**Solutions TP**

#Creation du modele additif

somme2 <- function(x)

{

##HM: patch pour gerer le fait que sobolEff retourne X comme une matrice

if(is.matrix(x) & ncol(x)==2){

x <- as.data.frame(x)

names(x) <- c("X1","X2")

}

## fin patch

if(!is.data.frame(x))

stop("x doit etre un data.frame")

if (! (is.numeric(x$X1) & is.numeric(x$X2)) )

stop("X1 et X2 doivent etre numeriques")

res <- x$X1 + x$X2

return(res)

}

# Sensitivity Analysis

# Additive and multiplicative models

# -------------------------------------------------------------

# 1 Study of the design of experiments

# -------------------------------------------------------------

library(sensitivity)

library(numbers)

# -------------------------------------------------------------

# 1 Study of the design of experiments

# -------------------------------------------------------------

Ns <- 10

# Construction des deux echantillons A et B (loi uniforme)

A <- data.frame(matrix(runif(4 \* Ns), nrow = Ns))

B <- data.frame(matrix(runif(4 \* Ns), nrow = Ns))

# Methode sobolEff

# ----------------------------

res\_SobolEff <- sobolEff(model=NULL,A,B)

# Composantes de lâ€™objet fourni en retour de sobolEff

names(res\_SobolEff)

str(res\_SobolEff)

# Taille du plan dâ€™experience

dim(res\_SobolEff$X)

# ensemble des valeurs

# on gele suivant la premiere dimension

plot(res\_SobolEff$X[1:Ns,1:2],xlim=c(0,1),ylim=c(0,1))

points(res\_SobolEff$X[(Ns+1):(2\*Ns),1:2],col='blue',pch=3)

# ----------------------------

# Methode sobolroalhs

# ----------------------------

res\_Sobolroalhs <- sobolroalhs(model=NULL, factors=4, Ns,order=1)

# sensitivity::sobolroalhs si versions locales concurrentes

# Taille du plan dâ€™experience

dim(res\_Sobolroalhs$X)

## representations 2D des valeurs echantillonnees

# ensemble des valeurs

plot(res\_Sobolroalhs$X[1:Ns,1:2],xlim=c(0,1),ylim=c(0,1))

points(res\_Sobolroalhs$X[(Ns+1):(2\*Ns),1:2],col='blue',pch=3)

# -------------------------------------------------------------

# 2.1 Additive model

# -------------------------------------------------------------

# Taille des deux echantillons

n<-1000

# ---------------

# Cas 1

# ---------------

# creation des echantillons

A <- data.frame(matrix(runif(2\*n,-1,1),nrow=n))

B <- data.frame(matrix(runif(2\*n,-1,1),nrow=n))

par(mfrow=c(1,2))

# indices d'ordre 1

res\_sobolEff <- sobolEff(model=somme2,A,B,order=1,nboot=100)

print(res\_sobolEff)

plot(res\_sobolEff)

# indices totaux

res\_soboltotal <- sobolEff(model=somme2,A,B,nboot=100,order=0)

print(res\_soboltotal)

plot(res\_soboltotal)

# ---------------

# Cas 2

# ---------------

# creation des echantillons

A<-data.frame(cbind(runif(n,0,2),runif(n,-1,1)))

B<-data.frame(cbind(runif(n,0,2),runif(n,-1,1)))

par(mfrow=c(1,2))

# indices d'ordre 1

res\_sobolEff <- sobolEff(model=somme2,A,B,nboot=100)

print(res\_sobolEff)

plot(res\_sobolEff)

# indices totaux

res\_soboltotal <- sobolEff(model=somme2,A,B,nboot=100,order=0)

print(res\_soboltotal)

plot(res\_soboltotal)

# ---------------

# Cas 3

# ---------------

# creation des echantillons

A <- data.frame(cbind(runif(n,-1,1),runif(n,-2,2)))

B <- data.frame(cbind(runif(n,-1,1),runif(n,-2,2)))

par(mfrow=c(1,2))

# indices d'ordre 1

res\_sobolEff <- sobolEff(model=somme2,A,B,nboot=100)

print(res\_sobolEff)

plot(res\_sobolEff)

# indices totaux

res\_soboltotal <- sobolEff(model=somme2,A,B,nboot=100,order=0)

print(res\_soboltotal)

plot(res\_soboltotal)

# ---------------

# Cas 4

# ---------------

# creation des echantillons

A <- data.frame(cbind(runif(n,-1,1),rnorm(n,0,2/sqrt(3))))

B <- data.frame(cbind(runif(n,-1,1),rnorm(n,0,2/sqrt(3))))

par(mfrow=c(1,2))

# indices d'ordre 1

res\_sobolEff <- sobolEff(model=somme2,A,B,nboot=100)

print(res\_sobolEff)

plot(res\_sobolEff)

# indices totaux

res\_soboltotal <- sobolEff(model=somme2,A,B,nboot=100,order=0)

print(res\_soboltotal)

plot(res\_soboltotal)

# -------------------------------------------------------------

# 2.2 Multiplicative model

# -------------------------------------------------------------

# Taille des deux echantillons

n<-1000

# ---------------

# Cas 1

# ---------------

# creation des echantillons

A <- data.frame(matrix(runif(2\*n,-1,1),nrow=n))

B <- data.frame(matrix(runif(2\*n,-1,1),nrow=n))

par(mfrow=c(1,2))

# indices d'ordre 1

res\_sobolEff <- sobolEff(model=produit2,A,B,nboot=100)

print(res\_sobolEff)

plot(res\_sobolEff)

# indices totaux

res\_soboltotal <- sobolEff(model=produit2,A,B,nboot=100,order=0)

print(res\_soboltotal)

plot(res\_soboltotal)

# ---------------

# Cas 2

# ---------------

# creation des echantillons

A <- data.frame(cbind(runif(n,0,2),runif(n,-1,1)))

B <- data.frame(cbind(runif(n,0,2),runif(n,-1,1)))

par(mfrow=c(1,2))

# indices d'ordre 1

res\_sobolEff <- sobolEff(model=produit2,A,B,nboot=100)

print(res\_sobolEff)

plot(res\_sobolEff)

# indices totaux

res\_soboltotal <- sobolEff(model=produit2,A,B,nboot=100,order=0)

print(res\_soboltotal)

plot(res\_soboltotal)

# ---------------

# Cas 3

# ---------------

# creation des echantillons

A <- data.frame(cbind(runif(n,0,2),rnorm(n,0,1/sqrt(3))))

B <- data.frame(cbind(runif(n,0,2),rnorm(n,0,1/sqrt(3))))

par(mfrow=c(1,2))

# indices d'ordre 1

res\_sobolEff <- sobolEff(model=produit2,A,B,nboot=100)

print(res\_sobolEff)

plot(res\_sobolEff)

# indices totaux

res\_soboltotal <- sobolEff(model=produit2,A,B,nboot=100,order=0)

print(res\_soboltotal)

plot(res\_soboltotal)

# -------------------------------------------------------------

# 3 g-Sobol function

# -------------------------------------------------------------

# Taille des deux echantillons

n <- 10000

# ---------------

# Cas 1 : a=c(1,99)

# ---------------

a <- c(1,99)

p <- length(a)

# valeurs thÃ©oriques

St <- sobol.gfunc(a)

St2 <- sobol.gfunc(a,order=2)

# fonction sobolroalhs

#ordre 1

xsobolroalhs <- sobolroalhs(model = g.func, factors = 2, N = 1000, order = 1, nboot=100,a=a)

print(xsobolroalhs)

plot(xsobolroalhs)

# ---------------

# Cas 2 : a=c(1,2,3,...,10)

# ---------------

a <- seq(1,10)

p <- length(a)

# valeurs thÃ©oriques

St <- sobol.gfunc(a)

St2 <- sobol.gfunc(a,order=2)

# fonction sobolroalhs

#ordre 1

xsobolroalhs <- sobolroalhs(model = g.func, factors = 10, N = 10000, order = 1, nboot=100,a=a)

print(xsobolroalhs)

plot(xsobolroalhs)

#ordre 2

xsobolroalhs <- sobolroalhs(model = g.func, factors = 10, N = 10000, order = 2, nboot=100,a=a)

print(xsobolroalhs)

plot(xsobolroalhs)

# ---------------

# Cas 3 : a=c(1,4,9,...,100)

# ---------------

a <- seq(1,10)^2

p <- length(a)

# valeurs thÃ©oriques

St <- sobol.gfunc(a)

St2 <- sobol.gfunc(a,order=2)

# fonction sobolroalhs

#ordre 1

xsobolroalhs=sobolroalhs(model = g.func, factors = 10, N = 1000, order = 1, nboot=100,a=a)

print(xsobolroalhs)

plot(xsobolroalhs)

#ordre 2

xsobolroalhs=sobolroalhs(model = g.func, factors = 10, N = 1000, order = 2, nboot=100,a=a)

print(xsobolroalhs)

plot(xsobolroalhs)

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

#

# Function: g.func.R

# Purpose: g-fonction

# f(X)=produit(fi(Xi)) i=1,...,p

#

# fi(Xi)=(|4Xi-2| + ai)/(1+ai) , ai > 0

#

# Xi: variables aleatoires independantes uniformement distribuees su [0,1]

#

# Author: L. Viry

# Modified:

# Date: 25/02/2013

#

# Entrees::

# X : echantillon

# a: tableau des parametres de la fonction G

#

# Sorties:

# y : valeurs de la fonction g sur l'echantillon X

#

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

g.func <- function (X,a=NULL)

{

if(is.null(a)) a <- c(1,99)

# a <- c(1,9)

# a <- c(0,0,0,1,1,1,9,9)

# a <- c(0, 1, 4.5, 9, 99, 99, 99, 99)

# a <- c(0,0,0,1,1,1,1,9,9,9,9,9)

if (ncol(X) != length(a))

stop("Le nombre de parametres de X est incompatible avec la taille de a")

p <- ncol(X)

y <- 1

for (j in 1:p) {

y <- y \* (abs(4 \* X[, j] - 2) + a[j])/(1 + a[j])

}

# Y <-as.numeric(y)

# return(Y)

return(y)

}

a <- seq(1,10)

p <- length(a)

# valeurs thÃ©oriques

#St <- sobol.gfunc(a)

#St2 <- sobol.gfunc(a,order=2)

# Calcul theorique des indices de sobol d'ordre 1 de la g-fonction

#

sobol.gfunc <- function(a, order=1)

{

# Nombre de parametres

p <-length(a)

sigma <- rep(0,p)

sig <-1

for (i in 1:p)

{

sigma[i] <- 1/(3\*(1+a[i])^2)

sig <- sig\*(sigma[i] + 1)

}

sig <- sig - 1

if(order==2){

sobol\_1 <- rep(0,p\*(p-1)/2)

i <- 1

j <- 1

I <- matrix(0,ncol=2,nrow=p\*(p-1)/2)

for (l in 1:(p\*(p-1)/2)) {

i <- i+(j==p)

j <- ((j==p)\*i + (j<p)\*j)+1

I[l,]=c(i,j)

sobol\_1[l] <- (sigma[i]+sigma[j]+sigma[i]\*sigma[j])/sig

}

}

else{

sobol\_1 <- rep(0,p)

for (i in 1:p){

sobol\_1[i] <- sigma[i]/sig

}

}

# xx <-list(a=a,sig=sig,sobol\_1=sobol\_1)

return(sobol\_1)

}