QMM DEA

Gloria Stephen

2022-10-31

library("Benchmarking")

## Loading required package: lpSolveAPI

## Loading required package: ucminf

## Loading required package: quadprog

##   
## Loading Benchmarking version 0.30h, (Revision 244, 2022/05/05 16:31:31) ...

## Build 2022/05/05 16:31:40

ddf\_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(ddf\_values) <- c("DMU","Staff\_Hours\_Per\_Day","Supplies\_Per\_Day","Reimbursed\_Patient\_Days","Privately\_Paid\_Patient\_Days")  
table.dataframe <- as.table(ddf\_values)  
table.dataframe

## 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 the Constant Returns to Scale (CRS)

x1 <- matrix(c(150,400,320,520,350,320,  
 0.2,0.7,1.2,2.0,1.2,0.7),ncol=2)  
y1 <- matrix(c(14000,14000,42000,28000,19000,14000,  
 3500,21000,10500,42000,25000,15000),ncol=2)  
colnames(y1) <- c("Reimbursed\_Patient\_Days","Privately\_Paid\_Patient\_Days")  
colnames(x1) <- c("Staff\_Hours\_Per\_Day","Supplies\_Per\_Day")  
DEA\_CRS <-dea(x1, y1, RTS = "crs")  
DEA\_CRS

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

peers(DEA\_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(DEA\_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

# It is observed that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient.\*

# That Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 that are the inefficient facilities.\*

# Facility 5 is 97.75 % is efficient leaving 2.25 % as inefficient and Facility 6 is 86.75 % efficient leaving 13.25 % as inefficient

#Calculating the Decreasing Returns to Scale (DRS)

DEA\_CRS <- dea(x1, y1, RTS = "drs")  
DEA\_CRS

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

peers(DEA\_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(DEA\_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

#DRS Observations # It is observed that simiarly that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient # It is also observed that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities # Facility 5 is 97.75 % efficient leaving 2.25 % as inefficient and Facility 6 is 86.75 % efficient leaving 13.25 % as inefficient

# Calculating the Increasing Returns to Scale (IRS)

DEA\_IRS <- dea(x1, y1, RTS = "irs")  
DEA\_IRS

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

peers(DEA\_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(DEA\_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

# The IRS Observations

# It is observed that simiarly that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient

# It is also observed that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities

# Facility 6 is 89.63 % efficient leaving 10.37 % as inefficient

# Calculating the Variable Returns to Scale (VRS)

DEA\_VRS <- dea(x1, y1, RTS = "vrs")  
DEA\_VRS

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

peers(DEA\_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(DEA\_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

# The VRS Observations

# It is observed that simiarly that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient

# It is also observed that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities

# Facility 6 is 89.63 % efficient leaving 10.37 % as inefficient.\*

# Calculating the Free Disposability Hull (FDH)

DEA\_FDH <- dea(x1, y1, RTS = "fdh")  
DEA\_FDH

## [1] 1 1 1 1 1 1

peers(DEA\_FDH)

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

lambda(DEA\_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

# The FDH Observations

# The DMUs are efficient, this is FDH follows this principle such that it detects a small level of efficiency

# Calculating the Free Replicability Hull (FRH)

# The FRH is calculated by specifying RTS = "add"  
DEA\_FRH <- dea(x1, y1, RTS = "add")  
DEA\_FRH

## [1] 1 1 1 1 1 1

peers(DEA\_FRH)

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

lambda(DEA\_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

# FRH Observations

# All the DMUs are efficient and it follows the no convexity assumption ensuring that the output is free from disposal and replication.\*

# Summary of Results - Inefficient DMU’s

ddf.summarise.inefficient <- matrix(c("CRS","DRS","IRS","VRS","FDH","FRH",  
 2,2,1,1,0,0,  
 "Facility 5 & 6", "Facility 5 & 6","Facility 6", "Facility 6", "-","-",  
 "97.75% & 86.7%","97.75% & 86.7%","89.63%","89.63%","-","-",  
 "Facility 1, 2 & 4","Facility 1, 2 & 4","Facility 1, 2 & 5","Facility 1, 2 & 5","-","-",  
 "0.2, 0.08, 0.54 and 0.34, 0.4, 0.13", "0.2, 0.08, 0.54 and 0.34, 0.4, 0.13", "0.4, 0.34 and 0.26", "0.4, 0.34 and 0.26", "-","-"),ncol=6,byrow=F)  
colnames(ddf.summarise.inefficient) <- c("RTS","Count\_Inefficient\_DMUs","Name\_DMUs","%\_Inefficiency","Peers","Lambda")  
as.table(ddf.summarise.inefficient)

## RTS Count\_Inefficient\_DMUs Name\_DMUs %\_Inefficiency Peers   
## A CRS 2 Facility 5 & 6 97.75% & 86.7% Facility 1, 2 & 4  
## B DRS 2 Facility 5 & 6 97.75% & 86.7% Facility 1, 2 & 4  
## C IRS 1 Facility 6 89.63% Facility 1, 2 & 5  
## D VRS 1 Facility 6 89.63% Facility 1, 2 & 5  
## E FDH 0 - - -   
## F FRH 0 - - -   
## Lambda   
## A 0.2, 0.08, 0.54 and 0.34, 0.4, 0.13   
## B 0.2, 0.08, 0.54 and 0.34, 0.4, 0.13   
## C 0.4, 0.34 and 0.26  
## D 0.4, 0.34 and 0.26   
## E -   
## F -

#Summary of Results - Efficient DMU’s

ddf.summarise.efficient <- matrix(c("CRS","DRS","IRS","VRS","FDH","FRH",  
"Facility 1, 2, 3 & 4","Facility 1, 2, 3 & 4","Facility 1, 2, 3, 4 & 5", "Facility 1, 2, 3, 4 & 5", "All DMUs", "All DMUs"), ncol = 2, byrow=F)  
colnames(ddf.summarise.efficient) <- c("RTS", "Efficient\_DMUs")  
as.table(ddf.summarise.efficient)

## RTS Efficient\_DMUs   
## A CRS Facility 1, 2, 3 & 4   
## B DRS Facility 1, 2, 3 & 4   
## C IRS Facility 1, 2, 3, 4 & 5  
## D VRS Facility 1, 2, 3, 4 & 5  
## E FDH All DMUs   
## F FRH All DMUs

#Interpretation of the DEA Analysis: # A brief gist of the scales are as follows

# Constant Returns to Scale (CRS) is regarded as the original scale and is utilized by the majority of businesses.

# The Decreasing, Increasing and Varying Returns to Scale (DRS, IRS, and VRS) dispersion scales aid in our understanding of what to increase and what to decrease dependent on input deployment.

# The Free Disposability and Free Replicability Hull (FDH & FRH), which makes no assumptions about convexity, is regarded as a non-parametric way to evaluate the effectiveness of DMUs.

# Constant Returns to Scale (CRS)

# The findings show that DMUs 1, 2, 3, and 4 are effective. Only 97.75% of DMU(5) and 86.7% of DMU(6) are effectively used. On the basis of our initial analysis, we discovered this. In addition, DMU(4peer )’s units are 1, 2, and 4, with respective weights of 0.2, 0.08, and 0.54. The peer units for DMU(6) are 1, 2, and 4, with respective weights of 0.34, 0.4, and 0.13.

# In general, CRS aids us in determining whether any potential DMUs may be scaled up or down; in this example, DMUs 1, 2, 3, and 4 can be scaled up.

# Decreasing Returns to Scale (DRS)

# The findings show that DMUs 1, 2, 3, and 4 are effective. Only 97.75% of DMU(5) and 86.7% of DMU(6) are efficient. On the basis of our initial analysis, we discovered this. In addition, DMU(4peer )’s units are 1, 2, and 4, with respective weights of 0.2, 0.08, and 0.54. The peer units for DMU(6) are 1, 2, and 4, with respective weights of 0.34, 0.4, and 0.13.

# By looking at the inefficient DMUs, in this case DMUs 5 and 6, we may determine whether there are any alternative DMUs where we can scale the processes. Since the CRS values are the base original, they can also be used to retrieve this item.

# IRS - Increasing Returns to Scale

# The outcomes show that DMUs 1, 2, 3, 4, and 5 are productive. Only 89.63% of the DMU(6) is effective. On the basis of our initial analysis, we discovered this. Additionally, DMU(6peer )’s units are 1, 2, and 5, with respective relative weights of 0.4, 0.34, and 0.26.

# By examining the efficiency scores, it enables any firm to determine whether it may arbitrarily expand its operations (Refer ddf.summarise.efficient table)

# VRS - Variable Returns to Scale

# The results indicate that DMUs 1, 2, 3, 4 and 5 are efficient. DMU(6) is only 89.63% efficient. This is what we found basis our initial analysis. Further, the peer units for DMU(6) are 1, 2 and 5 with relative weights of 0.4, 0.34 and 0.26 respectively

# Varying or Variable Returns to Scale helps us understand the scale of operations with variations towards the input and output factor either by increasing or decreasing or by employing both

# FDH - Free Disposability Hull

# The results show that all of the DMUs are efficient. This is mostly due to the assumption of no convexity, and this method enables the scale to capture even the tiniest amount of efficiency

# FRH - Free Replicability Hull

# The FRH findings show that all of the DMUs are efficient. This is mostly owing to the assumption of no convexity, and in general, this method allows the scale to capture even the lowest amount of efficiency that is devoid of replication and disposal.

# The peer values i.e. neighbors and lambda values i.e. weights of the peers would be only retrieved to the DMUs which are inefficient. Efficient DMUs don’t have peers and lambda weights

# Conclusion -

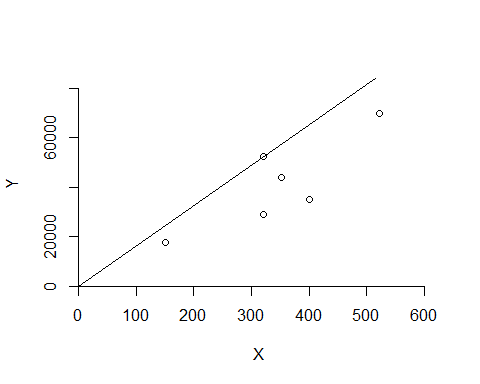
# It is must to note that DEA is a very useful tool to any firm in deciding which is the best DMU i.e. which of the Decision Making Unit has to be maximized so that there would be an increase, decrease or any kind of variations to the output by feeding input into it

# A company can decide upon which of the RTS it wants to employ i.e. Returns to Scale based on their requirements, each of these scales has it’s own importance

#Plotting the Graphs

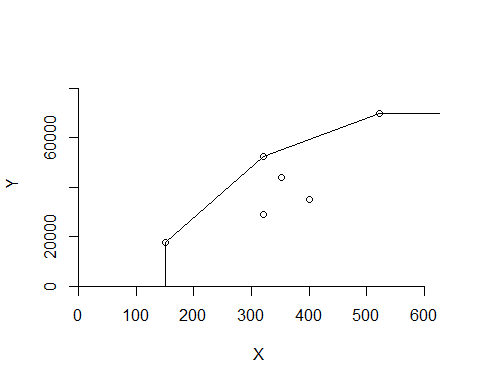
# CRS Plot

dea.plot(x1, y1, RTS='crs')



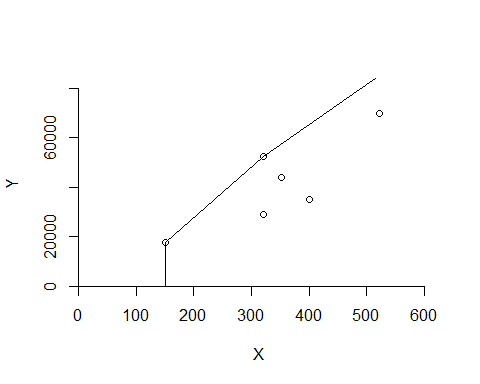
# DRS Plot

dea.plot(x1,y1,RTS="vrs")



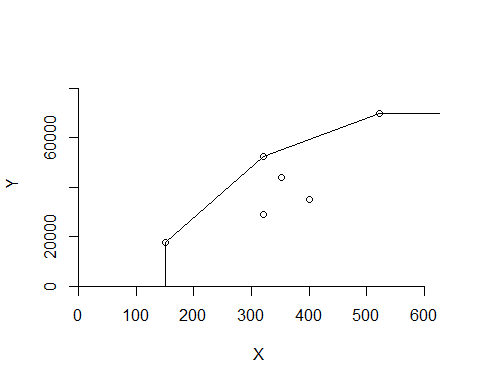
# IRS PLOT

dea.plot(x1,y1,RTS="irs")



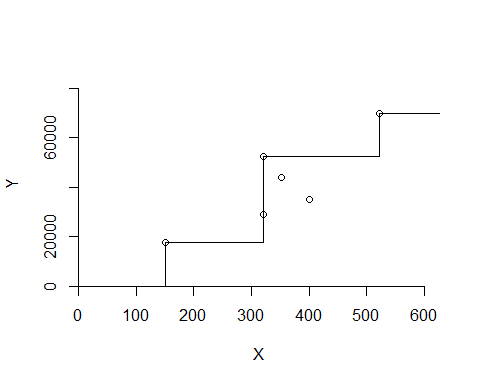
#VRS PLOT

dea.plot(x1,y1,RTS="vrs")



# FDH Plot

dea.plot(x1,y1,RTS="fdh")

 # FRH Plot

dea.plot(x1,y1,RTS="add")

