Data Envelopment Analysis(DEA)

Shashidhar Reddy Boreddy

2022-10-29

# Uploading The Required Libraries  
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

library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ readr 2.1.2 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

#Compute the Formulation  
#In this section, we will create a matrix and values  
#creating the vectors with our values  
input\_values <- matrix(c(150, 400, 320, 520, 350, 320,  
0.2, 0.7, 1.2, 2.0, 1.2, 0.7), ncol=2)  
output\_values <- matrix(c(14000, 14000, 42000, 28000, 19000, 14000,  
3500, 21000, 10500, 42000, 25000, 15000), ncol=2)  
# Defining column names  
colnames(input\_values) <- c("Staff Hours Per Day", "Supplies Per Day")  
colnames(output\_values) <- c("Reimbursed Patient-Days", "Privately Paid Patient-Days")  
#To view the input\_values values  
input\_values

## Staff Hours Per Day Supplies Per Day  
## [1,] 150 0.2  
## [2,] 400 0.7  
## [3,] 320 1.2  
## [4,] 520 2.0  
## [5,] 350 1.2  
## [6,] 320 0.7

# To view the output\_values values  
output\_values

## Reimbursed Patient-Days Privately Paid Patient-Days  
## [1,] 14000 3500  
## [2,] 14000 21000  
## [3,] 42000 10500  
## [4,] 28000 42000  
## [5,] 19000 25000  
## [6,] 14000 15000

#As we can see, we are obtaining the identical numbers as in the performance data table from the six nursing facilities operated by Hope Valley Health Care Association.  
#In the next part, we will conduct a Data Envelopment Analysis (DEA), which is an analytical method that may assist businesses in identifying and allocating resources to improve efficiency and practice.

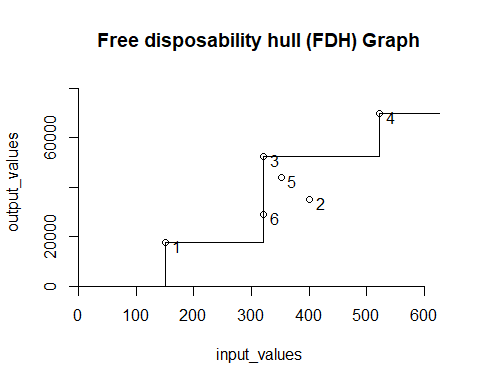
#FDH-based DEA Analysis  
#Now we'll use FDH to define and compute the DEA analysis.The Free disposability hull (FDH) is the assumption that undesirable inputs and outputs are discarded. "We can always create fewer outputs with more inputs because we have free disposability."  
#Giving the input\_values and output\_values.  
Anal.FDH<- dea(input\_values,output\_values,RTS = "FDH")  
#Making a data frame with efficiency values.  
EV.FDH <- as.data.frame(Anal.FDH$eff)  
# assigning an appropiate name  
colnames(EV.FDH) <- c("Efficiency\_FDH")  
# Identification of peers  
p.FDH <- peers(Anal.FDH)  
# assigning an appropiate name for peers  
colnames(p.FDH) <- c("Peer1\_FDH")  
# Using the lambda function, determine the relative weights assigned to peers.  
lambda\_FDH <- lambda(Anal.FDH)  
# giving Lambda exact column name  
colnames(lambda\_FDH) <- c("L1\_FDH", "L2\_FDH", "L3\_FDH", "L4\_FDH", "L5\_FDH", "L6\_FDH")  
# Creating a tabular data with peer, lambda, and efficiency  
peer\_lambda\_EV.FDH <- cbind(p.FDH, lambda\_FDH, EV.FDH)  
# displaying summary chart  
peer\_lambda\_EV.FDH

## Peer1\_FDH L1\_FDH L2\_FDH L3\_FDH L4\_FDH L5\_FDH L6\_FDH Efficiency\_FDH  
## 1 1 1 0 0 0 0 0 1  
## 2 2 0 1 0 0 0 0 1  
## 3 3 0 0 1 0 0 0 1  
## 4 4 0 0 0 1 0 0 1  
## 5 5 0 0 0 0 1 0 1  
## 6 6 0 0 0 0 0 1 1

#As we learnt in this session, peers are a good method to discover inefficient DMUs or units, and Lambda is a good approach to find inefficient units.When solving the DEA model, values are the raw weights assigned by peer units.The summary figure above indicates that each DMU or facility is operating at maximum capacity.efficiency. Because each peer was allocated a single unit, the Lambda and efficiency values are both 1as well as

# displaying the Plot results

dea.plot(input\_values,output\_values,RTS="FDH",ORIENTATION="in-out",txt=TRUE, xlab = "input\_values", ylab= "output\_values", main="Free disposability hull (FDH) Graph")

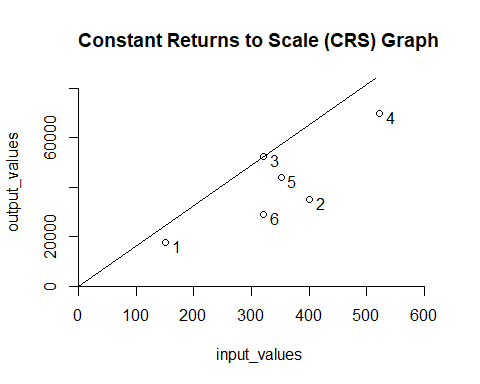


#CRS-based DEA Analysis  
#Now we'll formulate and calculate the DEA analysis with Constant Returns to Scale (CRS).The CRS is part of the scaling assumption, and it helps us to evaluate if there are any feasible scaling up or down combinations.  
# Giving the input\_values and output\_values  
Anal.CRS <- dea(input\_values,output\_values,RTS = "CRS")  
# observing the effciency values  
EV.CRS <- as.data.frame(Anal.CRS$eff)  
# assigning a appropiate name  
colnames(EV.CRS) <- c("Efficiency\_CRS")  
# Identification of peers  
Peers\_CRS <- peers(Anal.CRS)  
# assigning an appropiate name  
colnames(Peers\_CRS) <- c("Peer1\_CRS", "Peer2\_CRS", "Peer3\_CRS")  
# Identifying the relative weights given to the peers using lambda function  
lambda\_CRS <- lambda(Anal.CRS)  
# assign a appropiate column name for Lambda  
colnames(lambda\_CRS) <- c("L1\_CRS", "L2\_CRS", "L3\_CRS", "L4\_CRS")  
# Create a tabular data with peer, lambda, and efficiency  
Peers\_lambda\_EV.CRS <- cbind(Peers\_CRS, lambda\_CRS, EV.CRS)  
#displaying summary chart  
Peers\_lambda\_EV.CRS

## Peer1\_CRS Peer2\_CRS Peer3\_CRS L1\_CRS L2\_CRS L3\_CRS L4\_CRS  
## 1 1 NA NA 1.0000000 0.00000000 0 0.0000000  
## 2 2 NA NA 0.0000000 1.00000000 0 0.0000000  
## 3 3 NA NA 0.0000000 0.00000000 1 0.0000000  
## 4 4 NA NA 0.0000000 0.00000000 0 1.0000000  
## 5 1 2 4 0.2000000 0.08048142 0 0.5383307  
## 6 1 2 4 0.3428571 0.39499264 0 0.1310751  
## Efficiency\_CRS  
## 1 1.0000000  
## 2 1.0000000  
## 3 1.0000000  
## 4 1.0000000  
## 5 0.9774987  
## 6 0.8674521

#In terms of Constant Returns to Scale (CRS), the facilities 1, 2, 3, and 4 are utilizing all of their efficiency, as demonstrated by the lambdas and peers. Facility 5 and 6, on the other hand, require portions of 1, 2, and 4, as demonstrated by the peers and lambdas above. It suggests that these two facilities (5 and 6) have opportunity to improve because their efficiency rates are 97.74% and 86.74%, respectively.

# displaying the Plot results  
dea.plot(input\_values,output\_values,RTS="crs",ORIENTATION="in-out",  
txt=TRUE, xlab = 'input\_values', ylab= 'output\_values', main="Constant Returns to Scale (CRS) Graph")

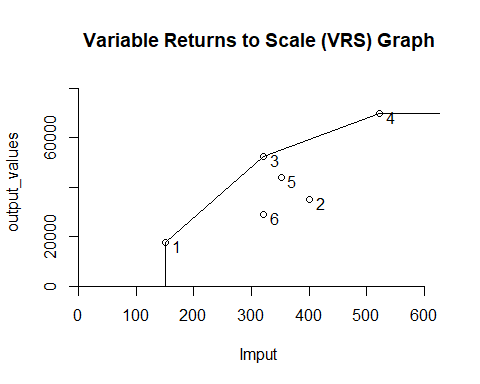


#VRS DEA Analysis  
#Now we will construct and calculate the DEA analysis utilizing Variable Returns to Scale (VRS). VRS is also part of the scaling assumption, and it aids in estimating the efficiency of the variables whether An increase or reduction is not proportionate.  
# Giving the input\_values and output\_values  
Anal.VRS <- dea(input\_values,output\_values,RTS = "VRS")  
# To observe effciency values  
EV.VRS <- as.data.frame(Anal.VRS$eff)  
# assigning an appropiate name  
colnames(EV.VRS) <- c("Efficiency\_VRS")  
# Identify the peers  
Peers\_VRS <- peers(Anal.VRS)  
# assigning an appropiate name  
colnames(Peers\_VRS) <- c("Peer1\_VRS", "Peer2\_VRS", "Peer3\_VRS")  
# Identify the relative weights given to the peers using lambda function  
lambda\_VRS <- lambda(Anal.VRS)  
# To assign an appropiate column name for Lambda  
colnames(lambda\_VRS) <- c("L1\_VRS", "L2\_VRS", "L3\_VRS", "L4\_VRS", "L5\_VRS")  
# Create a tabular data with peer, lambda, and efficiency  
peers\_lambda\_EV.VRS <- cbind(Peers\_VRS, lambda\_VRS, EV.VRS)  
#displaying summary chart  
peers\_lambda\_EV.VRS

## Peer1\_VRS Peer2\_VRS Peer3\_VRS L1\_VRS L2\_VRS L3\_VRS L4\_VRS L5\_VRS  
## 1 1 NA NA 1.0000000 0.0000000 0 0 0.0000000  
## 2 2 NA NA 0.0000000 1.0000000 0 0 0.0000000  
## 3 3 NA NA 0.0000000 0.0000000 1 0 0.0000000  
## 4 4 NA NA 0.0000000 0.0000000 0 1 0.0000000  
## 5 5 NA NA 0.0000000 0.0000000 0 0 1.0000000  
## 6 1 2 5 0.4014399 0.3422606 0 0 0.2562995  
## Efficiency\_VRS  
## 1 1.0000000  
## 2 1.0000000  
## 3 1.0000000  
## 4 1.0000000  
## 5 1.0000000  
## 6 0.8963283

#When we run the Variable Returns to Scale (VRS), we can see that facilities 1, 2, 3, 4, and 5 are all operating at full capacity or efficiency. This is not the case with facility 6, which has an efficiency of 89.63%. Peers and lambdas indicate that facility 6 requires parts of facilities 1, 2, and 5 to be more efficient.

# displaying the Plot results  
dea.plot(input\_values,output\_values,RTS="vrs",ORIENTATION="in-out",  
txt=TRUE, xlab ='Imput', ylab = 'output\_values', main ="Variable Returns to Scale (VRS) Graph")

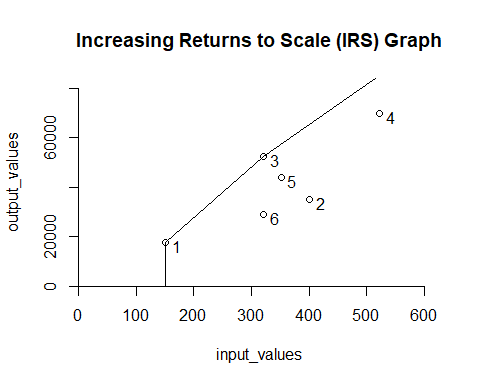


#IRS Analysis for DEA  
#Now we'll formulate and compute the DEA analysis with Increasing Returns to Scale (IRS).IRS shows whether or not the operation scale may be created.  
# Giving the input\_values and output\_values  
Anal.IRS <- dea(input\_values,output\_values,RTS = "irs")  
# observe effciency values  
EV.IRS <- as.data.frame(Anal.IRS$eff)  
# assigning an appropiate name  
colnames(EV.IRS) <- c("Efficiency\_IRS")  
# Identify the peers  
Peers\_IRS <- peers(Anal.IRS)  
# assign an appropiate name  
colnames(Peers\_IRS) <- c("Peer1\_IRS", "Peer2\_IRS", "Peer3\_IRS")  
# Identify the relative weights given to the peers using lambda function  
lambda\_IRS <- lambda(Anal.IRS)  
# assigning an appropiate column name for Lambda  
colnames(lambda\_IRS) <- c("L1\_IRS", "L2\_IRS", "L3\_IRS", "L4\_IRS", "L5\_IRS")  
# Create a tabular data with peer, lambda, and efficiency  
peers\_lambda\_EV.IRS <- cbind(Peers\_IRS, lambda\_IRS, EV.IRS)  
#displaying summary chart  
peers\_lambda\_EV.IRS

## Peer1\_IRS Peer2\_IRS Peer3\_IRS L1\_IRS L2\_IRS L3\_IRS L4\_IRS L5\_IRS  
## 1 1 NA NA 1.0000000 0.0000000 0 0 0.0000000  
## 2 2 NA NA 0.0000000 1.0000000 0 0 0.0000000  
## 3 3 NA NA 0.0000000 0.0000000 1 0 0.0000000  
## 4 4 NA NA 0.0000000 0.0000000 0 1 0.0000000  
## 5 5 NA NA 0.0000000 0.0000000 0 0 1.0000000  
## 6 1 2 5 0.4014399 0.3422606 0 0 0.2562995  
## Efficiency\_IRS  
## 1 1.0000000  
## 2 1.0000000  
## 3 1.0000000  
## 4 1.0000000  
## 5 1.0000000  
## 6 0.8963283

#Rising Returns to Scale (IRS) is the same as Variable Returns to Scale (VRS) since facilities 1, 2, 3, 4, and 5 are all operating at full capacity, while facility 6 requires improvements from units 1, 2, and 5 to enhance their efficiency, which is 89.63%.

# displaying the Plot results  
dea.plot(input\_values,output\_values,RTS="irs",ORIENTATION="in-out",  
txt=TRUE, xlab = 'input\_values', ylab= 'output\_values', main="Increasing Returns to Scale (IRS) Graph")

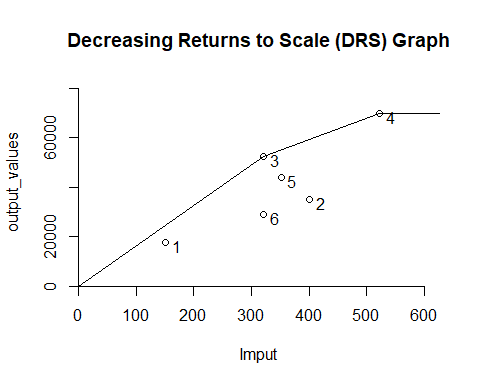


#DRS for DEA Analysis  
#Now we will develop and perform the DEA analysis based on Decreasing Returns to Scale (DRS).DRS is the inverse of IRS in that its purpose is to reduce the operation size on every potential productionprocess.  
# Giving the input\_values and output\_values  
Anal.DRS <- dea(input\_values,output\_values,RTS = "DRS")  
# To observe the effciency values  
EV.DRS <- as.data.frame(Anal.DRS$eff)  
# assigning an appropiate name  
colnames(EV.DRS) <- c("Efficiency\_DRS")  
# Identifying the peers  
Peers\_DRS <- peers(Anal.DRS)  
# assigning an appropiate name  
colnames(Peers\_DRS) <- c("Peer1\_DRS", "Peer2\_DRS", "Peer3\_DRS")  
# Identifying the relative weights given to the peers using lambda function  
lambda\_DRS <- lambda(Anal.DRS)  
# assign an appropiate column name for Lambda  
colnames(lambda\_DRS) <- c("L1\_DRS", "L2\_DRS", "L3\_DRS", "L4\_DRS")  
# Create a tabular data with peer, lambda, and efficiency  
peers\_lambda\_EV.DRS <- cbind(Peers\_DRS, lambda\_DRS, EV.DRS)  
#displaying summary chart  
peers\_lambda\_EV.DRS

## Peer1\_DRS Peer2\_DRS Peer3\_DRS L1\_DRS L2\_DRS L3\_DRS L4\_DRS  
## 1 1 NA NA 1.0000000 0.00000000 0 0.0000000  
## 2 2 NA NA 0.0000000 1.00000000 0 0.0000000  
## 3 3 NA NA 0.0000000 0.00000000 1 0.0000000  
## 4 4 NA NA 0.0000000 0.00000000 0 1.0000000  
## 5 1 2 4 0.2000000 0.08048142 0 0.5383307  
## 6 1 2 4 0.3428571 0.39499264 0 0.1310751  
## Efficiency\_DRS  
## 1 1.0000000  
## 2 1.0000000  
## 3 1.0000000  
## 4 1.0000000  
## 5 0.9774987  
## 6 0.8674521

#Decreasing Returns to Scale (DRS) is effective in facilities 1, 2, 3, and 4. There is need for improvement in facilities 5 and 6. Both require parts of facilities 1, 2, and 4 to attain their maximum efficiency of 1, as shown in the preceding table.

# displaying the Plot results  
dea.plot(input\_values,output\_values,RTS="drs",ORIENTATION="in-out",  
txt=TRUE, xlab = 'Imput', ylab= 'output\_values', main="Decreasing Returns to Scale (DRS) Graph")

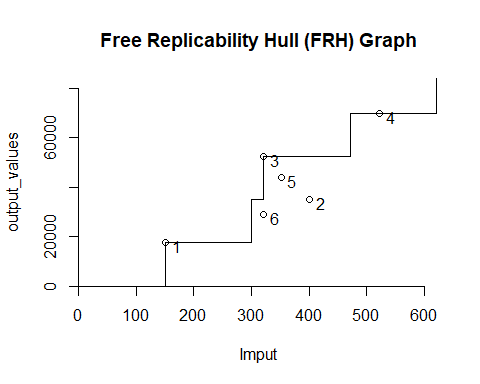


#FRH DEA Analysis  
#Now, we'll use Free Replicability Hull to define and perform the DEA analysis (FRH).Both FRH and FDH employ mixed integer programming, which requires the variables to be integers in order to find the best solution. FRH's purpose is to use random variables to substitute deterministic data.  
# Giving the input\_values and output\_values  
Anal.FRH <- dea(input\_values,output\_values,RTS = "add")  
# To observe effciency values  
EV.FRH <- as.data.frame(Anal.FRH$eff)  
# assigning an appropiate name  
colnames(EV.FRH) <- c("Efficiency\_FRH")  
# Identifying the peers  
Peers\_FRH <- peers(Anal.FRH)  
# assign an appropiate name  
colnames(Peers\_FRH) <- c("peer1\_FRH")  
# Identifying the relative weights given to the peers using lambda function  
lambda\_FRH <- lambda(Anal.FRH)  
# To assign an appropiate column name for Lambda  
colnames(lambda\_FRH) <- c("L1\_FRH", "L2\_FRH", "L3\_FRH", "L4\_FRH", "L5\_FRH", "L6\_FRH")  
# Create a tabular data with peer, lambda, and efficiency  
peers\_lambda\_EV.FRH <- cbind(Peers\_FRH, lambda\_FRH, EV.FRH)  
#displaying summary chart  
peers\_lambda\_EV.FRH

## peer1\_FRH L1\_FRH L2\_FRH L3\_FRH L4\_FRH L5\_FRH L6\_FRH Efficiency\_FRH  
## 1 1 1 0 0 0 0 0 1  
## 2 2 0 1 0 0 0 0 1  
## 3 3 0 0 1 0 0 0 1  
## 4 4 0 0 0 1 0 0 1  
## 5 5 0 0 0 0 1 0 1  
## 6 6 0 0 0 0 0 1 1

#All of the DMU in Free Replicability Hull (FRH) are extremely efficient. It behaves similarly to the Free disposability hull (FDH), in that each of its values has its own peer, lambas, and efficiency of 1.

# displaying the Plot results  
dea.plot(input\_values,output\_values,RTS="add",ORIENTATION="in-out",  
txt=TRUE, xlab = 'Imput', ylab= 'output\_values', main="Free Replicability Hull (FRH) Graph")



#Comparison of various assumptions  
#"All DEA models share the premise of estimating the technology using a minimal extrapolation technique," we taught in this session (DEA Slides). As we can see, FDH is the most compact technology set. It attempts to create fewer outputs (number of patientdays funded by third-party sources and number of patientdays reimbursed privately) with more than 12 inputs (staffing labor and the cost of supplies). Firms normally prefer the FDH model, however it has several downsides owing to its assumptions. As we can see, all of the efficiencies in this model are one, but when compared to other models, it is not as efficient as we believe since we identify areas/units for improvement.VRS is bigger than FDH because it "fills in" the gaps left by FDH.We can observe that unit 6's efficiency can be improved.The charts show that DRS and IRS are greater than VRS. DRS attempts to expand the set for lower input values, whereas the IRS attempts to expand the technology for higher input values. DRS suggests that units 5 and 6 might increase their efficiency, while IRS suggests that facility 6 could as well.CRS is the most extensive technology set, so we can examine whether there is any viable combination to scale up or down. Units 5 and 6 require improvement based on the efficiency numbers.Based on the arrow network we reviewed in class, FRH is larger than FDH but less than CRS.