#### ROTINA R PARA CÁLCULO DAS FRONTEIRAS DE EFICIÊNCIA - CUSTOS OPERACIONAIS EFICIENTES

#### ROTINA BÁSICA PARA REALIZAÇÃO DAS ESTIMAÇÕES POR ANÁLISE ENVOLTÓRIA DE DADOS

# Limpar os dados do R

rm(list=ls())

## O script utilizará os seguintes pacotes

if (!require(openxlsx)) {

install.packages('openxlsx')

library(openxlsx)

}

if (!require(Benchmarking)) {

install.packages('Benchmarking')

library(Benchmarking)

}

if (!require(SvDialogs)) {

install.packages('SvDialogs')

library(SvDialogs)

}

dlgMessage('Selecione o arquivo com as informações do Snis!')

base\_dados <- read.xlsx(file.choose(),sheet = "Snis", startRow = 15)

View(base\_dados)

names(base\_dados)

# Seleção de prestadores que ofertam os dois tipos de serviço (abastecimento de água e esgotamento sanitário)

base\_corte1 <- base\_dados[base\_dados$serv == 'Água e Esgoto',]

View(base\_corte1)

# Exclusão de autarquias

base\_corte2 <- base\_corte1[!(base\_corte1$natjur == 'Autarquia'),]

View(base\_corte2)

# Escala mínima de serviço (número de economias de água) de pelo menos 10% à da Copasa

econ\_agua\_med\_copasa <- mean (base\_corte2[base\_corte1$prest == "COPASA",25],na.rm=TRUE)

econativas\_a\_med <-with(base\_corte2, tapply(econativas\_a,prest,mean,na.rm=TRUE))

econativas\_a\_med <- data.frame(row.names(econativas\_a\_med),econativas\_a\_med)

names(econativas\_a\_med)[1] <- 'prest'

base\_corte2 <- merge(base\_corte2,econativas\_a\_med,by= 'prest')

base\_corte3 <- base\_corte2[base\_corte2$econativas\_a\_med>=econ\_agua\_med\_copasa\*0.1,]

View(base\_corte3)

# Nível mínimo de hidrometração = 70%.

hidrom\_med <- with(base\_corte3, tapply(hidrom, prest,mean,na.rm=TRUE))

hidrom\_med <- data.frame(row.names(hidrom\_med),hidrom\_med)

names(hidrom\_med)[1] <- 'prest'

base\_corte3 <- merge(base\_corte3,hidrom\_med,by= 'prest')

base\_corte4 <- base\_corte3[base\_corte3$hidrom\_med >= 70,]

View(base\_corte4)

## base\_final

base\_final <- base\_corte4

View(base\_final)

## Construção das perdas (1000 m³/ano)

base\_final$perdas = with(base\_final, volprod\_a + volimp\_a - volcons\_a)

## A modelagem DEA utiliza as médias para os anos de 2012 a 2014 para produtos e insumos

opex\_sem\_imp\_defl\_med <- with(base\_final, tapply(opex\_sem\_imp\_defl ,prest,mean,na.rm=TRUE))

econativas\_a\_med <- with(base\_final, tapply(econativas\_a,prest,mean,na.rm=TRUE))

econativas\_e\_med <- with(base\_final, tapply(econativas\_e,prest,mean,na.rm=TRUE))

voledt\_med <- with(base\_final, tapply(voledt,prest,mean,na.rm=TRUE))

perdas\_med <- with(base\_final, tapply(perdas,prest,mean,na.rm=TRUE))

prestadores <- unique(base\_final$prest)

## Escolhendo insumos e produtos

x = cbind (opex\_sem\_imp\_defl\_med ,perdas\_med)

y = cbind (econativas\_a\_med, econativas\_e\_med, voledt\_med)

## CALCULAR O DEA

dea\_model = dea(x,y,RTS ="irs","in")

eficiencia = eff(dea\_model)

is.vector(eficiencia)

tab\_eff <- data.frame(prestadores,eficiencia )

## Correção de viés com o algoritmo de Simar & Wilson (1998)

dea\_model\_ub <-dea.boot(x, y, NREP = 2000, EFF = NULL, RTS = "irs")

tab\_eff\_ub <- data.frame(prestadores,dea\_model\_ub$eff.bc,dea\_model\_ub$conf.int)

names(tab\_eff\_ub) <- c("Prestadores","Eficiências","LI Eficiências","LS Eficiências")

quantile(dea\_model\_ub$eff.bc, probs = 1)

hist(tab\_eff\_ub$Eficiências)

## Fazendo o bootstraping do Malmquist

Malmquist.boot <- function (x1,y1,x2,y2, rts="crs",orientation="in",NREP = 2000){

n = nrow(x1)

tal <- matrix(0,n,NREP)

epsilon <- matrix(0,n,NREP)

Malm <- matrix(0,n,NREP)

for (N in 1:NREP) {

dea1 = dea(x1,y1,RTS = rts,ORIENTATION = orientation)

dea2 = dea(x2,y2,RTS = rts,ORIENTATION = orientation)

## Transformando as distância de Farrel em distância de Shepard

Dt1\_t1 = 1/eff(dea1)

Dt2\_t2 = 1/eff(dea2)

## Contruindo a matriz delta da método dos reflexos

A = matrix(Dt1\_t1, ncol = 1)

B = matrix(Dt2\_t2, ncol = 1)

delta = cbind(rbind(A,2-A,2-A,A),rbind(B,B,2-B,2-B))

## Matrizes de correlação dos dados originais e dos reflexos

original <- cbind(A,B)

sigma <- cov(original)

reflexos <- cbind(2-A,B)

sigma\_r <- cov(reflexos)

## Selecionando aleatoriamente com reposição N elementos de delta

C = matrix (1, ncol = 2, nrow = n)

e <- matrix(0,n,2)

index <- sample(1:(4\*n),n,replace = TRUE)

delta\_star = delta[index,]

delta\_mean\_matrix = diag(colMeans(delta\_star))

## Fazendo o "random draw" da normal bivariada, utilizamos a decomposição de Cholesky

for (i in 1:n) {

if ( index[i] <= n | index[i] > 3\*n) {

M <- t(chol(sigma))

Z <- matrix(rnorm(2), 2, 1)

bvn2 <- t(M %\*% Z)

} else {

M <- t(chol(sigma\_r))

Z <- matrix(rnorm(2), 2, 1)

bvn2 <- t(M %\*% Z)

}

e[i,] <- bvn2

}

## Bandwith recomendado por Simar & Wilson (1999)

h = (4/(5\*n))^1/6

gamma = 1/sqrt(1+h^2)\*(delta\_star + h\*e - C %\*% delta\_mean\_matrix) + C%\*% delta\_mean\_matrix

gamma\_star = matrix (0, n, 2)

for (i in 1:n) {

for(j in 1:2){

if (gamma[i,j]>= 1) { gamma\_star[i,j] = gamma[i,j]

} else {

gamma\_star[i,j] = 2 - gamma[i,j]

}

}

}

x\_star1 = gamma\_star[,1]/original[,1]\*x1

x\_star2 = gamma\_star[,2]/original[,2]\*x2

Dt1t1\_boot = dea(x\_star1,y1, RTS = rts,ORIENTATION = orientation)

Dt2t2\_boot = dea(x\_star2,y2, RTS = rts,ORIENTATION = orientation)

Dt1t2 = dea(x\_star1,y1, XREF = x\_star2, YREF = y2, RTS = rts,ORIENTATION = orientation)

Dt2t1 = dea(x\_star2,y2,XREF = x\_star1, YREF = y1, RTS = rts,ORIENTATION = orientation)

tal[,N] = sqrt((Dt1t1\_boot$eff\*Dt2t1$eff)/(Dt1t2$eff\*Dt2t2\_boot$eff))

epsilon[,N] = Dt2t2\_boot$eff/Dt1t1\_boot$eff

}

Malm = epsilon \* tal

resultados <- list(malmquist = rowMeans(Malm), catch = rowMeans(epsilon), desloc = rowMeans(tal), malm.boot = Malm, catch.boot = epsilon, desloc.boot = tal)

return (resultados)

}

base\_12 <- subset (base\_final,base\_final$ano == 2012)

base\_14 <- subset (base\_final,base\_final$ano == 2014)

x\_12 = with(base\_12,cbind (opex\_sem\_imp\_defl,perdas))

y\_12 = with(base\_12,cbind (econativas\_a, econativas\_e, voledt))

x\_14 = with(base\_14,cbind (opex\_sem\_imp\_defl ,perdas))

y\_14 = with(base\_14,cbind (econativas\_a, econativas\_e, voledt))

modelo\_Malmquist <-Malmquist.boot (x\_12,y\_12,x\_14,y\_14,"crs","in")

deslocamento\_fronteira <- data.frame(prestadores,modelo\_Malmquist$desloc,t(apply(modelo\_Malmquist$desloc.boot,1,quantile,c(0.025,0.975))))

names(deslocamento\_fronteira)<- c("Prestadores","Deslocamento da Fronteira", "LI Deslocamento da Fronteira","LS Deslocamento da Fronteira")