Appendix 1: Simulation cohort: syntaxe of the model

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

# Impact of Gestational Diabetes Mellitus Screening Strategies on Perinatal Outcomes: a Simulation Study

# Coded by: Isaias Prestes; reviewed by Maicon Falavigna

# Performed on R version 2.11.1

# Packages required: hdrcde, boa

# Date: 01/06/12

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

# Settings

n <- 1000000 ;

alfa <- 0.05

# The Beta distribution with parameters shape1 = a and shape2 = b has density

#

# G(a+b)/(G(a)G(b))x^(a-1)(1-x)^(b-1)

#

# for a > 0, b > 0 and 0 = x = 1 where the boundary values at x=0 or x=1 are defined as by continuity (as limits).

# The mean is a/(a+b) and the variance is ab/((a+b)^2 (a+b+1)).

# Given a and b, return a array with mean, variance, deviation, lower endpoint and upper endpoint of the 100(1-alfa)% confidence interval for Beta distribution.

beta.parametros <- function(a, b, alfa, digits = 5) {

media <- a / (a+b)

variancia <- (a \* b)/ ( (a+b)^2 \* (a+b+1) )

liminf <- qbeta(alfa/2, a, b)

limsup <- qbeta(1 - alfa/2, a, b)

return( round( c("Mean"=media,"Var"=variancia,"SD"=variancia^.5, "LL"=liminf, "UL"=limsup) , digits=5) )

}

# Example :

beta.parametros(3, 6, alfa)

# Set RNG seed

set.seed(37589411)

# Values for PWHO

# PWHO <- seq(0.05,0.15,0.01)

PWHO <- 0.05

# Simulated parameters values

# LGA Birth (LGA)

LGA\_Iwho\_negative <- rbeta(n, 1.5\*900, 1.5\*9100 ) ; c <- 1.5; beta.parametros(c\*900, 9100\*c, alfa)

LGA\_logRRwho <- rnorm(n, 0.4256, 0.0494) ; # Distribution for log(RR)

LGA\_Ptreatment <- rbeta(n, 40.5, 4.5 ) ; c <- 4.5; beta.parametros(9\*c, 1\*c, alfa) # approximated solution

LGA\_logRPiadpsg <- rnorm(n, 0.4055, 0.0730) ; # Distribution for log(RP)

LGA\_logRRtreatment <- rnorm(n, -0.6095, 0.1245) ; # Distribution for log(RR)

LGA\_logRRiadpsg <- rnorm(n, 0.5503, 0.1558) ; # Distribution for log(RR)

# Preeclampsia (PE)

PE\_Iwho\_negative <- rbeta(n, 22.5, 477.5 ) ; c <- 0.5; beta.parametros(c\*45, c\*955, alfa)

PE\_logRRwho <- rnorm(n, 0.5260, 0.1288) ; # Distribution for log(RR)

PE\_Ptreatment <- rbeta(n, 40.5, 4.5 ) ; c <- 4.5; beta.parametros(9\*c, 1\*c, alfa) # approximated solution

PE\_logRPiadpsg <- rnorm(n, 0.4055, 0.0730) ; # Distribution for log(RP)

PE\_logRRtreatment <- rnorm(n, -0.4903, 0.1413) ; # Distribution for log(RR)

PE\_logRRiadpsg <- rnorm(n, 0.5374, 0.1108) ; # Distribution for log(RR)

# Caesarean section (CS)

CS\_Iwho\_negative <- rbeta(n, 11.7475, 51.7525 ) ; beta.parametros(11.7475, 51.7525, alfa)

CS\_logRRwho <- rnorm(n, 0.3144, 0.0508) ; # Distribution for log(RR)

CS\_Ptreatment <- rbeta(n, 40.5, 4.5 ) ; c <- 4.5; beta.parametros(9\*c, 1\*c, alfa) # approximated solution

CS\_logRPiadpsg <- rnorm(n, 0.4055, 0.0730) ; # Distribution for log(RP)

CS\_logRRtreatment <- rnorm(n, -0.1100, 0.0797) ; # Distribution for log(RR)

CS\_logRRiadpsg <- rnorm(n, 0.2086, 0.1035) ; # Distribution for log(RR)

# Set simulated block

set.idiaf <- function( PWHO, Iwho\_negative, logRRwho, Ptreatment, logRRtreatment, logRPiadpsg, logRRiadpsg) {

IDIAF <- ( ( PWHO \* Iwho\_negative \* exp(logRRwho) ) ) + ( ( 1 - PWHO ) \* Iwho\_negative ) / ( (1 - ( PWHO \* exp(logRPiadpsg) ) ) + ( exp(logRRiadpsg) \* PWHO \* exp(logRPiadpsg) ) )

return(IDIAF)

}

# Functions for model simulation

simula.nao.rastrear <- function ( n , PWHO, Iwho\_negative, logRRwho, Ptreatment, logRRtreatment, logRPiadpsg, logRRiadpsg) {

Iwho\_positive <- Iwho\_negative \* exp(logRRwho) # Iwho\_positive = Iwho\_negative \* RRwho

ans <- (

( PWHO \* Iwho\_positive ) + ( ( 1 - PWHO ) \* Iwho\_negative )

)

return(ans)

}

simula.who <- function ( n , PWHO, Iwho\_negative, logRRwho, Ptreatment, logRRtreatment, logRPiadpsg, logRRiadpsg) {

Iwho\_positive <- Iwho\_negative \* exp(logRRwho)

ans <- (

( ( 1 - PWHO ) \* Iwho\_negative ) +

( PWHO \* ( 1 - Ptreatment ) \* Iwho\_positive ) +

( PWHO \* Ptreatment \* exp(logRRtreatment) \* Iwho\_positive )

)

return(ans)

}

simula.iadpsg <- function ( n , PWHO, Iwho\_negative, logRRwho, Ptreatment, logRRtreatment, logRPiadpsg, logRRiadpsg) {

Piadpsg <- PWHO \* exp(logRPiadpsg) # Piadpsg = Pwho \* PRiadpsg

Iiadpsg\_negative <- ( ( PWHO \* Iwho\_negative \* exp(logRRwho) ) +

( ( 1 - PWHO) \* Iwho\_negative ) ) / ( ( ( 1 - (PWHO \* exp(logRPiadpsg) ) ) ) +

( PWHO \* exp(logRPiadpsg) \* exp(logRRiadpsg) ) )

Iiadpsg\_positive <- Iiadpsg\_negative \* exp(logRRiadpsg)

ans <- (

( ( 1 - Piadpsg ) \* Iiadpsg\_negative ) +

( Piadpsg \* Iiadpsg\_positive \* Ptreatment \* exp(logRRtreatment) ) +

( Piadpsg \* Iiadpsg\_positive \* ( 1 - Ptreatment ) )

)

return(ans)

}

# Looping PWHO values

datasummary <- array(,10)

datasummary.diff <- array(,10)

datasummary.table.diff <- array(,3)

names(datasummary) <- c("PWHO", "Strategy", "Min.", "1st Qu.", "Median", "Mean", "3rd Qu.", "Max.")

names(datasummary.diff) <- c("PWHO", "Strategy", "Min.", "1st Qu.", "Median", "Mean", "3rd Qu.", "Max.")

for ( i in PWHO ) {

arrayPWHO <- rep(i, n)

# Compute values for each screening strategies

resultados.IADPSG.LGA <- simula.iadpsg( n , arrayPWHO, LGA\_Iwho\_negative, LGA\_logRRwho, LGA\_Ptreatment, LGA\_logRRtreatment, LGA\_logRPiadpsg, LGA\_logRRiadpsg)

resultados.IADPSG.PE <- simula.iadpsg( n , arrayPWHO, PE\_Iwho\_negative, PE\_logRRwho, PE\_Ptreatment, PE\_logRRtreatment, PE\_logRPiadpsg, PE\_logRRiadpsg)

resultados.IADPSG.CS <- simula.iadpsg( n , arrayPWHO, CS\_Iwho\_negative, CS\_logRRwho, CS\_Ptreatment, CS\_logRRtreatment, CS\_logRPiadpsg, CS\_logRRiadpsg)

resultados.NR.LGA <- simula.nao.rastrear( n , arrayPWHO, LGA\_Iwho\_negative, LGA\_logRRwho, LGA\_Ptreatment, LGA\_logRRtreatment, LGA\_logRPiadpsg, LGA\_logRRiadpsg)

resultados.NR.PE <- simula.nao.rastrear( n , arrayPWHO, PE\_Iwho\_negative, PE\_logRRwho, PE\_Ptreatment, PE\_logRRtreatment, PE\_logRPiadpsg, PE\_logRRiadpsg)

resultados.NR.CS <- simula.nao.rastrear( n , arrayPWHO, CS\_Iwho\_negative, CS\_logRRwho, CS\_Ptreatment, CS\_logRRtreatment, CS\_logRPiadpsg, CS\_logRRiadpsg)

resultados.WHO.LGA <- simula.who( n , arrayPWHO, LGA\_Iwho\_negative, LGA\_logRRwho, LGA\_Ptreatment, LGA\_logRRtreatment, LGA\_logRPiadpsg, LGA\_logRRiadpsg)

resultados.WHO.PE <- simula.who( n , arrayPWHO, PE\_Iwho\_negative, PE\_logRRwho, PE\_Ptreatment, PE\_logRRtreatment, PE\_logRPiadpsg, PE\_logRRiadpsg)

resultados.WHO.CS <- simula.who( n , arrayPWHO, CS\_Iwho\_negative, CS\_logRRwho, CS\_Ptreatment, CS\_logRRtreatment, CS\_logRPiadpsg, CS\_logRRiadpsg)

resultado.final <- data.frame(

"PWHO" = arrayPWHO,

"NR.LGA" = resultados.NR.LGA,

"NR.PE" = resultados.NR.PE,

"NR.CS" = resultados.NR.CS,

"WHO.LGA" = resultados.WHO.LGA,

"WHO.PE" = resultados.WHO.PE,

"WHO.CS" = resultados.WHO.CS,

"IADPSG.LGA" = resultados.IADPSG.LGA,

"IADPSG.PE" = resultados.IADPSG.PE,

"IADPSG.CS" = resultados.IADPSG.CS

)

write.csv2( resultado.final , paste("table\_data\_PWHO\_", i,".csv", sep="") )

# Compute differences

Delta.NR.WHO.LGA <- resultados.NR.LGA - resultados.WHO.LGA

Delta.NR.IADPSG.LGA <- resultados.NR.LGA - resultados.IADPSG.LGA

Delta.WHO.IADPSG.LGA <- resultados.WHO.LGA - resultados.IADPSG.LGA

Delta.NR.WHO.PE <- resultados.NR.PE - resultados.WHO.PE

Delta.NR.IADPSG.PE <- resultados.NR.PE - resultados.IADPSG.PE

Delta.WHO.IADPSG.PE <- resultados.WHO.PE - resultados.IADPSG.PE

Delta.NR.WHO.CS <- resultados.NR.CS - resultados.WHO.CS

Delta.NR.IADPSG.CS <- resultados.NR.CS - resultados.IADPSG.CS

Delta.WHO.IADPSG.CS <- resultados.WHO.CS - resultados.IADPSG.CS

# Compute p-values empirical # Pr { standart strategy > alternative strategy }

comptype <- c("NR.WHO.LGA","NR.IADPSG.LGA","WHO.IADPSG.LGA","NR.WHO.PE","NR.IADPSG.PE","WHO.IADPSG.PE","NR.WHO.CS","NR.IADPSG.CS","WHO.IADPSG.CS")

resultados.pvalue <- array(,0)

for (i in comptype) {

ans <- c(i, eval(parse(text=paste("sum( ifelse(Delta.",i ," < 0 , 1, 0) ) / n",sep=""))) )

resultados.pvalue <- rbind(resultados.pvalue, ans)

print( ans )

}

write.csv2( resultados.pvalue , paste("Summary\_pvalues.csv", sep=""))

# NNS for screening strategies

NNS.Delta.NR.WHO.LGA <- 1 / Delta.NR.WHO.LGA

NNS.Delta.NR.IADPSG.LGA <- 1 / Delta.NR.IADPSG.LGA

NNS.Delta.WHO.IADPSG.LGA <- 1 / Delta.WHO.IADPSG.LGA

NNS.Delta.NR.WHO.PE <- 1 / Delta.NR.WHO.PE

NNS.Delta.NR.IADPSG.PE <- 1 / Delta.NR.IADPSG.PE

NNS.Delta.WHO.IADPSG.PE <- 1 / Delta.WHO.IADPSG.PE

NNS.Delta.NR.WHO.CS <- 1 / Delta.NR.WHO.CS

NNS.Delta.NR.IADPSG.CS <- 1 / Delta.NR.IADPSG.CS

NNS.Delta.WHO.IADPSG.CS <- 1 / Delta.WHO.IADPSG.CS

# Compute binary variables for proportions

Bin.Delta.NR.WHO.LGA <- ifelse(Delta.NR.WHO.LGA > 0, 1, 0)

Bin.Delta.NR.IADPSG.LGA <- ifelse(Delta.NR.IADPSG.LGA > 0, 1, 0)

Bin.Delta.WHO.IADPSG.LGA <- ifelse(Delta.WHO.IADPSG.LGA > 0, 1, 0)

Bin.Delta.NR.WHO.PE <- ifelse(Delta.NR.WHO.PE > 0, 1, 0)

Bin.Delta.NR.IADPSG.PE <- ifelse(Delta.NR.IADPSG.PE > 0, 1, 0)

Bin.Delta.WHO.IADPSG.PE <- ifelse(Delta.WHO.IADPSG.PE > 0, 1, 0)

Bin.Delta.NR.WHO.CS <- ifelse(Delta.NR.WHO.CS > 0, 1, 0)

Bin.Delta.NR.IADPSG.CS <- ifelse(Delta.NR.IADPSG.CS > 0, 1, 0)

Bin.Delta.WHO.IADPSG.CS <- ifelse(Delta.WHO.IADPSG.CS > 0, 1, 0)

comptype <- c("NR.WHO.LGA","NR.IADPSG.LGA","WHO.IADPSG.LGA","NR.WHO.PE","NR.IADPSG.PE","WHO.IADPSG.PE","NR.WHO.CS","NR.IADPSG.CS","WHO.IADPSG.CS")

for (i in comptype) {

ans <- eval(parse(text=paste("table(Bin.Delta.",i,")",sep="")))

print(ans)

if ( length(ans) < 2 ) ans <- c("0" = 0, ans[1] )

datasummary.table.diff <- rbind( datasummary.table.diff, as.array(c(i,ans)) )

}

# Save data

datasummary.table.diff <- datasummary.table.diff[ !is.na(datasummary.table.diff[,1]) ,]

datasummary.table.diff <- data.frame(

"Comparação" = as.character(datasummary.table.diff[,1]),

"Freq 0" = as.numeric(as.character(datasummary.table.diff[,2])),

"Freq 1" = as.numeric(as.character(datasummary.table.diff[,3])),

"Fr 0" = (100 \* (as.numeric(as.character(datasummary.table.diff[,2])) / n )),

"Fr 1" = (100 \* (as.numeric(as.character(datasummary.table.diff[,3])) / n ))

)

write.csv2( datasummary.table.diff , "Estatísticas\_Binarias\_Prop\_Comp.csv", row.names=FALSE )

resultado.final.diff <- data.frame(

"PWHO" = arrayPWHO,

"NR vs WHO - LGA" = Delta.NR.WHO.LGA,

"NR vs IADPSG - LGA" = Delta.NR.IADPSG.LGA,

"WHO vs IADPSG - LGA" = Delta.WHO.IADPSG.LGA,

"NR vs WHO - PE" = Delta.NR.WHO.PE,

"NR vs IADPSG - PE" = Delta.NR.IADPSG.PE,

"WHO vs IADPSG - PE" = Delta.WHO.IADPSG.PE,

"NR vs WHO - CS" = Delta.NR.WHO.CS,

"NR vs IADPSG - CS" = Delta.NR.IADPSG.CS,

"WHO vs IADPSG - CS" = Delta.WHO.IADPSG.CS

)

write.csv2( resultado.final.diff , paste("table\_data\_Differences\_PWHO\_", i,".csv", sep=""), row.names=FALSE )

# Compute summary tables

datasummary <- rbind( datasummary, c( i, "NR.LGA", c(summary(resultado.final$"NR.LGA",)) ,

quantile(resultado.final$"NR.LGA", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "NR.PE", c(summary(resultado.final$"NR.PE")) ,

quantile(resultado.final$"NR.LGA", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "NR.CS", c(summary(resultado.final$"NR.CS")) ,

quantile(resultado.final$"NR.LGA", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "WHO.LGA", c(summary(resultado.final$"WHO.LGA")) ,

quantile(resultado.final$"WHO.LGA", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "WHO.PE", c(summary(resultado.final$"WHO.PE")) ,

quantile(resultado.final$"WHO.PE", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "WHO.CS", c(summary(resultado.final$"WHO.CS")) ,

quantile(resultado.final$"WHO.CS", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "IADPSG.LGA", c(summary(resultado.final$"IADPSG.LGA")) ,

quantile(resultado.final$"IADPSG.LGA", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "IADPSG.PE", c(summary(resultado.final$"IADPSG.PE")) ,

quantile(resultado.final$"IADPSG.PE", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary <- rbind( datasummary, c( i, "IADPSG.CS", c(summary(resultado.final$"IADPSG.CS")) ,

quantile(resultado.final$"IADPSG.CS", probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

# Compute summary tables for differences

datasummary.diff <- rbind( datasummary.diff, c( i, "NR vs WHO - LGA", c(summary(Delta.NR.WHO.LGA)) ,

quantile(Delta.NR.WHO.LGA, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary.diff <- rbind( datasummary.diff, c( i, "NR vs IADPSG - LGA", c(summary(Delta.NR.IADPSG.LGA)) ,

quantile(Delta.NR.IADPSG.LGA, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

# Compute summary : screening strategies WHO vs IADPSG for LGA

datasummary.diff <- rbind( datasummary.diff, c( i, "WHO vs IADPSG - LGA", c(summary(Delta.WHO.IADPSG.LGA)) ,

quantile(Delta.WHO.IADPSG.LGA, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary.diff <- rbind( datasummary.diff, c( i, "NR vs WHO - PE", c(summary(Delta.NR.WHO.PE)) ,

quantile(Delta.NR.WHO.PE, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary.diff <- rbind( datasummary.diff, c( i, "NR vs IADPSG - PE", c(summary(Delta.NR.IADPSG.PE)) ,

quantile(Delta.NR.IADPSG.PE, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

# Compute summary : screening strategies WHO vs IADPSG for PE

datasummary.diff <- rbind( datasummary.diff, c( i, "WHO vs IADPSG - PE", c(summary(Delta.WHO.IADPSG.PE)) ,

quantile(Delta.WHO.IADPSG.PE, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary.diff <- rbind( datasummary.diff, c( i, "NR vs WHO - CS", c(summary(Delta.NR.WHO.CS)) ,

quantile(Delta.NR.WHO.CS, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

datasummary.diff <- rbind( datasummary.diff, c( i, "NR vs IADPSG - CS", c(summary(Delta.NR.IADPSG.CS)) ,

quantile(Delta.NR.IADPSG.CS, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

# Compute summary : screening strategies WHO vs IADPSG for CS

datasummary.diff <- rbind( datasummary.diff, c( i, "WHO vs IADPSG - CS", c(summary(Delta.WHO.IADPSG.CS)) ,

quantile(Delta.WHO.IADPSG.CS, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

}

# Save summary data

datasummary <- datasummary[ !is.na(datasummary[,1]) ,]

datasummary.diff <- datasummary.diff[ !is.na(datasummary.diff[,1]) ,]

datasummary <- data.frame(datasummary)

names(datasummary) <- c(names(datasummary)[1:(length(names(datasummary))-2)],

paste("Percentil ",round((alfa/2)\*100, digits=1),sep=""), paste("Percentil ",round((1-alfa/2)\*100, digits=1),sep=""))

datasummary.diff <- data.frame(datasummary.diff)

names(datasummary.diff) <- c(names(datasummary.diff)[1:(length(names(datasummary))-2)],

paste("Percentil ",round((alfa/2)\*100, digits=1),sep=""), paste("Percentil ",round((1-alfa/2)\*100, digits=1),sep=""))

write.csv2( datasummary , paste("Summary\_statistics.csv", sep=""), row.names=FALSE )

write.csv2( datasummary.diff , paste("Summary\_statistics\_for\_differences.csv", sep=""), row.names=FALSE )

# Summary - distribution - Compute the Highest Posterior Density Interval (HPD)

require(hdrcde)

require(boa)

comptype <- c("NR.LGA","NR.PE","NR.CS","WHO.LGA","WHO.PE","WHO.CS","IADPSG.LGA","IADPSG.PE","IADPSG.CS")

for (i in comptype) {

print(paste("Computing HPD Interval for ",i,sep=""))

graphname <- paste("grafico\_",i,".png", sep="")

png(file.path(paste(getwd(),"//saida", sep=""),graphname))

ans <- hdr.den(resultado.final[,i])

dev.off()

# Using boa package

ans <- boa.hpd(resultado.final[,i], alpha = 0.05)

write.csv2( ans , paste("HPD ",i,".csv", sep=""))

print( ans )

}

comptype <- c("NR.WHO.LGA","NR.IADPSG.LGA","WHO.IADPSG.LGA","NR.WHO.PE","NR.IADPSG.PE","WHO.IADPSG.PE","NR.WHO.CS","NR.IADPSG.CS","WHO.IADPSG.CS")

for (i in comptype) {

print(paste("Computing HPD Interval for Difference ",i,sep=""))

graphname <- paste("grafico\_IDiff\_",i,".png", sep="")

png(file.path(paste(getwd(),"//saida", sep=""),graphname))

ans <- eval(parse(text=paste("hdr.den(Delta.",i,")",sep="")))

dev.off()

# Using boa package

ans <- eval(parse(text=paste("boa.hpd(Delta.",i,", alpha = 0.05)",sep="")))

write.csv2( ans , paste("HPD Diff ",i,".csv", sep=""))

print( ans )

}

comptype <- c("NR.WHO.LGA","NR.IADPSG.LGA","WHO.IADPSG.LGA","NR.WHO.PE","NR.IADPSG.PE","WHO.IADPSG.PE","NR.WHO.CS","NR.IADPSG.CS","WHO.IADPSG.CS")

for (i in comptype) {

print(paste("Computing HPD Interval for NNS",i,sep=""))

graphname <- paste("grafico\_NNS\_Diff\_",i,".png", sep="")

png(file.path(paste(getwd(),"//saida", sep=""),graphname))

ans <- eval(parse(text=paste("plot(density(NNS.Delta.",i,"))",sep="")))

dev.off()

# Using boa package

ans <- eval(parse(text=paste("boa.hpd(NNS.Delta.",i,", alpha = 0.05)",sep="")))

write.csv2( ans , paste("HPD Diff ",i,".csv", sep=""))

print( ans )

}

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

# Sensitivity analysis - HAPO #

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

# Screening strategies settings

LGA\_Ptreatment <- rbeta(n, 40.5, 4.5 ) ; c <- 4.5; beta.parametros(9\*c, 1\*c, alfa) # melhor solução encontrada

LGA\_logRRtreatment <- rnorm(n, -0.6095, 0.1245) ; # A distribuição de log(RR)

PE\_Ptreatment <- rbeta(n, 40.5, 4.5 ) ; c <- 4.5; beta.parametros(9\*c, 1\*c, alfa) # melhor solução encontrada

PE\_logRRtreatment <- rnorm(n, -0.4903, 0.1413) ; # A distribuição de log(RR)

CS\_Ptreatment <- rbeta(n, 40.5, 4.5 ) ; c <- 4.5; beta.parametros(9\*c, 1\*c, alfa) # melhor solução encontrada

CS\_logRRtreatment <- rnorm(n, -0.1100, 0.0797) ; # A distribuição de log(RR)

# Common parameters settings

Pwho\_HAPO <- 0.114

Piadpsg\_HAPO <- 0.161

LGA\_Iwhopos\_HAPO <- 0.137

PE\_Iwhopos\_HAPO <- 0.076

CS\_Iwhopos\_HAPO <- 0.244

LGA\_Iiadpsgpos\_HAPO <- 0.162

PE\_Iiadpsgpos\_HAPO <- 0.091

CS\_Iiadpsgpos\_HAPO <- 0.244

# Compute incidence reduction - WHO

RI\_LGA\_WHO <- Pwho\_HAPO \* LGA\_Iwhopos\_HAPO \* LGA\_Ptreatment \* (1-exp(LGA\_logRRtreatment))

NNS\_LGA\_WHO <- 1 / RI\_LGA\_WHO

RI\_PE\_WHO <- Pwho\_HAPO \* PE\_Iwhopos\_HAPO \* PE\_Ptreatment \* (1-exp(PE\_logRRtreatment))

NNS\_PE\_WHO <- 1 / RI\_PE\_WHO

RI\_CS\_WHO <- Pwho\_HAPO \* CS\_Iwhopos\_HAPO \* CS\_Ptreatment \* (1-exp(CS\_logRRtreatment))

NNS\_CS\_WHO <- 1 / RI\_CS\_WHO

# Compute incidence reduction - IADPSG

RI\_LGA\_IADPSG <- Piadpsg\_HAPO \* LGA\_Iiadpsgpos\_HAPO \* LGA\_Ptreatment \* (1-exp(LGA\_logRRtreatment))

NNS\_LGA\_IADPSG <- 1 / RI\_LGA\_IADPSG

RI\_PE\_IADPSG <- Piadpsg\_HAPO \* PE\_Iiadpsgpos\_HAPO \* PE\_Ptreatment \* (1-exp(PE\_logRRtreatment))

NNS\_PE\_IADPSG <- 1 / RI\_PE\_IADPSG

RI\_CS\_IADPSG <- Piadpsg\_HAPO \* CS\_Iiadpsgpos\_HAPO \* CS\_Ptreatment \* (1-exp(CS\_logRRtreatment))

NNS\_CS\_IADPSG <- 1 / RI\_CS\_IADPSG

# Save data

resultado.final.sa <- data.frame(

"RI\_LGA\_WHO" = RI\_LGA\_WHO,

"NNS\_LGA\_WHO" = NNS\_LGA\_WHO,

"RI\_PE\_WHO" = RI\_PE\_WHO,

"NNS\_PE\_WHO" = NNS\_PE\_WHO,

"RI\_CS\_WHO" = RI\_CS\_WHO,

"NNS\_CS\_WHO" = NNS\_CS\_WHO,

"RI\_LGA\_IADPSG" = RI\_LGA\_IADPSG,

"NNS\_LGA\_IADPSG" = NNS\_LGA\_IADPSG,

"RI\_PE\_IADPSG" = RI\_PE\_IADPSG,

"NNS\_PE\_IADPSG" = NNS\_PE\_IADPSG,

"RI\_CS\_IADPSG" = RI\_CS\_IADPSG,

"NNS\_CS\_IADPSG" = NNS\_CS\_IADPSG

)

write.csv2( resultado.final.sa , "table\_data\_sensitivity analysis.csv" )

# Compute summary tables

tabela.resumo.sa <- array(,9)

vet.nomeselementos <- c("RI\_LGA\_WHO","NNS\_LGA\_WHO","RI\_PE\_WHO","NNS\_PE\_WHO","RI\_CS\_WHO","NNS\_CS\_WHO",

"RI\_LGA\_IADPSG", "NNS\_LGA\_IADPSG", "RI\_PE\_IADPSG", "NNS\_PE\_IADPSG", "RI\_CS\_IADPSG", "NNS\_CS\_IADPSG")

for (i in vet.nomeselementos) {

ans <- eval(parse(text=i))

tabela.resumo.sa <- rbind( tabela.resumo.sa, c( i, summary(ans), quantile(ans, probs = c(alfa/2, 1 - alfa/2), names = FALSE ) ) )

}

tabela.resumo.sa <- tabela.resumo.sa[ !is.na(tabela.resumo.sa[,1]) ,]

tabela.resumo.sa <- data.frame(tabela.resumo.sa)

names(tabela.resumo.sa) <- c("Estatística/Estratégia", names(tabela.resumo.sa)[2:(length(names(tabela.resumo.sa))-2)],

paste("Percentil ",round((alfa/2)\*100, digits=1),sep=""), paste("Percentil ",round((1-alfa/2)\*100, digits=1),sep=""))

write.csv2( tabela.resumo.sa , paste("Summary\_AS\_HAPO.csv", sep=""), row.names=FALSE )

# Summary - Highest Posterior Density Interval (HPD)

vet.nomeselementos1 <- c("RI\_LGA\_WHO","RI\_PE\_WHO","RI\_CS\_WHO","RI\_LGA\_IADPSG", "RI\_PE\_IADPSG", "RI\_CS\_IADPSG")

vet.nomeselementos2 <- c("NNS\_LGA\_WHO", "NNS\_LGA\_IADPSG","NNS\_PE\_WHO", "NNS\_PE\_IADPSG", "NNS\_CS\_WHO", "NNS\_CS\_IADPSG")

vet.nomeselementos <- c(vet.nomeselementos1, vet.nomeselementos2)

for (i in vet.nomeselementos) {

print(paste("HPD Interval for ",i," | Sensitivity analysis HAPO",sep=""))

graphname <- paste("grafico\_AS\_HAPO\_",i,".png", sep="")

png(file.path(paste(getwd(),"//saida", sep=""),graphname))

ans <- eval(parse(text=paste("plot(density(",i,"))",sep="")))

dev.off()

ans <- eval(parse(text=paste("boa.hpd(",i,", alpha = 0.05)",sep="")))

write.csv2( ans , paste("HPD Sensitivity analysis HAPO ",i,".csv", sep=""))

print( ans )

}

dev.off()

# End of the syntaxe

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