assignment 4

joshna katta

2022-10-31

knitr::opts\_chunk$set(message = FALSE)  
knitr::opts\_chunk$set(warning = FALSE)

library("Benchmarking")

data.df.values <- matrix(c("Facility 1","Facility 2","Facility 3","Facility 4","Facility 5", "Facility 6",  
 150,400,320,520,350,320,  
 0.2,0.7,1.2,2.0,1.2,0.7,  
 14000,14000,42000,28000,19000,14000,  
 3500,21000,10500,42000,25000,15000), ncol=5, byrow=F)  
colnames(data.df.values) <- c("DMU", "Staff\_Hours\_Per\_Day","Supplies\_Per\_Day","Reimbursed\_Patient\_Days","Privately\_Paid\_Patient\_Days")  
table.df <- as.table(data.df.values)  
table.df

## DMU Staff\_Hours\_Per\_Day Supplies\_Per\_Day Reimbursed\_Patient\_Days  
## A Facility 1 150 0.2 14000   
## B Facility 2 400 0.7 14000   
## C Facility 3 320 1.2 42000   
## D Facility 4 520 2 28000   
## E Facility 5 350 1.2 19000   
## F Facility 6 320 0.7 14000   
## Privately\_Paid\_Patient\_Days  
## A 3500   
## B 21000   
## C 10500   
## D 42000   
## E 25000   
## F 15000

***Calculating Constant Returns to Scale (CRS)***

x <- matrix(c(150,400,320,520,350,320,  
 0.2,0.7,1.2,2.0,1.2,0.7),ncol=2)  
y <- matrix(c(14000,14000,42000,28000,19000,14000,  
 3500,21000,10500,42000,25000,15000),ncol=2)  
colnames(y) <- c("Reimbursed\_Patient\_Days","Privately\_Paid\_Patient\_Days")  
colnames(x) <- c("Staff\_Hours\_Per\_Day","Supplies\_Per\_Day")  
D\_E\_A\_crs<-dea(x, y, RTS = "crs")  
D\_E\_A\_crs

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

peers(D\_E\_A\_crs)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 1 2 4  
## [6,] 1 2 4

lambda(D\_E\_A\_crs)

## L1 L2 L3 L4  
## [1,] 1.0000000 0.00000000 0 0.0000000  
## [2,] 0.0000000 1.00000000 0 0.0000000  
## [3,] 0.0000000 0.00000000 1 0.0000000  
## [4,] 0.0000000 0.00000000 0 1.0000000  
## [5,] 0.2000000 0.08048142 0 0.5383307  
## [6,] 0.3428571 0.39499264 0 0.1310751

*CRS Observations -* *1. We get to see that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient.* *2. Also, we get to see that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities.* *3. Facility 5 is 97.75 % efficient leaving 2.25 % as inefficient and Facility 6 is 86.75 % efficient leaving 13.25 % as inefficient.*

***Calculating Decreasing Returns to Scale (DRS)***

D\_E\_A\_drs <- dea(x, y, RTS = "drs")  
D\_E\_A\_drs

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

peers(D\_E\_A\_drs)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 1 2 4  
## [6,] 1 2 4

lambda(D\_E\_A\_drs)

## L1 L2 L3 L4  
## [1,] 1.0000000 0.00000000 0 0.0000000  
## [2,] 0.0000000 1.00000000 0 0.0000000  
## [3,] 0.0000000 0.00000000 1 0.0000000  
## [4,] 0.0000000 0.00000000 0 1.0000000  
## [5,] 0.2000000 0.08048142 0 0.5383307  
## [6,] 0.3428571 0.39499264 0 0.1310751

*DRS Observations -* *1. We get to see that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient.* *2. Also, we get to see that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities.* *3. Facility 5 is 97.75 % efficient leaving 2.25 % as inefficient and Facility 6 is 86.75 % efficient leaving 13.25 % as inefficient.*

***Calculating Increasing Returns to Scale (IRS)***

D\_E\_A\_irs <- dea(x, y, RTS = "irs")  
D\_E\_A\_irs

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

peers(D\_E\_A\_irs)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 5 NA NA  
## [6,] 1 2 5

lambda(D\_E\_A\_irs)

## L1 L2 L3 L4 L5  
## [1,] 1.0000000 0.0000000 0 0 0.0000000  
## [2,] 0.0000000 1.0000000 0 0 0.0000000  
## [3,] 0.0000000 0.0000000 1 0 0.0000000  
## [4,] 0.0000000 0.0000000 0 1 0.0000000  
## [5,] 0.0000000 0.0000000 0 0 1.0000000  
## [6,] 0.4014399 0.3422606 0 0 0.2562995

*IRS Observations -* *1. We get to see that Facility 1, Facility 2, Facility 3, Facility 4 and Facility 5 are efficient.* *2. Also, we get to see that Facility 1, Facility 2 and Facility 5 are the peer members for Facility 6 which is the only inefficient facility.* *3. Facility 6 is 89.63 % efficient leaving 10.37 % as inefficient.*

***Calculating Variable Returns to Scale (VRS)***

D\_E\_A\_vrs <- dea(x, y, RTS = "vrs")  
D\_E\_A\_vrs

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

peers(D\_E\_A\_vrs)

## peer1 peer2 peer3  
## [1,] 1 NA NA  
## [2,] 2 NA NA  
## [3,] 3 NA NA  
## [4,] 4 NA NA  
## [5,] 5 NA NA  
## [6,] 1 2 5

lambda(D\_E\_A\_vrs)

## L1 L2 L3 L4 L5  
## [1,] 1.0000000 0.0000000 0 0 0.0000000  
## [2,] 0.0000000 1.0000000 0 0 0.0000000  
## [3,] 0.0000000 0.0000000 1 0 0.0000000  
## [4,] 0.0000000 0.0000000 0 1 0.0000000  
## [5,] 0.0000000 0.0000000 0 0 1.0000000  
## [6,] 0.4014399 0.3422606 0 0 0.2562995

*VRS Observations -* *1. We get to see that Facility 1, Facility 2, Facility 3, Facility 4 and Facility 5 are efficient.* *2. Also, we get to see that Facility 1, Facility 2 and Facility 5 are the peer members for Facility 6 which is the only inefficient facility.* *3. Facility 6 is 89.63 % efficient leaving 10.37 % as inefficient.*

***Calculating Free Disposability Hull (FDH)***

D\_E\_A\_fdh <- dea(x, y, RTS = "fdh")  
D\_E\_A\_fdh

## [1] 1 1 1 1 1 1

peers(D\_E\_A\_fdh)

## peer1  
## [1,] 1  
## [2,] 2  
## [3,] 3  
## [4,] 4  
## [5,] 5  
## [6,] 6

lambda(D\_E\_A\_fdh)

## L1 L2 L3 L4 L5 L6  
## [1,] 1 0 0 0 0 0  
## [2,] 0 1 0 0 0 0  
## [3,] 0 0 1 0 0 0  
## [4,] 0 0 0 1 0 0  
## [5,] 0 0 0 0 1 0  
## [6,] 0 0 0 0 0 1

*FDH Observations -* *All the DMUs are efficient. This is basically due to the principal which FDH technique follows thereby detecting even a small level of efficiency.*

***Calculating Free Replicability Hull (FRH)***

#here FRH is calculated by specifying RTS = "add"  
D\_E\_A\_frh <- dea(x, y, RTS = "add")  
D\_E\_A\_frh

## [1] 1 1 1 1 1 1

peers(D\_E\_A\_frh)

## peer1  
## [1,] 1  
## [2,] 2  
## [3,] 3  
## [4,] 4  
## [5,] 5  
## [6,] 6

lambda(D\_E\_A\_frh)

## L1 L2 L3 L4 L5 L6  
## [1,] 1 0 0 0 0 0  
## [2,] 0 1 0 0 0 0  
## [3,] 0 0 1 0 0 0  
## [4,] 0 0 0 1 0 0  
## [5,] 0 0 0 0 1 0  
## [6,] 0 0 0 0 0 1