BA64018_Assignment4

Ruthvick Bulagakula

2023-10-29

The Hope Valley Health Care Association owns and operates six nursing homes in neighboring states. Their efficiency has been assessed using two inputs and two outputs. Staffing labor (measured in average hours per day) and supply costs (in thousands of dollars per day) are the inputs. The outputs are the number of patient days reimbursed by third-party sources and the number of patient days reimbursed privately. The table below contains a summary of performance data:

Calling Benchmarking library

```
library(Benchmarking)
```

```
## Loading required package: lpSolveAPI
```

```
## Loading required package: ucminf
```

```
## Loading required package: quadprog
```

Creating data frame

```
DMU = c("Facility1", "Facility2", "Facility3", "Facility4", "Facility5", "Facility6")

StaffHoursPerDay = c(100,300,320,500,350,340)

SuppliesPerDay = c(0.3,0.6,1.2,2,1.4,0.7)

ReimbursedPatientDays = c(15000, 15000, 40000, 28000, 20000, 14000)

PrivatelyPaidPatientDays = c(3500,20000,11000,42000,25000,15000)

data = data.frame(DMU, StaffHoursPerDay, SuppliesPerDay, ReimbursedPatientDays, PrivatelyPaidPatientDays

data
```

```
##         DMU StaffHoursPerDay SuppliesPerDay ReimbursedPatientDays
## 1 Facility1              100            0.3                 15000
## 2 Facility2              300            0.6                 15000
## 3 Facility3              320            1.2                 40000
## 4 Facility4              500            2.0                 28000
## 5 Facility5              350            1.4                 20000
## 6 Facility6              340            0.7                 14000
##   PrivatelyPaidPatientDays
## 1                     3500
## 2                    20000
## 3                    11000
## 4                    42000
## 5                    25000
## 6                    15000
```

Data interpretation

Hope Valley Health Care Association has six facilities, each with two inputs and two outputs. The inputs are the number of labor hours per day and the cost of supplies, while the outputs are the number of patients reimbursed by third parties and the number of patients reimbursed privately.

Matrices for input and output

```
input_ss = matrix(c(100,300,320,500,350,340,
                  0.3,0.6,1.2,2,1.4,0.7), byrow= FALSE, nrow =6)

colnames(input_ss) = c("StaffHoursPerDay", "SuppliesPerDay")

rownames(input_ss) = c("Facility1", "Facility2", "Facility3", "Facility4", "Facility5", "Facility6")

input_ss
```

```
##           StaffHoursPerDay SuppliesPerDay
## Facility1              100            0.3
## Facility2              300            0.6
## Facility3              320            1.2
## Facility4              500            2.0
## Facility5              350            1.4
## Facility6              340            0.7
```

```
output_rs = matrix(c(15000, 15000, 40000, 28000, 20000, 14000,
                   3500,20000,11000,42000,25000,15000), byrow=FALSE, nrow=6)

rownames(output_rs) = c("Facility1", "Facility2", "Facility3", "Facility4", "Facility5", "Facility6")

colnames(output_rs) = c("ReimbursedPatientPerDay", "PrivatelyPaidPatientDays")

output_rs
```

```
##           ReimbursedPatientPerDay PrivatelyPaidPatientDays
## Facility1                   15000                     3500
## Facility2                   15000                    20000
## Facility3                   40000                    11000
## Facility4                   28000                    42000
## Facility5                    20000                    25000
## Facility6                    14000                    15000
```

1. Create and run DEA analyses using all of the FDH, CRS, VRS,IRS, DRS, and FRH DEA assumptions.
2. Calculate the Peers and Lambdas under each of the preceding assumptions.
3. Create a table to summarize your findings.
4. Compare and contrast the above results

Calculate the efficiency for all the facilities using CRS assumption

CRS Efficiency

```
effi = dea(input_ss, output_rs, RTS = "CRS")

effi
```

```
## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6
##    1.0000    1.0000    0.8793    1.0000    0.8942    0.7048
```

The efficiency of a DMU can be defined as the ratio between its actual and optimal performance as shown by the efficiency frontier. The efficiency score ranges from 0 to 1, with 1 representing optimal efficiency. The CRS assumption (proportional inputs to outputs) is used in the DEA analysis. It calculates efficiency scores for DMUs while accounting for changes in input and output levels. The CRS assumption yields non-unity efficiency scores, which reflect relative efficiency.

F3, F5, and F6 have lower efficiency than the other facilities.F3 has an efficiency of 87.93, F5 has an efficiency of 89.42, and F6 has an efficiency of 70.48, whereas F1, F2, and F4 have 1 efficiency.

1st Assumption: FDH

The term free disposability hull refers to the ability to discard unnecessary or unwanted inputs and outputs. That is if the facility uses more inputs to produce less output or produces the same amount of output with more input, it is said to dispose of excess inputs or outputs.

FDH Efficiency

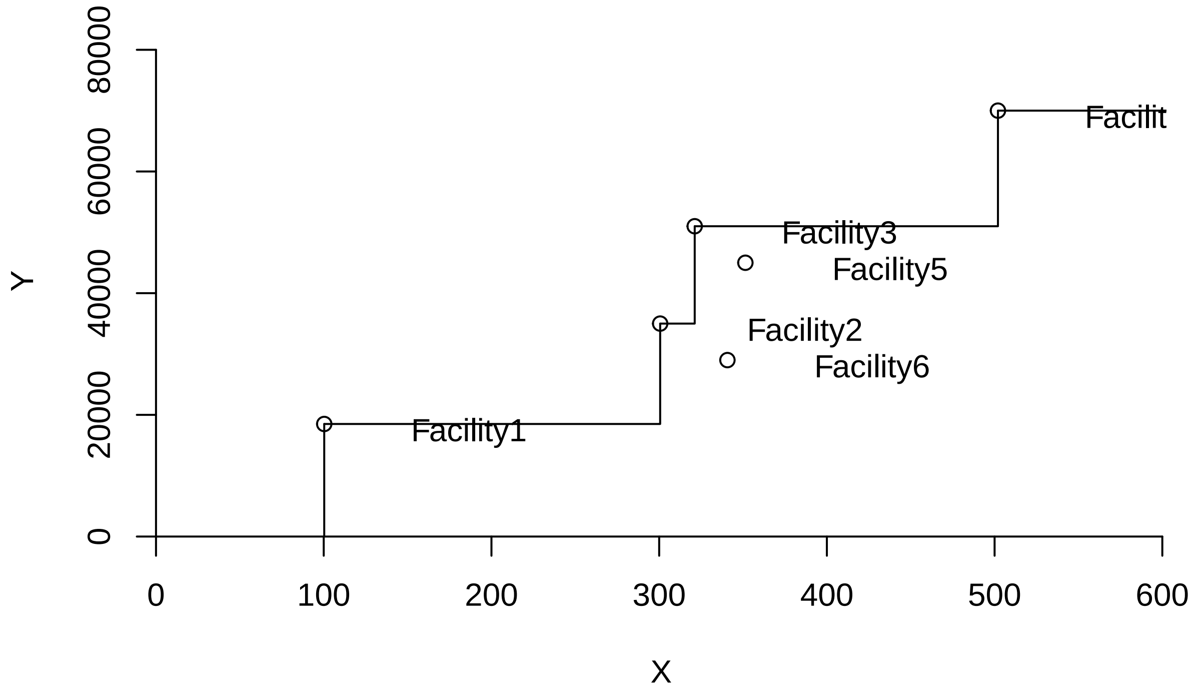
```
FreeDH = dea(input_ss, output_rs, RTS ="FDH" )

FreeDH
```

```
## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8824
```

According to FDH assumptions, facility 6 is 88.24 efficient, while all other facilities are 1 efficient.

FDH Plot

```
dea.plot(input_ss, output_rs,RTS="FDH",ORIENTATION="in-out",txt=rownames(input_ss))
```



FDH Peers

Peers are a group of efficient DMUs that are used to evaluate inefficient DMUs. That is, efficient DMUs are regarded as a model or benchmark for other DMUs.

```
peers(FreeDH)
```

```
##           peer1
## Facility1     1
## Facility2     2
## Facility3     3
## Facility4     4
## Facility5     5
## Facility6     2
```

Facility1, Facility2, Facility3, Facility4, and Facility5 all have their own facilities as peers, implying that all five facilities are efficient, whereas Facility6 has Facility2 as a peer, implying that Facility2 is a benchmark for Facility6.

FDH Lambda

The Lambda value represents how much a less efficient facility can learn from its peers.

```
lambda(FreeDH)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1           1            0            0            0            0
## Facility2           0            1            0            0            0
## Facility3           0            0            1            0            0
## Facility4           0            0            0            1            0
## Facility5           0            0            0            0            1
## Facility6           0            1            0            0            0
```

Facility6 is the only less efficient facility here, and it only has one peer, Facility2. Facility6 is 88.24% efficient, but it falls short by 11.76%. In this case, the lambda value of Facility6 is 1, implying that Facility2 can compensate for Facility6's 11.76% inefficiency.

According to FDH, there are 5 efficient facilities (Facility1, Facility2, Facility3, Facility4, and Facility5) and 1 inefficient facility (Facility6), with only Facility6 requiring a peer, which is Facility2. Because it has only one peer, it is possible that that peer can reduce Facility6's total lack of efficiency (11.76%).

2nd Assumption: CRS

Constant returns to scale assume that if the input increases by one unit, the output will increase by the same number of units.

CRS Efficiency

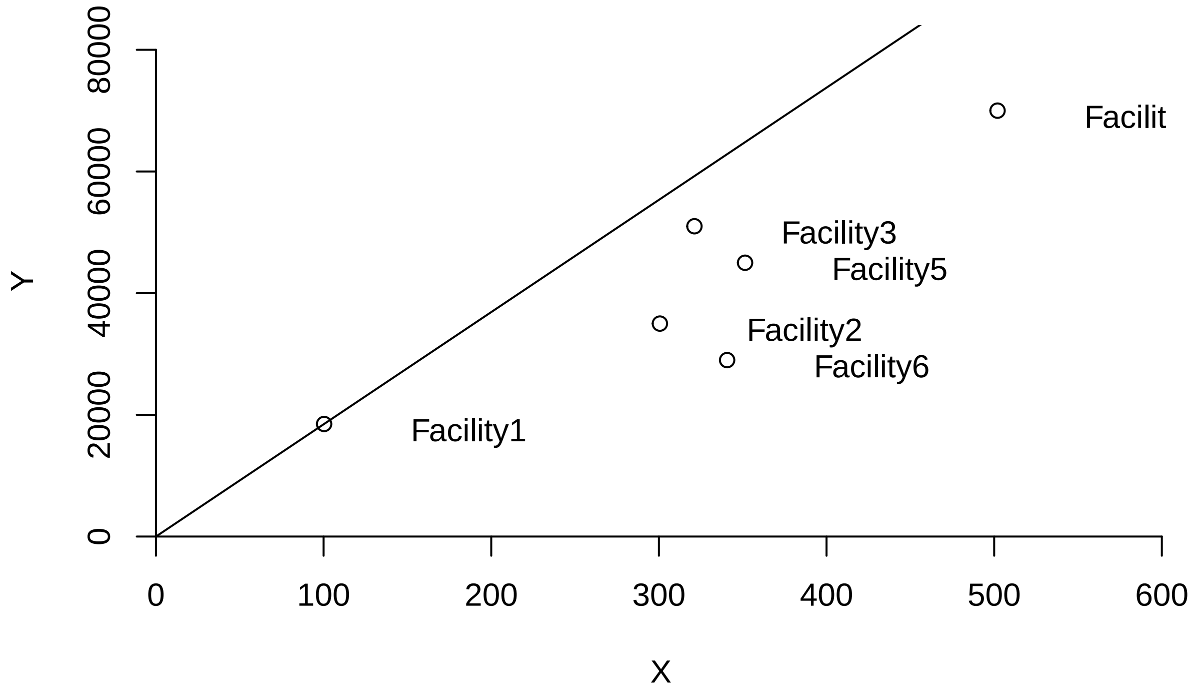
```
ConstantRS = dea(input_ss, output_rs,RTS="CRS")

ConstantRS
```

```
## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6
##    1.0000    1.0000    0.8793    1.0000    0.8942    0.7048
```

We can see that Facility1, Facility2, and Facility4 are efficient, with a value of 1 indicating that they are 100% efficient, whereas Facility3, Facility5, and Facility6 are not. Facility3 has an efficiency of 87.93%, Facility5 has an efficiency of 89.42%, and Facility6 has an efficiency of 70.48%.

CRS Plot

```
dea.plot(input_ss, output_rs,RTS="CRS",ORIENTATION="in-out",txt=rownames(input_ss))
```



CRS Peers

```
peers(ConstantRS)
```

```
##           peer1 peer2
## Facility1     1    NA
## Facility2     2    NA
## Facility3     1     4
## Facility4     4    NA
## Facility5     1     4
## Facility6     1     2
```

Only Facility3, Facility5, and Facility6 have peers here. Facility3 and Facility5 have two peers, Facility1 and Facility4, while Facility6 has two peers, but they are Facility1 and Facility2.

CRS Lambda

```
lambda(ConstantRS)
```

```
##           L_Facility1 L_Facility2 L_Facility4
## Facility1   1.0000000   0.0000000  0.00000000
## Facility2   0.0000000   1.0000000  0.00000000
## Facility3   2.5789474   0.0000000  0.04699248
## Facility4   0.0000000   0.0000000  1.00000000
## Facility5   0.2631579   0.0000000  0.57330827
## Facility6   0.2222222   0.7111111  0.00000000
```

Facility3 has two peers with weights of 2.57 of Facility1, 0.04 of Facility4, 0.26 of Facility1, 0.57 of Facility4, and Facility6 has two peers with weights of 0.22 of Facility1, 0.7 of Facility2.

summary:

3rd Assumption: DRS

Decreasing returns to scale assumes that increasing the input increases the output by less than the amount increased in input.

DRS Efficiency

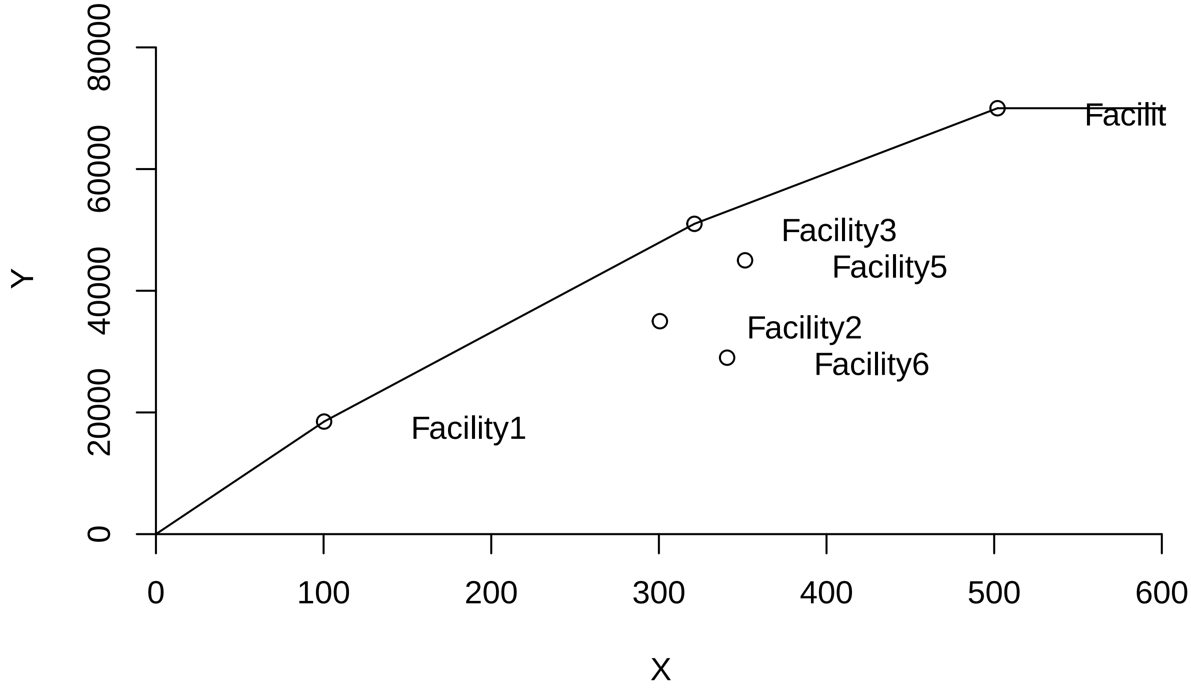
```
DecreasingRS = dea(input_ss, output_rs, RTS = "DRS")

DecreasingRS
```

```
## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6
##    1.0000    1.0000    1.0000    1.0000    0.8942    0.7048
```

Except for Facility5 and Facility6, all other facilities in the DRS assumption are efficient. Facility5 and Facility6 are both 89.42% and 70.48% efficient.

DRS Plot

```
dea.plot(input_ss, output_rs, RTS = "DRS", ORIENTATION = "in-out", txt = rownames(input_ss))
```



DRS Peers

```
peers(DecreasingRS)
```

```
##           peer1 peer2
## Facility1     1    NA
## Facility2     2    NA
## Facility3     3    NA
## Facility4     4    NA
## Facility5     1     4
## Facility6     1     2
```

Facility5 and Facility6 require peers because they are less efficient. Facility5's inefficiency is reduced by Facility1, Facility4, and Facility6's inefficiency is reduced by Facility1, Facility2.

DRS Lambda

```
lambda(DecreasingRS)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4
## Facility1   1.0000000   0.0000000           0   0.0000000
## Facility2   0.0000000   1.0000000           0   0.0000000
## Facility3   0.0000000   0.0000000           1   0.0000000
## Facility4   0.0000000   0.0000000           0   1.0000000
## Facility5   0.2631579   0.0000000           0   0.5733083
## Facility6   0.2222222   0.7111111           0   0.0000000
```

Facility1 and Facility4 reduce inefficiency by 0.26 and 0.57, respectively, while Facility6 reduces inefficiency by 0.22 and 0.71.

4th Assumption: IRS

Increasing returns to scale assumes that increasing the input increases the output by a proportion greater than the increase in inputs.

IRS Efficiency

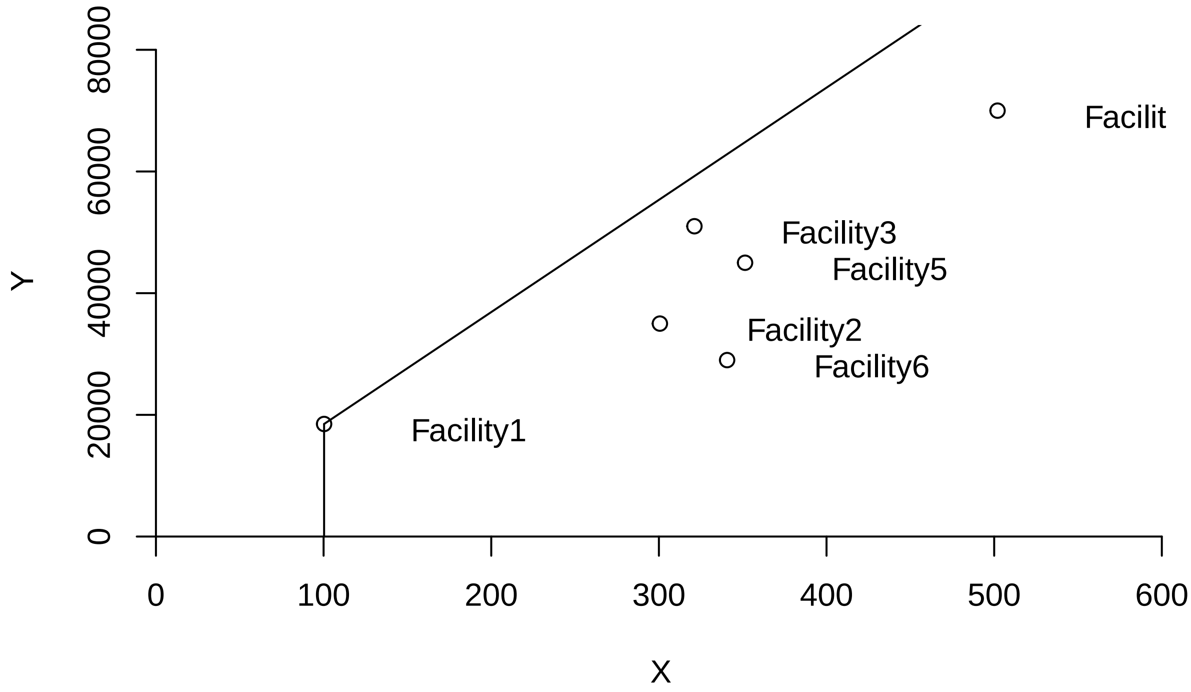
```
IncreasingRS = dea(input_ss, output_rs, RTS = "IRS")

IncreasingRS
```

```
## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6
##    1.0000    1.0000    0.8793    1.0000    0.9239    0.7273
```

According to IRS assumptions, Facility1, Facility2, and Facility4 are 100% efficient, while Facility3 is 87.93% efficient, Facility5 is 92.39% efficient, and Facility6 is 72.73% efficient.

IRS Plot

```
dea.plot(input_ss, output_rs, RTS = "IRS", ORIENTATION = "in-out", txt = rownames(input_ss))
```



IRS Peers

```
peers(IncreasingRS)
```

```
##           peer1 peer2
## Facility1     1    NA
## Facility2     2    NA
## Facility3     1     4
## Facility4     4    NA
## Facility5     1     4
## Facility6     1     2
```

Facility3's peers are 1,4; Facility5's peers are 1,4; and Facility6's peers are Facility1,Facility6; this means that Facility3,Facility5, and Facility6 are inefficient and will learn from their peers.

IRS Lambda

```
lambda(IncreasingRS)
```

```
##           L_Facility1 L_Facility2 L_Facility4
## Facility1   1.0000000   0.0000000  0.00000000
## Facility2   0.0000000   1.0000000  0.00000000
## Facility3   2.5789474   0.0000000  0.04699248
## Facility4   0.0000000   0.0000000  1.00000000
## Facility5   0.4415584   0.0000000  0.55844156
## Facility6   0.3030303   0.6969697  0.00000000
```

Facility3 has two peers with weights of 2.57 from Facility1, 0.04 from Facility4, Facility5 has two peers with weights of 0.44 from Facility1, 0.55 from Facility4, and Facility6 has two peers with weights of 0.3 and 0.69.

5th Assumption: VRS

VRS assumes constant returns to scale (CRS), while allowing for the possibility that some DMUs may operate with increasing returns to scale (IRS) while others with decreasing returns to scale (DRS).

VRS Efficiency

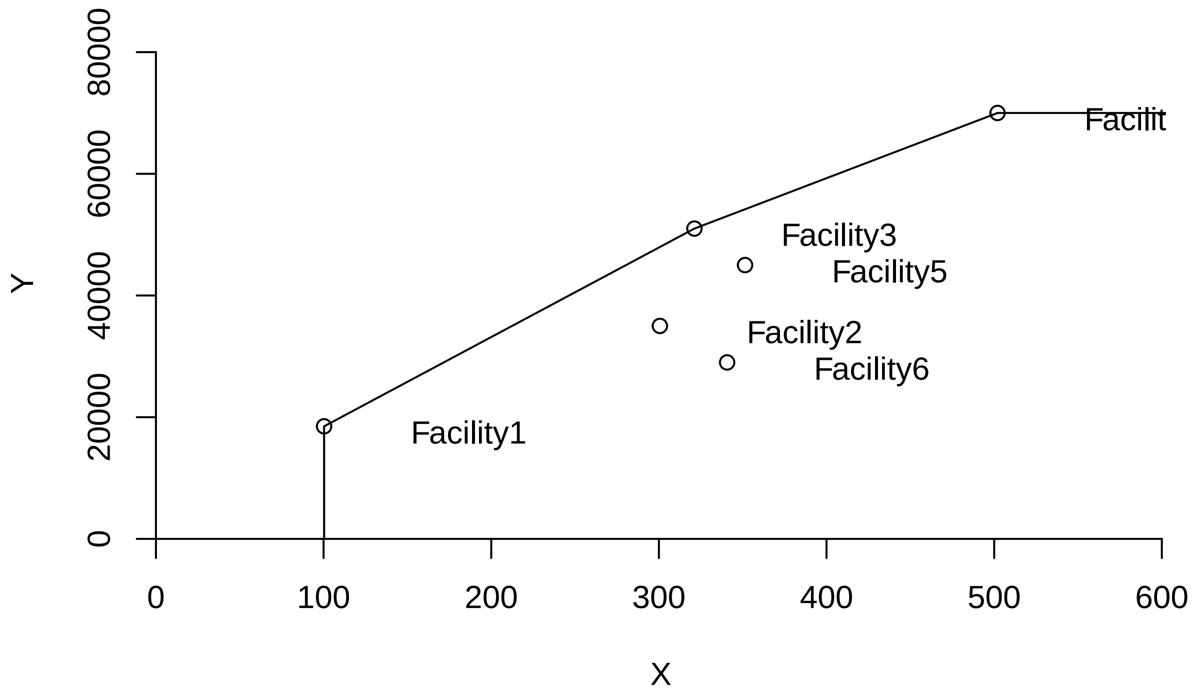
```
VariableRS = dea(input_ss, output_rs, RTS = "VRS")

VariableRS
```

```
## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6
##    1.0000    1.0000    1.0000    1.0000    0.9239    0.7273
```

Only Facility5 and Facility6 are less efficient, with 92.39% and 72.73% efficiency, respectively, whereas the others are 100% efficient.

VRS Plot

```
dea.plot(input_ss, output_rs, RTS = "VRS", ORIENTATION = "in-out", txt=rownames(input_ss))
```



VRS Peers

```r
peers(VariableRS)
```

```
##           peer1 peer2
## Facility1     1    NA
## Facility2     2    NA
## Facility3     3    NA
## Facility4     4    NA
## Facility5     1     4
## Facility6     1     2
```

As Facility5 and Facility6 are less efficient, they have Facility1,Facility4 as peers for Facility5 and Facility1, Facility2 as peers for Facility6.

VRS Lambda

```r
lambda(VariableRS)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4
## Facility1   1.0000000   0.0000000           0   0.0000000
## Facility2   0.0000000   1.0000000           0   0.0000000
## Facility3   0.0000000   0.0000000           1   0.0000000
## Facility4   0.0000000   0.0000000           0   1.0000000
## Facility5   0.4415584   0.0000000           0   0.5584416
## Facility6   0.3030303   0.6969697           0   0.0000000
```

Facility5 has a peer weight of 0.44 from Facility1, 0.55 from Facility4, and Facility6 has a peer weight of 0.3 from Facility1 and 0.69 from Facility2.

Sixth assumption: FRH

The full replicability hull provides a comprehensive understanding of the feasible and efficient set of inputs and outputs for the facilities (DMUs) under consideration.

FRH Efficiency

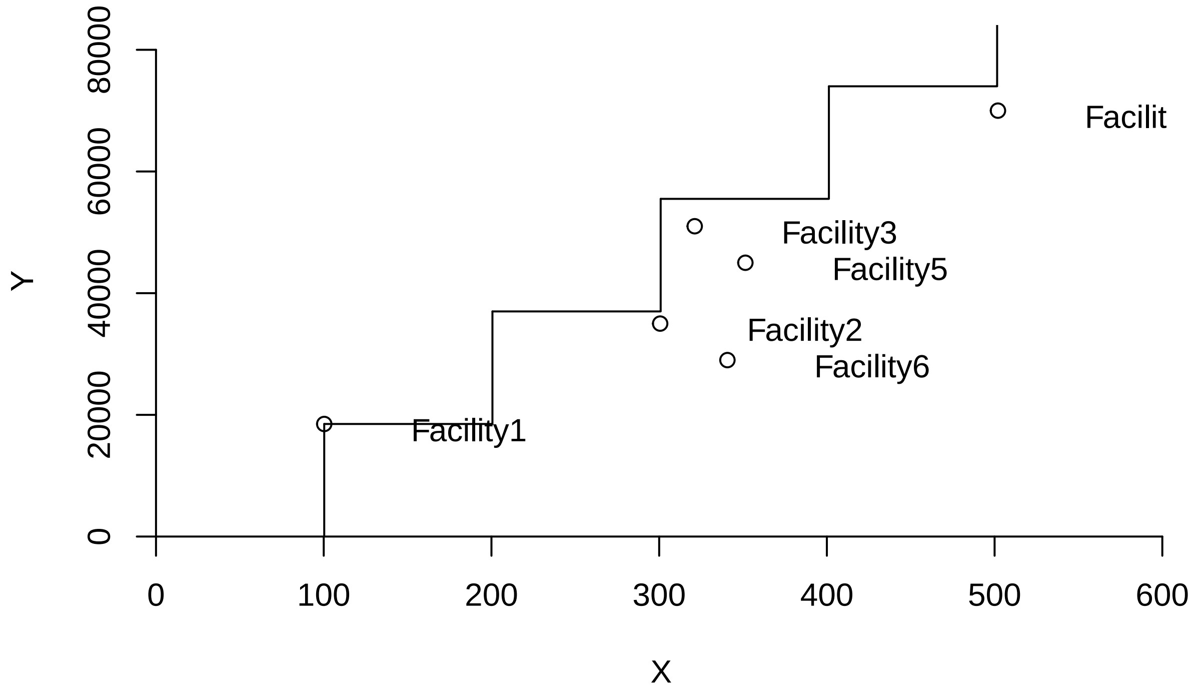
```r
FullRH = dea(input_ss, output_rs, RTS= "add")

FullRH
```

```
## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6
##    1.0000    1.0000    1.0000    1.0000    1.0000    0.8824
```

Except for Facility6, all other facilities are completely efficient. Only Facility6 is more efficient at 88.24%.

FRH Plot

```r
dea.plot(input_ss, output_rs, RTS = "add", ORIENTATION = "in-out", txt = rownames(input_ss))
```



FRH Peers

```
peers(FullRH)
```

```
##           peer1
## Facility1     1
## Facility2     2
## Facility3     3
## Facility4     4
## Facility5     5
## Facility6     2
```

Facility1, Facility2, Facility3, Facility4, and Facility5 all have their own facilities as peers, implying that all five facilities are efficient, whereas Facility6 has Facility2 as a peer, implying that Facility2 is a benchmark for Facility6.

FRH Lambda

```
lambda(FullRH)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1           1           0           0           0           0
## Facility2           0           1           0           0           0
## Facility3           0           0           1           0           0
## Facility4           0           0           0           1           0
## Facility5           0           0           0           0           1
## Facility6           0           1           0           0           0
```

Facility6 is the only less efficient facility here, and it only has one peer, Facility2. Facility6 is 0.88 efficient, but it falls short by 1.1. In this case, the lambda value of Facility6 is 1, implying that Facility2 can compensate for Facility6's 1.1 inefficiency.

```
new_data = data.frame(DMU= c("Facility1", "Facility2", "Facility3", "Facility4", "Facility5", "Facility
                       e=c(1.0000, 1.0000, 0.8793, 1.0000 ,0.8942 ,0.7048 ),
                       FDH=c(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 0.8824),
                       CRS=c(1.0000, 1.0000, 0.8793, 1.0000, 0.8942, 0.7048),
                       DRS=c(1.0000, 1.0000, 1.0000, 1.0000, 0.8942, 0.7048),
                       IRS=c(1.0000, 1.0000, 0.8793, 1.0000, 0.9239, 0.7273),
                       VRS=c(1.0000, 1.0000, 1.0000, 1.0000, 0.9239, 0.7273),
                       FRH=c(1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 0.8824)
                       )
new_data
```

```
##         DMU      e    FDH    CRS    DRS    IRS    VRS    FRH
## 1 Facility1 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
## 2 Facility2 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
## 3 Facility3 0.8793 1.0000 0.8793 1.0000 0.8793 1.0000 1.0000
## 4 Facility4 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000
## 5 Facility5 0.8942 1.0000 0.8942 0.8942 0.9239 0.9239 1.0000
## 6 Facility6 0.7048 0.8824 0.7048 0.7048 0.7273 0.7273 0.8824
```

```
peers = data.frame(DMU= c("Facility1", "Facility2", "Facility3", "Facility4", "Facility5", "Facility6")
                   peers_FDH=c("Facility1", "Facility2", "Facility3", "Facility4", "Facility5", "Facilit
                   L_FDH=c(0,0,0,0,0,1),
                   peers_CRS=c("Facility1", "Facility2", "Facility1, Facility4", "Facility4", "Facility
                   L_CRS=c(0,0,"2.57, 0.04",0, "0.26, 0.57","0.22, 0.71"),
                   peers_DRS=c("Facility1", "Facility2", "Facility3", "Facility4","Facility1, Facility4
                   L_DRS=c(0,0,0,0,"0.26, 0.57", "0.22, 0.71"),
                   peers_IRS=c("Facility1", "Facility2","Facility1, Facility4","Facility4", "Facility1,
                   L_IRS=c(0,0,"2.57, 0.04",0,"0.44, 0.55","0.30,0.69"),
                   peers_VRS=c("Facility1", "Facility2", "Facility3", "Facility4","Facility1, Facility4
                   L_VRS=c(0,0,0,0,"0.44, 0.55", "0.30, 0.69"),
                   peers_FRH=c("Facility1", "Facility2", "Facility3", "Facility4","Facility5","Facility
                   L_FRH=c(0,0,0,0,0,1))
peers
```

```
##         DMU peers_FDH L_FDH            peers_CRS      L_CRS
## 1 Facility1 Facility1     0            Facility1          0
## 2 Facility2 Facility2     0            Facility2          0
## 3 Facility3 Facility3     0 Facility1, Facility4 2.57, 0.04
## 4 Facility4 Facility4     0            Facility4          0
## 5 Facility5 Facility5     0 Facility1, Facility4 0.26, 0.57
## 6 Facility6 Facility2     1 Facility1, Facility2 0.22, 0.71
##              peers_DRS      L_DRS            peers_IRS      L_IRS
## 1            Facility1          0            Facility1          0
## 2            Facility2          0            Facility2          0
## 3            Facility3          0 Facility1, Facility4 2.57, 0.04
## 4            Facility4          0            Facility4          0
## 5 Facility1, Facility4 0.26, 0.57 Facility1, Facility4 0.44, 0.55
## 6 Facility1, Facility2 0.22, 0.71 Facility1, Facility2  0.30,0.69
##              peers_VRS      L_VRS peers_FRH L_FRH
## 1            Facility1          0 Facility1     0
## 2            Facility2          0 Facility2     0
## 3            Facility3          0 Facility3     0
## 4            Facility4          0 Facility4     0
## 5 Facility1, Facility4 0.44, 0.55 Facility5     0
## 6 Facility1, Facility2 0.30, 0.69 Facility2     1
```

```
lambda_inefficient = data.frame(DMU= c("Facility1", "Facility2", "Facility3", "Facility4", "Facility5",
                                    L_FDH=c(0,0,0,0,0,1),
                                    L_CRS=c(0,0,"2.57, 0.04",0, "0.26, 0.57","0.22, 0.71"),
                                    L_DRS=c(0,0,0,0,"0.26, 0.57", "0.22, 0.71"),
                                    L_IRS=c(0,0,"2.57, 0.04",0,"0.44, 0.55","0.30,0.69"),
                                    L_VRS=c(0,0,0,0,"0.44, 0.55", "0.30, 0.69"),
                                    L_FRH=c(0,0,0,0,0,1))

lambda_inefficient
```

```
##          DMU L_FDH      L_CRS      L_DRS      L_IRS      L_VRS L_FRH
## 1 Facility1     0          0          0          0          0     0
## 2 Facility2     0          0          0          0          0     0
## 3 Facility3     0 2.57, 0.04          0 2.57, 0.04          0     0
## 4 Facility4     0          0          0          0          0     0
## 5 Facility5     0 0.26, 0.57 0.26, 0.57 0.44, 0.55 0.44, 0.55     0
## 6 Facility6     1 0.22, 0.71 0.22, 0.71  0.30,0.69 0.30, 0.69     1
```

Compare and constrast the results

Summary:

Full disposal hull (FDH)

FDH is used to assess the efficiency of decision-making units.There are five efficient facilities (F1, F2, F3, F4, and F5) and one less efficient unit (F6). In this context, “efficient” means that these facilities are operating at peak efficiency and maximizing resource utilization.F6 has an efficiency of 88.24%. Because F6 only has one peer, F2, it is possible that F2 can assist F6 in improving its efficiency.The statement implies that if F2 assists F6, it may be able to close the 11.76% efficiency gap of F6.

Constant returns to scale (CRS)

The returns to scale in CRS are constant, which means that the output increases in proportion to the increase in input.

The DMUs F1, F2, and F4 are all 100% efficient, whereas F3, F5, and F6 are less efficient, with F3 being 87.93% efficient, F5 being 89.42% efficient, and F6 being 70.48% efficient. F3, F5, AND F6 have peers because they are less efficient. F3 has two peers, F1 and F4, with weights of 2.57 and 0.04 respectively. F5 has two peers with weights of 0.26 and 0.57 from F1 and F4, respectively. F6 also has two peers, F1 and F2, with weights of 0.22 and 0.71, respectively.

Here, DMU F1, F2 and F4 are highly efficient, so they can be scaled up or down.

Decreasing returns to scale (DRS)

DRS considers scale effects on efficiency and recognizes that efficiency may change as a decision-making unit scales its operations. The output will be increased by a smaller proportion than the input.

Except for F5 and F6, all of the other facilities (DMUs) are efficient, with a value of 1. F5 has an efficiency of 89.42% and F6 has an efficiency of 70.48%. F5’s peers are F1 and F4 with weights of 0.26 and 0.57, respectively, and F6’s peers are F1 and F2 with weights of 0.22 and 0.71.

We can conclude from the results that the inefficient facilities can be determined if there are any alternate DMUS to scale the process.

Increasing returns to scale (IRS)

The IRS assumes that as a facility or decision-making unit (DMU) grows in size, it becomes more efficient, and efficiency is positively correlated with scale.

The results show that F1, F2, and F4 are productive, while F3, F5, and F6 are less productive, yielding 87.93%, 92.39% and 72.73% respectively. Each of the less productive facilities has two peers. F3 has F1 and F4 with weights of 2.57 and 0.04 respectively, F5 has F1 and F4 with weights of 0.44 and 0.55, and F6 has F1 and F2 with weights of 0.3 and 0.69.

The weighting of peers suggests that these less productive facilities may be able to learn and improve their efficiency by benchmarking against more productive peers.

Varying returns to scale (VRS)

It is recognized in the VRS approach that the efficiency of a facility or decision-making unit (DMU) may vary with its scale of operations, that is, it may increase or decrease.

Only F5 92.39% and F6 72.73% are less efficient in VRS than F1, F2, F3, and F4. Each facility that is less efficient has two peers. F1 and F4 have weights of 0.44 and 0.55, respectively, and are peers of F5. F1 and F2 are F6's peers, with weights of 0.3 and 0.69, respectively.

Variable returns to scale allow the scale of operations to change in response to changes in the input and output components by increasing or decreasing.

Free Replicability hull (FRH)

The FRH results show that all facilities are effective except F6, which has only one peer, F2. The weight of the F6's peer is one, indicating that the only peer is responsible for improving the inefficient unit.

Conclusion

F1, F2, and F4 are efficient because all of the assumptions result in a value of 1 that is 100%, whereas F3, F5, and F6 have a varying value with each assumption. The results show that certain facilities consistently maintain high efficiency levels, whereas other facilities exhibit varying efficiency values based on the specific models or assumptions used in the investigation. This implies that various circumstances or events may have an impact on the efficiency of the latter group, resulting in variations in their evaluated performance.

DEA is an effective tool for benchmarking and performance evaluation because it allows entities and organizations to assess their effectiveness and identify areas for improvement. It is commonly used in performance appraisals, efficiency analyses, and strategic decision-making. DEA provides decision-makers with knowledge on how to allocate resources, increase output, and optimize operations by focusing on underperforming regions.