Simulation Methods

Zeyu CHEN,Valentin GUYONNEAU ,Sovathan SOUN

2019/2/6

Numero Du Groupe n:11

Zeyu CHEN n:10 3 3

Valentin GUYONNEAU n:12 3 3

Sovathana SOUN n:13 3 3

EX1.1

P(U<=p) = p donc P(X=1) = p P(X=0) = 1-p

donc , X suis bien une loi de Bernoulli

EX1.2 et EX1.3

```
rbern <- function(N,p)
{
  X <- rep(0,N)
  for(i in 1:N)
  {
  u <- runif(1)
  if(u <= p)
    X[i] = 1
  }
  return(X)
}
X_100 <- rbern(100,0.3)

X_1000 <- rbern(1000,0.3)

X_10000 <- rbern(10000,0.3)


sum(X_100)
```

```
## [1] 23
```

```
sum(X_1000)
```

```
## [1] 304
```

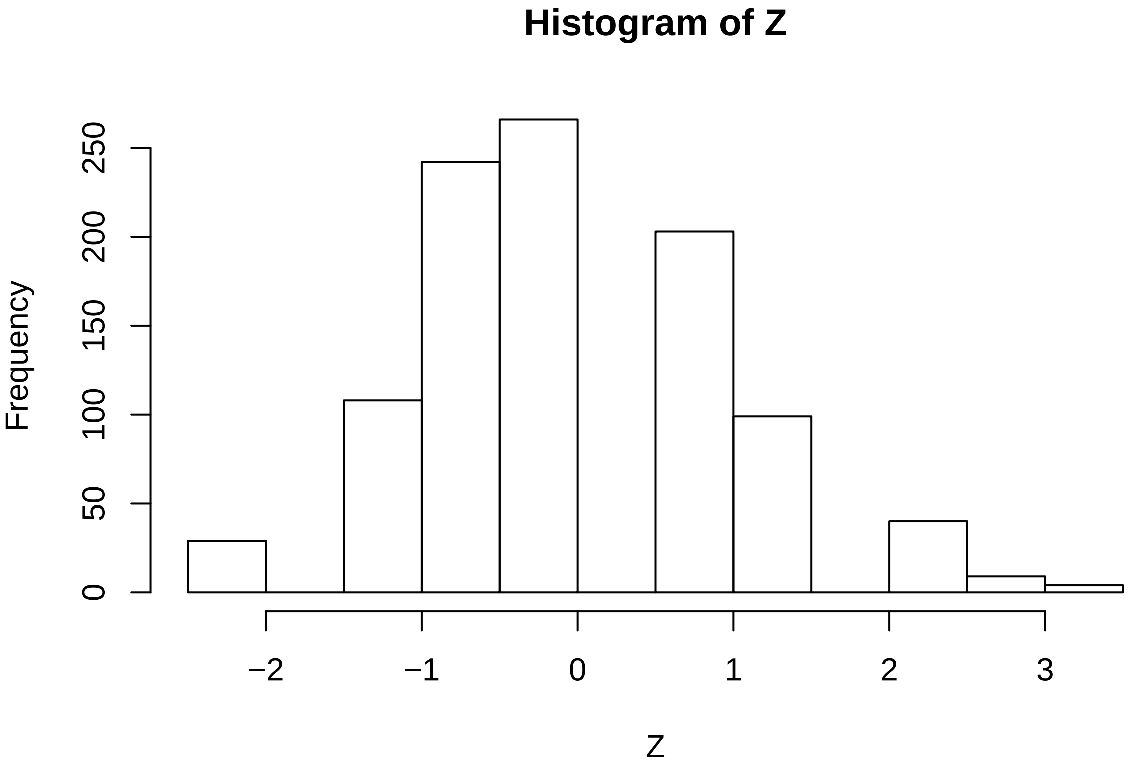
```
sum(X_10000)
```

```
## [1] 2992
```

EX1.4.a

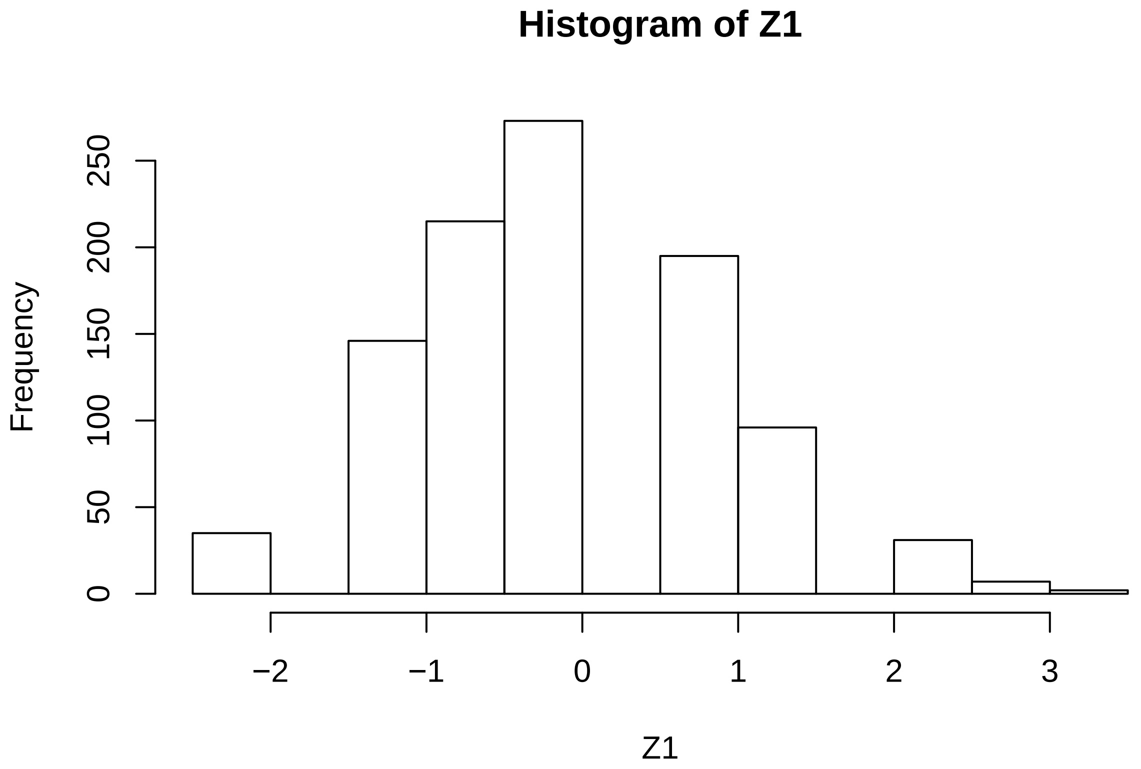
```
rgaus <- function (N,n,p)
{
  Z <- vector(length = N)
  for(i in 1:N)
  {
    X <- rbern(n,p)
    Z[i] <- sqrt(n)*(mean(X)-p)/sqrt(p*(1-p))
  }
  return(Z)
}

Z <- rgaus(1000,10,0.3)
hist(Z)
```

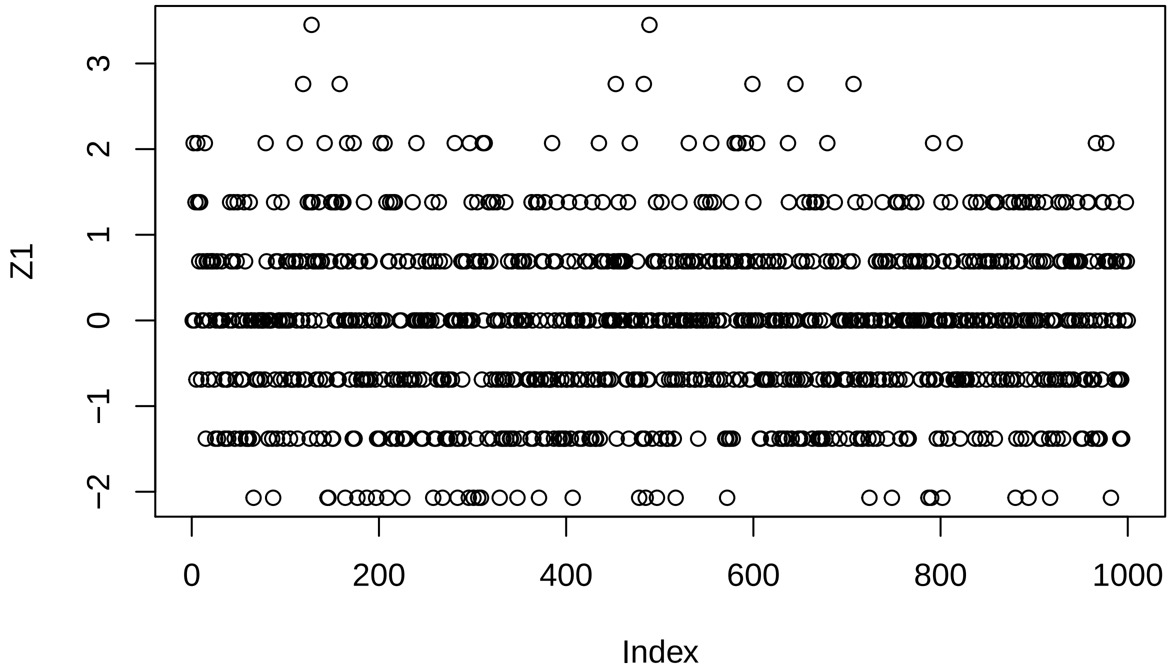


###EX1.4.b
et EX1.4.c

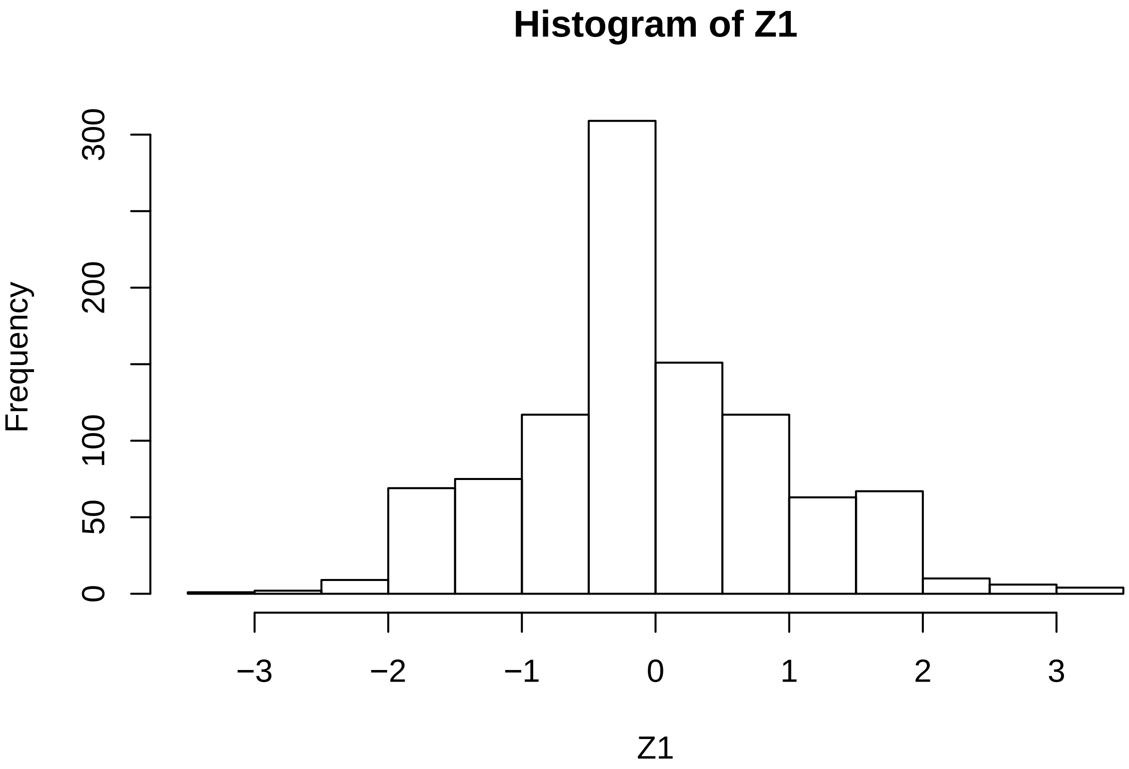
```
Z1 <- rgaus(1000,10,0.3)
hist(Z1)
```



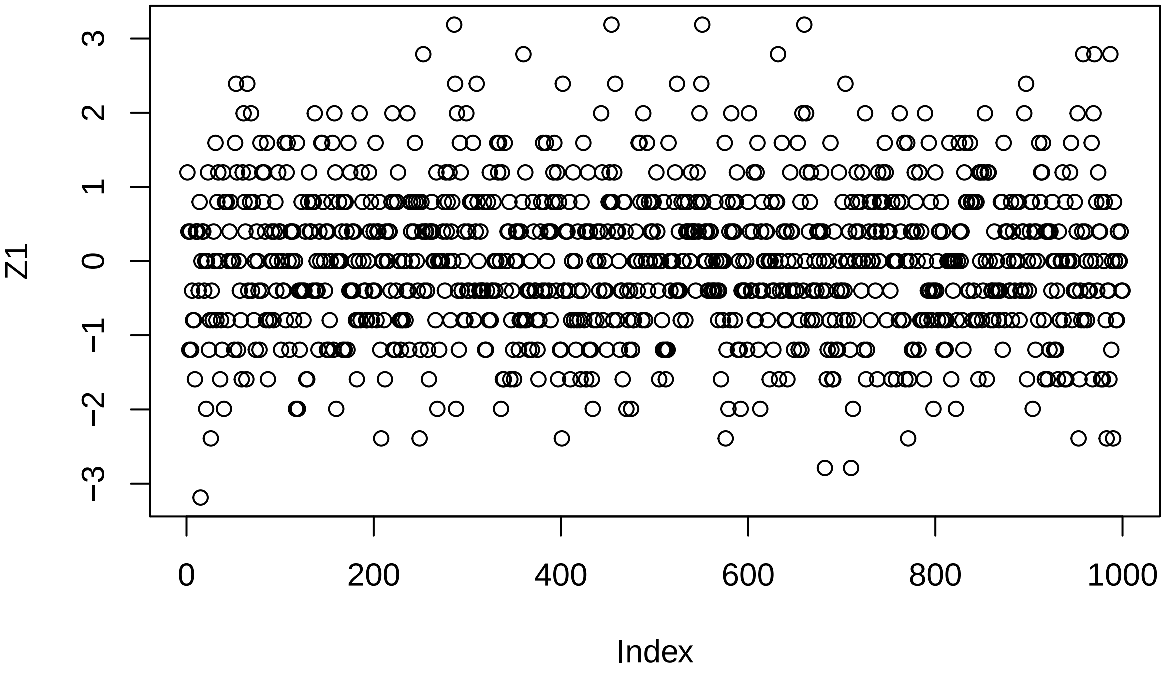
```
plot(Z1)
```



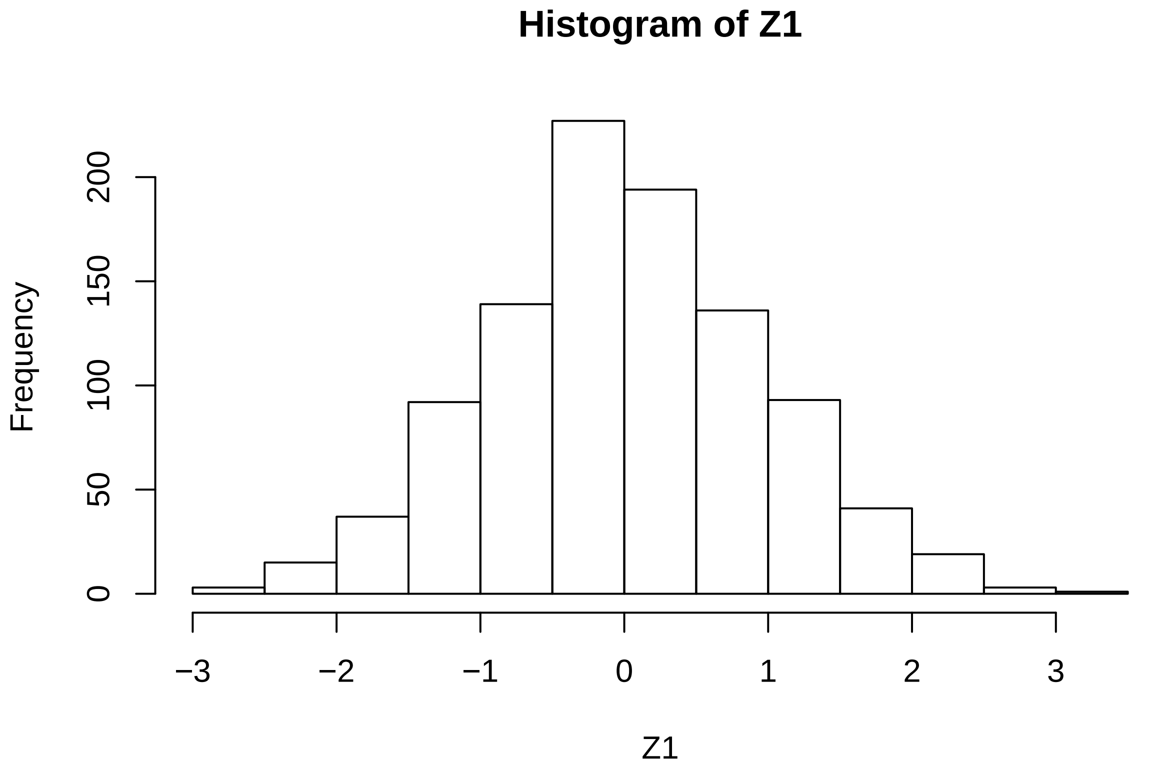
```
Z1 <- rgaus(1000,30,0.3)
hist(Z1)
```



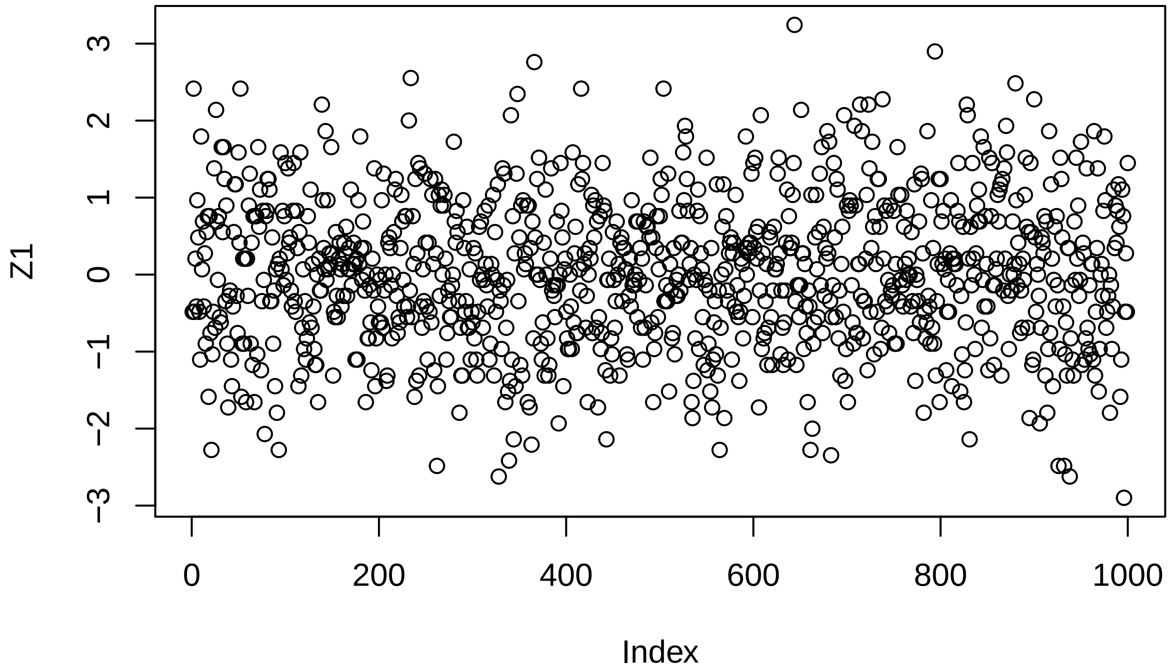
```r
plot(Z1)
```



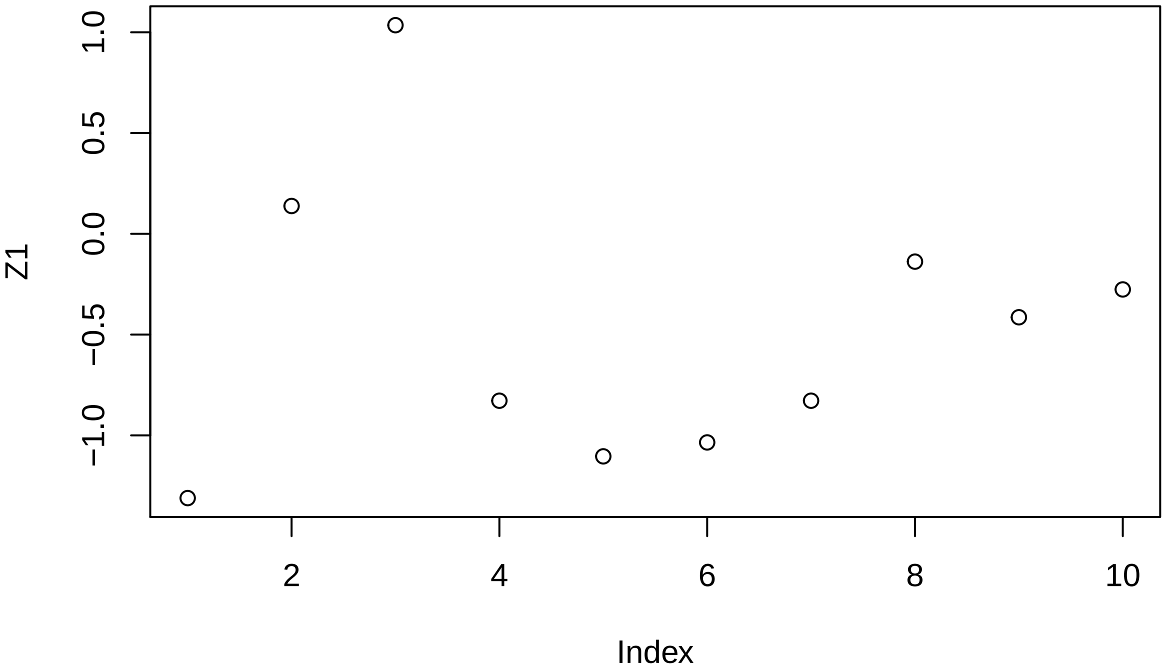
```r
Z1 <- rgaus(1000,1000,0.3)
hist(Z1)
```



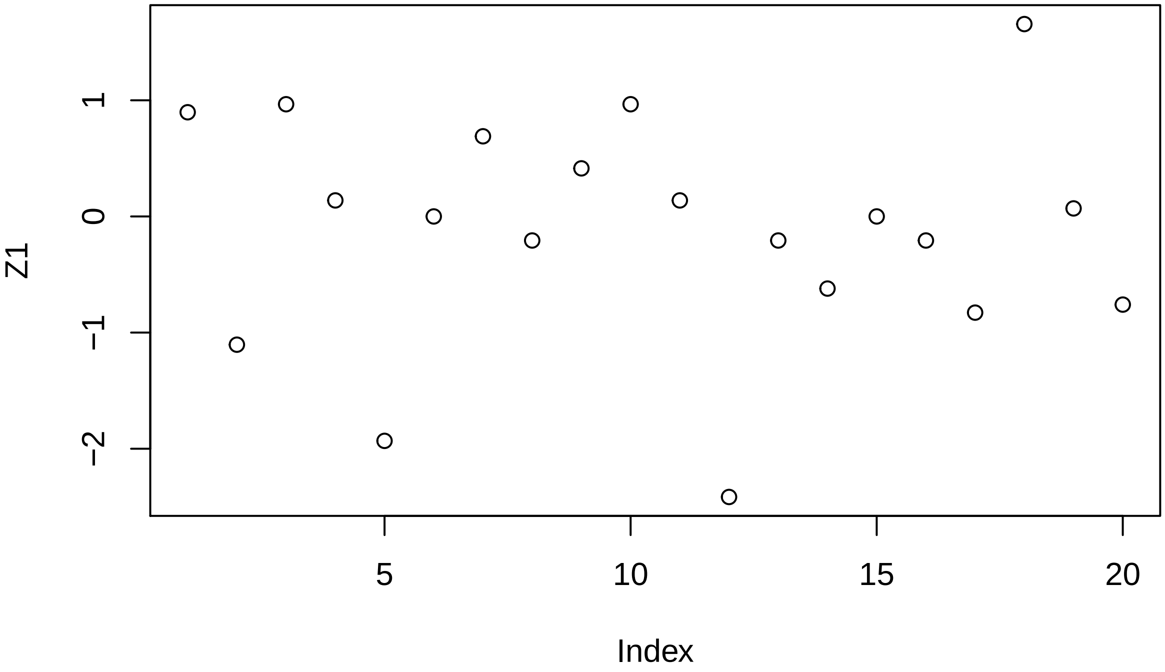
```
plot(Z1)
```



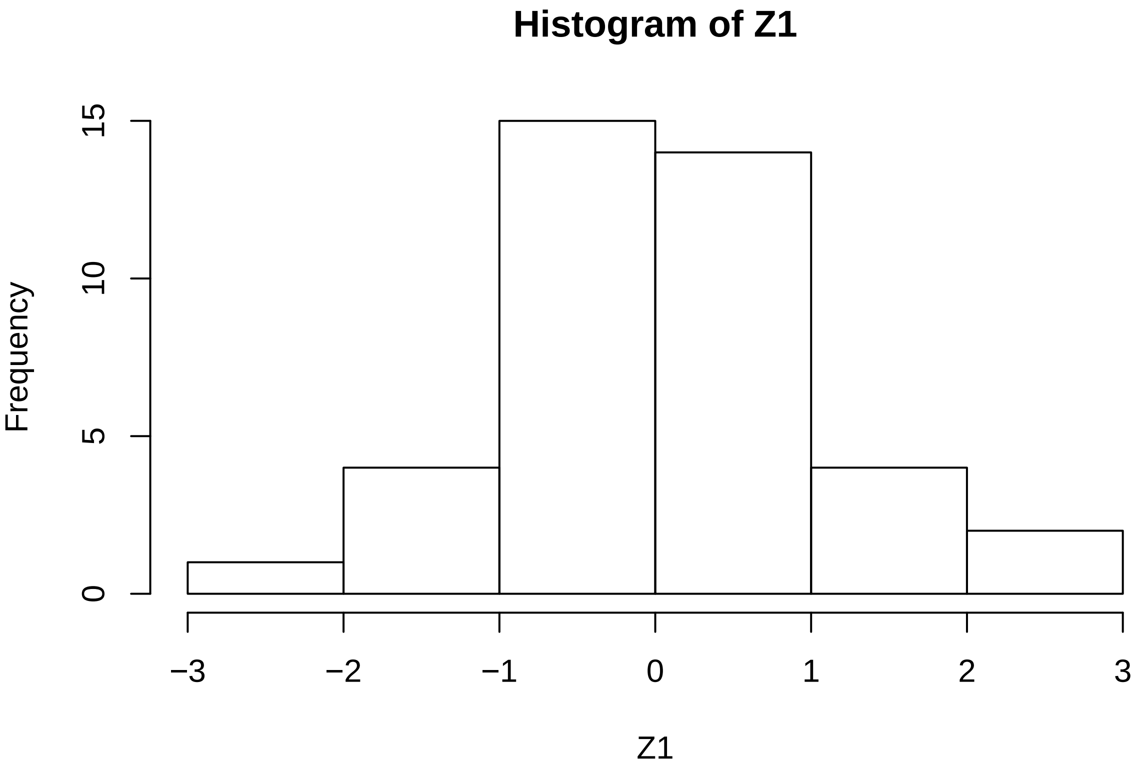
```
Z1 <- rgaus(10,1000,0.3)
plot(Z1)
```



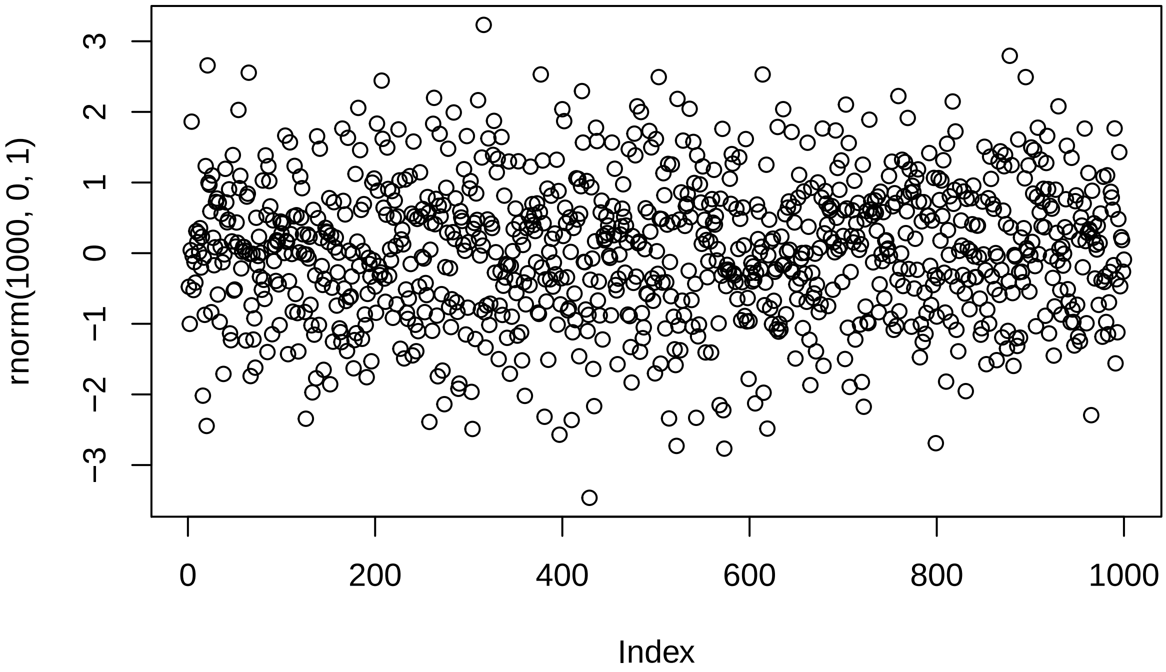
```
Z1 <- rgaus(20,1000,0.3)
plot(Z1)
```



```
Z1 <- rgaus(40,1000,0.3)
hist(Z1)
```



```
plot(rnorm(1000,0,1))
```



```
# Donc ,plus N et n sont grand , plus le graph resemble une loi normal
```

EX2.1

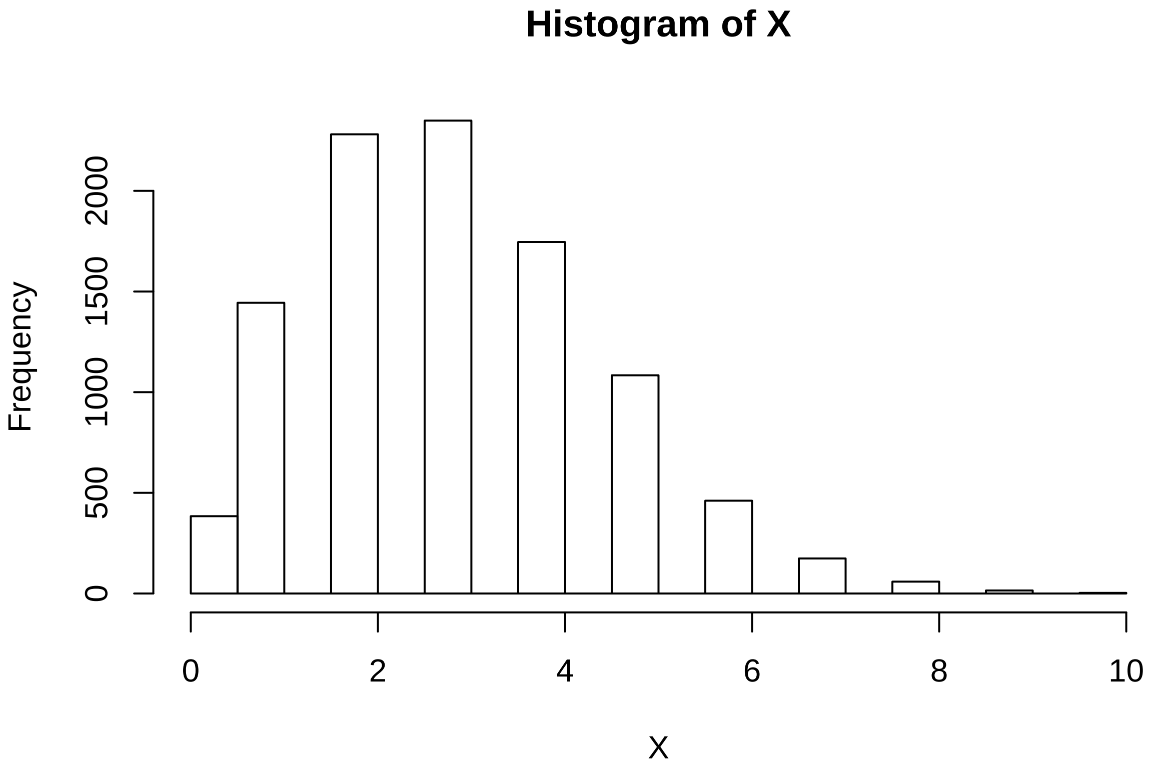
$$P(X=k)=\{sum(U_k)=k\}=C_n^k p^k q^{(1-k)}$$

EX2.2

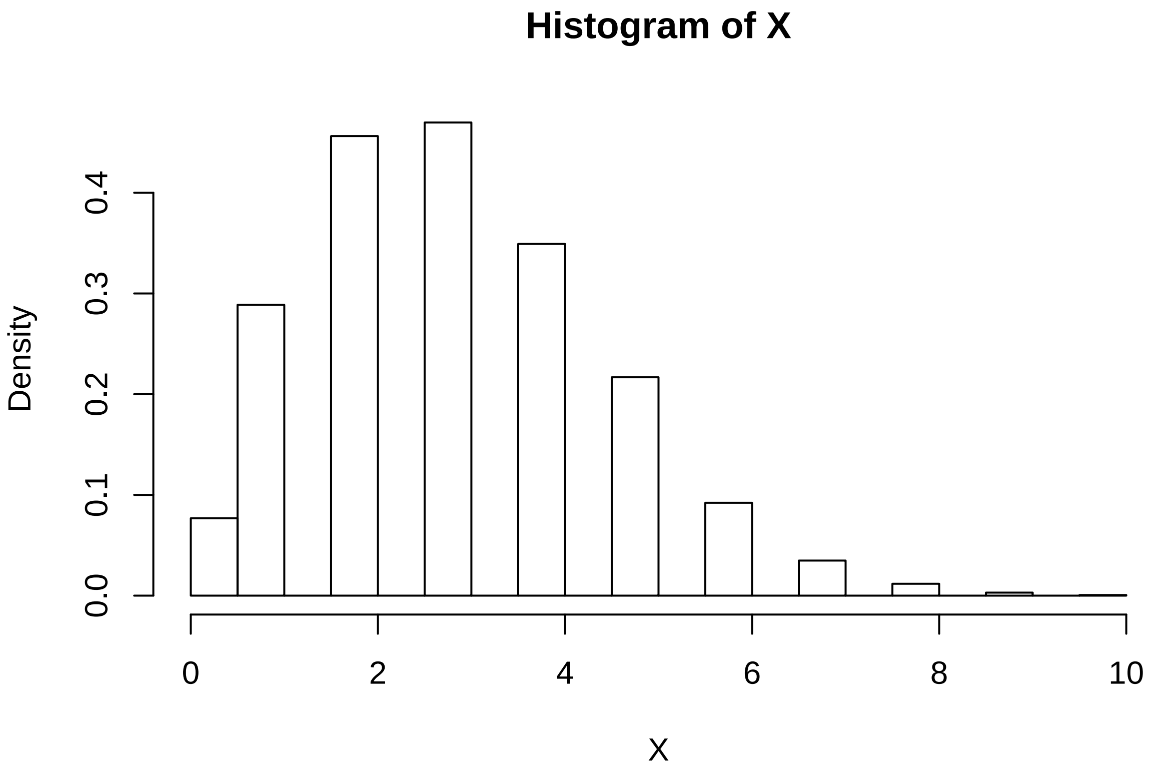
```
rbi <- function (N,n,p)
{
  X <- vector(length = N)
  U <- vector(length = n)
  for(i in 1:N)
  {
    for(k in 1:n)
    {
       U[k] <- as.numeric(runif(1)<p)
    }
    X[i] =sum(U)
  }
  return(X)
}
X<-rbi(10000,30,0.1)
```

EX2.3 et 2.4

```
hist(X)
```



```
hist(X,freq = FALSE)
```

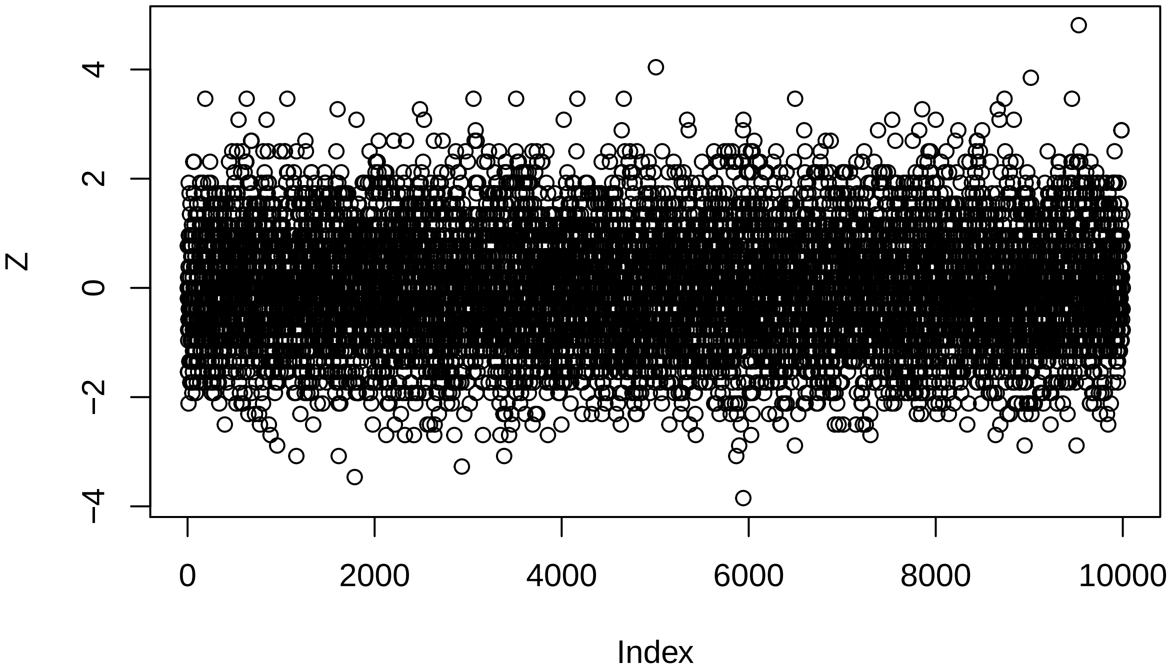


```
mean(X)
```

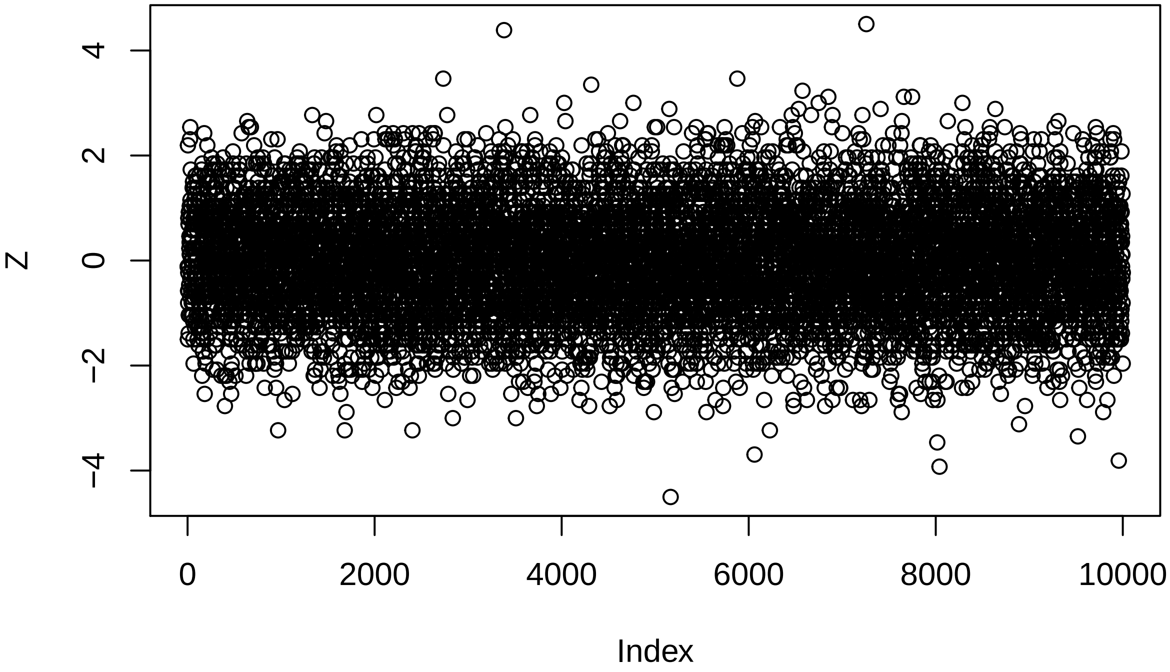
```
## [1] 3.0078
```

EX2.5 et 2.6

```
rgaus_binom <- function (N,k,n,p)
{
  Z <- vector(length = N)
  for(i in 1:N)
  {
    X <- rbi(k,n,p)
    Z[i] <- sqrt(k)*(mean(X)-n*p)/sqrt(n*p*(1-p))
  }
  return(Z)
}
Z <- rgaus_binom(10000,10,30,0.1)
plot(Z)
```



```r
Z <- rgaus_binom(10000,10,30,0.5)
plot(Z)
```



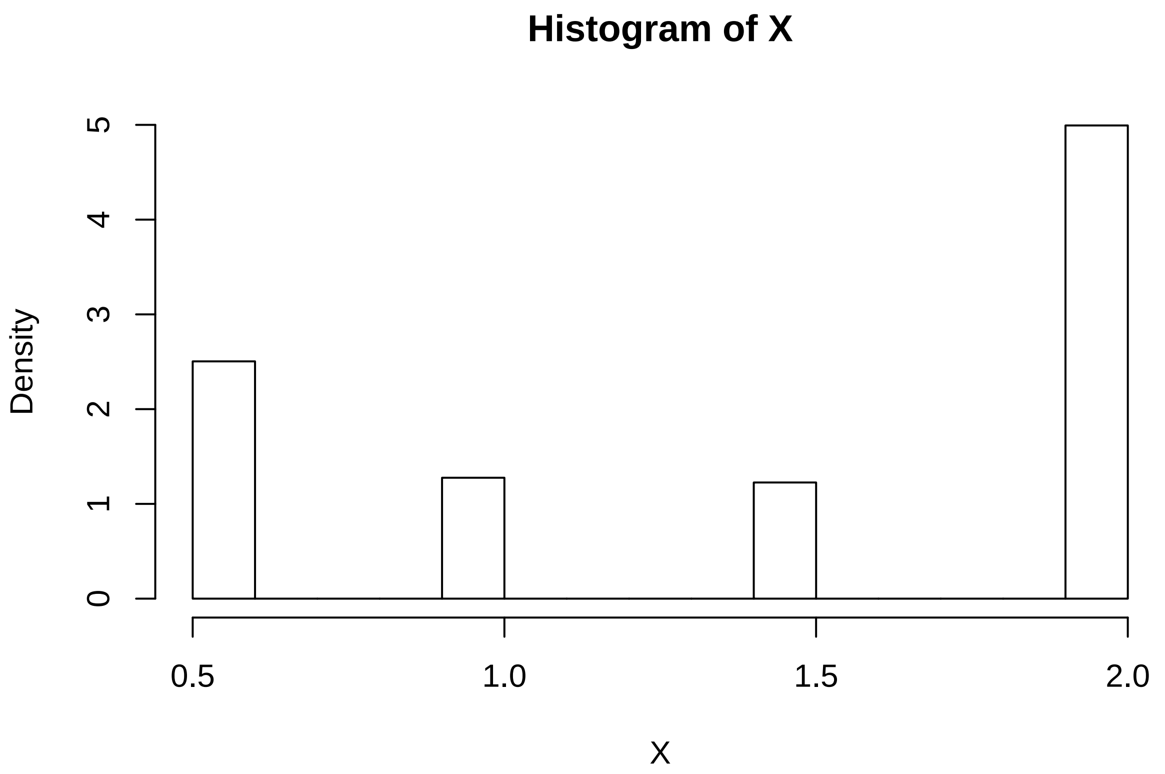
EX3

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

```r
p <- vector(length = 4)
p[1]=1/4
p[2]=1/8
p[3]=1/8
p[4]=1/2
c <- rep(0,4)
for(i in 1:4)
{
  for(j in 1:i)
    {
      c[i] <-c[i]+p[j]
    }
}

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)
}
X<- rloidiscret(10000)

hist(X,freq = FALSE)
```



EX4

```
#definir les fontion pour generer les rv de Poisson
rpoisson <- function(Num,lambda)
{
  N=10
  X <- vector(length = Num)
  p <- vector(length = N)
  c <- rep(0,N)
  for(i in 1:N)
  {
    p[i]<- exp(-lambda)* lambda^(i-1)/factorial(i-1)
  }
  for(i in 1:N)
  {
    for(j in 1:i)
    {
      c[i]= c[i] + p[j]
    }
  }
  for(i in 1:Num)
  {
  u <- runif(1)
  if(u <= c[N])
  {
    k <- 1
    while (u > c[k])
      k<-k+1
    X[i] <- k-1
  }
  else
  {
    k <- N
    pn <- p[N]
    cn <- c[N]
    while (u>cn)
    {
      k<- k+1
      pn <- lambda * pn/k
      cn <- cn+pn
    }
    X[i] <- k-1
  }

  }
  return(X)
}

#applique TCL
rgaus_poisson<- function (N,k,lambda)
{
  Z <- vector(length = N)
  for(i in 1:N)
  {
    X <- rpoisson(k,lambda)
    Z[i] <- sqrt(k)*(mean(X)-lambda)/sqrt(lambda)
  }
  return(Z)
}
```

```
#definir les fontion pour generer les rv d'exponential
rexpoentielle <- function(N,lambda)
{
  for(i in 1:N)
  {
  u <- runif(1)
  X[i] <- -log(u)/lambda
  }
  return(X)
}

#applique TCL
rgaus_exp<- function (N,k,lambda)
{
  Z <- vector(length = N)
  for(i in 1:N)
  {
    X <- rpoisson(k,lambda)
    Z[i] <- sqrt(k)*(mean(X)-1/lambda)/sqrt(1/lambda^2)
  }
  return(Z)
}
```

```
#definir les fontion pour generer les rv de Weibull
rweibull <- function(N,lambda,a)
{
  for(i in 1:N)
  {
  u <- runif(1)
  X[i] <- (-lambda*log(1-u))^(1/a)
  }
  return(X)
}

#applique TCL
rgaus_weibull<- function (N,k,lambda,a)
{
  Z <- vector(length = N)
  mu <- lambda*gamma(1+1/a)
  var <- lambda^2*gamma(1+2/a)-mu^2
  for(i in 1:N)
  {
    X <- rweibull(k,lambda,a)
    Z[i] <- sqrt(k)*(mean(X)-mu)/sqrt(var)
  }
  return(Z)
}
```

```
#Finally , we can write the funtion which consider all those distribution
plot_lois <- function(t)
{
    if(t==1)
    {
    p <- readline("input p (0<= p <=1):")
    p <- as.double(p)
    Z <- rgaus(1000,30,p)
    par(mfrow=c(1,2))
    hist(Z,main = "hist of sample of Bernouilli distribution")
    hist(rnorm(1000),main =" hist of standard normal distribution")
    }
    if(t==2)
    {
    p<- readline('input the p (0<= p <=1):')
    p<- as.double(p)
    n<- readline('input the n :')
    n<- as.double(n)
    Z <- rgaus_binom(1000,30,n,p)
    par(mfrow=c(1,2))
    hist(Z,main = "hist of sample of binomial distribution")
    hist(rnorm(1000),main =" hist of standard normal distribution")
    }
   if(t==3)
   {
    lambda<- readline('input the lambda :')
    lambda<- as.double(lambda)
    Z <- rgaus_poisson(1000,30,lambda)
    par(mfrow=c(1,2))
    hist(Z,main = "hist of sample of poisson distribution")
    hist(rnorm(1000),main =" hist of standard normal distribution")
   }
   if(t==4)
   {
    lambda<- readline('input the lambda :')
    lambda<- as.double(lambda)
    Z <- rgaus_exp(1000,30,lambda)
    par(mfrow=c(1,2))
    hist(Z,main = "hist of sample of Exponential distribution")
    hist(rnorm(1000),main =" hist of standard normal distribution")
   }
   if(t==5)
   {
    lambda<- readline('input the lambda :')
    lambda<- as.double(lambda)
    a<- readline('input the a :')
    a<- as.double(a)
    Z <- rgaus_weibull(1000,30,lambda,a)
    par(mfrow=c(1,2))
    hist(Z,main = "hist of sample of Weibull distribution")
    hist(rnorm(1000),main =" hist of standard normal distribution")
   }
}


t=''
while(class(t)=="character" &&  t!=1 && t!=2 && t!=3 && t!=4 && t!=5)
{
t<-readline("1: Bernouilli 2: binomial 3: Poisson 4: Exponentielle 5: Weibull: ")
}
plot_lois(t)
```