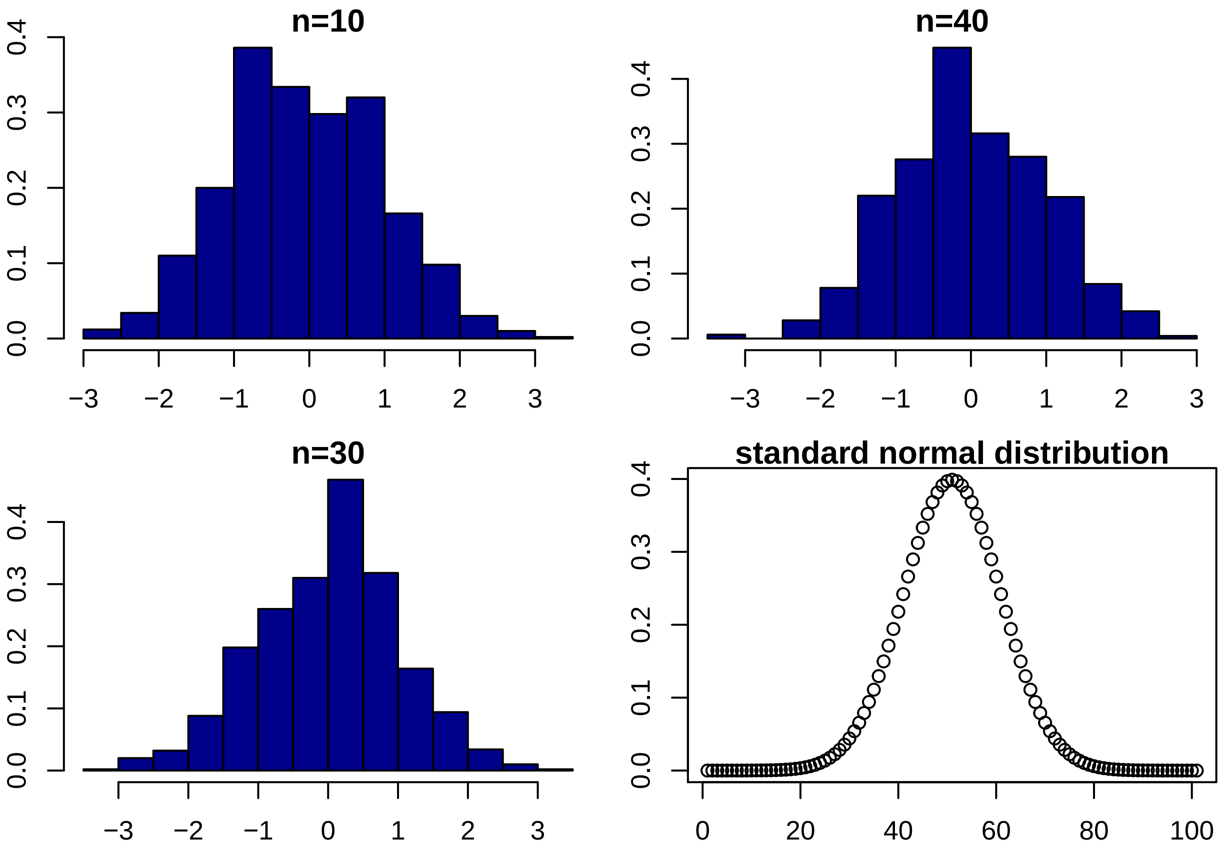
```
a <- vector(length = 4)
a[1]=-0.5
a[2]=-1
a[3]=1.5
a[4]=2
p <- vector(length = 4)
p[1]=1/4
p[2]=1/4
p[3]=1/8
p[4]=3/8

#On calcule le mu et sigma par les formules
mu <- a[1]*p[1]+a[2]*p[2]+a[3]*p[3]+a[4]*p[4]
sig<- sqrt((a[1]^2)*p[1]+(a[2]^2)*p[2]+(a[3]^2)*p[3]+(a[4]^2)*p[4]-(mu^2))

c <- rep(0,4)
for(i in 1:4)
{
  for(j in 1:i)
    {
      c[i] <-c[i]+p[j]
    }
}

#La fonction pour generer un échantillon Xn
rloidiscret <- function(N)
{
  X <- vector(length = N)
  for(i in 1:N)
  {
    u <- runif(1)
    k<-1
    while (u>c[k])
      k<-k+1
    X[i] <- a[k]
  }
  return(X)
```

```
}

#La fonction pour generer un échantillon Zn
rZn <- function(N,n)
{
  Z <- vector(length = N)
  for(i in 1:N)
  {
    X <- rloidiscret(n)
    Z[i] <- sqrt(n)*(mean(X)-mu)/sig
  }
   return(Z)
}

par(mfcol=c(2,2))
par(mar = c(2, 2, 1, 1))
Z_10<- rZn(10,10)
plot(Z_10)

Z_30<- rZn(10,30)
plot(Z_30)

Z_40<- rZn(10,40)
plot(Z_40)
```



EX1.2

```
N <- 1000
Z_10<- rZn(N,10)
Z_30<- rZn(N,30)
Z_40<- rZn(N,40)
x = seq(-5,5,0.1)
pi=3.1415
fx <- 1/sqrt(2*pi)*exp(-1/2*(x^2))
par(mfcol=c(2,2))
par(mar = c(2.7, 2, 1, 1))
hist(Z_10,freq= FALSE, col= "Darkblue",main = "n=10", ylab= "Frequence")
hist(Z_30,freq= FALSE, col= "Darkblue",main = "n=30", ylab= "Frequence")
hist(Z_40,freq= FALSE, col= "Darkblue",main = "n=40", ylab= "Frequence")
plot(fx ,main = "standard normal distribution")
```



EX1.3

On constate que plus n est grand , plus le plot ressemble à la loi normale centrée réduite

EX2.1

Pour choisir des valeur pour que ρ prenne y , il faut just fixer trois valeur parmi ces quatre sigma, donc , on suppose que $\sigma_{11} = 1$, $\sigma_{22} = 1$, $\sigma_{21} = 0$ et $\sigma_{12} = x$

Et après on a

$$\sigma_{12} = x = \frac{y}{\sqrt{1-y^2}}$$

```r
x_sol <- function(y)
{
  return (y/sqrt(1-y^2))
}
sig_12_0.05 = x_sol(0.05)
sig_12_0.5 = x_sol(0.5)
sig_12_0.95 = x_sol(0.95)
sig_12_0.05
```

```
## [1] 0.05006262
```

```r
sig_12_0.5
```

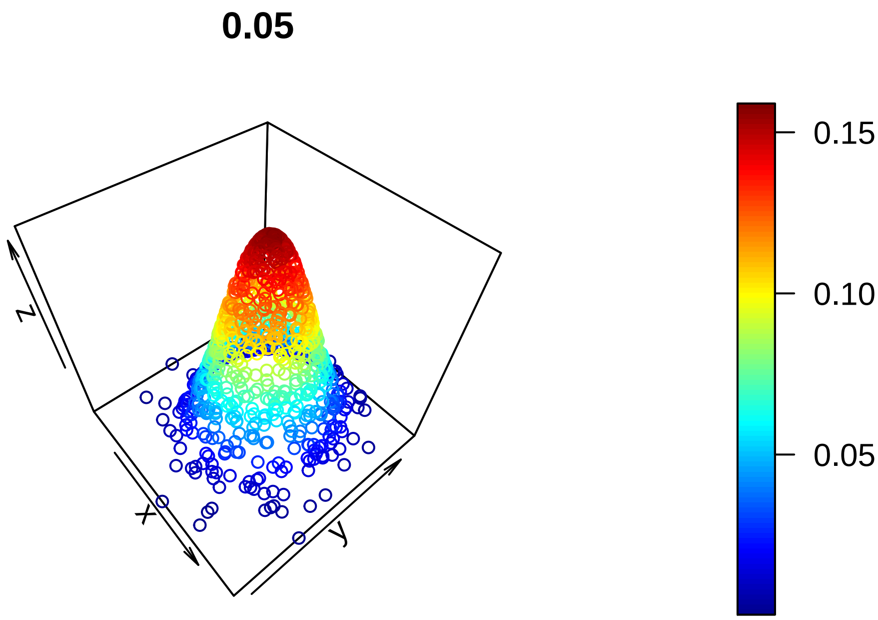
```
## [1] 0.5773503
```

```r
sig_12_0.95
```

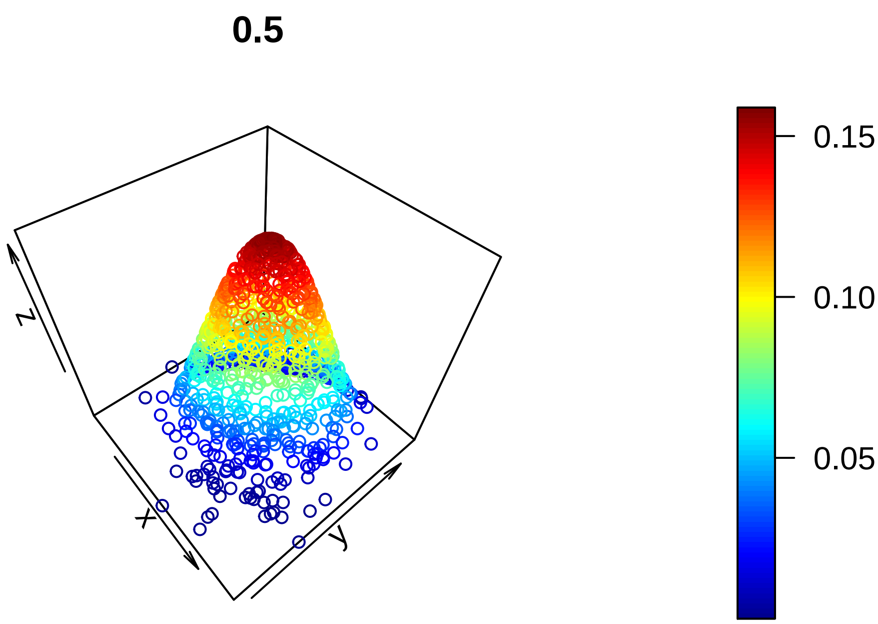
```
## [1] 3.042435
```

EX2.2

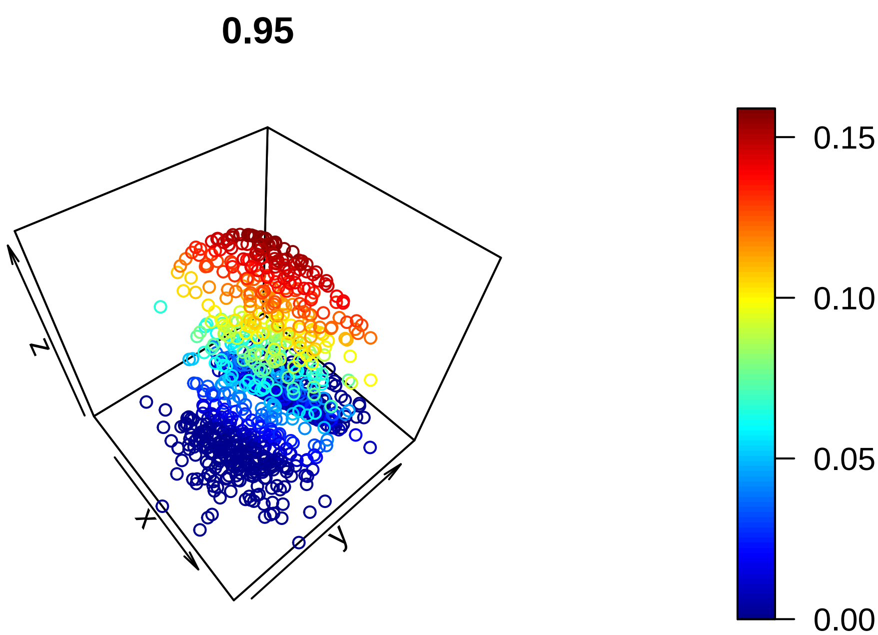
```r

sig11<-1
sig21<-0
sig22<-1
#function pour plot f_Z d'apres le rho different
plot_f_z1_z2 <- function(rho,z1,z2)
{
  rho<-rho
  if(rho==0.05)
    sig12<-sig_12_0.05
  if(rho==0.5)
    sig12<-sig_12_0.5
  if(rho==0.95)
    sig12<-sig_12_0.95
  sig1 <- sqrt(sig11^2+sig12^2)
  sig2 <- sqrt(sig21^2+sig22^2)
  z <- (z1^2)/(sig1^2)-(2*rho*z1*z2)/(sig1*sig2)+(z2^2)/(sig2^2)
  f <- 1/(2*pi*sig1*sig2*sqrt(1-rho^2))*exp(-z/(2*(1-rho^2)))
  plot3D::points3D(z1,z2,f,main=rho,phi=50, theta=50,cex=0.8)
}

z1 = r_normal(1000)
z2 = r_normal(1000)
plot_f_z1_z2(0.05,z1,z2)
```




```
plot_f_z1_z2(0.5,z1,z2)
```




```
plot_f_z1_z2(0.95,z1,z2)
```



EX2.3

```
Mu <- c(0,0)


r_z1_z2 <- function(N,mu = Mu,sigma = Sigma)
{
  #R <- rexpoentielle(N,1/2)
  pi <- 3.1415926
 # Theta <- runif(N,min=0,max = 2*pi)
  X1 <- r_normal(1000)
  X2 <- r_normal(1000)
  Z1 <- mu[1]+sigma[1,1]*X1+sigma[1,2]*X2
  Z2 <- mu[2]+sigma[2,1]*X1+sigma[2,2]*X2
  Z <- array(data=c(Z1,Z2),dim=c(N,2))
  return (Z)
}

Sigma_0.05 <- matrix(c(sig11,sig_12_0.05,sig21,sig22),2,2)
Sigma_0.5 <- matrix(c(sig11,sig_12_0.5,sig21,sig22),2,2)
Sigma_0.95 <- matrix(c(sig11,sig_12_0.95,sig21,sig22),2,2)

#le resulata quand on prends rho =0.05
r_z1_z2(10,Mu,Sigma_0.05)
```

```
##              [,1]        [,2]
##  [1,] -2.17555609  0.37778527
##  [2,] -1.06488369 -2.16593316
##  [3,]  0.39551914 -0.22582559
##  [4,] -0.05248020 -1.10444008
##  [5,] -0.09008725 -1.08737146
##  [6,] -0.50330033 -0.25753140
```

```
##  [7,]  0.66062922  0.53871732
##  [8,] -0.09023214 -0.60691702
##  [9,]  0.66578255 -0.06743265
## [10,] -0.06246367 -0.02564031
```

```r
#le resulata quand on prends rho =0.5
r_z1_z2(10,Mu,Sigma_0.5)
```

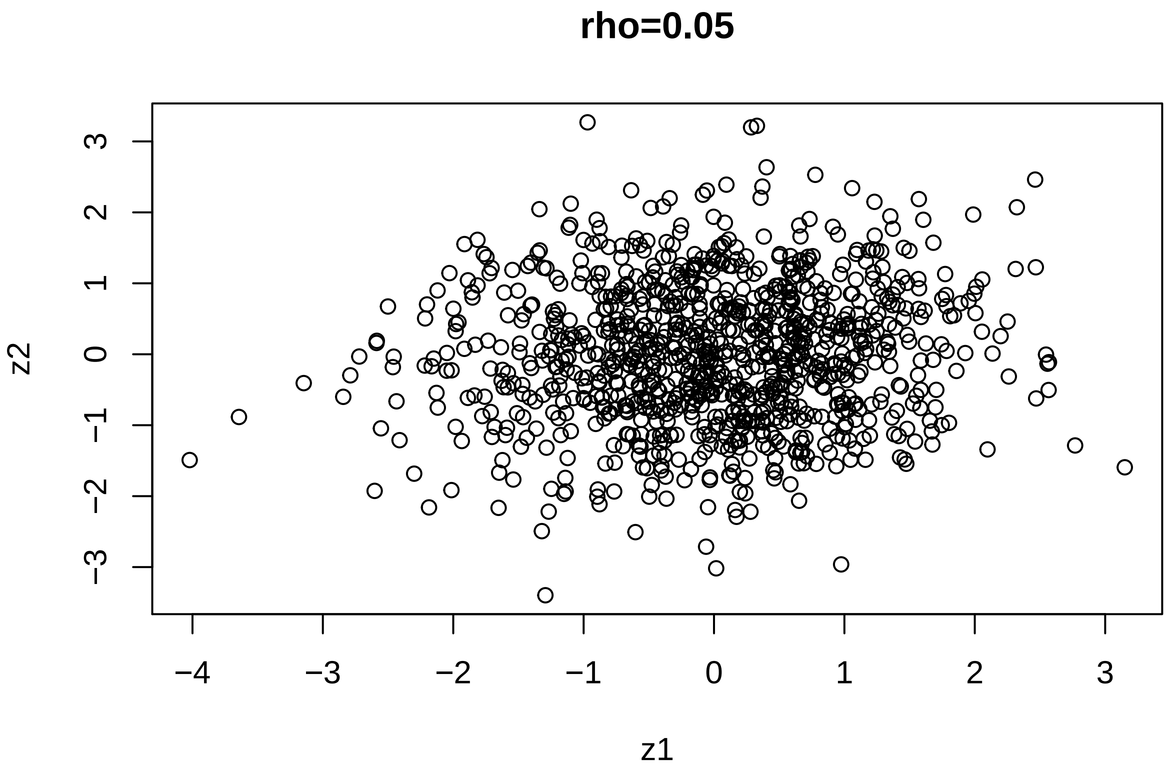
```
##             [,1]       [,2]
##  [1,]  1.2770543 -0.3276311
##  [2,]  1.1896555 -0.2460476
##  [3,]  0.5387074  1.6820726
##  [4,] -1.7735630  2.1654366
##  [5,]  0.8167551  0.6694246
##  [6,]  1.9193017 -0.1444315
##  [7,]  0.4449434  2.6325206
##  [8,] -1.1313972  1.2002345
##  [9,] -0.4816638  1.1048254
## [10,] -0.6638515  1.0811551
```

```r
#le resulata quand on prends rho =0.95
r_z1_z2(10,Mu,Sigma_0.95)
```

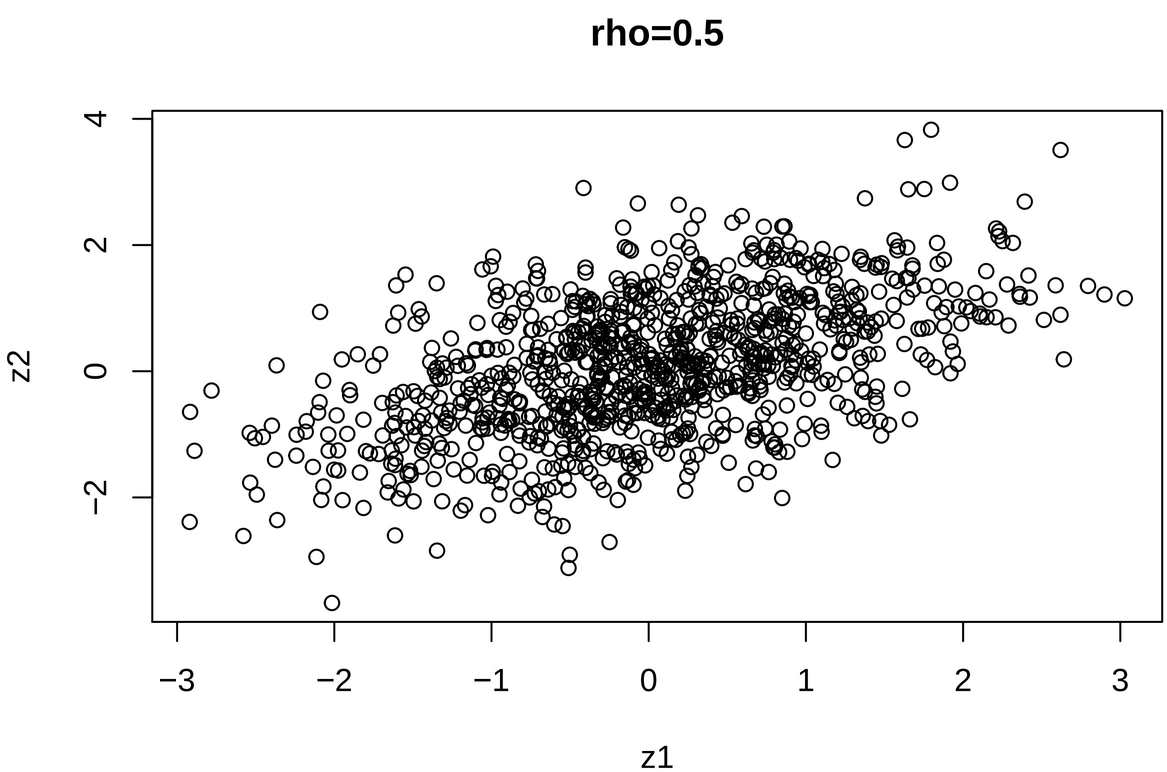
```
##              [,1]        [,2]
##  [1,]  0.06881001 -0.26760016
##  [2,] -1.03978813  0.26664096
##  [3,] -1.93125390  0.99072487
##  [4,] -1.87799589 -0.79714989
##  [5,] -0.12478526 -0.25354870
##  [6,]  1.28501218  1.97101159
##  [7,] -1.28344212  0.05621238
##  [8,] -0.73977503 -0.63049255
##  [9,]  1.78351719  0.23425883
## [10,] -1.22849940  1.84651423
```

EX2.4

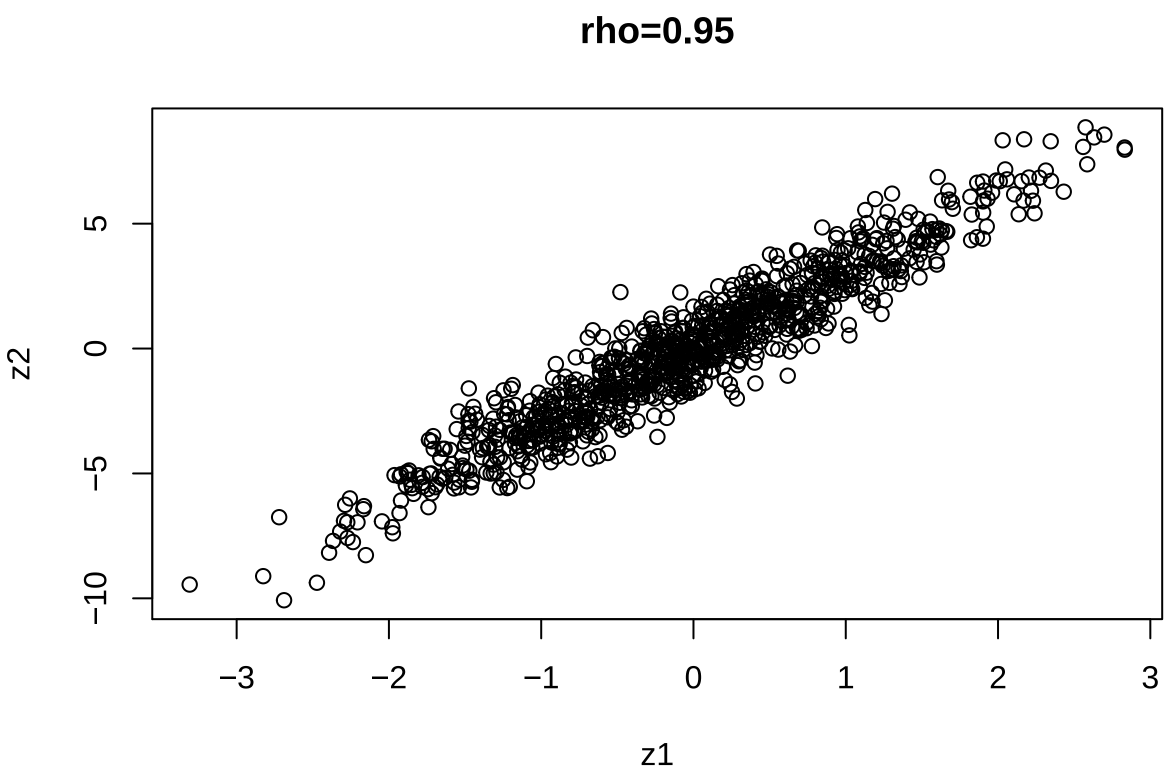
```r
N = 1000

plot(r_z1_z2(N,Mu,Sigma_0.05),ylab = "z2",xlab="z1",main = "rho=0.05")
```




```
plot(r_z1_z2(N,Mu,Sigma_0.5),ylab = "z2",xlab="z1",main = "rho=0.5")
```

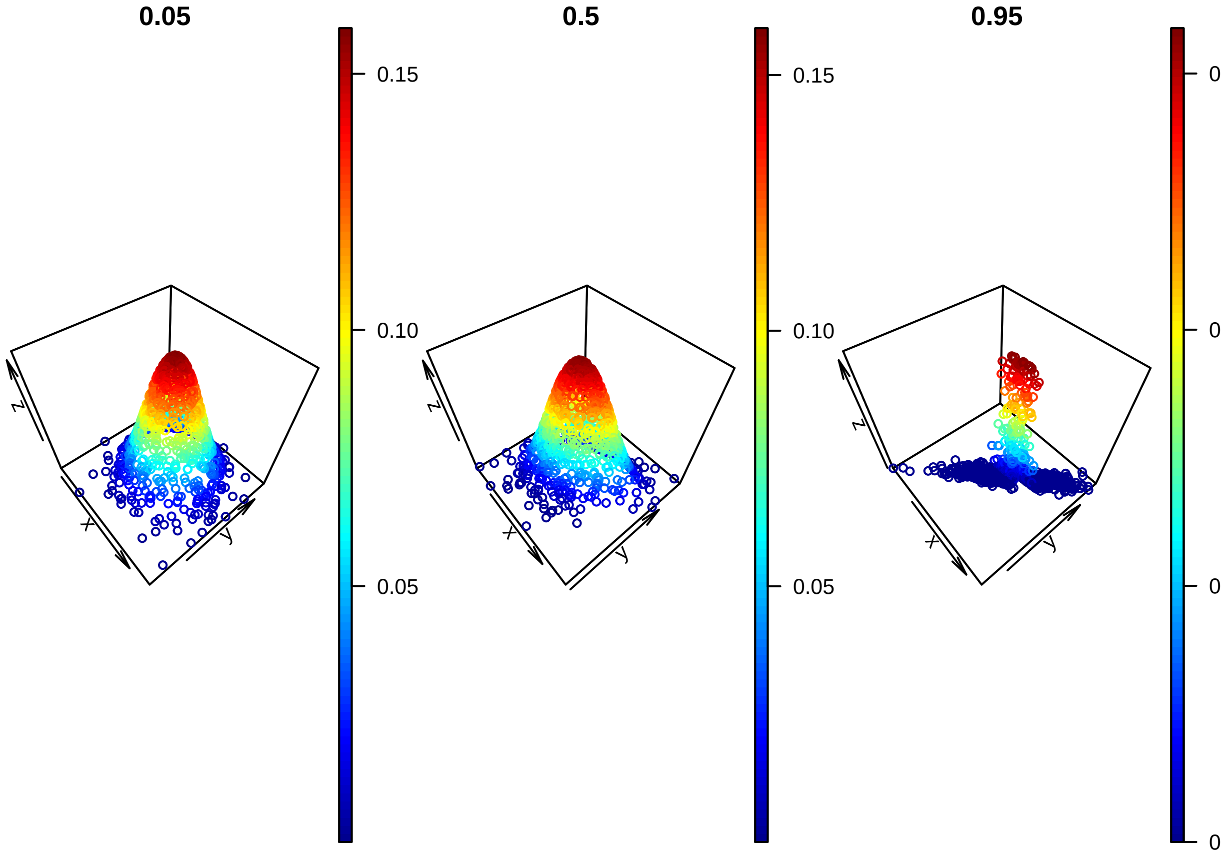



```
plot(r_z1_z2(N,Mu,Sigma_0.95),ylab = "z2",xlab="z1",main = "rho=0.95")
```

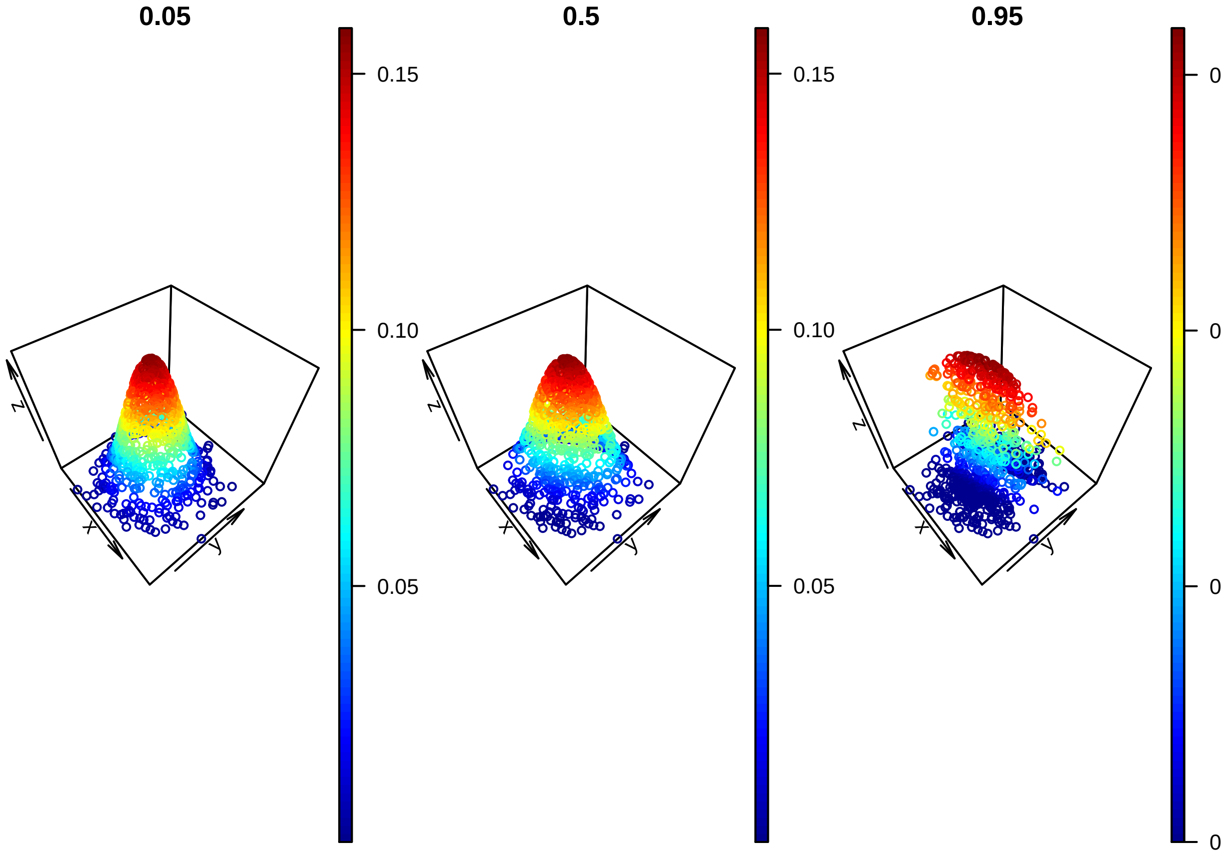


EX2.5

```
N=1000

#On plot pour les densites empirique
par(mfcol=c(1,3))
par(mar = c(1, 1, 1, 1))
Z <- r_z1_z2(N,Mu,Sigma_0.05)
z1 = Z[,1]
z2 = Z[,2]
plot_f_z1_z2(0.05,z1,z2)

Z <- r_z1_z2(N,Mu,Sigma_0.5)
z1 = Z[,1]
z2 = Z[,2]
plot_f_z1_z2(0.5,z1,z2)

Z <- r_z1_z2(N,Mu,Sigma_0.95)
z1 = Z[,1]
z2 = Z[,2]
plot_f_z1_z2(0.95,z1,z2)
```




```
#On plot pour les densites theoriques
par(mfcol=c(1,3))
z1 = r_normal(1000)
z2 = r_normal(1000)
plot_f_z1_z2(0.05,z1,z2)
plot_f_z1_z2(0.5,z1,z2)
plot_f_z1_z2(0.95,z1,z2)
```



EX3.1

$E[X] = 1$, on suppose que $P(Y = -6) = x$ et $P(Y = 6) = y$, on les trouve facilement que $x = \frac{1}{4}$ $y = \frac{5}{12}$

EX3.2

$$Var[X] = E(X^2) - E[X]^2 = 29$$

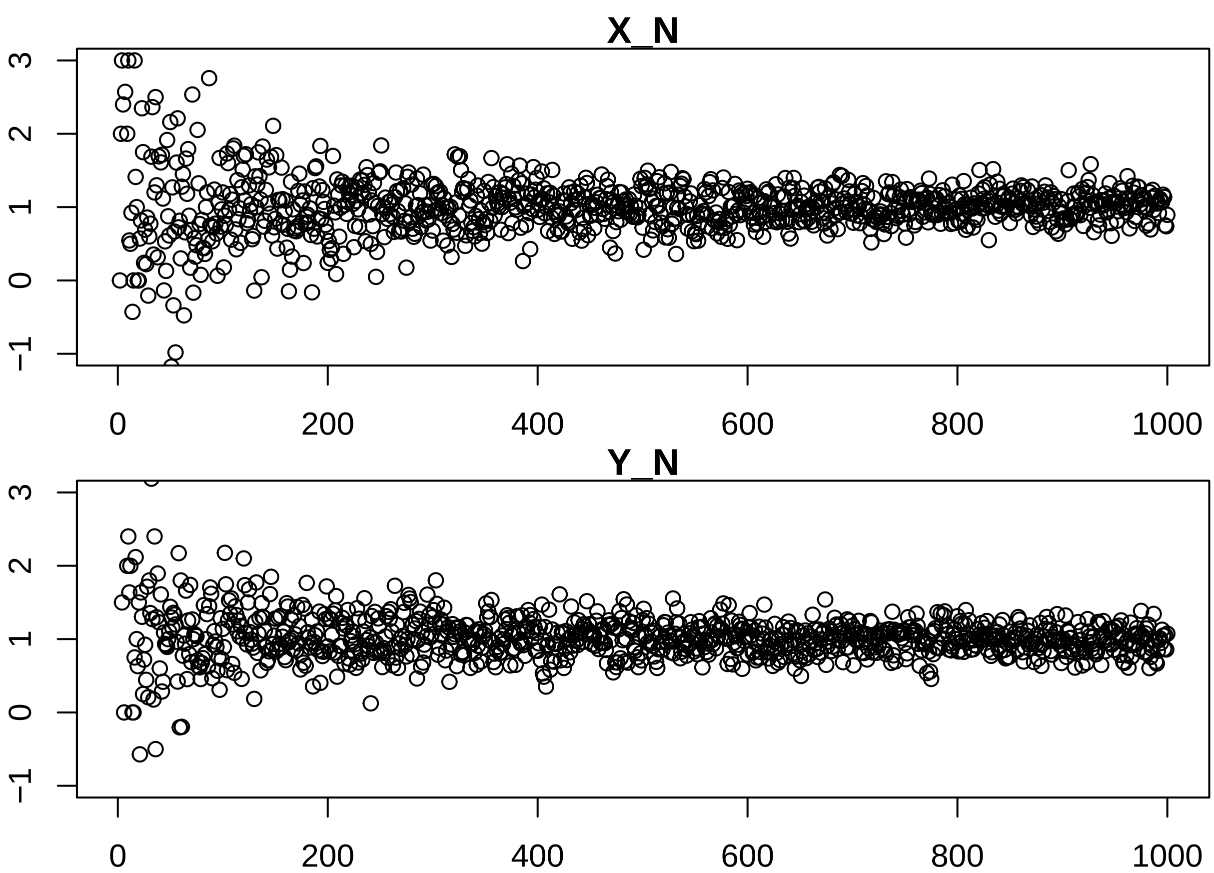
$$Var[Y] = E(Y^2) - E[Y]^2 = 23$$

On constate que Var[Y] < Var[X], Parce que la distribution de Y est plus uniforme

EX3.3

```
r_X_Y <- function(p1,p2,N)
{
  X <- rep(0,N)
  for(i in 1:N)
  {
  u <- runif(1)
  if(u <= p1)
    X[i] = -6
  else if(u<=p2+p1)
```

```
    X[i] = 0
  else
    X[i]= 6
  }
  return(X)
}

plot_X_Y<- function(p1,p2,N)
{
  X <- rep(0,N)
  for(i in 1:N)
  {
    X[i] <-mean(r_X_Y(p1,p2,i))
  }
  return (X)
}

N=1000

par(mfcol=c(2,1))
par(mar = c(2, 2, 1, 1))
X_N <- plot_X_Y(1/3,1/6,N)
plot(X_N,ylim=c(-1,3),main="X_N")
Y_N <- plot_X_Y(1/4,1/3,N)
plot(Y_N,ylim=c(-1,3),main="Y_N")
```

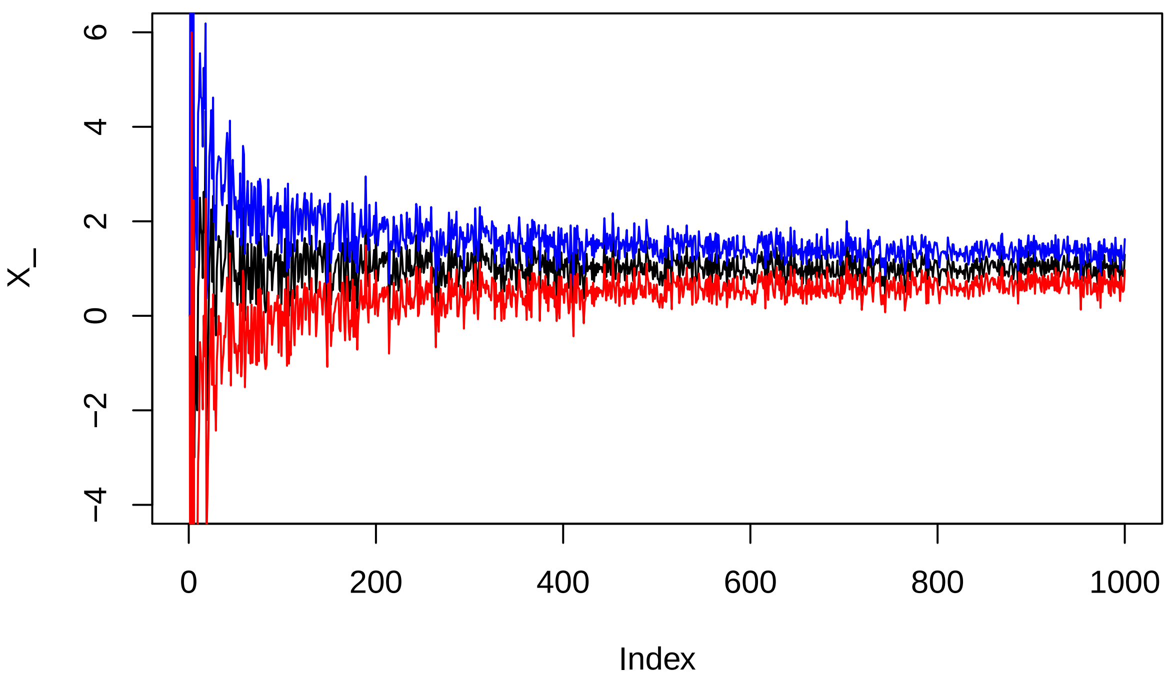


EX3.4

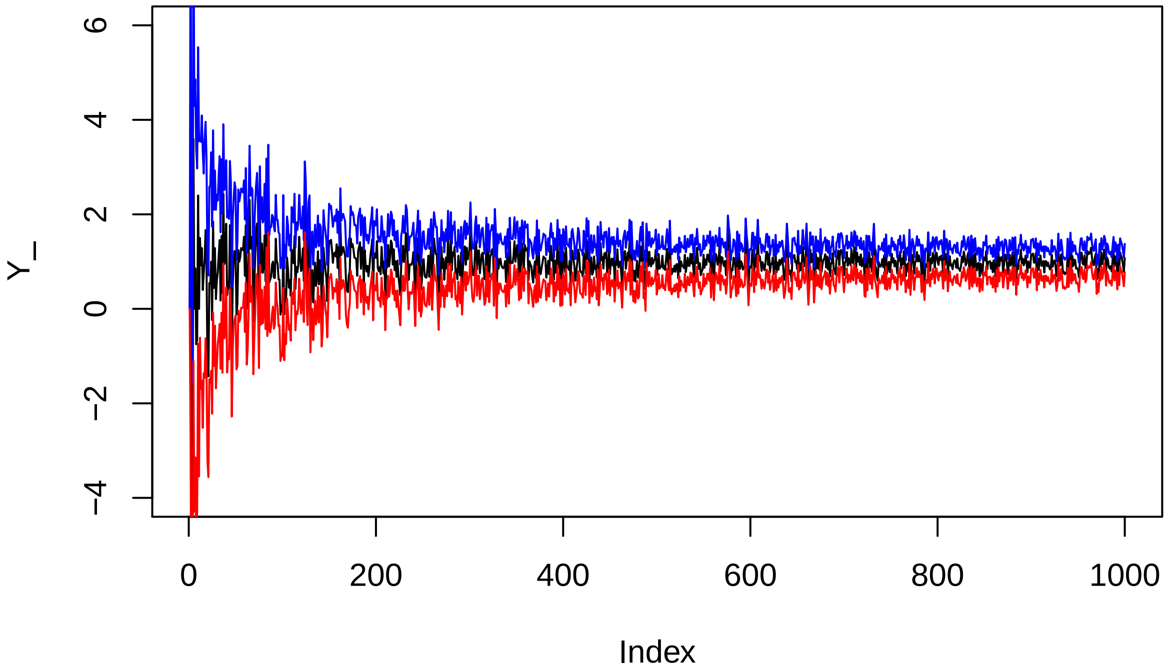
$$IC = [\bar{X}_N - C_\alpha \frac{S_N}{\sqrt{N}}, \bar{X}_N + C_\alpha \frac{S_N}{\sqrt{N}}]$$

On veut que ça soit IC de 95% , ou $C_\alpha = 1.96$

```
N <- 1000
X_ <- rep(0,N)
Y_ <- rep(0,N)
Inf_Ic_X <- rep(0,N)
Sup_Ic_X <- rep(0,N)
Inf_Ic_Y <- rep(0,N)
Sup_Ic_Y <- rep(0,N)
  for(i in 2:N)
  {
    X <- r_X_Y(1/3,1/6,i)
    Y <- r_X_Y(1/4,1/3,i)
    X_[i] <- mean(X)
    Y_[i] <- mean(Y)
    S_N_X<- sqrt(sum((X-X_[i])^2)/(i-1))
    Inf_Ic_X[i] <- X_[i] - 1.96*S_N_X/sqrt(i)
    Sup_Ic_X[i] <- X_[i] + 1.96*S_N_X/sqrt(i)
    S_N_Y<- sqrt(sum((Y-Y_[i])^2)/(i-1))
    Inf_Ic_Y[i] <- Y_[i] - 1.96*S_N_Y/sqrt(i)
    Sup_Ic_Y[i] <- Y_[i] + 1.96*S_N_Y/sqrt(i)
  }

#On plot pour X bar
plot(X_,type="l",ylim =c(-4,6))
lines(Sup_Ic_X,col="blue")
lines(Inf_Ic_X,col="red")
```



```
#On plot pour Y bar
plot(Y_,type="l",ylim =c(-4,6))
lines(Sup_Ic_Y,col="blue")
lines(Inf_Ic_Y,col="red")
```




```
#on constate que Y bar converge plus vite que X ver la moyenne theorique
```

EX4

```
#Step 1 , find pi
Q <- t(matrix(data =c(0.2,0.1,0.5,0.2,0.1,0.6,0.1,0.2,0.3,0.1,0.2,0.4,0.1,0.1,0.1,0.7),nrow = 4,ncol= 4

pi = Q
for(i in 1:500)
{
  pi <- pi%*%Q
}
pi <- pi[1,]


#Step 2 , set X0
X0 <-1

#Step 3 , built Xn
n = 100000
Xn <- rep(1,n)

Xn[1]<-X0
#la fonction pour generer Yn
rYn <- function(X)
{
```

```r
  Yn <- 1
  u <- runif(1)
  c <- Q[X,]

  if(u <= c[1])
    Yn=1
  else if(u<=c[2]+c[1])
    Yn=2
  else if(u<=c[2]+c[1]+c[3])
    Yn=3
  else
    Yn=4

  return(Yn)
}

for(i in 2:n)
{
  u <- runif(1)
  x <- Xn[i-1]
  Yn <- rYn(x)
  h = min(1,(pi[Yn]*Q[Yn,Xn[i-1]])/(pi[Xn[i-1]]*Q[Xn[i-1],Yn]))
  if(u < h)
    Xn[i]=Yn
  else
    Xn[i]=Xn[i-1]
}

#step 4, we have now Markov chain Xn ,using the ergodic theorem now

Approx <- mean(Xn^2)

#Donc , notre estimation est :
Approx
```

```
## [1] 10.13449
```