```
#        Cours de Stat Mass 07: Methode de Monte-Carlo - un theorème d'Archimède
 help("runif")
# dunif(x, min = 0, max = 1, log = FALSE)
# punif(q, min = 0, max = 1, lower.tail = TRUE, log.p = FALSE)
# qunif(p, min = 0, max = 1, lower.tail = TRUE, log.p = FALSE)
# runif(n, min = 0, max = 1)
# Arguments
# x, q vector of quantiles
# p vector of probabilities.
# n number of observations. If length(n) > 1, the length is taken to be the number required.
library(rgl) # raster graphic library: se télécharge à
# http://cran.r-project.org/bin/windows/contrib/3.1/rgl_0.95.1201.zip

# On tire un echantillon de points du demi-cube des z dans [0,1]
n=10000;a=-1;b=+1;h=1
x=runif(n,a,b);
y=runif(n,a,b);
z=runif(n,0,h);
MonNuage=data.frame(x,y,z)
# Représentations graphiques des trois solides, en couleur
plot3d(MonNuage)
aspect3d("iso") # pour avoir des axes orthonormés
# help("plot3d")
couleurCyl <- ifelse(x^2+y^2<1, "green", "red")
plot3d(MonNuage,col=couleurCyl)
couleurCone <- ifelse(x^2+y^2<z^2, "blue", "NA")
plot3d(MonNuage,col=couleurCone,size=1)
couleurSphere <- ifelse(x^2+y^2+z^2<1, "yellow", "NA")
plot3d(MonNuage,col=couleurSphere,size=1)
aspect3d("iso")
# dénombrements des points dans chaque solide
Cyl=(x^2+y^2<1);nbCyl=sum(Cyl);nbCyl
Cone=x^2+y^2<z^2;nbCone=sum(Cone);nbCone
Sphere=x^2+y^2+z^2<1;nbSphere=sum(Sphere);nbSphere
# autre solution:
# r2=x^2+y^2
# R2=x^2+y^2+z^2
# nbCyl=sum(r2<1);nbCyl
# nbCone=sum(r2<z^2);nbCone
# nbSphere=sum(R2<1);nbSphere

# Le theorème d'Archimède:
nbCone+nbSphere
nbCyl
# erreur relative
(nbCone+nbSphere-nbCyl)/n
# un calcul de Pi par Monte-Carlo
# nbCyl/n ~ vol(Cyl)/vol(total)=Pi*R^2*h/4*Pi*R^2*h=Pi/4, donc
# Pi ~ nbCyl/n*4
Pi= nbCyl/n*4 ; Pi

# histogramme des résidus de vol(cone)+vol(sphere)=vol(cylindre)+resisu
K=1000;Res=numeric(K) # crée la variable Res; sinon "Erreur  objet 'Res' introuvable"
for (k in 1:K) # avec K=1000 cette boucle s'execute en une demi-minute environ
{x=runif(n,a,b);
y=runif(n,a,b);
z=runif(n,0,h);
Cyl=(x^2+y^2<1);nbCyl=sum(Cyl);
Cone=x^2+y^2<z^2;nbCone=sum(Cone);
Sphere=x^2+y^2+z^2<1;nbSphere=sum(Sphere)

# residus de Volume(Cone)+Volume(Sphere)=Volume(Cylindre)+Residu
Res[k]<- (nbCone+nbSphere-nbCyl)/n
} #fin de la boucle " for (k in 1:K) "
hist(Res)
boxplot(Res)
hist(Res,freq=FALSE) # cette option donne un histogramme d'aire totale 1
mu=mean(Res);mu
sigma=sd(Res);sigma
curve(dnorm(x,mu,sigma),add=TRUE,col="red") # ajoute (add=TRUE) le graphe de x->dnorm(x)
```