ANEXO III-B - RStudio OPEX Regulatório

```
# Limpa os dados do R
rm(list=ls(all=TRUE))

# Instalaçaão dos pacotes estatísticos (necessário apenas uma vez a instalação)
install.packages("Benchmarking", repos = "https://cloud.r-project.org")
install.packages("mvtnorm")
install.packages("openxlsx")
install.packages("readxl")

# Carrega os pacotes necessários
library(Benchmarking)
library(mvtnorm)
library(openxlsx)
library(readxl)

# Definir diretório de trabalho

setwd("COLOCAR O DIRETÓRIO DA PASTA”)

# Importar arquivo em formato xlsx

base_dados <- read_excel("base_dados.xlsx",skip = 1)

# Visualização dos dados

View(base_dados)

##DEA variáveis com valor médio do período de 2020 a 2024
# Define variaveis inputs e outputs DEA
x <- as.matrix(base_dados[,c("DEX_TOTAL - CORR")])
y <-
as.matrix(base_dados[,c("LIG_ATIV_AG","LIG_ATIV_ES","ECO_ATIV_AG","ECO_ATIV_ES
","VOL_PROD_AG","VOL_COL_ES","VOL_TRAT_ES","REDE_AG","REDE_ESG",
"PERDAS_DISTRIB","PERDAS_LIG")])

#Atualizando vari?veis da base de acordo com o modelo utilizado:
empresas=as.matrix(base_dados[,c("Estado","Sigla do Prestador")])
variaveis=cbind(empresas,y,x)
base_dados=as.data.frame(variaveis)

# Calcula o DEA
base_dados$DEA <- dea(X=x, Y=y, RTS="irs", ORIENTATION="in")$eff
```

```
# Correção do pacote Benchmarking

dea.boot <- function(X,Y, NREP=200, EFF=NULL, RTS="vrs",
                ORIENTATION="in", alpha=0.05, XREF=NULL, YREF=NULL, FRONT.IDX=NULL,
                EREF=NULL, DIRECT=NULL, TRANSPOSE=FALSE, SHEPHARD.INPUT=TRUE,
LP=FALSE,
                CONTROL=NULL)
{
 if ( !is.null(DIRECT) )
  stop("Use of DIRECT does not yet work in dea.boot")

 ## ---- LINHA CORRIGIDA (era c("fdh","vrs","drs","crs","irs","irs","add")) ----
 rts_ <- c("fdh","vrs","drs","crs","irs","__nao_usado__","add")
 ## ------------------------------------------------------------------------

 if ( is.numeric(RTS) )  {
  RTStemp <- rts_[1+RTS] # the first fdh is number 0
  RTS <- RTStemp
 }
 RTS <- tolower(RTS)
 rts <- which(RTS == rts_) -1  # Foerste i RTS 'fdh' er 0

 if ( rts < 1 || rts > 4 )
  stop("Invalid value of RTS in call to boot:",RTS)

 orientation <- c("in-out","in","out","graph")
 if ( is.numeric(ORIENTATION) )  {
  ORIENTATION_ <- orientation[ORIENTATION+1]  # "in-out" er nr. 0
  ORIENTATION <- ORIENTATION_
 }
 ORIENTATION <- tolower(ORIENTATION)
 orientation <- which(ORIENTATION == orientation) -1
 if ( orientation < 1 || 3 < orientation )
  stop("Invalid value of ORIENTATION in call to boot")

 if (LP) print(paste("rts =",RTS," orientation=",ORIENTATION), quote=FALSE)
 if (LP) print(paste("rts =",rts," orientation=",orientation), quote=FALSE)

 if ( !is(X, "matrix") )
  stop("X is not a matrix")
 if ( !is(Y, "matrix") )
  stop("Y is not a matrix")
 if ( !is.null(XREF) && !is(XREF, "matrix") )
  stop("XREF is not a matrix")
```

```
 if ( !is.null(YREF) && !is(YREF, "matrix") )
  stop("YREF is not a matrix")
 if ( !is.null(EREF) && (is.null(XREF) || is.null(YREF)) )
  stop("When EREF is present then XREF and YREF must also be present")

 .xyref.missing <- FALSE
 if ( missing(XREF) || is.null(XREF) ) {
  .xyref.missing <- TRUE
  XREF <- X
 }
 if ( missing(YREF) || is.null(YREF) ) {
  .xyref.missing <- TRUE && .xyref.missing
  YREF <- Y
 }

 if ( TRANSPOSE ) {
  X <- t(X)
  Y <- t(Y)
  XREF <- t(XREF)
  YREF <- t(YREF)
  if ( !is.null(DIRECT) & is(DIRECT, "matrix") )
    DIRECT <- t(DIRECT)
 }

 if ( length(FRONT.IDX) > 0 ) {
  if ( !is.vector(FRONT.IDX) & !(ifelse(is.matrix(FRONT.IDX), dim(FRONT.IDX)[2]==1,
FALSE) ))
    stop("FRONT.IDX is not a vector or collumn matrix in 'dea'")
  XREF <- XREF[FRONT.IDX,, drop=FALSE]
  YREF <- YREF[FRONT.IDX,, drop=FALSE]
 }

 rNames <- rownames(XREF)
 if ( is.null(rNames) & !is.null(rownames(YREF)) )
  rNames <- rownames(YREF)

 K <- dim(X)[1]    # number of units, firms, DMUs
 Kr <- dim(XREF)[1] # number of units,units in the reference technology

 if ( is.null(EFF) )
  eff <- dea(X,Y, RTS, ORIENTATION, XREF, YREF, FAST=TRUE, CONTROL=CONTROL)
 else if ( is(EFF, "Farrell") )
  eff <- EFF$eff
 else
  eff <- EFF
 if ( length(eff) != K )
```

```
  stop("The length of EFF must be the number of firms in X and Y")

 if ( is.null(EREF) ) {
  if ( identical(X, XREF) & identical(Y, YREF) ) {
   EREF <- eff
  } else {
   EREF <- dea(XREF, YREF, RTS, ORIENTATION, FAST=TRUE, CONTROL=CONTROL)
  }
 }

 if (!all(!is.na(eff))) {
  nr <- which(is.na(eff))
  msg <- paste0("The dea calculation has NA values for firms ",
           paste(nr, collapse=", "), ".\n",
           "For bootstrap either remove these firms from the data, use a (another)
scaling\n",
           "of the data, or use a scaling option using CONTROL, cf. the manual for
'dea'.\n")
  stop(msg)
 }
 if (!all(!is.na(EREF))) {
  nr <- which(is.na(EREF))
  msg <- paste0("The dea calculation has NA values for firms ",
           paste(nr, collapse=", "), " in the reference technology.\n",
           "For bootstrap either remove these firms from the reference data, use a
(another) scaling\n",
           "of the data, or use a scaling option using CONTROL, cf. the manual for
'dea'.\n")
  stop(msg)
 }

 if (ORIENTATION=="out") farrell<-TRUE else farrell<-FALSE

 dist <- eff
 if ( !farrell ) dist <- 1/dist

 zeff <- eff[ dist > 1 + 1e-6 ]
 if ( length(zeff) == 0 ) {
  cat("No unit with efficiency different from 1.0000.\n", quote=FALSE)
  stop("The range of efficiencies is degenrate, 'dea.boot' stops.")
 }

 neff <- c(zeff,2-zeff)

 adjust <- sd(dist)/sd(neff) * (length(neff)/length(dist))^(1/5)
```

```
  std <- sd(neff)
  iqr <- IQR(neff)/1.349
  if ( iqr > 1e-6 && iqr < std ) std <- iqr
  h0 <- .9 * std * length(neff)^(-1/5)
  h <- adjust * h0
  if (LP) cat("Bandwidth =",h,"\n")

  boot <- matrix(NA, nrow=K, ncol=NREP)

  if ( ORIENTATION == "in" ) {
    for ( b in 1:NREP ) {
      estar <- dea.sample(eff, h, Kr)
      xstar <- EREF/estar * XREF
      boot[,b] <- dea(X,Y, RTS, ORIENTATION, XREF=xstar, YREF, FAST=TRUE,
CONTROL=CONTROL)
    }
  } else if ( ORIENTATION == "out" ) {
    for ( b in 1:NREP ) {
      estar <- dea.sample(eff, h, Kr)
      ystar <- EREF/estar * YREF
      boot[,b] <- dea(X,Y, RTS, ORIENTATION, XREF, YREF=ystar, FAST=TRUE,
CONTROL=CONTROL)
    }
  } else if ( ORIENTATION == "graph" ) {
    for ( b in 1:NREP ) {
      estar <- dea.sample(eff, h, Kr)
      xstar <- EREF/estar * XREF
      ystar <- estar/EREF * YREF
      boot[,b] <- dea(X,Y, RTS, ORIENTATION, XREF=xstar, YREF=ystar,
                FAST=TRUE, CONTROL=CONTROL)
    }
  } else {
    stop("Unknown ORIENTATION for dea.boot")
  }

  if ( SHEPHARD.INPUT & ORIENTATION != "out" ) {
    eff <- 1/eff
    boot <- 1/boot
  }

  bias <- rowMeans(boot, na.rm=TRUE) - eff
  eff.bc <- eff - bias

  ci_ <- t(apply(eff-boot, 1, function(x) {
    quantile(x, probs=c(0.5*alpha, 1-0.5*alpha), type=9, na.rm=TRUE)}))
  ci <- eff + ci_
```

```
  if (SHEPHARD.INPUT & ORIENTATION!="out") {
    eff <- 1/eff
    boot <- 1/boot
    eff.bc <- 1/eff.bc
    ci <- t(apply(1/ci, 1, rev))
    bias <- eff - eff.bc
  }

  var <- apply(boot,1, function(x) {var(x, na.rm=TRUE)})

  bb <- list(eff=eff, eff.bc=eff.bc,
             bias=bias,
             var=var,
             conf.int=ci,
             eref=EREF, boot=boot)

  return( bb )
}

dea.sample <- function(e, h, K=NULL) {
  if ( min(e) < 1-1e-6 ) {
    e <- 1/e
    farrell <- TRUE
  } else {
    farrell <- FALSE
  }
  if ( is.null(K) )
    K <- length(e)

  espejl <- c(e, 2-e)
  beta <- sample(espejl, K, replace=TRUE)
  etilde <- beta + h * rnorm(K)
  estar <- mean(beta) + (etilde-mean(beta)) / sqrt(1+h^2/var(espejl))
  estar <- ifelse(estar < 1, 2-estar, estar)

  if ( farrell ) {
    estar <- 1/estar
  }
  return(estar)
}

## Correção de viés com o algoritmo de Simar e Wilson (bootstrap)
dea_boot <- dea.boot(X=x, Y=y, NREP = 2000, EFF = NULL, RTS= "irs")

base_dados$inf <- dea_boot$conf.int[,1]/dea_boot$eff.bc*base_dados$DEA
```

```
base_dados$sup <- dea_boot$conf.int[,2]/dea_boot$eff.bc*base_dados$DEA


# Bootstrap Arsesp
base_dados$DEA_unbiased <- dea_boot$eff.bc
base_dados$boot.arsesp <-
base_dados$DEA_unbiased/max(base_dados$DEA_unbiased)

write.csv2(base_dados, "resultado-2.csv")

write.xlsx(base_dados, "resultado-2.xlsx", overwrite = TRUE)
```