```
#############################################################################

#  Practical works: Design of numerical experiments


# Bertrand Iooss - Fabrice Gamboa - Barranquilla - 2015


# S O L U T I O N S


#############################################################################


rm(list=ls())

graphics.off()


#################################################################


#1 a)


library(DiceDesign)

help(package="DiceDesign")

?mindist


critere=0

for (i in 1:10000)

{

  x = matrix(runif(n=2*9),nrow=9,ncol=2)

  if ( mindist(x) > critere )

  {

    x1 = x
```

```
   critere = mindist(x)
  }
}

x11()
par(mfrow=c(1,2))
plot(x1,main="Random maximin")
plot(x,main="random")

######################
#1 b)

?factDesign

x2 = factDesign(2,3)
x11()
par(mfrow=c(2,2))
plot(x1,main="Random maximin")
plot(x,main="Random")
plot(x2$design,main="Factorial")
print(c("critere maximin(Random) :", mindist(x)))
print(c("critere maximin(Random maximin) :", mindist(x1)))
print(c("critere maximin(Factorial) :", mindist(x2$design)))


# Bad uni-dimensional sub-projections
# Number of points cannot be ordinary
```

```
# Too many points when dimension increases

################################################################

#2 a)

library(randtoolbox)

help(package="randtoolbox")

?sobol

x3 = randtoolbox::sobol(9,2)

plot(x3,main="Sobol")

print(c("critere maximin(Sobol) :", mindist(x3)))

########################

#2 b)

?discrepancyCriteria

print("Random")

print(discrepancyCriteria(x,type=c('M2','C2')))

print("Random maximin")

print(discrepancyCriteria(x1,type=c('M2','C2')))

print("Factorial")

print(discrepancyCriteria(x2$design,type=c('M2','C2')))

print("Sobol")

print(discrepancyCriteria(x3,type=c('M2','C2')))

######################
```

```
#2 c)


x4 = halton(200,8)

pairs(x4,pch =".",cex=3)


######################

#2 d)


x5 = sobol(200,8)

pairs(x5,pch =".",cex=3)


######################

#2 e)

?rss2d

rss2d(x5,lower=rep(0,8),upper=rep(1,8))


############################################################

#3 a)


lhs <- function(N,p){

 ran = matrix(runif(N*p),nrow=N,ncol=p)   # tirage de N x p valeurs selon loi U[0,1]

 x = matrix(0,nrow=N,ncol=p)  # construction de la matrice x


 for (i in 1:p) {

    idx = sample(1:N)       # vecteur de permutations des entiers {1,2,□,N}

    P = (idx-ran[,i]) / N    # vecteur de probabilités

    x[,i] <- qunif(P)  }
```

```
  return(x)}



x1 = lhs(20,2)


################################
#3 b)


# method i)


x0 = matrix(runif(n=2*20),nrow=20,ncol=2)


critere=0
for (i in 1:10000)
{
  x = lhs(20,2)
  if ( mindist(x) > critere )
  {
    x2 = x
    critere = mindist(x)
  }
}


# method ii)


xinit <- lhs(20,2)
x3 <- maximinSA_LHS(xinit)$design
```

```
plot(x3)


# Comparisons


print(c("criterion maximin(Random design) :", mindist(x0)))

print(c("criterion maximin(Standard LHS) :", mindist(x1)))

print(c("criterion maximin(Maximin LHS) :", critere))

print(c("criterion maximin(SA maximin LHS) :", mindist(x3)))


x11()

par(mfcol=c(2,2))

plot(x0,main="Random")

plot(x1,main="Random LHS")

plot(x2,main="Maximin LHS")

plot(x3,main="SA maximin LHS")
```