Some numerical methods with R

Tom Rohmer

HAL Id: hal-04043102
https://hal.inrae.fr/hal-04043102

Submitted on 23 Mar 2023

Some numerical methods with R

Tom Rohmer (tom.rohmer@inrae.fr)

2023-03-09

Contents

1 Generating distributions

In the following exercises, we will use the R functions:

- **rnorm, rexp, rbinom, rpois** to **simulate** random variables with respectively Gaussian, exponential, binomial and poisson distributions
- **pnorm, pexp, pbinom, ppois** to to evaluate the **c.d.f.** of random variables with respectively Gaussian, exponential, binomial and poisson distributions
- **qnorm, qexp, qbinom, qpois** to to evaluate the **quantile function**, i.e. the inverse of the c.d.f. of random variables with respectively Gaussian, exponential, binomial and poisson distributions
- **dnorm, dexp, dbinom, dpois** to to evaluate the **density or p.m.f.** of random variables with respectively Gaussian, exponential, binomial and poisson distributions

1.1 Simulation of classical distributions

First, fix the seed by using the following command

```
set.seed(1)
```

- Using the R functions rexp, rnorm, rbinom, generate i.i.d. samples X_1, X_2 and X_3 of size $n = 10000$ respectively from exponential distribution with shame parameter $\lambda = 2$, from Gaussian distribution with mean value $\mu = 3$ and standard deviation $sd = 2$ and from Bernoulli distribution with probability of success $p = 0.25$.

- Create a data frame X (R command data.frame) which contains X_1, X_2 and X_3. Print the top of the data frame (use head).
- Using the R function apply, Calculate the mean value and the empirical variance of the samples. Compare with the theoretical values. Evaluate the Pearson's correlation between the 3 samples and also the covariance matrix (use $cov(X)$).
- Plot the histogram of the distributions of the exponential et Gaussian distribution and add the corresponding density curves (use hist(X1, freq=FALSE))
- print the table of frequency (use table(X3)/n) and compare these values with p and $1-p$

```r
n=10000

X1<-rexp(n,2)
X2<-rnorm(n,3,2)
X3<-rbinom(n,1,0.25)

X=data.frame(exp=X1,Gauss=X2,Bernouilli=X3) ###concatenation de X1, X2 et X3
head(X)  ##Print the head of the matrix (or dataset) X
```

```
##          exp    Gauss Bernouilli
## 1 0.37759092 4.869048          0
## 2 0.59082139 3.580626          0
## 3 0.07285336 3.387249          0
## 4 0.06989763 5.147031          1
## 5 0.21803431 2.286873          0
## 6 1.44748427 3.790633          0
```

```r
apply(X,2,mean)
```

```
##       exp     Gauss Bernouilli
## 0.4991806 3.0078905  0.2482000
```

```r
apply(X,2,var)
```

```
##       exp     Gauss Bernouilli
## 0.2578852 4.0025920  0.1866154
```

```r
cov(X)
```

```
##                   exp        Gauss   Bernouilli
## exp        0.25788516 -0.023411310 -0.006222800
## Gauss     -0.02341131  4.002592022 -0.005486475
## Bernouilli -0.00622280 -0.005486475  0.186615422
```
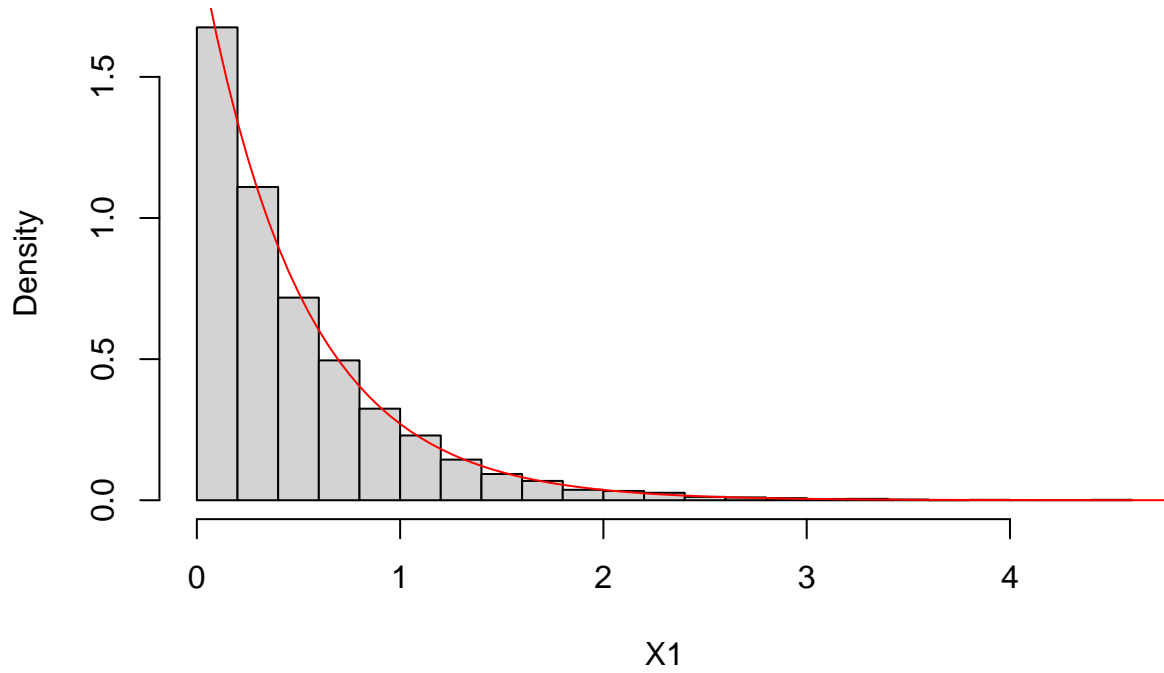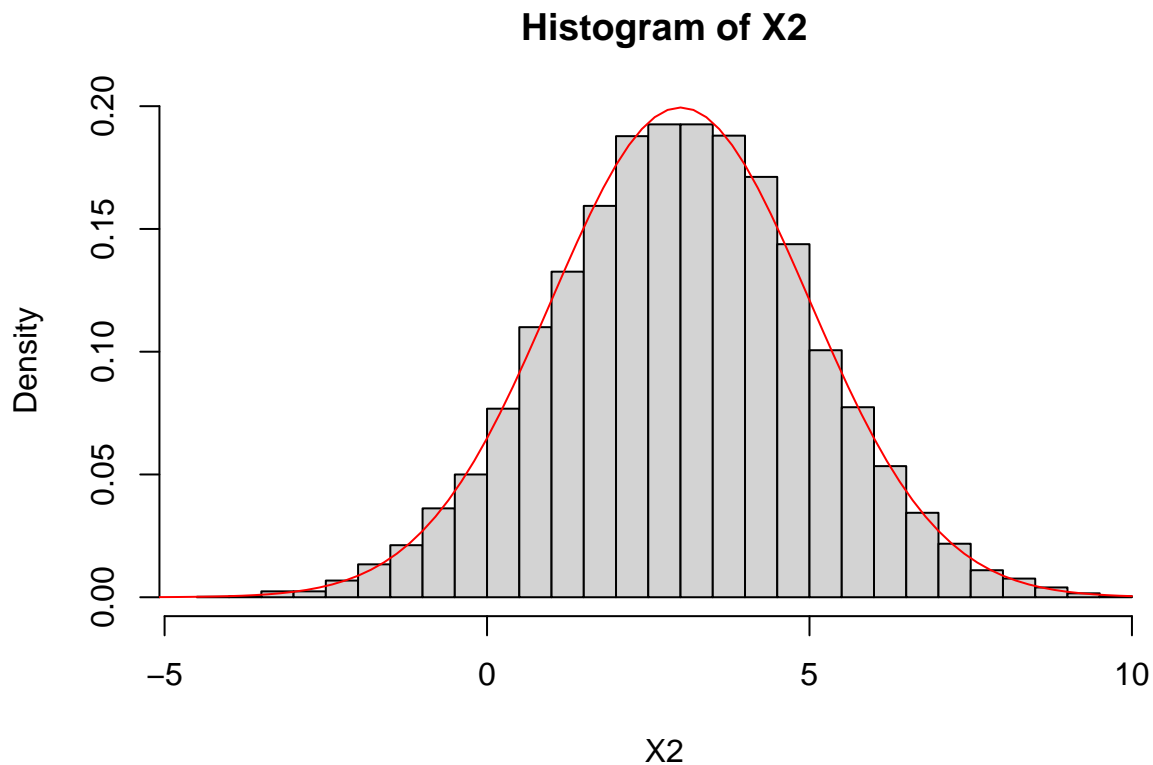
```r
hist(X1,breaks=30,freq=FALSE)
plot(function(x) dexp(x,2),add=TRUE,col="red",xlim=c(0,5))
```

Histogram of X1


```
hist(X2,breaks=30,freq=FALSE)
plot(function(x) dnorm(x,3,2),add=TRUE,col="red",xlim=c(-10,10))
```

Histogram of X2


```
table(X3)/n
```

```
## X3
##      0      1
## 0.7518 0.2482
```

1.2 Simulating distributions using uniform samplings

Let U a random variable distributed from the uniform distribution on $[0, 1]$. Then for any cumulative distribution function F, the random variable

$$Z = F^{-1}(U) \quad \text{has distribution } F.$$

Using this result, simulate the same exponential, Gaussian samples and answer the same questions as exercise 1. Use the r function runif, qexp, qnorm, qbinom

```
U1=runif(n)
U2=runif(n)
U3=runif(n)
X1= qexp(U1, 2)
X2= qnorm(U2,3,2)
X3=qbinom(U3,1,0.25)
X=data.frame(exp=X1,Gauss=X2,Bernouilli=X3) ###concatenation de X1, X2 et X3
head(X)   ##Affiche le début de la matrice X
```

```
##          exp     Gauss Bernouilli
## 1 0.01972026 4.5880807          0
## 2 0.18615195 0.7783434          0
```

```
## 3 0.59454820 2.5494842            0
## 4 0.08673629 6.3367267            0
## 5 0.91940992 2.3569812            1
## 6 0.78985083 4.1971632            0
```

```
apply(X,2,mean)
```

```
##         exp      Gauss Bernouilli
##    0.507474   3.009853   0.247900
```

```
apply(X,2,var)
```

```
##         exp      Gauss Bernouilli
##   0.2578895  4.0610531   0.1864642
```
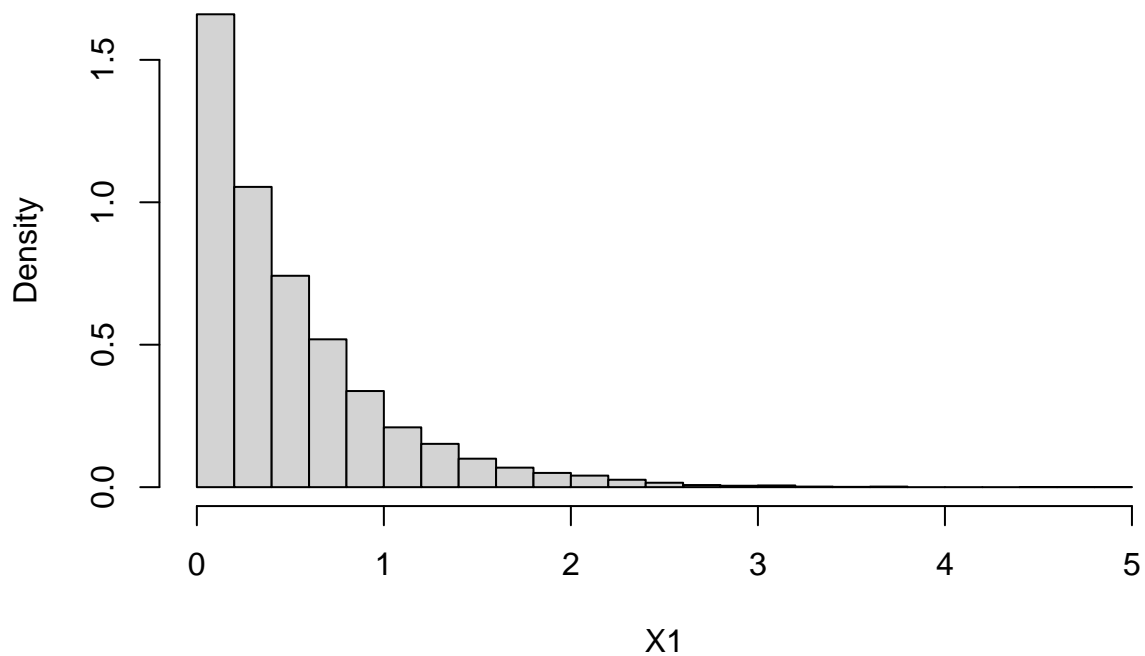
```
cor(X)
```

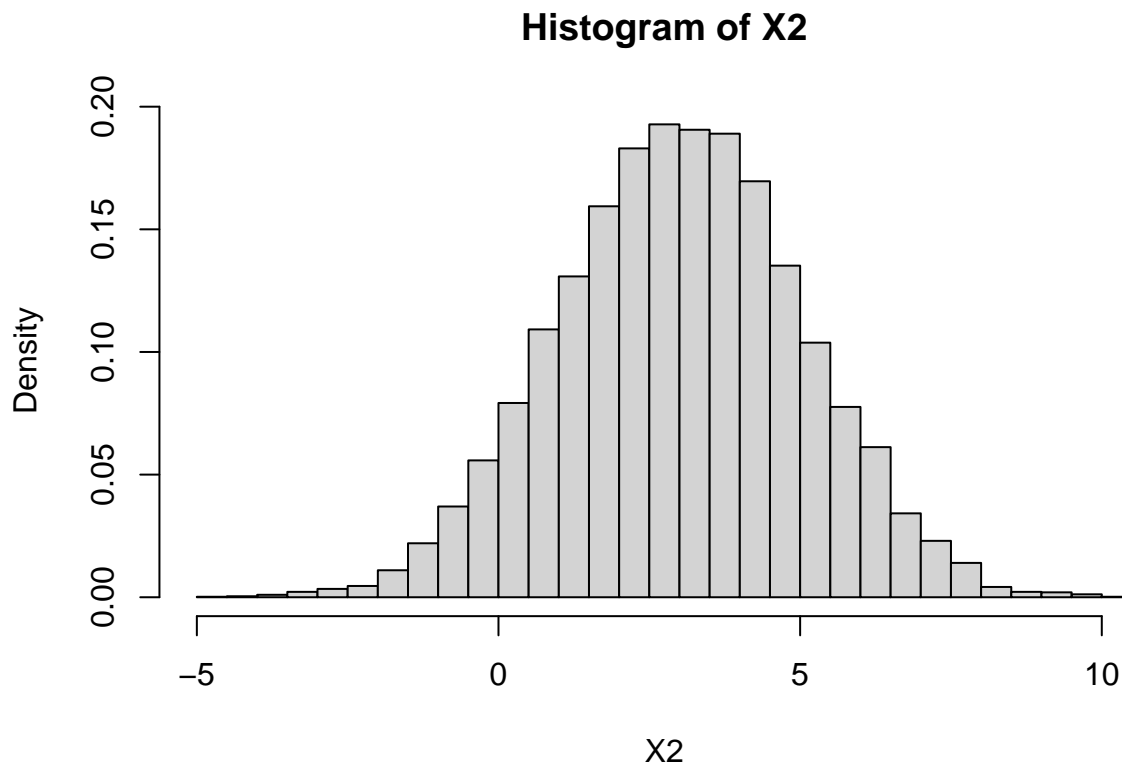```
exp   Gauss   Bernouilli
exp   1.0000000000 -0.0009199897 -0.0121168116
Gauss   -0.0009199897 1.0000000000 -0.0009433278
Bernouilli -0.0121168116 -0.0009433278 1.0000000000
```

```
hist(X1,breaks=30,freq=FALSE)
```

Histogram of X1


```
hist(X2,breaks=30,freq=FALSE)
```

Histogram of X2


```
table(X3)
```

```
## X3
##    0    1
## 7521 2479
```

Let X a continuous random variable with c.d.f. F. Then the random variable $Z = F(X)$ is uniformly distributed on $[0,1]$.
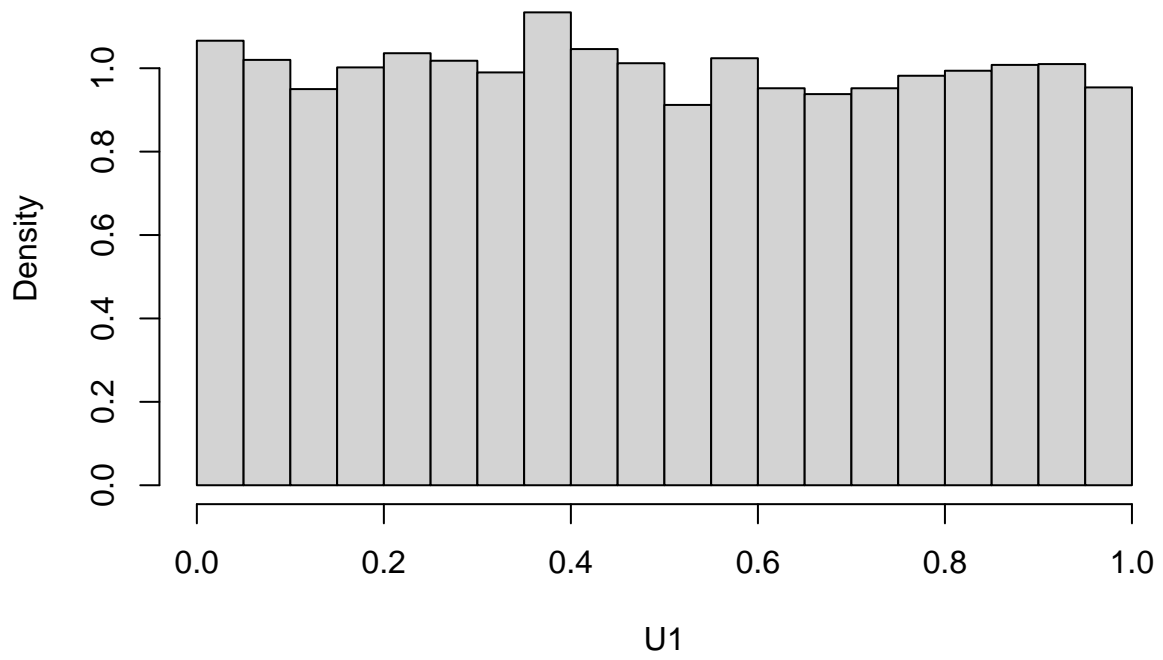
Using this result, from a Gaussian distribution, then from an exponential distribution, simulate a uniform sample on $[0,1]$.

```
X1=rnorm(n,2,3)
X2=rexp(n,3)

U1=pnorm(X1,2,3)
U2=pexp(X2,3)

hist(U1,breaks=30,freq=FALSE)
```

Histogram of U1

```
hist(U2,breaks=30,freq=FALSE)
```

Histogram of U2
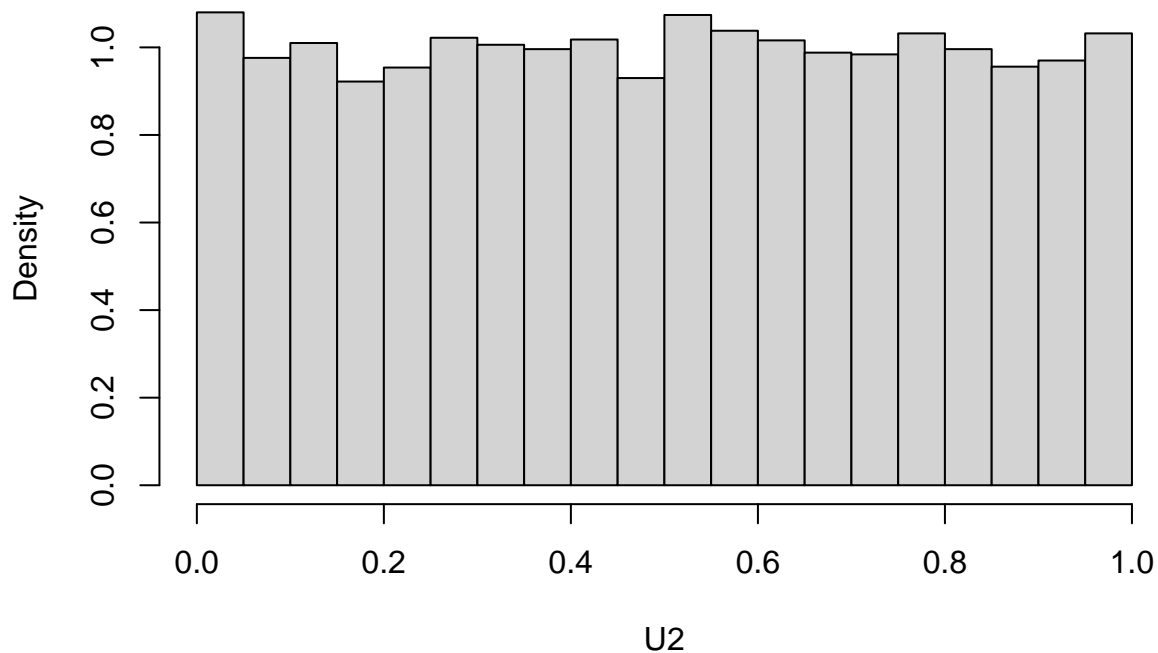
1.3 Theorem Central Limit (TCL), convergence in distribution

Let $X_1, \ldots, X_n$ an i.i.d. sample with expectation $\mathbb{E}(X) = \mu$ and variance $Var(X) = \sigma^2$. Then,

$$\sqrt{n}(\bar{X}_n - \mu) \to \mathcal{N}(0, \sigma^2) \quad \text{as soons as } n \to \infty$$

We say that the mean value $\bar{X}_n$ converges in distribution to a Gaussian distribution.

Using a loop 'for', simulate $M = 10^4$ samples of size $n = 1000$ from a Poisson distribution with parameter $\lambda = 5$ and verify the theorem. Plot the histogram of the statistic with the density of the Gaussian.

```
M=10^4;n=10^3
Z=NULL;
for(m in 1:M){
  X=rpois(n,5)
  Z[m]=sqrt(n)*(mean(X) - 5)
}
hist(Z,freq=FALSE)
plot(function(x) dnorm(x,0,sqrt(5)),add=TRUE,col="red",xlim=c(-6,6) )
```

Histogram of Z
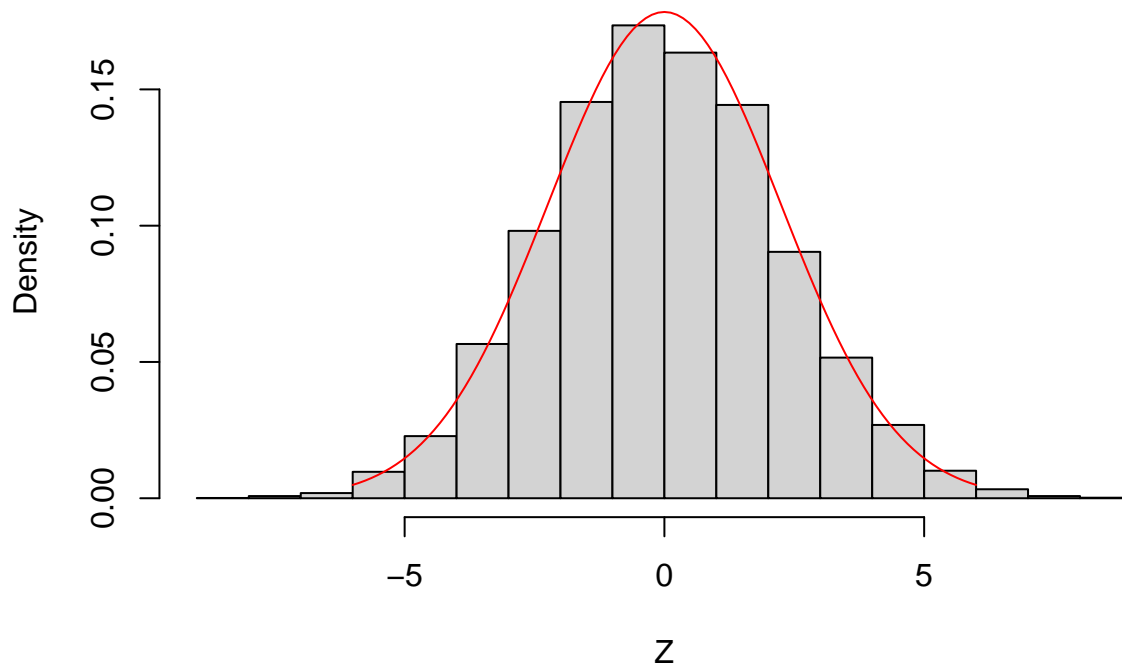
Note that pour a Poisson distribution the expectation and the variance is $\lambda = 5$, so the standard deviation is $\sqrt{\lambda} = 5$.

try now with a exponential distribution with rate parameter $\lambda = 5$. Note that for an exponential distribution the expectation is $1/\lambda = 1/5$, and the variance is $1/\lambda^2 = 1/25$ (so the sd is $1/5$)

```
M=10^4;n=10^3
Z=NULL;
for(m in 1:M){
  X=rexp(n,5)
  Z[m]=sqrt(n)*(mean(X) - 1/5)
}
hist(Z,freq=FALSE)
plot(function(x) dnorm(x,0,sqrt(1/25)),add=TRUE,col="red",xlim=c(-1,1) )
```
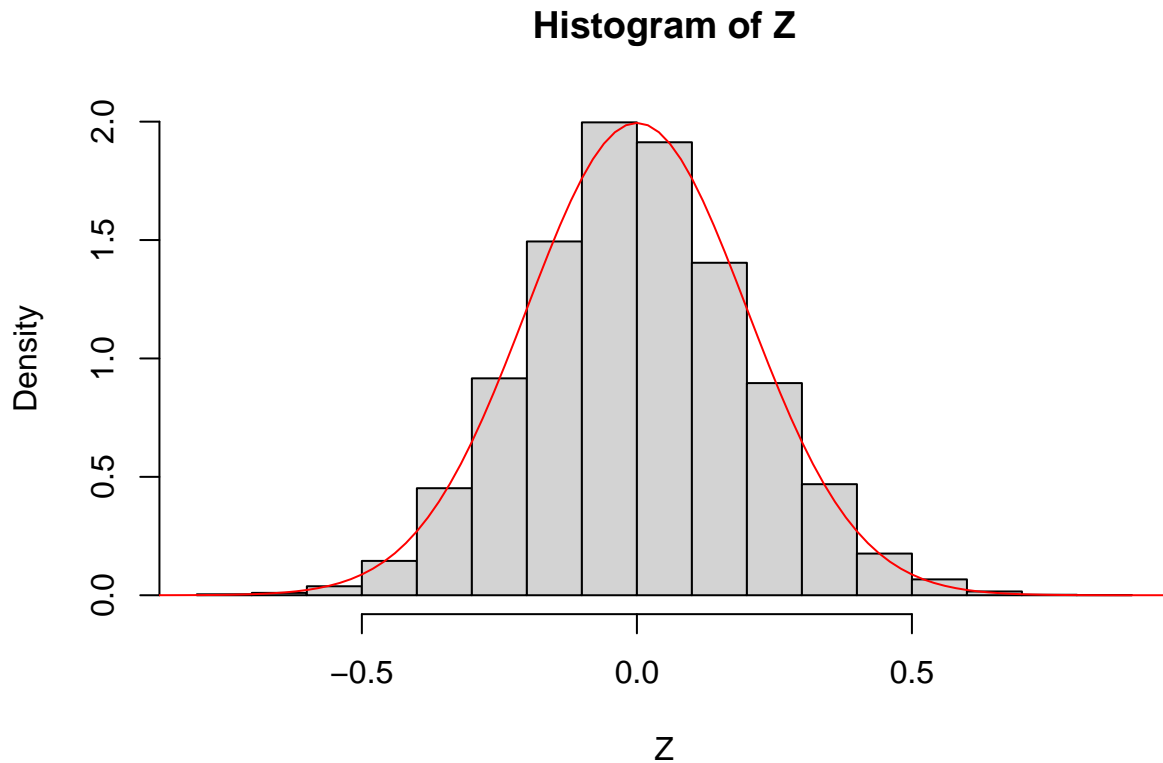
Histogram of Z

1.4 Law of large numbers (LLN), almost surely convergence

Let $X_1, \ldots, X_n$ an i.i.d. sample with expectation μ. Then

$$P(\bar{X}_n \to \mu) = 1.$$

We say that $\bar{X}_n$ converges almost surely (a.s.) to μ.

simulate a sample of size $n = 10^4$ of exponential distribution with parameter $\lambda = 2$. For k varying from 1 to n, evaluate the mean value $Z_k = \bar{X}_k$. Plot the sequence Z_k and add the horizontal line $y = 0.5$ (use abline(h=0.5)).

```
set.seed(2)
n=10^4
Z=NULL;
X=rexp(n,2)
for(k in 1:n){
  Z[k]=mean(X[1:k])
}
plot(Z)
abline(h=1/2,col="red")
```
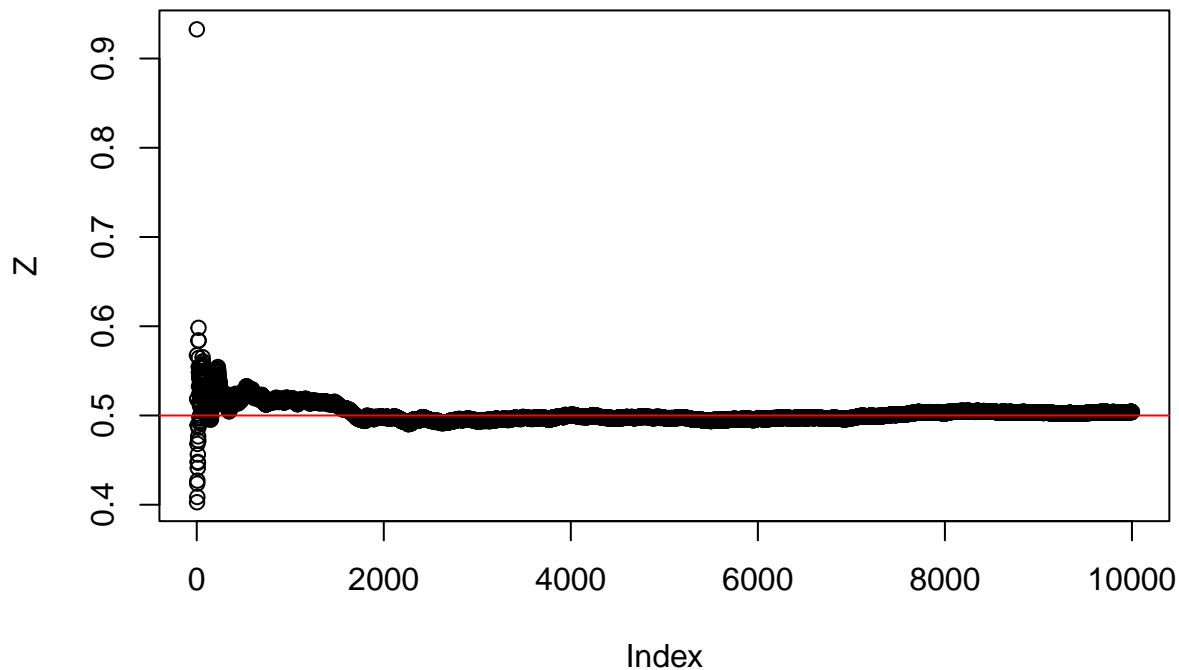
1.5 Multivariate Gaussian distribution

1.5.1 Cholesky decomposition

Any symmetric definite-positive matrix Σ can to be decompose in the product of an lower triangular matrix L (composed with 0 above the diagonal) and his transposed:

$$\Sigma = LL^T.$$

Find these two triangular matrices require specific algebra algorithm (refereed as the Cholesky decomposition).

Let

$$\Sigma = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 1 \end{pmatrix}$$

Using the eigen R function that provides the eigen value of a matrix, check that the matrix Σ is definite-positive (Note that a symmetric matrix is positive definite if and only if all the eigen values of Σ are strictly positives). Use the R function chol to get the lower triangular matrix. Make the matricial product to check that $\Sigma = LL^T$. The R matricial product is %*% and the transpose operation is t().

```
sigma=matrix(c(1,0.4,0.4,1),2,2)
eigen(sigma)
```

```
## eigen() decomposition
## $values
## [1] 1.4 0.6
##
## $vectors
##            [,1]        [,2]
```

```
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

```
L=t(chol(sigma))
L%*%t(L)
```

```
##      [,1] [,2]
## [1,]  1.0  0.4
## [2,]  0.4  1.0
```

1.5.2 Simulating of the multivariate Gaussian distribution

Let $X = (X_1, X_2)$ be a <u>random vector</u>. We say that X is multivariate Gaussian with mean value $m = (m_1, m_2)$ and covariance matrix Σ if the density of X is

$$f(x, m, \Sigma) = \frac{1}{(2\pi)^{n/2}\sqrt{\det\Sigma}} \exp\left(-\frac{1}{2}(x-m)^t\Sigma^{-1}(x-m)\right), \quad x \in \mathbb{R}^2$$

To simulate a multivariate Gaussian sample with mean value $m = (m_1, m_2)$ and covariance matrix Σ we can use the Cholesky decomposition. In fact, Consider Z_1 and Z_2 two independent random variable from the Gaussian distribution with mean value 0 and sd 1. Let $Z = (Z_1, Z_2)$. Then we have the property that the vector $X = L^T Z + m$ is multivariate Gaussian with covariance matrix Σ and mean value m.

- Generate a sample of size n from the multivariate Gaussian distribution with covariance Σ and mean value $m = c(0,0)$. To do that, use the cholesky decomposition (**optional**) or use the R package mvtnorm:

```
##or using mvtnorm
library(mvtnorm)
sigma=matrix(c(1,0.4,0.4,1),2,2) #matrix sigma
m=c(0,0) #mean value vector m
X<-rmvnorm(M,m,sigma)
```

- Plot the sample and the two histograms

- Calculate the mean value and the empirical variance of the two component of the samples and the empirical covariance between the two components.

- Start again with

$$\Sigma_2 = \begin{pmatrix} 1 & 0.1 \\ 0.1 & 1 \end{pmatrix} \text{ and } \Sigma_3 = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}.$$
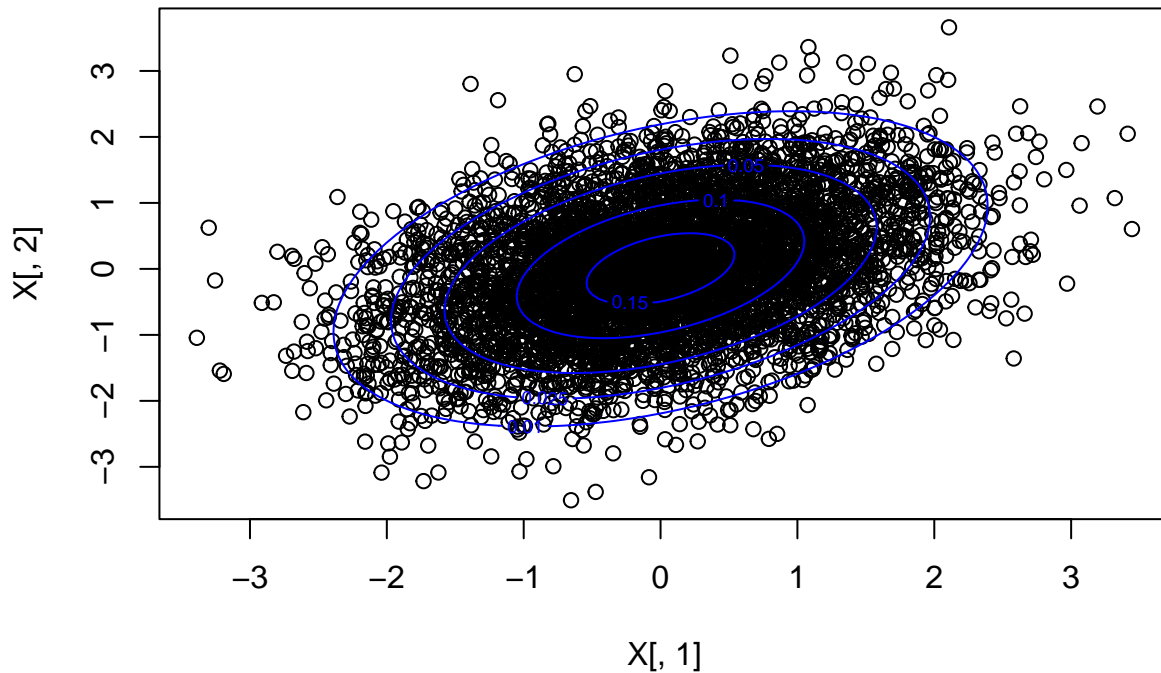
- Use the following code to add the 'contour-plot' of the distribution

```
library(rvinecopulib)
```

```
## Warning: le package 'rvinecopulib' a été compilé avec la version R 4.1.3
```

```
plot(X[,1],X[,2],main=paste("rho=",0.4))
Cn <- bicop_dist(family = "gaussian", rotation = 0, parameters =0.4)
contour(Cn,col="blue",add=TRUE)
```

rho= 0.4

1.6 Other multivariate distributions

Simulate a bivariate sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ in which $X_1, \ldots, X_n$ are i.i.d. and follow a exponential distribution with shape parameter $\lambda = 3$, $Y_1, \ldots, Y_n$ are i.i.d. and follow a Gaussian distribution white mean value $\mu = 2$ and variance $\sigma^2 = 1$, and such that X_1 and X_2 be correlated

To do that:

- Simulate a $(X_1, Y_1), \ldots, (X_n, Y_n)$ following a multivariate Gaussian distribution with mean value $m = (2, 2)$ and covariance 0.4 using the R package mvtnorm and the function rmvnorm.
- Transform $X_1, \ldots, X_n$ in order to obtain an uniform sample following Exercise 2.
- Transform the uniform sample in order to obtain an exponential sample following Exercise 2.
- plot Y vs X. Plot the histogram of $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$
- print the Pearson's correlation and Spearman's correlation.

```
M=10000
rho=0.4
K = matrix(c(1,rho,rho,1),2,2) ##matrice de covariance souhaitée

##in the same way of the previous exercise

L=chol(K) ##matrice triangulaire supérieur
L=t(L) ##matrice triangulaire inférieur
X=matrix(rep(0,M*2),M,2) ##Initialisation de la matrice
for(m in 1:M)
X[m,]=L%*%rnorm(2) + c(2,2) ##Remplissage de la matrice X
```

```
##or using mvtnorm
library(mvtnorm)
X<-rmvnorm(M,c(2,2),K)

X[,1]<-pnorm(X[,1],2,1)
X[,1]<-qexp(X[,1],3)

hist(X[,1],freq=FALSE)
plot(function(x) dexp(x,3),xlim=c(0,3),add=TRUE,col="red")
```
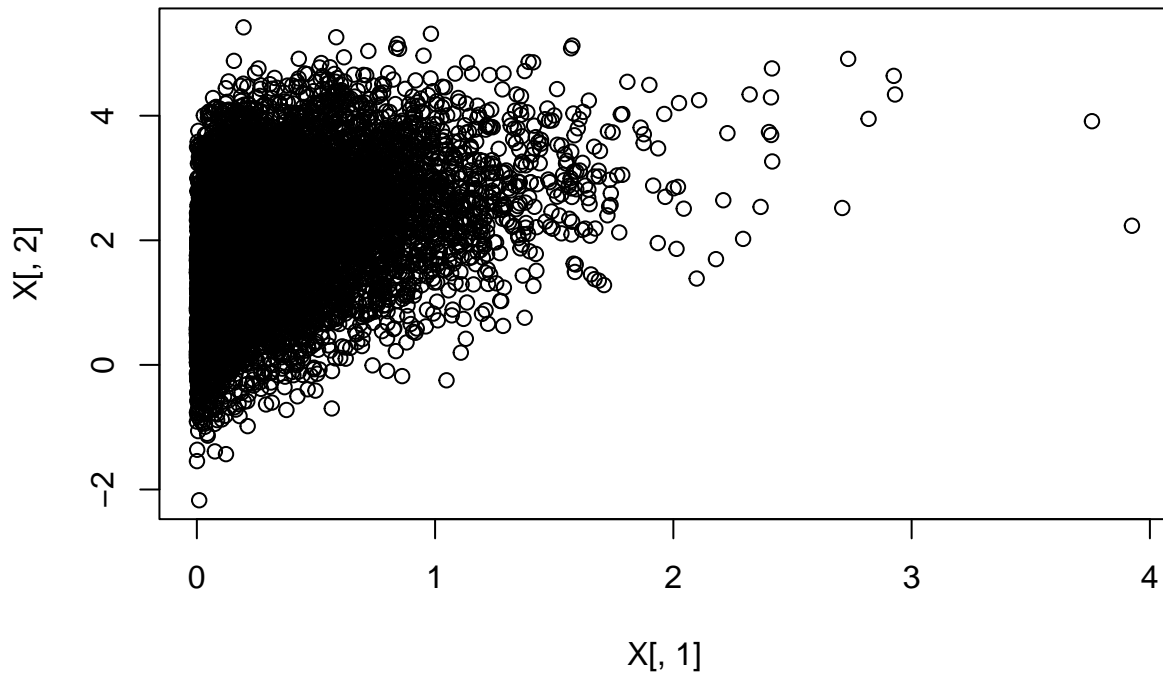
Histogram of X[, 1]


```
plot(X[,1],X[,2])
```
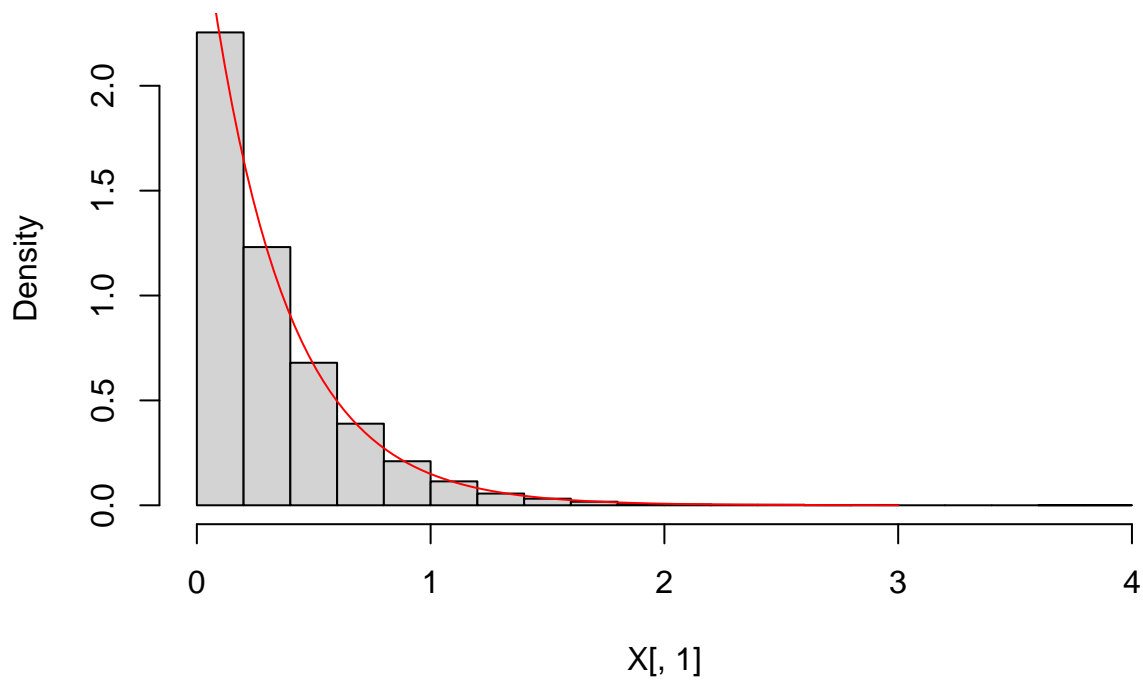
```r
cor(X[,1],X[,2])
```

```
## [1] 0.3643721
```

```r
cor(X[,1],X[,2],method="spearman")
```

```
## [1] 0.3909433
```

- Use the following R code to have the scatterplot of the distribution and the histograms:

```r
xhist <- hist(X[,1], plot=FALSE)
yhist <- hist(X[,2], plot=FALSE)
top <- max(c(xhist$density, yhist$density))
nf <- layout(matrix(c(2,0,1,3),2,2,byrow=TRUE), c(3,1), c(1,3), TRUE)
par(mar=c(4,4,1,1))
#image(vx,vy,z,col=rev(heat.colors(101)))
plot(X,cex=0.2)
#points(X,cex=.2)
par(mar=c(0,3,1,1))
barplot(xhist$density, axes=FALSE, ylim=c(0, top), space=0)
lines((density(X[,1])$x-xhist$breaks[1])/diff(xhist$breaks)[1],
      dexp(density(X[,1])$x,3),col="red")
par(mar=c(3,0,1,1))
barplot(yhist$density, axes=FALSE, xlim=c(0, top), space=0,
        horiz=TRUE)
lines(dnorm(density(X[,2])$x,2,1),(density(X[,2])$x-yhist$breaks[1])/
        diff(yhist$breaks)[1],col="red")
```
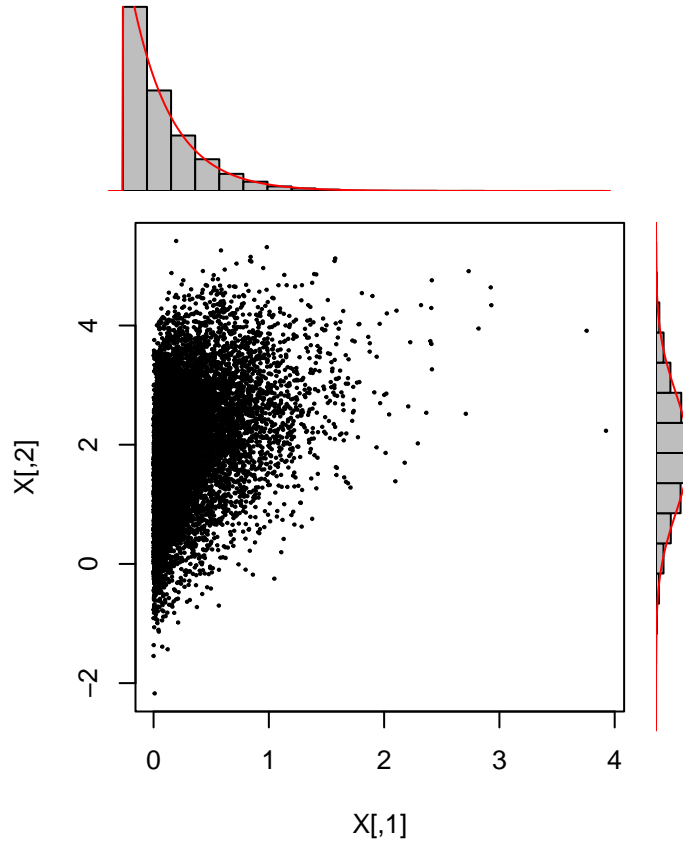
2 Maximum likelihood estimator

2.1 log-Likelihood

Let $y_1, \ldots, y_n$ a i.i.d. sample with density (or p.m.f.) f with parameter θ. The likelihood of the sample $y_1, \ldots, y_n$ is

$$\mathcal{L}(\theta; y_1, \ldots, y_n) = \prod_{i=1}^{n} \mathcal{L}(\theta; y_i)$$

where $\mathcal{L}(\theta; y_i)$ is the function $\theta \mapsto f(y_i, \theta)$. Generally, Instead of the likelihood function, we consider the log-likelihood:

$$\log \mathcal{L}(\theta; y_1, \ldots, y_n) = \log \left(\prod_{i=1}^{n} \mathcal{L}(\theta; y_i) \right) = \sum_{i=1}^{n} \log \mathcal{L}(\theta; y_i).$$

The Maximum Likelihood Estimation (MLE) consists to maximizing the log-likelihood of the sample, (or minimizing $-\log \mathcal{L}(\theta; y_1, \ldots, y_n)$). With some distributions, it can be done to solve the scoring equations, that are

$$\frac{\partial}{\partial \theta} \log \mathcal{L}(\theta; y_1, \ldots, y_n) = 0.$$

for many distributions, the scoring equations do not have an explicit solution, and an optimization procedure can be down to obtain the MLE, use `optim` in `R`.

- Let $y_1, \ldots, y_n$ be a sample from a Gaussian distribution with mean value μ and variance σ^2. Found the MLE of μ and σ^2.

16

$$\log \mathcal{L}(\mu, \sigma^2; y_1, \ldots, y_n) = \sum_{i=1}^{n} \log \mathcal{L}(\mu, \sigma^2; y_i) = \sum_{i=1}^{n} \log \left(\frac{1}{\sqrt{2\pi\sigma^2}} exp \left(-\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right)$$

$$= -\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \sum_{i=1}^{n} \left(\frac{(y_i - \mu)^2}{2\sigma^2} \right)$$

So

$$\frac{\partial}{\partial \mu} \log L(\mu, \sigma^2; y_1, \ldots, y_n) = 0 \Leftrightarrow \frac{\partial}{\partial \mu} \left(-\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \sum_{i=1}^{n} \left(\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right) = 0$$

$$\Leftrightarrow \frac{\partial}{\partial \mu} \sum_{i=1}^{n} \left(\frac{(y_i - \mu)^2}{2\sigma^2} \right) = 0$$

$$\Leftrightarrow -\frac{1}{\sigma^2} \sum_{i=1}^{n} (y_i - \mu) = 0$$

$$\Leftrightarrow \mu = \bar{y}_n$$

$$\frac{\partial}{\partial \sigma^2} \log L(\mu, \sigma^2; y_1, \ldots, y_n) = 0 \Leftrightarrow \frac{\partial}{\partial \sigma^2} \left(-\frac{n}{2} \log(2\pi) - \frac{n}{2} \log(\sigma^2) - \sum_{i=1}^{n} \left(\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right) = 0$$

$$\Leftrightarrow \frac{\partial}{\partial \sigma^2} \left(-\frac{n}{2} \log(\sigma^2) - \sum_{i=1}^{n} \left(\frac{(y_i - \mu)^2}{2\sigma^2} \right) \right) = 0$$

$$\Leftrightarrow -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{n} (y_i - \mu)^2 = 0$$

$$\Leftrightarrow \sigma^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y}_n)$$

```
n=10^4
Y<-rnorm(n,2,2)
mu.MLE<-mean(Y)
sigma2.MLE<-var(Y)
mu.MLE;sigma2.MLE
```

```
## [1] 1.98438
```

```
## [1] 4.060696
```

- Generate a sample (use `rnorm` function) of size $n = 10^6$ from a Gaussian distribution with mean value $\mu = 2$ and variance $\sigma^2 = 4$. Estimate the MLE and illustrate the a.s. convergence of the estimator, that is to say the convergence of the estimator when the sample size increase

```
Me<-Va<-NULL
for(m in 1:M){
Me[m]=mean(Y[1:m])
Va[m]=var(Y[1:m])
}

plot(Me)
abline(h=2,col="red")
```
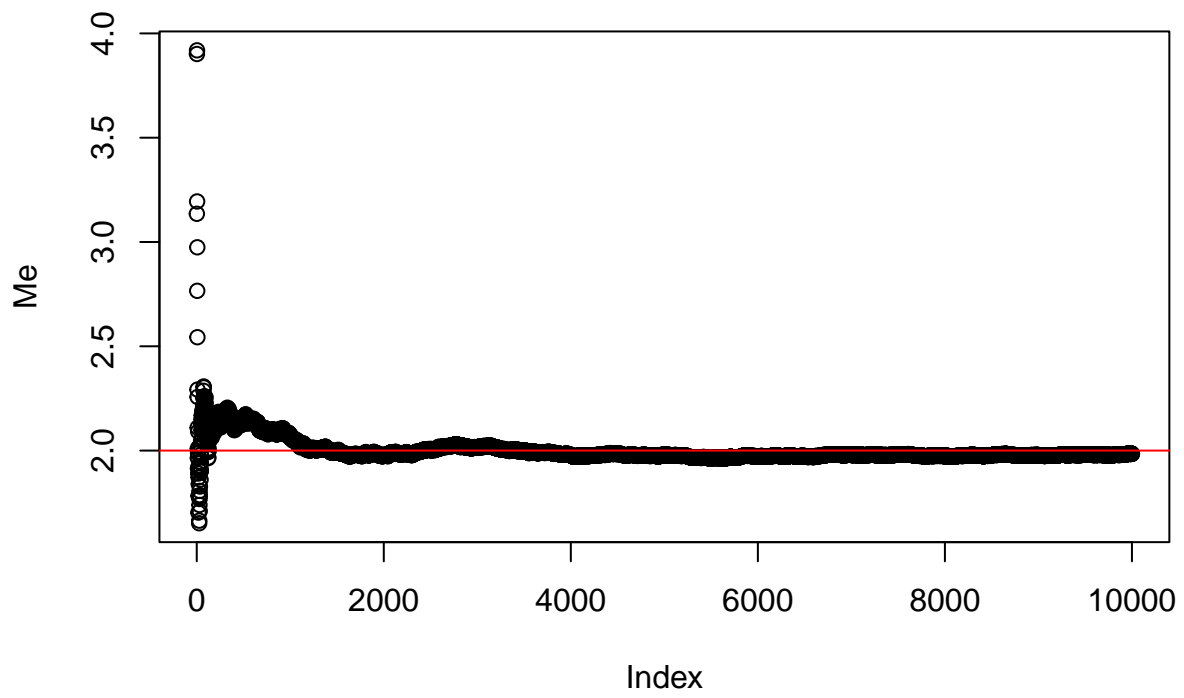
```
plot(Va)
abline(h=4,col="red")
```
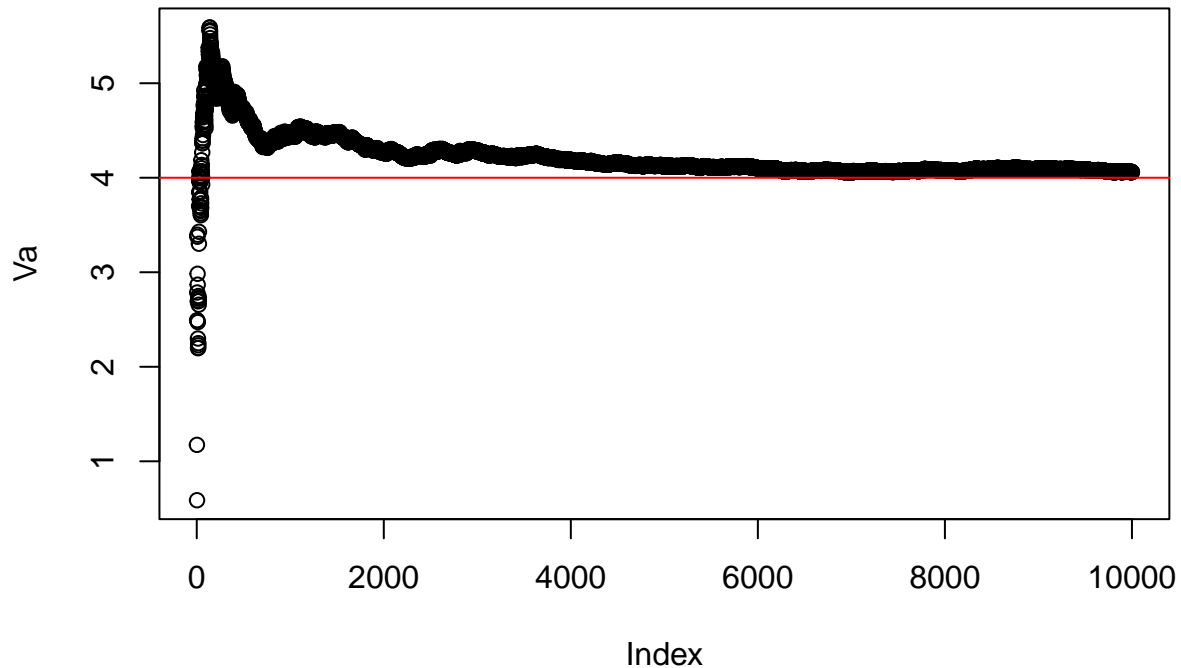
2.2 Gamma distribution

The density of the Gamma distribution with shape parameter α and rate parameter β is defined by

$$f(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} exp(-\beta x), \quad x > 0, \alpha > 0, \beta > 0,$$

where the Gamma function Γ is defined by

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt.$$

Note that $\Gamma(z+1) = z\Gamma(z)$ and $\Gamma(1) = 1$. If Y follows a Gamma distribution with shape parameter α and rate parameter β,

$$\mathbb{E}(Y) = \frac{\alpha}{\beta}, \quad Var(Y) = \frac{\alpha}{\beta^2}.$$

- Check that for $\alpha = 1$, the Gamma distribution is the exponential distribution.

- Plot the density of the Gamma distribution for many parameter α and β (use `dgamma` function).

```
plot(function(x) dgamma(x,1,3),xlim=c(0,5))
plot(function(x) dgamma(x,2,0.5),xlim=c(0,5),col="red",add=TRUE)
plot(function(x) dgamma(x,4,6),xlim=c(0,5),col="blue",add=TRUE)
plot(function(x) dgamma(x,0.5,6),xlim=c(0,5),col="green",add=TRUE)
```
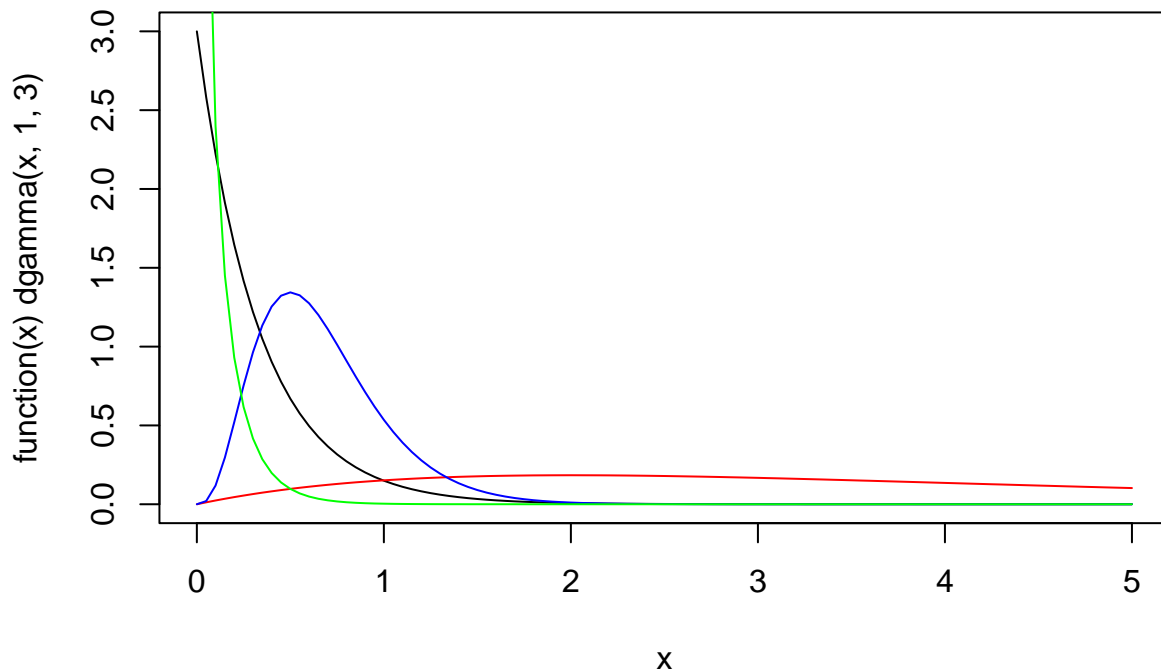
- Generate a sample (use `rgamma` function) of size $n = 10^6$ from a gamma distribution with shape value $\alpha = 3$ and rate parameter $\beta = 5$. Estimate the MLE using the R function `optim`.

```
Y<-rgamma(M,3,5)
```

```
optim(f=function(x) -sum(dgamma(Y,x[1],x[2],log=TRUE)),par=c(0.1,0.1))$par
```

```
## [1] 2.989211 4.927904
```

3 Dichotomy algorithm (bisection method)

The objective of the exercise is to find the value of x such that $F(x) = P(X \leq x) = 0.975$, that is to say $F^{-1}(0.975)$ where X is a random variable following a standard Gaussian distribution, using a dichotomy algorithm

```
plot(function(x) pnorm(x),xlim=c(-6,6))
abline(h=0.975,col="red",lty=2)
abline(v=qnorm(0.975),col="blue",lty=2)
points(c(qnorm(0.975),0.01),pch="?",col="blue")
```
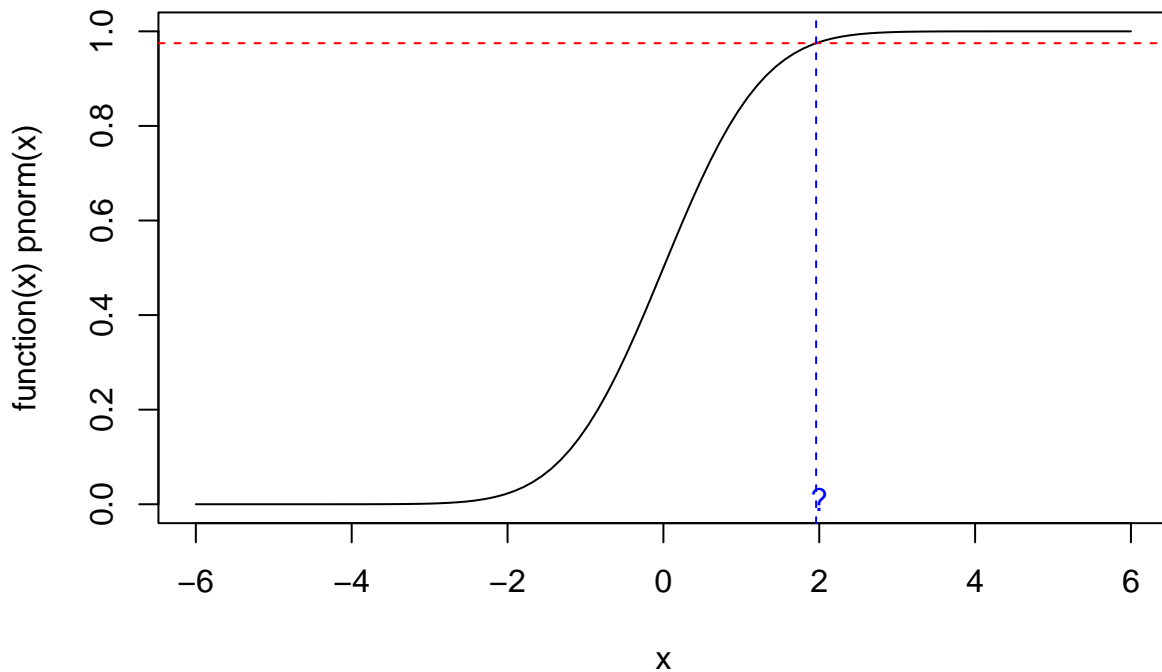
In a equivalent way, we have to solve the equation

$$F(x) - 0.975 = 0$$

Because the c.d.f. are strictly increasing and the c.d.f. of the Gaussian distribution is continuous, we can numerically solve this equation using a dichotomy algorithm.

Define the function $Z(y) = F(y) - 0.975$ and start by define an initial interval $[a, b]$ such that the solution of the equation satisfies $a \leq x \leq b$, that is to say $Z(a)Z(b) < 0$.

```
a=-6; b=6
Z<-function(y) pnorm(y)-0.975
Z(a)*Z(b)
```

```
## [1] -0.024375
```

Then consider $c = \frac{a+b}{2}$ and the interval $[a, c]$. If the solution x is in the interval $[a, c]$, necessarily $Z(a)Z(c) < 0$. If the solution x is in the interval $[c, b]$, $Z(a)Z(c) \geq 0$. Hence, if $Z(a)Z(c) < 0$, we can look for a solution in $[a, c]$ and else we can look for a solution in $[c, b]$.\

Consider now the middle of the new interval and find a smaller interval which contains x. Reiterate the algorithm as many times as necessary until you get an interval smaller than an ϵ, chosen by user. (We will consider $\epsilon = 10^{-7}$). The final value for the estimation of x will be the middle of the considered interval.

To do that:

- define a maximum number of iteration, $maxit = 1000$ and do a loop from 1 to maxit.
- Use the following command to stop the algorithm:

```
if(abs(b-a)<eps) break;
```

Then

- print the number of iteration to reach the solution
- Compare the solution with the true value that is $qnorm(0.975)$
- print the convergence of the algorithm

```
maxit=1000
eps=10^{-7}
c=NULL
for(m in 1:maxit)
{
  c[m]=(a+b)/2
  if(Z(a)*Z(c[m]) <0) b<-c[m]
  if(Z(a)*Z(c[m]) >=0) a<-c[m]

  if(abs(b-a)<eps) break;
}
c[m]=(a+b)/2
#print(c[m])
#print(qnorm(0.975))

plot(c, type='l',ylab="value",xlab="iteration")
abline(h=qnorm(0.975),col="red",lty=2)
```
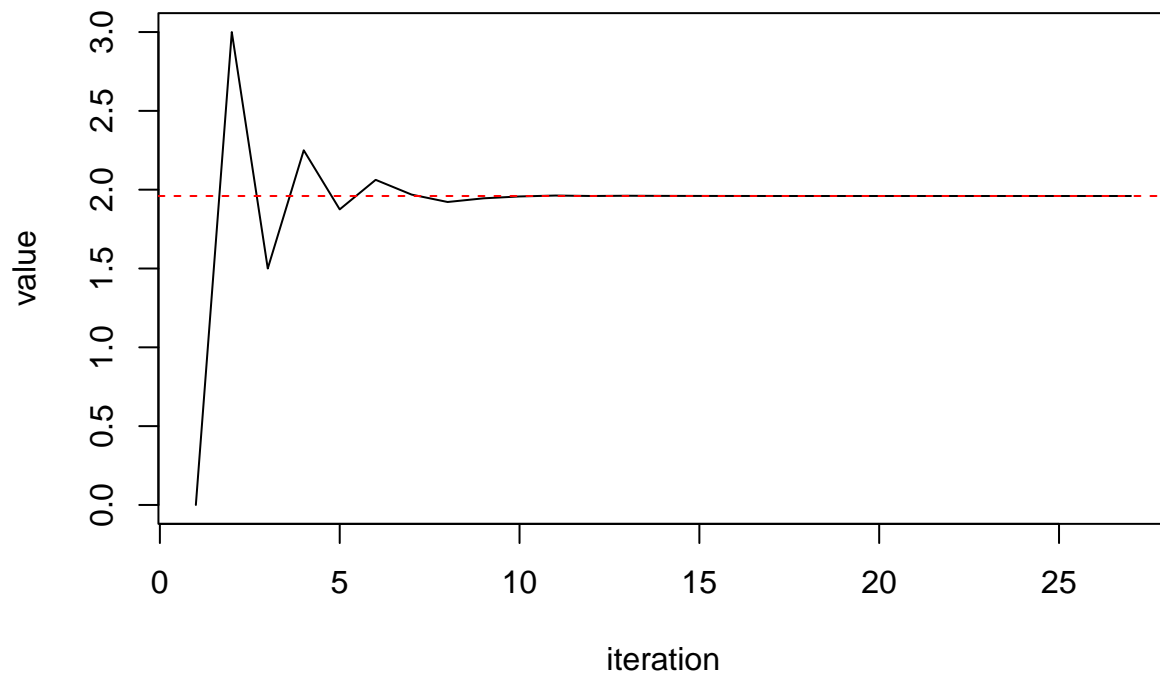
```
cat(paste("we used",m,"iterations to get the solution",c[m]))
```
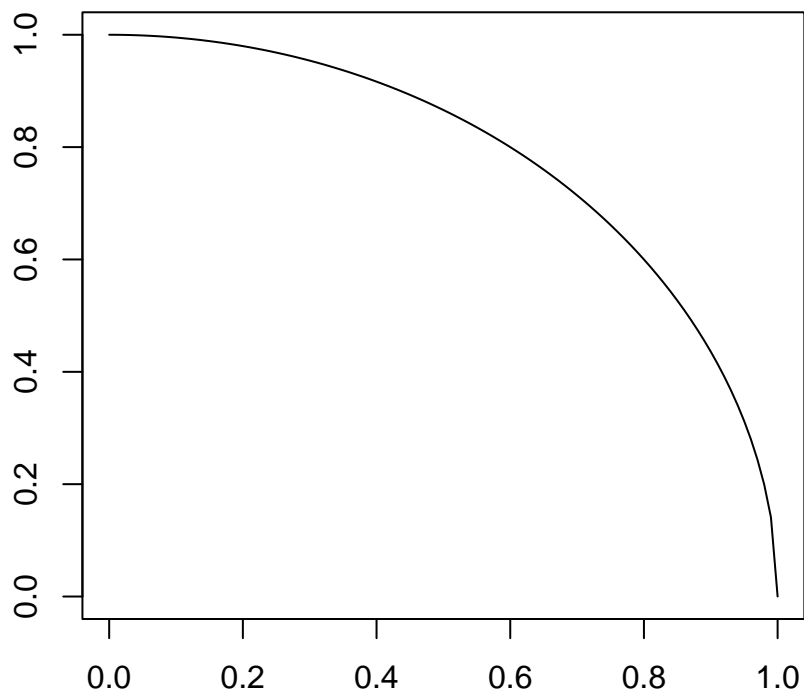
```
## we used 27 iterations to get the solution 1.95996402204037
```

4 Monte Carlo approximation

4.1 Monte Carlo approximation of π

Let recall that the equation of the unit circle (circle with a radius of 1) is given by $x^2 + y^2 = 1$. In the following Figure, the circle equation $y = \sqrt{1 - x^2}$ is plotted on [0,1].

```
plot(function(x) sqrt(1-x^2),ylab="",xlab="")
```


The objective of this exercise is to approximate π. Let's start by sampling random vectors $U_1 = (U_{1,1}, U_{1,2})$ and $U_2 = (U_{2,1}, U_{2,2})$ according to independent uniform variables on $[0, 1]$. Then we look for the position of U_1 and U_2 in the unit square:

```
set.seed(1)
plot(function(x) sqrt(1-x^2),ylab="",xlab="")
U1=runif(2)
U2=runif(2)
points(U1[1],U1[2],col="red",pch=3)
text(U1[1]+0.04,U1[2]+0.01,"U1",col="red")
segments(x0=0,y0=U1[2],x1=U1[1],y1=U1[2],lty=2,col="red")
```
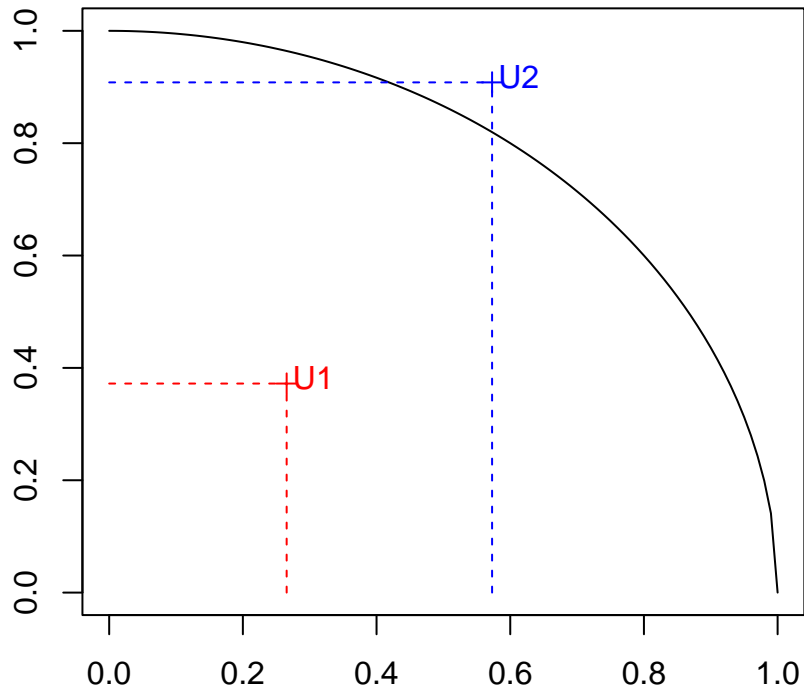
```
segments(x=U1[1],y0=0,x1=U1[1],y1=U1[2],lty=2,col="red")

points(U2[1],U2[2],col="blue",pch=3)
text(U2[1]+0.04,U2[2]+0.01,"U2",col="blue")
segments(x0=0,y0=U2[2],x1=U2[1],y1=U2[2],lty=2,col="blue")
segments(x=U2[1],y0=0,x1=U2[1],y1=U2[2],lty=2,col="blue")
```


In this example,

- $U_1 = (U_{1,1}, U_{1,2}) = (0.27, 0.37)$ is under the curve of the quarter circle: $U_{1,1}^2 + U_{1,2}^2 = 0.27^2 + 0.37^2 = 0.21 < 1$.
- $U_2 = (U_{2,1}, U_{2,2}) = (0.57, 0.91)$ is above the curve of the quarter circle: $U_{2,1}^2 + U_{2,2}^2 = 0.57^2 + 0.91^2 = 1.15 > 1$.

Remember that the area of the circle is π so the area of the quarter circle is $\pi/4$. What is the probability for any simulated $U = (U_1, U_2)$ where U_1 and U_2 are uniform, to be under the curve of the quarter circle?

To estimate $\pi/4$, Monte Carlo procedure consists in simulating M random vectors $U_1 = (U_{1,1}, U_{1,2}), \ldots, U_M = (U_{M,1}, U_{M,2})$, according to uniform distributions on $[0,1]$, and evaluate the proportion of the points U_m which are under the curve of the quarter circle, that is to say, such that $U_{m,1}^2 + U_{m,2}^2 < 1$.

- Using $M = 10^4$, simulate M such random vectors
- Plot the points which are under the curve of the quarter circle in red and the points which are above the curve of the quarter circle in blue:

```
set.seed(1)
M=10^4
U<-matrix(runif(2*M),M,2)
plot(function(x) sqrt(1-x^2),ylab="",xlab="",lwd=3)
points(U[U[,1]^2+U[,2]^2<1,],col="red",pch='.')
points(U[U[,1]^2+U[,2]^2>1,],col="blue",pch='.')
```
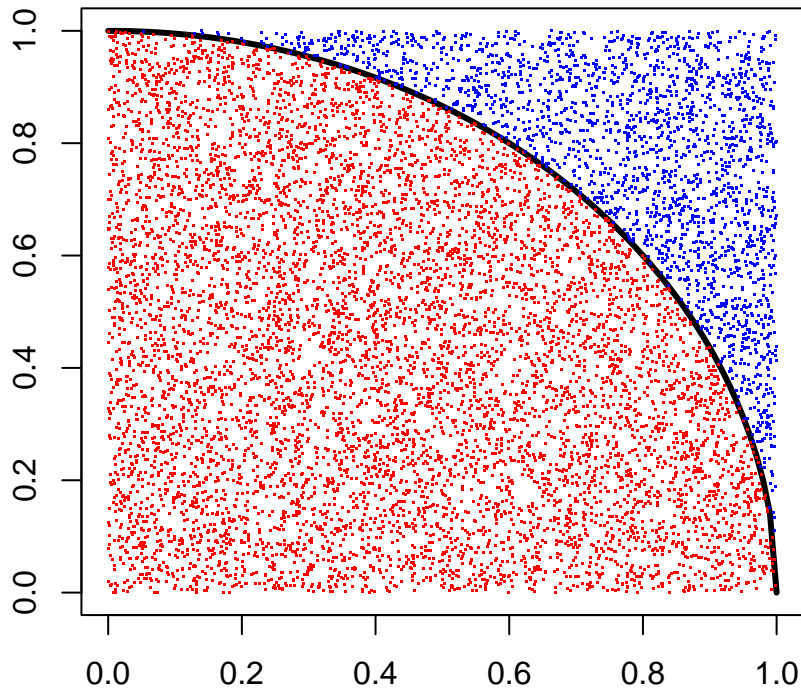


```
MC.pi=4*mean(U[,1]^2+U[,2]^2<1)
#print(MC.pi)
```

Then, $\pi/4$ can be approximated by the average of points which are under the curve of the quarter circle (among the total number of point)

$$\pi/4 \approx \frac{\text{number of red points}}{\text{total number of points}}$$

- Evaluate $\pi/4$ then π

- Illustrate the (almost surely) convergence of the Monte Carlo estimator of π. (Note that in R, you can access to the true value of π using the R command pi):
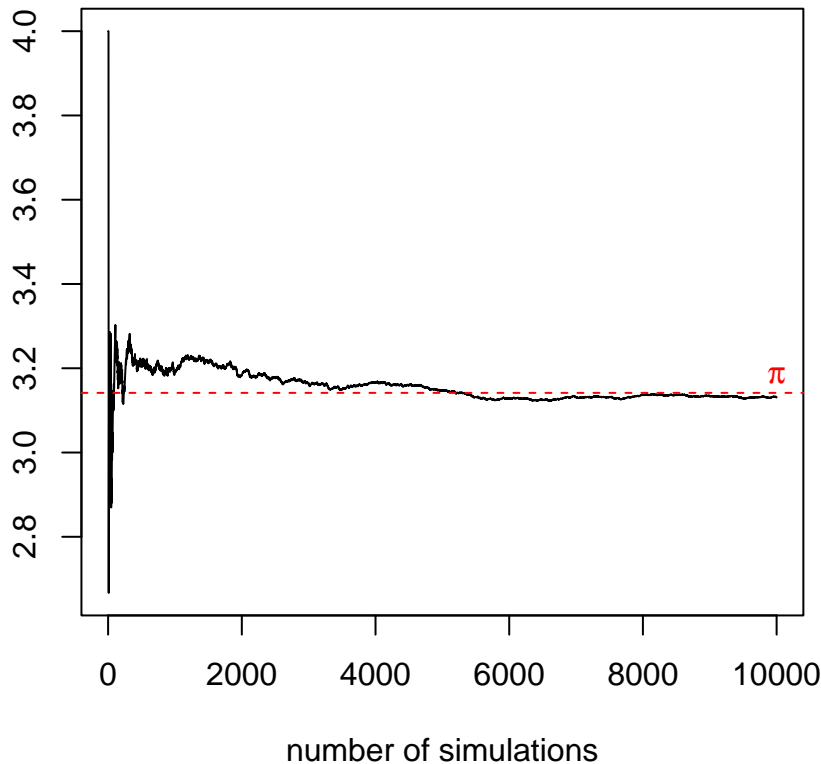
```
Z=NULL
for(k in 1:M)
Z[k] =4*mean(U[1:k,1]^2+U[1:k,2]^2<1)
```

```
plot(Z,type='l',xlab="number of simulations",ylab="")
abline(h=pi,col="red",lty=2)
text(10000+0.04,pi+0.04,expression(pi),col="red")
```
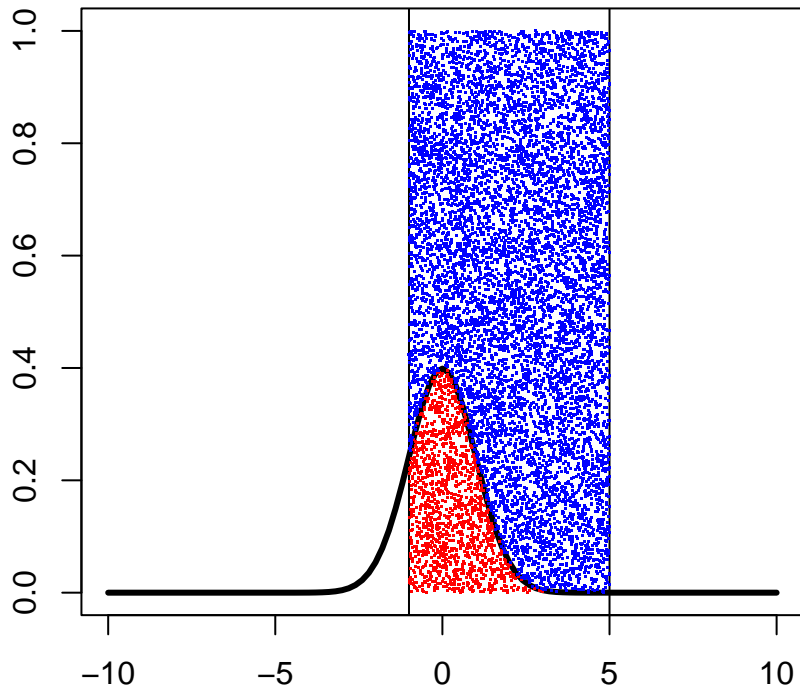

number of simulations

4.2 Monte Carlo approximation of the area under the Gaussian density

In the the same way of exercise 1, propose a Monte Carlo(MC) algorithm to evaluate the area under the Gaussian density between $a = -1$ and $b = 5$, that is to say the probability $P(a \leq X \leq b) = P(X \leq b) - P(X \leq a)$ for X a random variable with standard Gaussian distribution.

- Plot the density of the Gaussian distribution on the interval $(-10, 10)$.
- In the same graphic, add the vertical lines $a = -1$ and $b = 5$.
- Add the points of the MC algorithm which are under the Gaussian curve in red and the points of the MC algorithm which are above the Gaussian curve in blue

$$P(a \leq X \leq b) \approx \frac{\text{number of red points}}{\text{total number of points}} \times (b - a)$$

- Evaluate the value of the area under the Gaussian density between $a = -1$ and $b = 5$ and compare with the theoretical value obtained using *pnorm*
- Illustrate the (almost surely) convergence of the Monte Carlo estimation of the area under the Gaussian curve.

```
## [1] 0.8346
## [1] 0.8413445
```