```
## Constante ds Chasapis versus Pinelis
########

# répertoire
setwd("C:/Users/gauthe01/Documents/4. Recherche bureau/1. Themes/Concentration mesure")

#Packages
library(extraDistr) #pour rademacher

###  Paramètres
#pas des t
length_t=100
#nombre de vecteurs déterministes "a" de dimension "d"
#n<-50
#dimension "d" de chaque vecteur "a"
#d<-2
#Nombre d'itérations de la méthode de Monte Carlo (MC)
#N<-10000
#Constantes
#C<-0.5/((1-pnorm(sqrt(2),0,1))) #supposition
#Ctest1<-2*exp(3)/9
#Ctestp<-3*exp(2)/4

graphe<-function(x, length_t=100, N=10000, Ctest1=2*exp(3)/9){
     d<-x[1]
     n<-x[2]

###On simule l'ensemble des n vecteurs de taille déterministes de manière aléatoire (:-)), unif entre -100 et 100
  a<-matrix(runif(n*d, min=-100, max=100), ncol=n, nrow=d) #n vecteurs (en colonne) de taille d (en ligne)

### Calcul numérique de la combinaison linéaires des rademacher et de sgaussiennes en d dimensions

  #Initialisation du tableau des résultats de MC. Indique la valeur de la norme de la somme des rademacher pondérée
  normecombi_rade <- matrix(rep(0,N), ncol=N) # 1 ligne, N colonnes
  normecombi_gauss<- matrix(rep(0,N), ncol=N) # 1 ligne, N colonnes

  #Boucle pour MC
  for (i in 1:N) {
    #Génération aléatoire de n rademacker et gauss indép.
      rade<-matrix(rsign(n), ncol=n, nrow=1)  # 1 ligne, n colonnes
      gauss<-matrix(rnorm(n), ncol=n, nrow=1)  # 1 ligne, n colonnes
    #Calcul de la combinaison linéaire de la somme des a_i rademacher_i et gauss_i respect.
      combi_rade<-a%*%t(rade) # d lignes, 1 colonne
      combi_gauss<-a%*%t(gauss) # d lignes, 1 colonne
    #Calcul de la norme euclidienne des combinaisons de rademacher et des gaussiennes. vecteurs de taille N
      normecombi_rade[1,i]<-sqrt(t(combi_rade)%*%combi_rade)
      normecombi_gauss[1,i]<-sqrt(t(combi_gauss)%*%combi_gauss)
        }
#normecombi_rade
#normecombi_gauss

### On met cela en pourcentage entre 0 et 1

points_t<-seq(0.0000001, (max(normecombi_gauss, normecombi_rade)+1), l=length_t)

### calcul des proba de réaliser l'évenement norme > à t

proba_rade<-matrix(rep(0,length_t), ncol=length_t) # 1 ligne, N colonnes
proba_gauss<-matrix(rep(0,length_t), ncol=length_t ) # 1 ligne, N colonnes

for (k in 1:length_t) {
proba_rade[,k]<-mean(normecombi_rade[1,]>points_t[k])
proba_gauss[,k]<-mean(normecombi_gauss[1,]>points_t[k])
}
proba_rade
#proba_gauss

#### Tracés

plot(points_t, proba_rade, col='red',t='l',main="" , xlab="t", ylab="tail estimator ")
title(main=c(paste("n=",x[2],"and p=",x[1])))
lines(points_t, Ctest1*proba_gauss, col="green", t="l")
}



#graphe(c(2,50))

par(mfrow=c(2,4))
title(main="In p dimension, red line : combinaison of n Rademacher,  green line : with constant C, combinaison of n Gaussian"
```

```
#1ere coordonnée=dimension p, 2nde coordonnée=taille de l'échantillon n
graphe(c(10,10))
graphe(c(50,10))
graphe(c(200,10))
graphe(c(1000,10))


graphe(c(10,50))
graphe(c(50,50))
graphe(c(200,50))
graphe(c(1000,50))


par(mfrow=c(2,3))
title(main="In p dimension, red line : combinaison of n Rademacher,  green line : with constant C, combinaison of n Gaussian"

graphe(c(200,200))
graphe(c(1000,200))
graphe(c(5000,200))

graphe(c(200,1000))
graphe(c(1000,1000))
graphe(c(5000,1000))


### différence entre les survies des combi de rademacher et les combi de gauss
#cbind(max=max(Ctest*proba_gauss - proba_rade),min=min(Ctest*proba_gauss - proba_rade))
```