TPSTAT_CHEN_Zeyu_GROUP_2.1

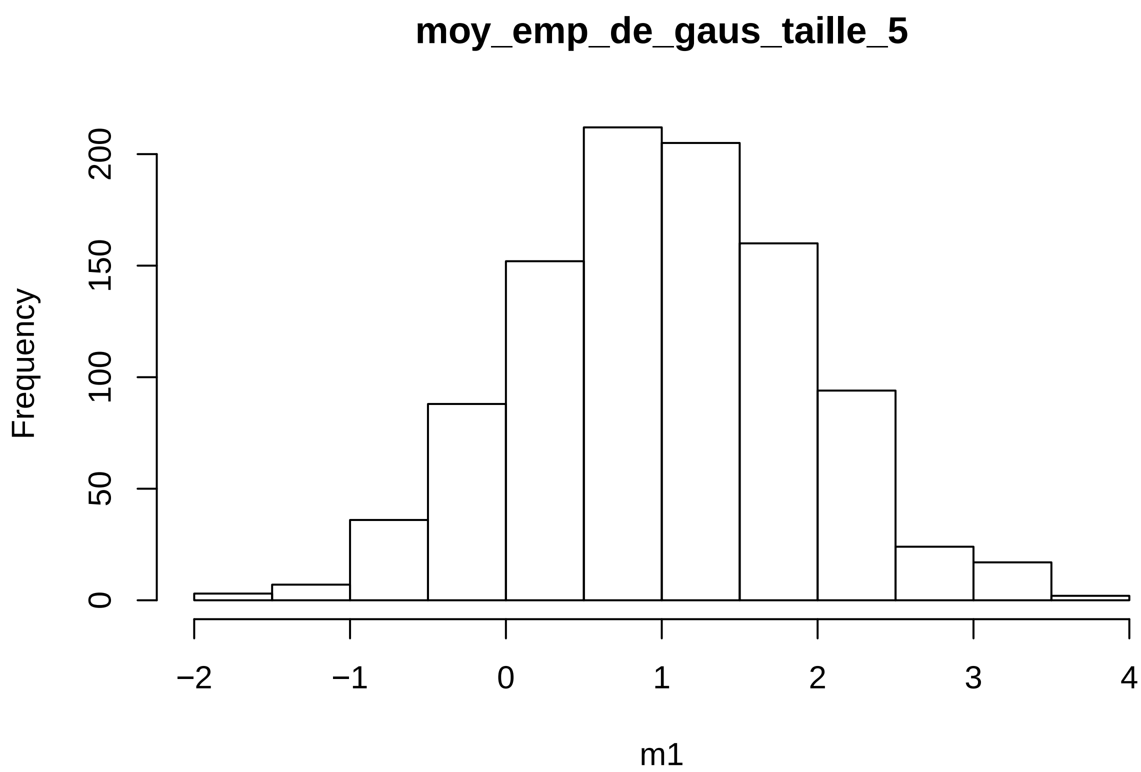
Zeyu

2018/3/5

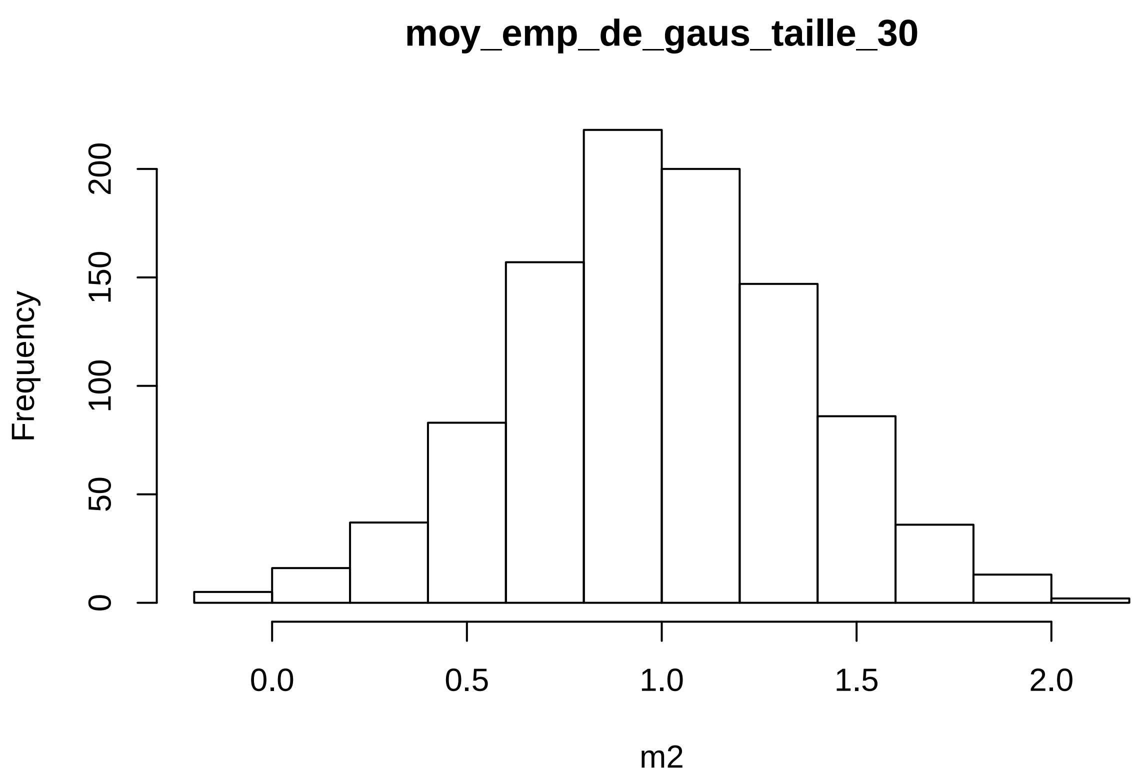
1.Echantillon, Théorème Central Limite, Estimation Monte Carlo

<–1–>

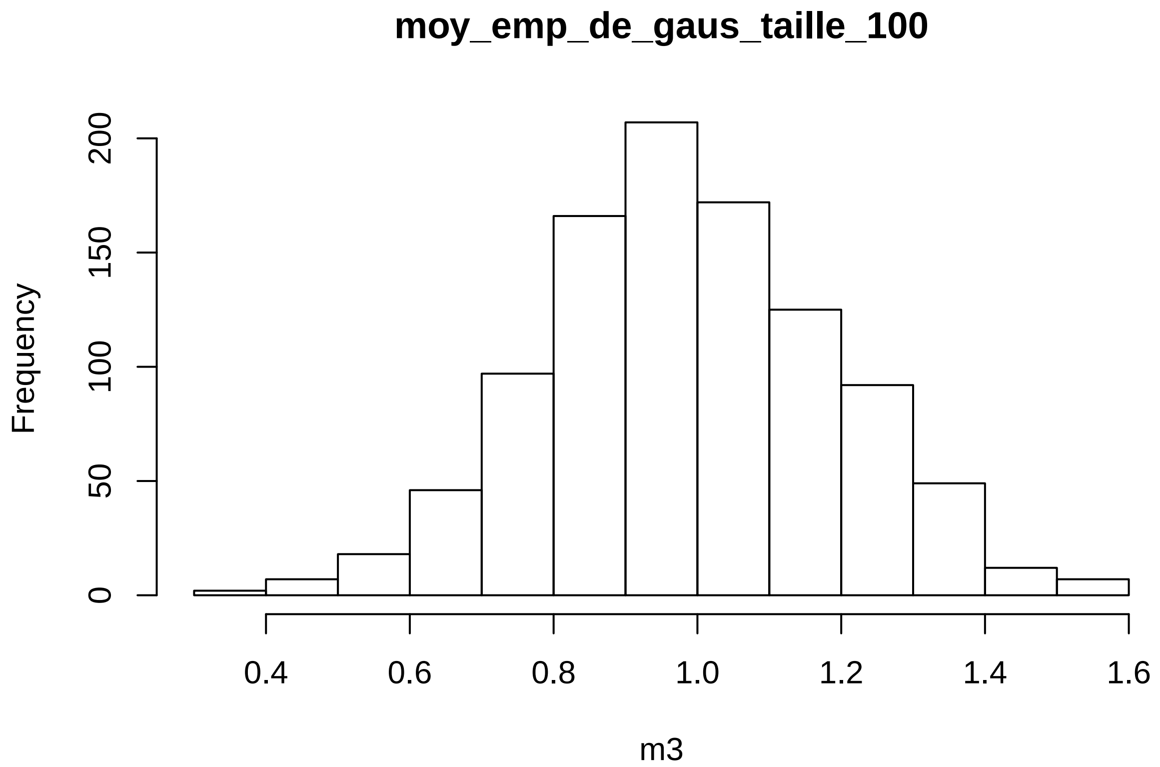
```
# N est la fois d'échantillons
# f est la fonction qu'on va simuler
# n taille d'échantillon, u est mu, o est sigma
sim.fun <-function (N,f,n,u,o)
  {
    sample<-1:N
    for (i in 1:N) {
        sample[i] <-f(n,u,o)
     }
    return(sample)
}

moyenne=function(n,mu,sigma)
  {
  r=rnorm(n,mu,sigma);
  return(mean(r));
}

varience=function(n,mu,sigma)
  {
  r=rnorm(n,mu,sigma);
  return(var(r));
}
m1 = sim.fun(1000,moyenne,5,1,2)
m2 = sim.fun(1000,moyenne,30,1,2)
m3 = sim.fun(1000,moyenne,100,1,2)
v1 = sim.fun(1000,varience,5,1,2)
v2 = sim.fun(1000,varience,30,1,2)
v3 = sim.fun(1000,varience,100,1,2)
hist(m1,freq =T ,main ="moy_emp_de_gaus_taille_5")
```



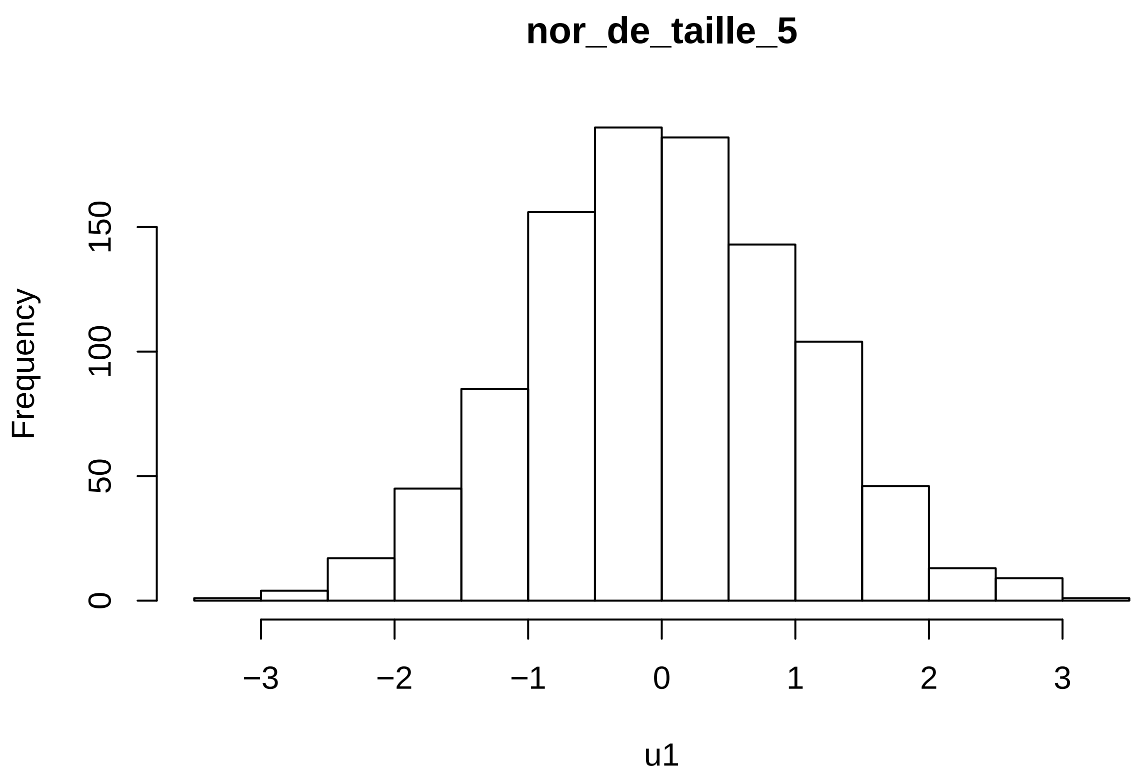

```r
hist(m2,freq =T ,main ="moy_emp_de_gaus_taille_30")
```



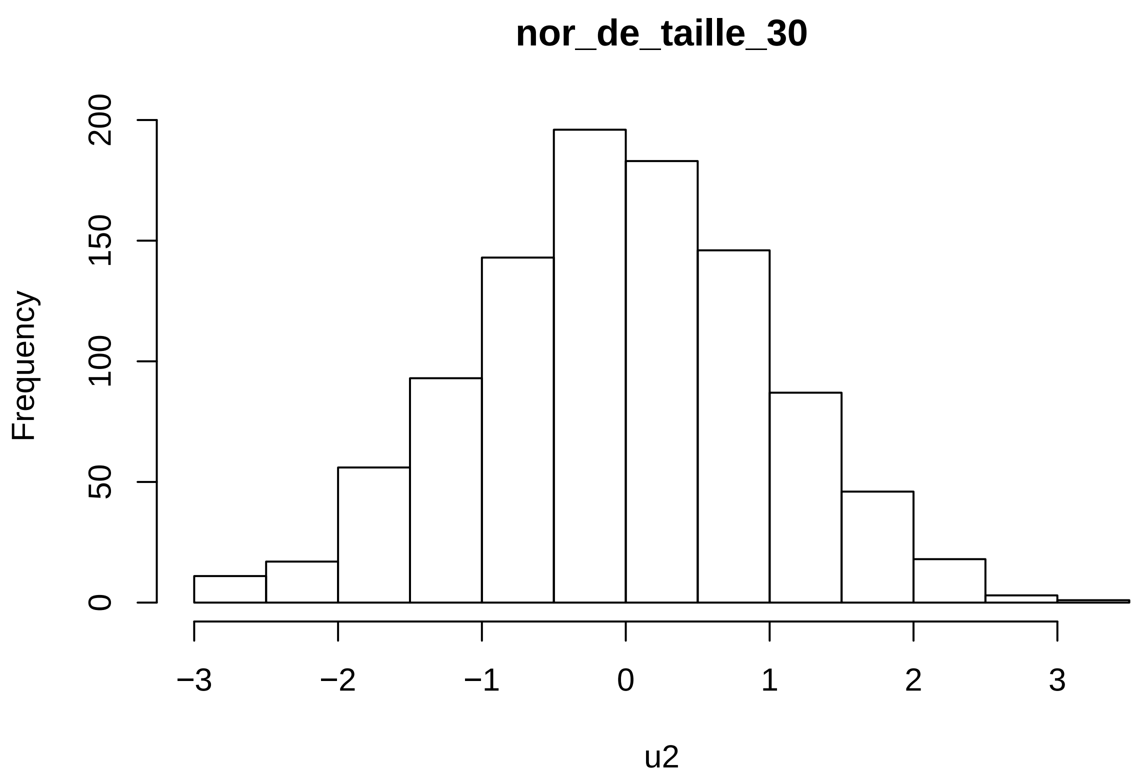

```r
hist(m3,freq =T ,main ="moy_emp_de_gaus_taille_100")
```



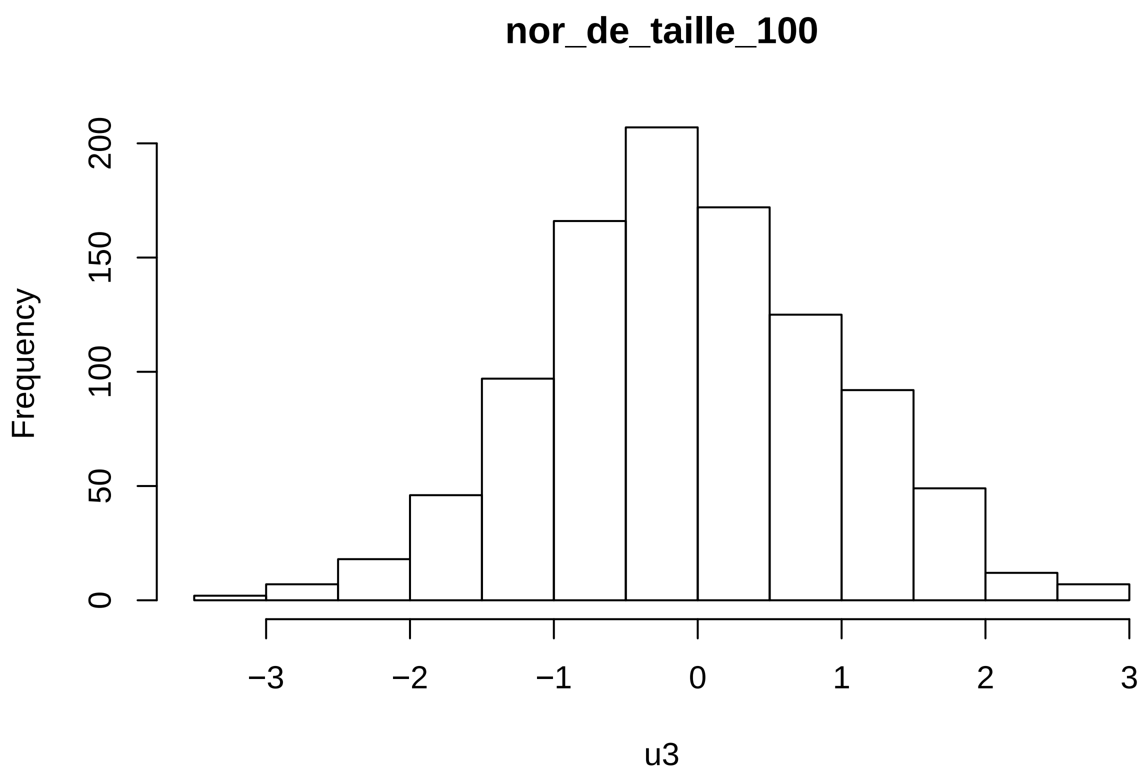

```r
renormalisation<-function(N,moyenne_epq,n,mu,sigma)
{
  for ( i in 1:N)
  {
    moyenne_epq[i]=(moyenne_epq[i]-mu)/(sigma/sqrt(n));
  }
  return(moyenne_epq)
}

u1 = renormalisation(1000,m1,5,1,2);
u2 = renormalisation(1000,m2,30,1,2);
u3 = renormalisation(1000,m3,100,1,2);
hist(u1,freq =T ,main ="nor_de_taille_5")
```




```r
hist(u2,freq =T ,main ="nor_de_taille_30")
```




```r
hist(u3,freq =T ,main ="nor_de_taille_100")
```



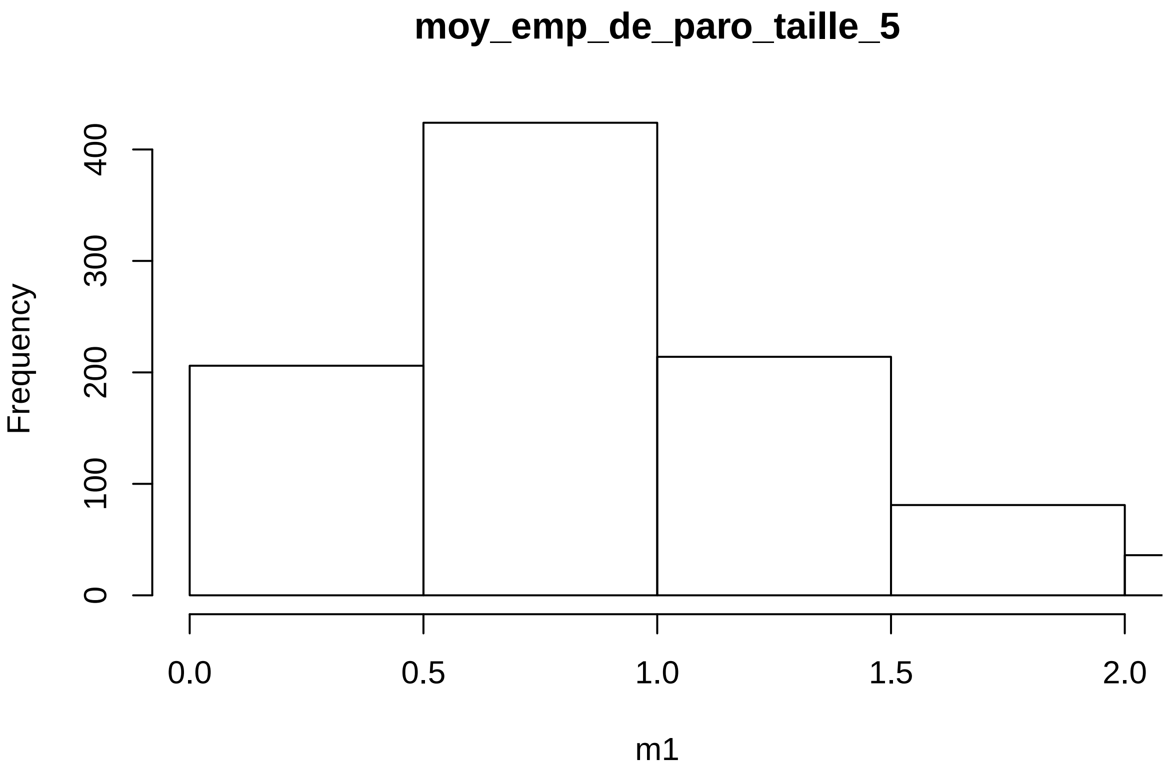
<−2−>

```
library(rmutil)
```

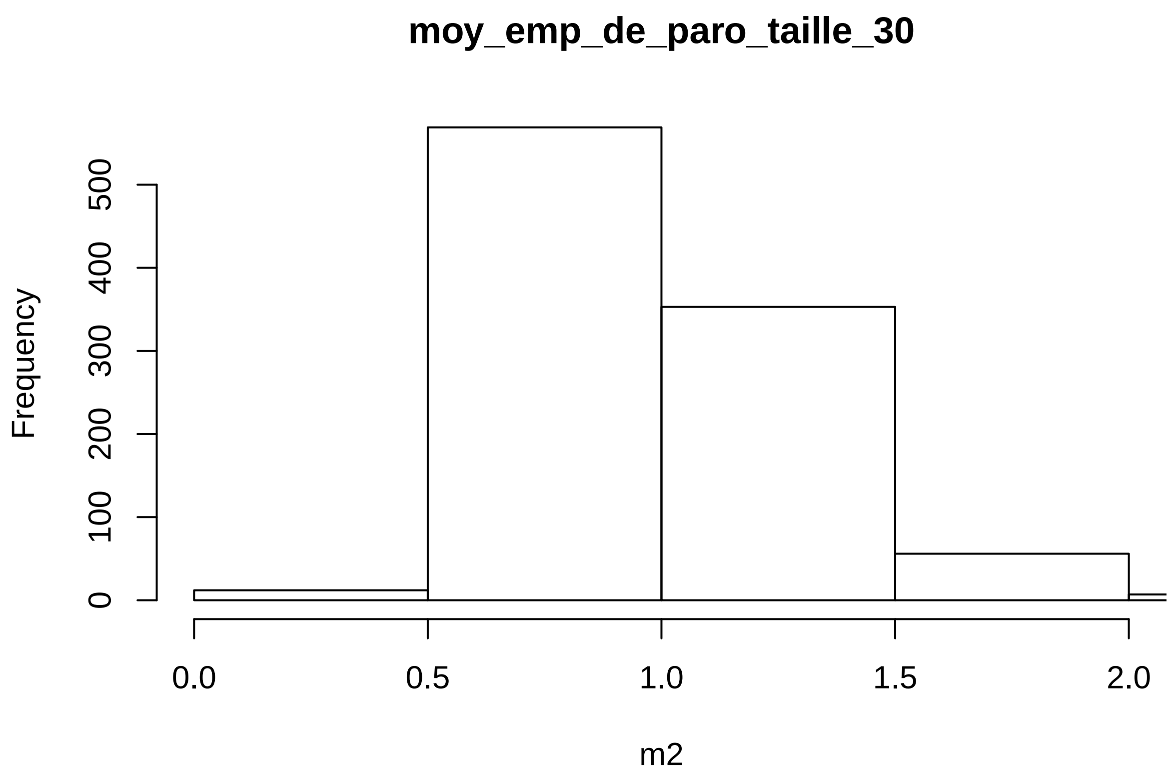
```
##
## Attaching package: 'rmutil'
## The following object is masked from 'package:stats':
##
##     nobs
```

```
# N est la fois d'échantillons
# f est la fonction qu'on va simuler
# n taille d'échantillon
sim.fun <-function (N,f,n,m,s)
  {
    sample<-1:N
    for (i in 1:N) {
        sample[i] <-f(n,m,s)
     }
    return(sample)
}
moyenne=function(n,m,s)
  {
  r=rpareto(n,m,s);
  return(mean(r));
}

varience=function(n,m,s)
```

```
  {
  r=rpareto(n,m,s);
  return(var(r));
}
m1 = sim.fun(1000,moyenne,5,1,3)
m2 = sim.fun(1000,moyenne,30,1,3)
m3 = sim.fun(1000,moyenne,100,1,3)
v1 = sim.fun(1000,varience,5,1,3)
v2 = sim.fun(1000,varience,30,1,3)
v3 = sim.fun(1000,varience,100,1,3)
hist(m1,freq =T ,xlim=c(0,2),main ="moy_emp_de_paro_taille_5")
```




```
hist(m2,freq =T ,xlim=c(0,2),main ="moy_emp_de_paro_taille_30")
```



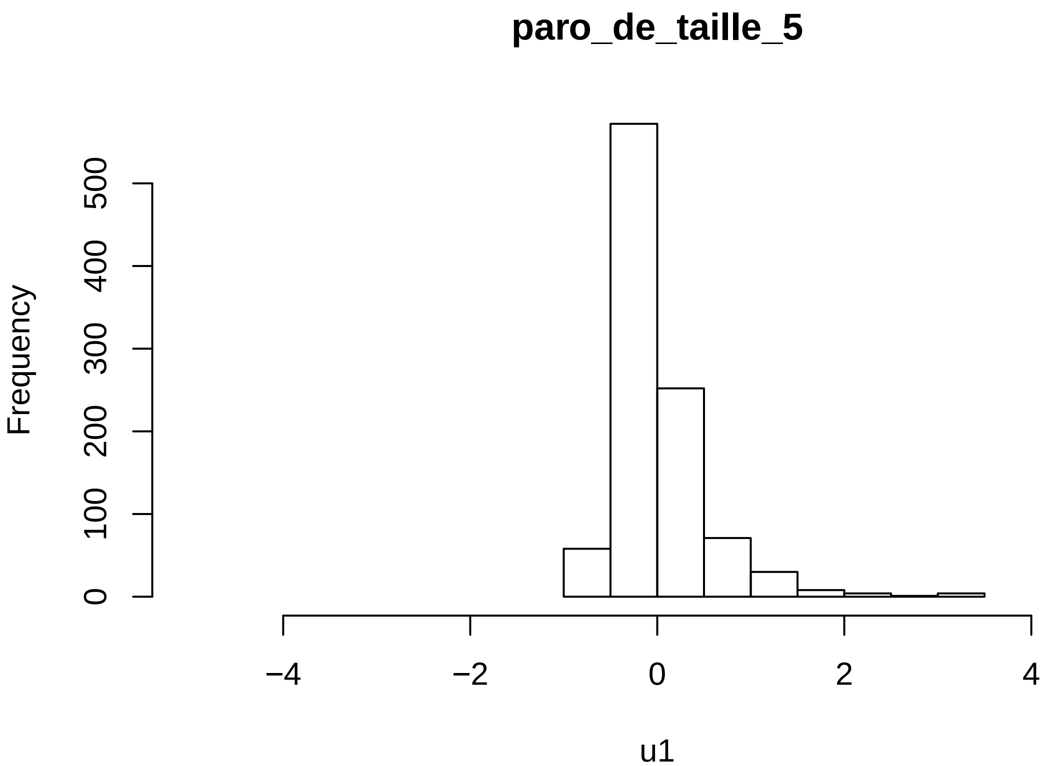

```
hist(m3,freq =T ,xlim=c(0,2),main ="moy_emp_de_paro_taille_100")
```


```
renormalisation<-function(N,moy_emp,n,mu,sigma)
{
  for ( i in 1:N)
  {
```

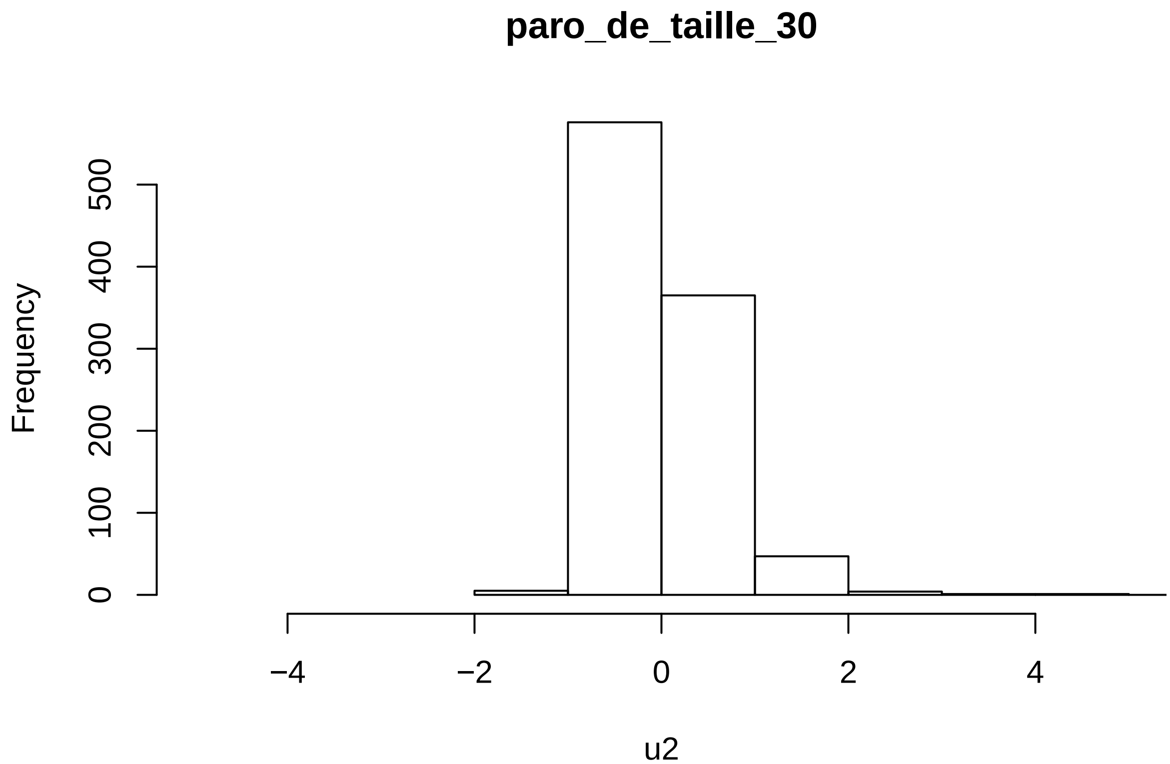
```
    moy_emp[i]=(moy_emp[i]-mu)/(sigma/sqrt(n));
  }
  return(moy_emp)
}
```

$$a_n = m, b_n = s$$

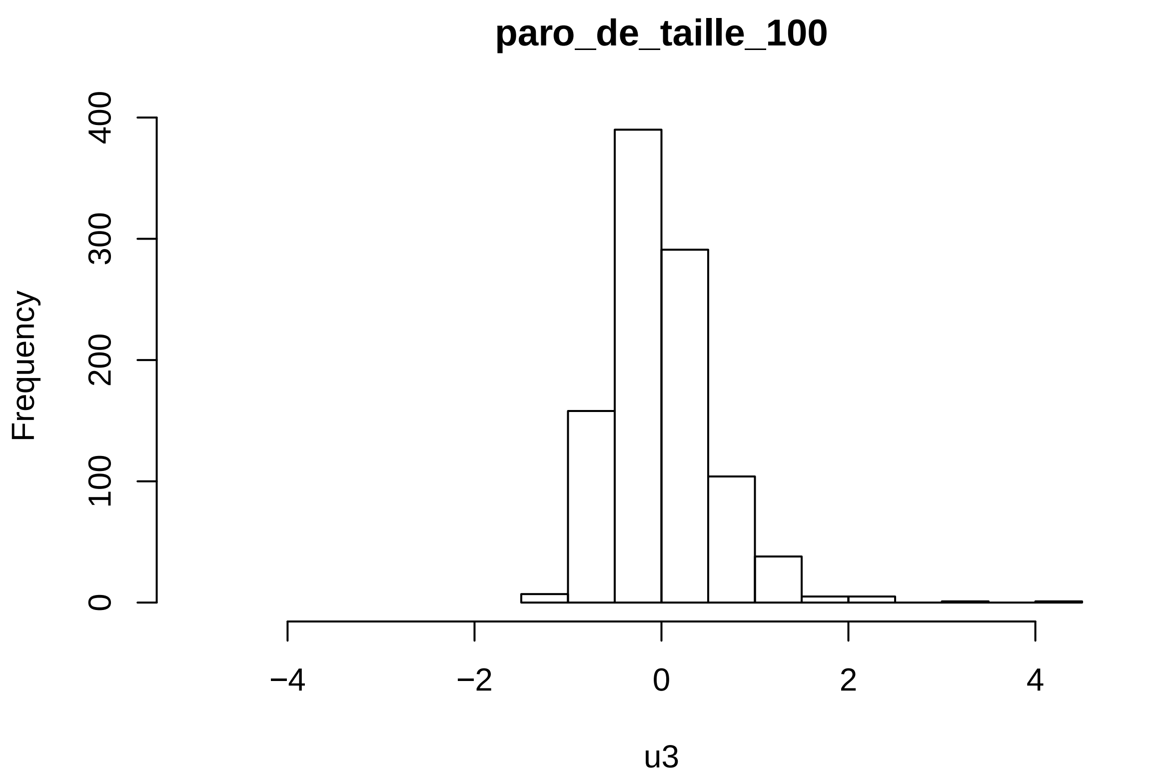
```
u1 = renormalisation(1000,m1,5,1,3);
u2 = renormalisation(1000,m2,30,1,3);
u3 = renormalisation(1000,m3,100,1,3);
hist(u1,freq =T ,xlim=c(-5,5),main ="paro_de_taille_5")
```




```
hist(u2,freq =T ,xlim=c(-5,5),main ="paro_de_taille_30")
```

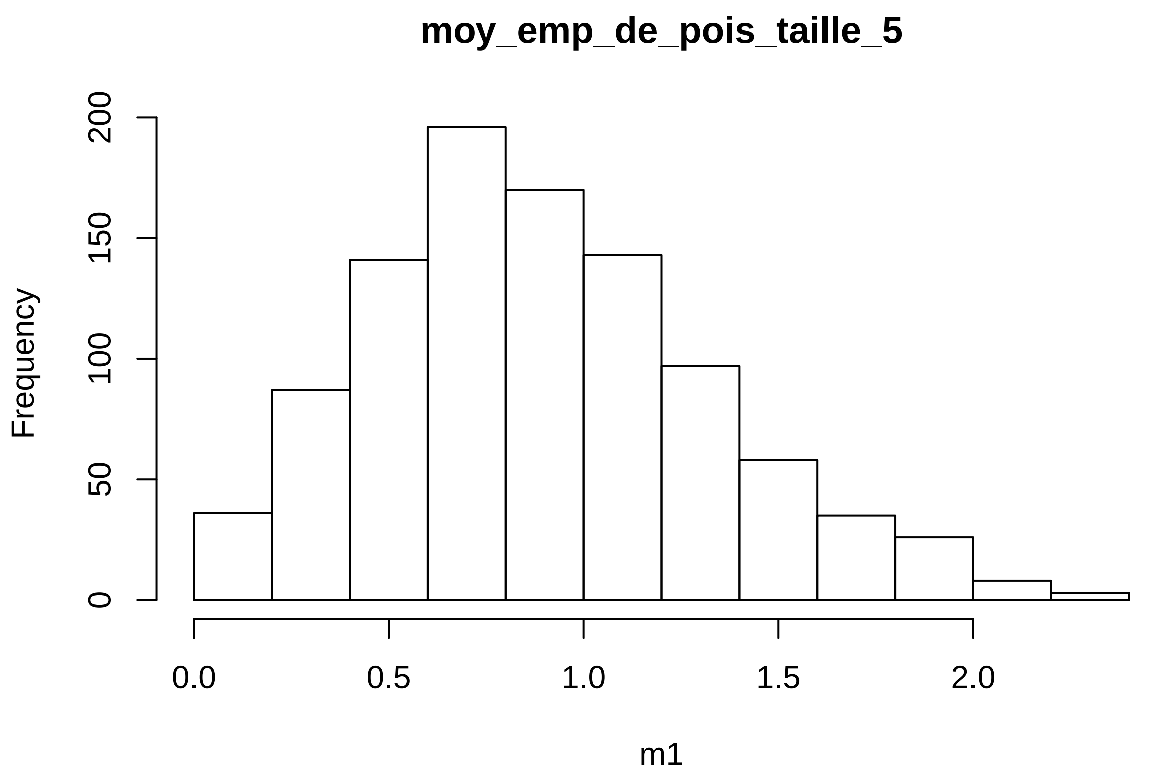



```
hist(u3,freq =T ,xlim=c(-5,5),main ="paro_de_taille_100")
```

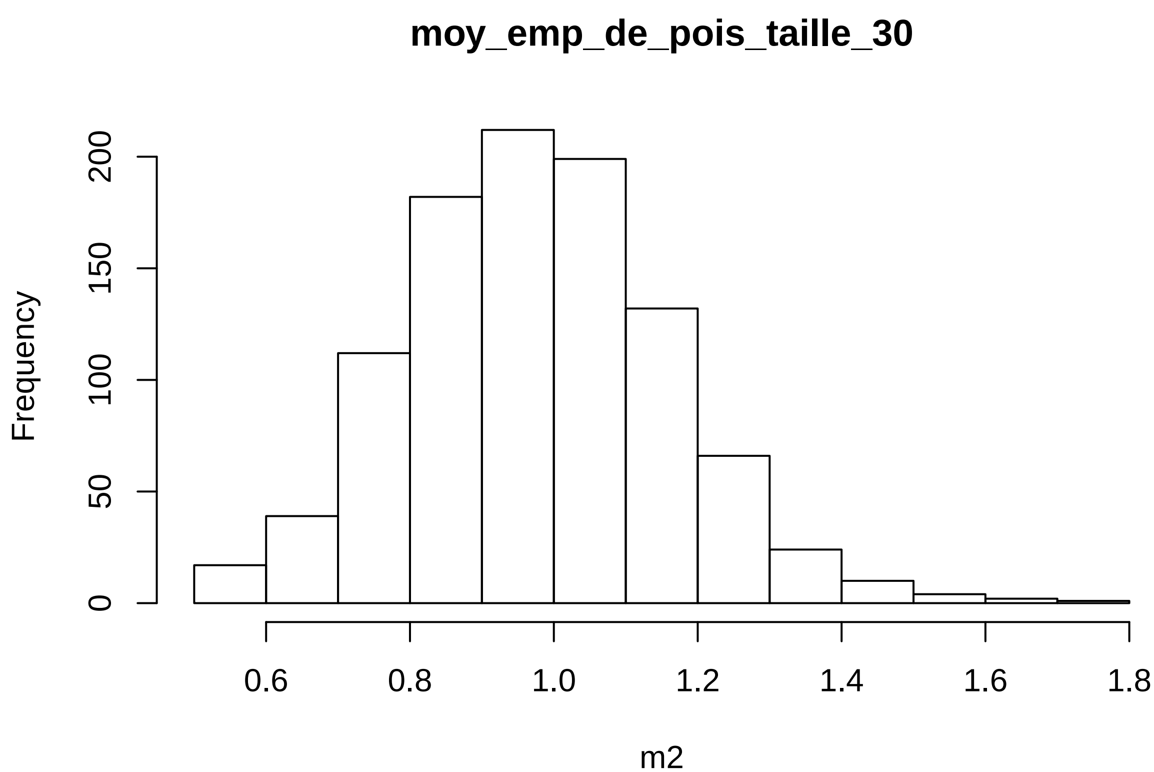


<–3–>

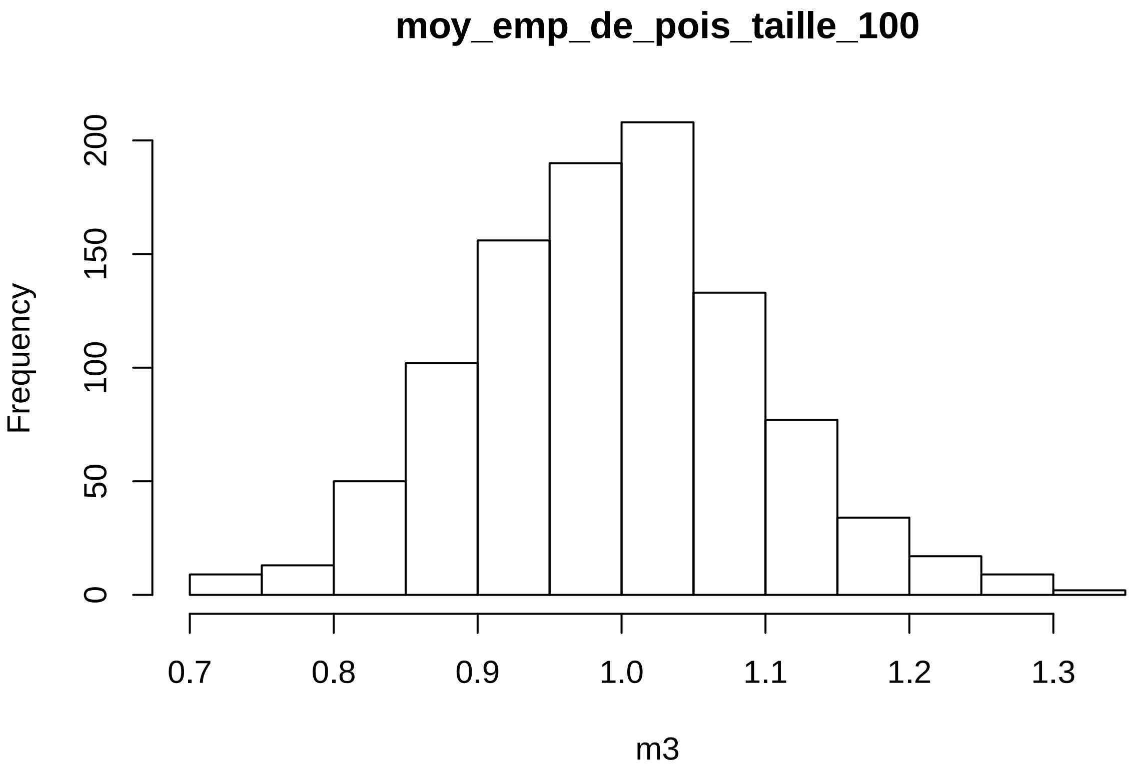
```r
# N est la fois d'échantillons
# f est la fonction qu'on va simuler
# n taille d'échantillon
sim.fun <-function (N,f,n,lamda)
  {
    sample<-1:N
    for (i in 1:N) {
        sample[i] <-f(n,lamda)
     }
    return(sample)
}
moyenne=function(n,lamda)
  {
  r=rpois(n,lamda);
  return(mean(r));
}

varience=function(n,lamda)
  {
  r=rpois(n,lamda);
  return(var(r));
}
m1 = sim.fun(1000,moyenne,5,1)
m2 = sim.fun(1000,moyenne,30,1)
m3 = sim.fun(1000,moyenne,100,1)
v1 = sim.fun(1000,varience,5,1)
v2 = sim.fun(1000,varience,30,1)
v3 = sim.fun(1000,varience,100,1)
hist(m1,freq =T ,main ="moy_emp_de_pois_taille_5")
```



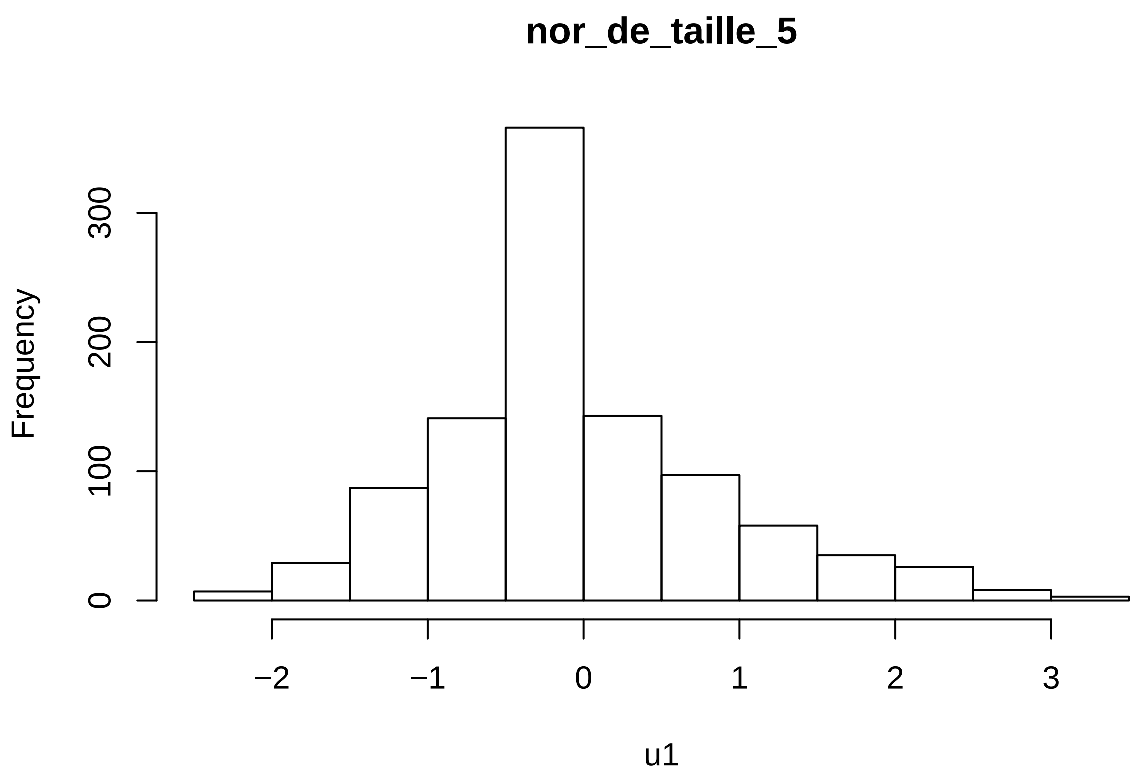

```r
hist(m2,freq =T ,main ="moy_emp_de_pois_taille_30")
```



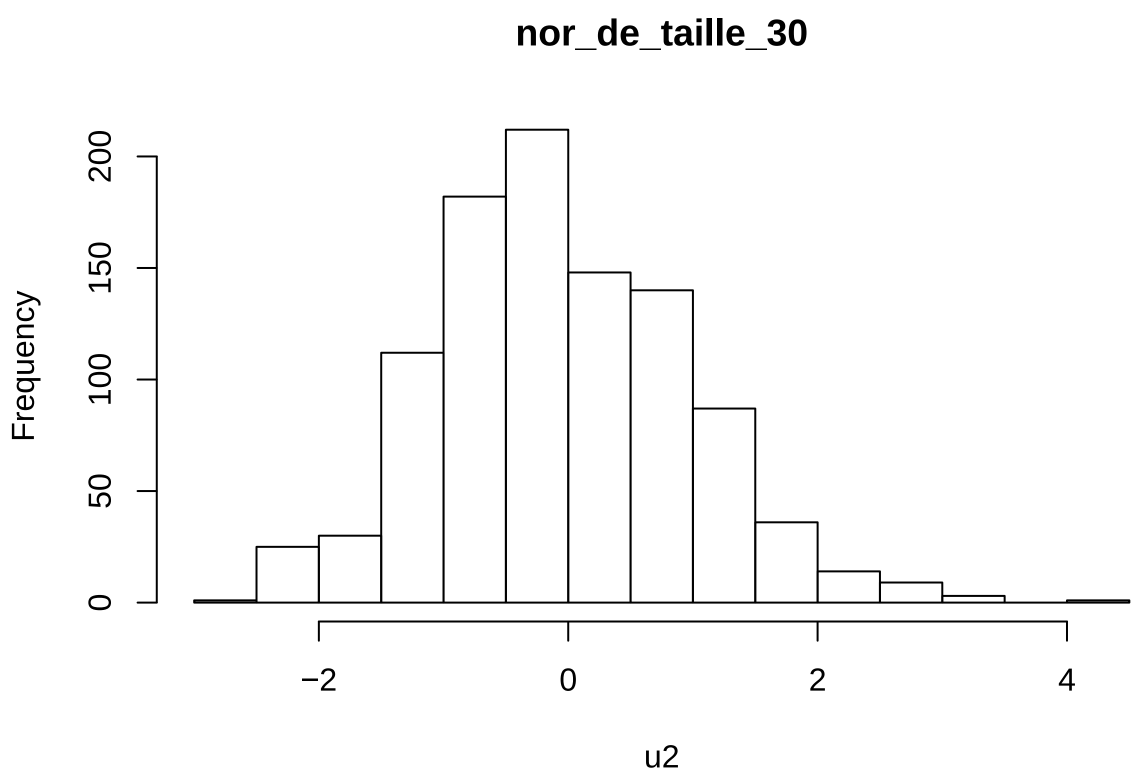

```r
hist(m3,freq =T ,main ="moy_emp_de_pois_taille_100")
```



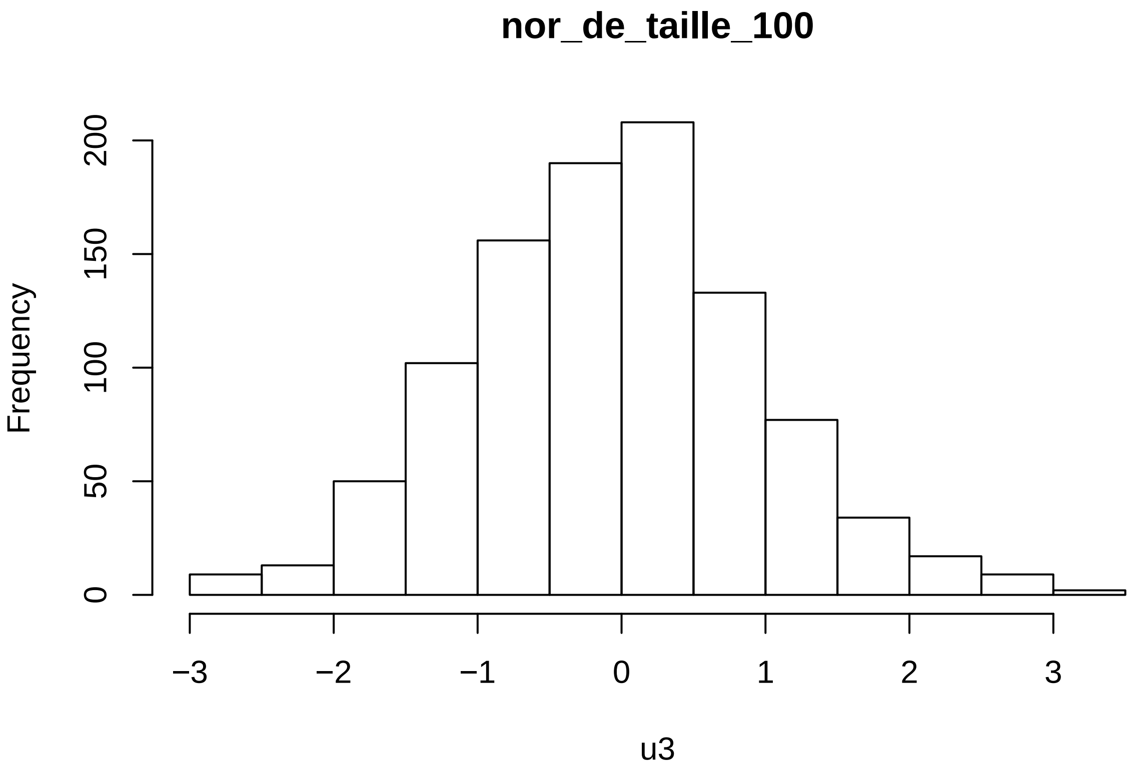

```
renormalisation<-function(N,moy_emp,n,mu,sigma)
{
  for ( i in 1:N)
  {
    moy_emp[i]=(moy_emp[i]-mu)/(sigma/sqrt(n));
  }
  return(moy_emp)
}

u1 = renormalisation(1000,m1,5,1,1);
u2 = renormalisation(1000,m2,30,1,1);
u3 = renormalisation(1000,m3,100,1,1);
hist(u1,freq =T ,main ="nor_de_taille_5")
```




```r
hist(u2,freq =T ,main ="nor_de_taille_30")
```




```r
hist(u3,freq =T ,main ="nor_de_taille_100")
```



<–4–>

quand on fais N fois échantillon (x1,x2..xn) iid de n'importe quel loi, on trouve que toute moyenne $\bar{X}_n$ des échantillons tends vers une variable aléatoire gausienne. En plus, plus N est grand, plus la distribution d'échantillonage est similaire à gaussienne avec μ = E(x) var = var(x). Par ailleurs,

$$Z_n = \frac{\bar{X}_n - \mu}{\sigma/\sqrt{n}}$$

$$\lim_{n\to\infty} P(Z_n \leq z) = \Phi(z)$$

$\Phi(z)$ suis la **loi normale centrée réduite** $\mathcal{N}(0,1)$

2.Moyenne et dispersion

<–1–>

l'inégalité de Chebychef dans les cas Gaussien est

$$P(|X-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$

l'inégalité de Chebychef dans les cas Poisson est

$$P(|X-\lambda| \geq k\lambda) \leq \frac{1}{k^2}$$

parce que l'espérence et la varience de Poison sont tous λ

<–**2**–>

2.a

$$P(|X-\mu| \geq \sigma) \ = E(I_{|X-\mu|\geq k\sigma})$$

avec I_A est une fonction indicatrice

2.b

```
#calcule le borne par Monte Carlo
#parma r une échantillon qu'on a obeteun,n la taille,mu espérence
#return la probabilté de déviation d'une v.a de sa moyenne
library(rmutil)
  Prob_devia <- function(n,r,mu,delta)
  {
    res<- 0
    for (i in 1:n) {
        if (abs(r[i]-mu) >= delta)
          res<-res+1
    }
    return(res/n)
  }
  r1<-rnorm(100000,mean = 1,sd = 1)
  r2<-rpareto(10000,m=1,s=2)
  r3<-rpois(10000,lambda=1)
```

On calcule par la fonction P_deviation les probabilités de déviation d'une v.a de sa moyenne

Dans le cas Gaussien avec mean=1 sd=1 $P(|X-1| \geq 2)$

```
list("Gaussien "= Prob_devia(100000,r1,1,2))
```

```
## $`Gaussien `
## [1] 0.04603
```

Dans le cas Pareto avec m=1 s=1 $P(|X-1| \geq 2) =$

```
list("Pareto"=Prob_devia(10000,r2,1,2))
```

```
## $Pareto
## [1] 0.0629
```

Dans le cas Poisson avec lambda=1$P(|X-1| \geq 2) =$

```
list("Poisson"=Prob_devia(10000,r3,1,2))
```

```
## $Poisson
## [1] 0.0804
```

La précison de cette estimation est cinq

2.c

Remarque: Quand on prend un échantillon avec taille N, $V[\bar{X}_N] = \frac{1}{N}V[X]$

```r
Borne_Chebcf <- function (delta,sigma)
{
  return ((sigma/delta)**2);
}
```

Quand le cas Gaussien avec $\delta = 1$ $\mu = 0$ $\sigma = 1$

```r
x1=Borne_Chebcf(1,1);
r= rnorm(10000,0,1)
x2=Prob_devia(10000,r,0,1);
list("Chebychev"=x1,"Monte Carlo"=x2)
```

```
## $Chebychev
## [1] 1
##
## $`Monte Carlo`
## [1] 0.3238
```

Quand le cas Gaussien avec $\delta = 1$ $\mu = 0$ $\sigma = 0.5$

```r
x1=Borne_Chebcf(1,0.5);
r= rnorm(10000,0,0.5)
x2=Prob_devia(10000,r1,0,1);
list("Chebychev"=x1,"Monte Carlo"=x2)
```

```
## $Chebychev
## [1] 0.25
##
## $`Monte Carlo`
## [1] 0.5212
```

Quand le cas Gaussien avec $\delta = 2$ $\mu = 1$ $\sigma = 1$

```r
x1=Borne_Chebcf(2,1);
r= rnorm(10000,1,1)
x2=Prob_devia(10000,r1,1,1);
list("Chebychev"=x1,"Monte Carlo"=x2)
```

```
## $Chebychev
## [1] 0.25
##
## $`Monte Carlo`
## [1] 0.3116
```

Quand le cas Poisson avec $\delta = 2$ $\lambda = 1$

```r
x1=Borne_Chebcf(2,1);
r= rpois(10000,1)
x2=Prob_devia(10000,r,1,2);
list("Chebychev"=x1,"Monte Carlo"=x2)
```

```
## $Chebychev
## [1] 0.25
##
## $`Monte Carlo`
## [1] 0.0799
```

Quand le cas Poisson avec $\delta = 2$ $\lambda = 0.5$

```
x1=Borne_Chebcf(2,0.5);
r= rpois(10000,0.5)
x2=Prob_devia(10000,r,0.5,2);
list("Chebychev"=x1, "Monte Carl" = x2)
```

```
## $Chebychev
## [1] 0.0625
##
## $`Monte Carl`
## [1] 0.0142
```

d

Le cas Gaussien $X \sim N(\mu, \sigma)$ $P(X \geq t) \leq exp(-\frac{(t-\mu)^2}{2\sigma^2})$ Donc : $P(X - \mu \geq \delta) \leq exp(-\frac{\delta^2}{2\sigma^2})$ car ici $t = \mu + \delta$

```
Borne_Chernoff_Gaus<- function(sigma,delta)
{
  borne= exp(-(delta**2)/(2*(sigma**2)));
  return (borne);
}

Prob_devia_sans_abs <- function(n,r,mu,delta)
  {
    res<- 0
    for (i in 1:n) {
        if ((r[i]-mu) >= delta)
          res<-res+1
    }
    return(res/n)
  }
```

Quand le cas Gaussien avec $\delta = 1$ $\mu = 0$ $\sigma = 1$

```
x1 = Borne_Chernoff_Gaus(1,1)
r= rnorm(10000,0,1)
x2=Prob_devia_sans_abs(10000,r,0,1);
list("Chernoff"=x1,"Monte Carlo"=x2)
```

```
## $Chernoff
## [1] 0.6065307
##
## $`Monte Carlo`
## [1] 0.1609
```

Quand le cas Gaussien avec $\delta = 2$ $\mu = 1$ $\sigma = 3$

```
x1 = Borne_Chernoff_Gaus(3,2)
r= rnorm(10000,1,3)
x2=Prob_devia_sans_abs(10000,r,1,2);
list("Chernoff"=x1,"Monte Carlo"=x2)
```

```
## $Chernoff
## [1] 0.8007374
##
## $`Monte Carlo`
## [1] 0.2518
```

Quand le cas Gaussien avec $\delta = 3 \;\; \mu = 1 \;\; \sigma = 1$

```
x1 = Borne_Chernoff_Gaus(1,3)
r= rnorm(10000,1,1)
x2=Prob_devia_sans_abs(10000,r,1,3);
list("Chernoff"=x1,"Monte Carlo"=x2)
```

```
## $Chernoff
## [1] 0.011109
##
## $`Monte Carlo`
## [1] 0.0015
```

remarque: $P(X \geq t) \leq exp(-\mu h(\frac{t}{\mu})$ avec $h(x) = (1+x)log(1+x) - x$

Quand le cas Poisson avec $\delta = 3 \;\; \lambda = 1$

```
Borne_Chernoff_Pois<- function(lambda,delta)
{
  borne= exp(-lambda*(1+(delta/lambda)*log(1+(delta/lambda))-(delta/lambda)));
  return (borne);
}
x1 = Borne_Chernoff_Pois(1,3)
r= rpois(10000,1)
x2=Prob_devia_sans_abs(10000,r,1,3);
list("Chernoff"=x1,"Monte Carlo"=x2)
```

```
## $Chernoff
## [1] 0.115454
##
## $`Monte Carlo`
## [1] 0.0217
```

Quand le cas Poisson avec $\delta = 2 \;\; \lambda = 2$

```
x1 = Borne_Chernoff_Pois(2,2)
r= rpois(10000,2)
x2=Prob_devia_sans_abs(10000,r,2,2);
list("Chernoff"=x1,"Monte Carlo"=x2)
```

```
## $Chernoff
## [1] 0.25
##
## $`Monte Carlo`
## [1] 0.1389
```

<–3–>

3.a

Remarque: $P(\bar{X}_n - \mu \geq \delta) \leq exp(-\frac{n(\mu+\delta)^2}{2\sigma^2})$

```
# n est la fois d'échantillons
Borne_Chernoff_Gaus_n <- function(n,mu,sigma,delta)
{
  borne= exp(-n*((mu+delta)**2)/(2*(sigma**2)));
  return (borne);
}
```

```r
Borne_Chernoff_Pois_n<- function(n,lambda,delta)
{
  borne= exp(n*(-lambda*(1+(delta/lambda)*log(1+(delta/lambda))-(delta/lambda))));
  return (borne);
}




x1=Borne_Chernoff_Gaus_n(20,0,1,1)
x2=Borne_Chernoff_Gaus_n(100,0,1,1)
x3=Borne_Chernoff_Gaus_n(1000,0,1,1)
y1=Borne_Chernoff_Pois_n(20,1,1)
y2=Borne_Chernoff_Pois_n(100,1,1)
y3=Borne_Chernoff_Pois_n(1000,1,1)
list("Gaus_20"=x1,"Gaus_100"=x2,"Gaus_1000"=x3)
```

```
## $Gaus_20
## [1] 4.539993e-05
##
## $Gaus_100
## [1] 1.92875e-22
##
## $Gaus_1000
## [1] 7.124576e-218
```

```r
list("Pois_20"=y1,"Pois_100"=y2,"Pois_1000"=y3)
```

```
## $Pois_20
## [1] 9.536743e-07
##
## $Pois_100
## [1] 7.888609e-31
##
## $Pois_1000
## [1] 9.332636e-302
```

3.b

estimateur de μ on met $\mu = la\ moyenne\ de\ Esprence\ empirique$

```r
#
esti_de_mu <-function (N,mu,sigma)
  {
    sample<-array(1:N,c(N,20))
    moy <- 1:N
    for (i in 1:N) {
        sample[i,] <-rnorm(20,mu,sigma)
        for(j in 1:20)
        {
          sum <- 0
          moy[i]= sum+sample[i,j]
        }
        moy[i]=moy[i]/20
```

```
    }
    somme <- 0
    for(i in 1:N){
        somme=somme+moy[i]
    }
    return(somme/N)
}
list("estimateur de mu quand N est 20"= esti_de_mu(20,0,1))
```

```
## $`estimateur de mu quand N est 20`
## [1] 0.007826292
```

```
list("estimateur de mu quand N est 100"= esti_de_mu(100,0,1))
```

```
## $`estimateur de mu quand N est 100`
## [1] -0.001802835
```

```
list("estimateur de mu quand N est 1000"= esti_de_mu(1000,0,1))
```

```
## $`estimateur de mu quand N est 1000`
## [1] -0.00225964
```

estimateur de λ on met $\lambda = la\ moyenne\ de\ Esprence\ empirique$

```
esti_de_lambda <-function (N,lambda)
  {
    sample<-array(1:N,c(N,20))
    moy <- 1:N
    for (i in 1:N) {
        sample[i,] <-rpois(20,lambda)
        for(j in 1:20)
        {
          sum <- 0
          moy[i]= sum+sample[i,j]
        }
        moy[i]=moy[i]/20
    }
    somme <- 0
    for(i in 1:N){
        somme=somme+moy[i]
    }
    return(somme/N)
}
list("estimateur de mu quand N est 20"= esti_de_lambda(20,1))
```

```
## $`estimateur de mu quand N est 20`
## [1] 0.055
```

```
list("estimateur de mu quand N est 100"= esti_de_lambda(100,1))
```

```
## $`estimateur de mu quand N est 100`
## [1] 0.0525
```

```
list("estimateur de mu quand N est 1000"= esti_de_lambda(1000,1))
```

```
## $`estimateur de mu quand N est 1000`
## [1] 0.05025
```

Remarque:

On peut remarquer que plus la fois d'échantillon est grande,plus l'estimateur est proche que le vrai espérence

<–4–>

4.a

```
moyenne_empirique<- function(n,vec)
{
  sum <- 0
  for(i in vec)
    {
    sum=i+sum
  }
  return(sum/n)
}

for (n in c(20,100,1000,10000)){
  cauchy <- rcauchy(n,0,1)
  moy_emp <- moyenne_empirique(n,cauchy)
  cat("n=",n,"la moyenne empirique est",moy_emp,"\n")
}
```

```
## n= 20 la moyenne empirique est 0.4714217
## n= 100 la moyenne empirique est -0.2543739
## n= 1000 la moyenne empirique est -0.7597563
## n= 10000 la moyenne empirique est -0.2062845
```

La moyenne empirique ne converge pas selon "n"

4.b

on considère le "standard Cauchy distribution",c'est à dire $\theta = 0$, la fonction caractéristique sera $\Phi(t) = exp(-|t|)$, on trouve qu'il n'est pas dérivable en 0 ,c'est à dire son ésperence n'exsite pas .

4.c

```
for(n in c(20,100,1000,10000)){
  cauchy <- rcauchy(n,0,1)
  s <- sort(cauchy)
  cat("la médiane de taille n=",n,"est",s[n/2+1],"\n")
}
```

```
## la m<U+00E9>diane de taille n= 20 est 0.4400188
## la m<U+00E9>diane de taille n= 100 est -0.1022014
## la m<U+00E9>diane de taille n= 1000 est 0.06145867
## la m<U+00E9>diane de taille n= 10000 est -0.01320312
```

on mets estimateur de $\theta =$ médiane