

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_corte1 <- subset(base_dados, serv == 'Água e Esgoto')
```

```
View(base_corte1)
```

Exclusão de autarquias

```
base_corte2 <- base_corte2 <- subset(base_corte1, natjur != 'Autarquia')
```

```
View(base_corte2)
```

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

```
base_corte2B <- subset(base_corte2, prest == 'COPASA')
```

```
econ_agua_med_copasa <- mean(base_corte2B$econativas_a)
```

```
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 <- aggregate(hidrom ~ prest, data=base_corte3, FUN=mean)
```

```
names(hidrom_med)[2] <- 'hidrom_med'
```

```
base_corte3 <- merge(base_corte3,hidrom_med,by= 'prest')
```

```

base_final <- subset(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

base_medias <- aggregate(cbind(opex_sem_imp_defl, econativas_a,
econativas_e, voledt, perdas) ~ prest, data=base_final, FUN=mean)
x <- as.matrix( base_medias[,c("opex_sem_imp_defl", "perdas")] )
y <- as.matrix( base_medias[,c("econativas_a", "econativas_e", "voledt")] )

## 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 bootstrapping 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)
    ## Construindo 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] %in% c(1:n, 31:45) ) {
    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 = 1/dea(x_star1, y1, RTS = rts, ORIENTATION = orientation)$eff
Dt2t2_boot = 1/dea(x_star2, y2, RTS = rts, ORIENTATION = orientation)$eff
Dt1t2_boot = 1/dea(x_star1, y1, XREF = x_star2, YREF = y2, RTS = rts,
ORIENTATION = orientation)$eff
Dt2t1_boot = 1/dea(x_star2, y2, XREF = x_star1, YREF = y1, RTS = rts,
ORIENTATION = orientation)$eff
tal[,N] = sqrt( ((Dt2t1_boot/Dt2t2_boot)*(Dt1t1_boot/Dt1t2_boot)) )
epsilon[,N] = Dt2t2_boot/Dt1t1_boot }
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_frenteira <-
data.frame(prestadores,modelo_Malmquist$desloc,t(apply(modelo_Malmquist$desloc.boot,1,quantil
e,c(0.025,0.975))))
names(deslocamento_frenteira)<- c("Prestadores","Deslocamento da Fronteira", "LI Deslocamento da
Fronteira","LS Deslocamento da Fronteira")

```