Assig 04-AMM

Chathurani Ekanayake

2023-10-28

import required packages

```
#install.packages("Benchmarking")
library(Benchmarking)
```

```
## Warning: package 'Benchmarking' was built under R version 4.2.3

## Loading required package: lpSolveAPI

## Warning: package 'lpSolveAPI' was built under R version 4.2.3

## Loading required package: ucminf

## Warning: package 'ucminf' was built under R version 4.2.3

## Loading required package: quadprog
```

```
if(!require(knitr)){
  library(knitr)
}
```

```
## Loading required package: knitr
```

```
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.2.3
```

Data

```
df <- data.frame(
  DMU = c("Facility 1", "Facility 2", "Facility 3", "Facility 4", "Facility 5", "Facility 6" ),
  Staff_hours_per_day = c(100,300,320,500,350,340),
  Supplies_per_day = c(0.3,0.6,1.2,2,1.4,0.7),
  Reimbursed_patient_days = c(15000,15000,40000,28000,20000,14000),
  Privetly_paid_patient_days = c(3500,20000,11000,42000,25000,15000)
)
```

```
kable(df,format = "pandoc",caption = "Hope Valley Health Care Association")
```

Table 1: Hope Valley Health Care Association

DMU	Staff_hours_per_day	Supplies_per_day	Reimbursed_patient_days	Privetly_paid_patient_days
Facility 1	100	0.3	15000	3500
Facility 2	300	0.6	15000	20000
Facility 3	320	1.2	40000	11000
Facility 4	500	2.0	28000	42000
Facility 5	350	1.4	20000	25000
Facility 6	340	0.7	14000	15000

create the matrix with given data

x- contains input data (Supplies per day,Staff hours per day) y- contains output data (Reimbursed patient days, Privetly paid patient days)

```
x <- matrix(c(100,300,320,500,350,340,
             0.3,0.6,1.2,2,1.4,0.7), ncol = 2)
y <- matrix(c(15000,15000,40000,28000,20000,14000,
              3500,20000,11000,42000,25000,15000), ncol = 2)
colnames(x) <- c("Supplies_per_day","Staff_hours_per_day")
colnames(y) <- c("Reimbursed_patient_days", "Privetly_paid_patient_days")
```

Questions 1. Formulate and perform DEA analysis under all DEA assumptions of FDH, CRS, VRS, IRS, DRS, and FRH. 2. Determine the Peers and Lambdas under each of the above assumptions 3. Summarize your results in a tabular format 4. Compare and contrast the above result

DEA Analysis using FDH

Now, we are going to formulate and compute the DEA analysis under the assumption of FDH. The Free Disposability Hull (FDH) is a way to measure how efficient a company is at producing outputs with its inputs. It assumes that a company can always get rid of unwanted inputs and outputs without cost.

input and output values

```
fdh <- dea(x,y,RTS = "fdh")
```

calculate efficiency and name the column

```
eff_fdh <- as.data.frame(fdh$eff)
colnames(eff_fdh)<- c("efficiency_fdh")
eff_fdh
```

```
##   efficiency_fdh
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
## 5      1.0000000
## 6      0.8823529
```

According to the above results, under the FDH assumption, Facilities 1 to 5 are operating at full efficiency, while Facility 6 is slightly less efficient (0.8823529) in its resource utilization. An efficiency score of 1.0000000 represents full efficiency, and scores lower than 1 indicate some level of inefficiency. Facility 6 has room for improvement in its input-output efficiency compared to the other facilities.

In the case of the sixth nursing home, it is possible that the nursing home could reduce its staffing hours or the cost of supplies without sacrificing quality. For example, the nursing home could implement new efficiency measures or negotiate better prices with suppliers.

Find the peers

```
peer_fdh<-peers(fdh)
colnames(peer_fdh)<- c("peer_fdh")
peer_fdh
```

```
##      peer_fdh
## [1,]        1
## [2,]        2
## [3,]        3
## [4,]        4
## [5,]        5
## [6,]        2
```

According to the results of peer column, Facility 1 is compared to Facility 1 itself, and its peer index is 1. Facility 2 is compared to Facility 2 itself, and its peer index is 2. Facility 3 is compared to Facility 3 itself, and its peer index is 3. Facility 4 is compared to Facility 4 itself, and its peer index is 4. Facility 5 is compared to Facility 5 itself, and its peer index is 5. Facility 6 is compared to Facility 2, and its peer index is 2. The peer index of 1 for Facilities 1, 2, 3, 4, and 5 means that they are considered efficient (i.e., they are their own peers), while Facility 6 is compared to Facility 2 as its peer. Facility 6 may not be as efficient as the others, as it's not considered its own peer under the FDH assumption.

Find the lambda

```
lambda_fdh <- lambda(fdh)
lambda_fdh
```

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

According to the above results, L1, L2, L3, L4, L5: These columns represent the facilities (Facility 1 to Facility 5) that serve as peers for each of the six facilities, including Facility 6.

Facility 1,2,3,4 and 5 are 100% efficient, and its efficiency score is calculated by comparing it to itself.Facility 6 is less efficient than the other facilities, and its efficiency score is calculated by comparing it to Facility 2. This means that Facility 6 could improve its efficiency by using its inputs and outputs more efficiently, or by finding ways to reduce its inputs or increase its outputs.

Tabular data for FDH

```r
tabular_fdh <- cbind(peer_fdh,lambda_fdh,eff_fdh)
tabular_fdh
```

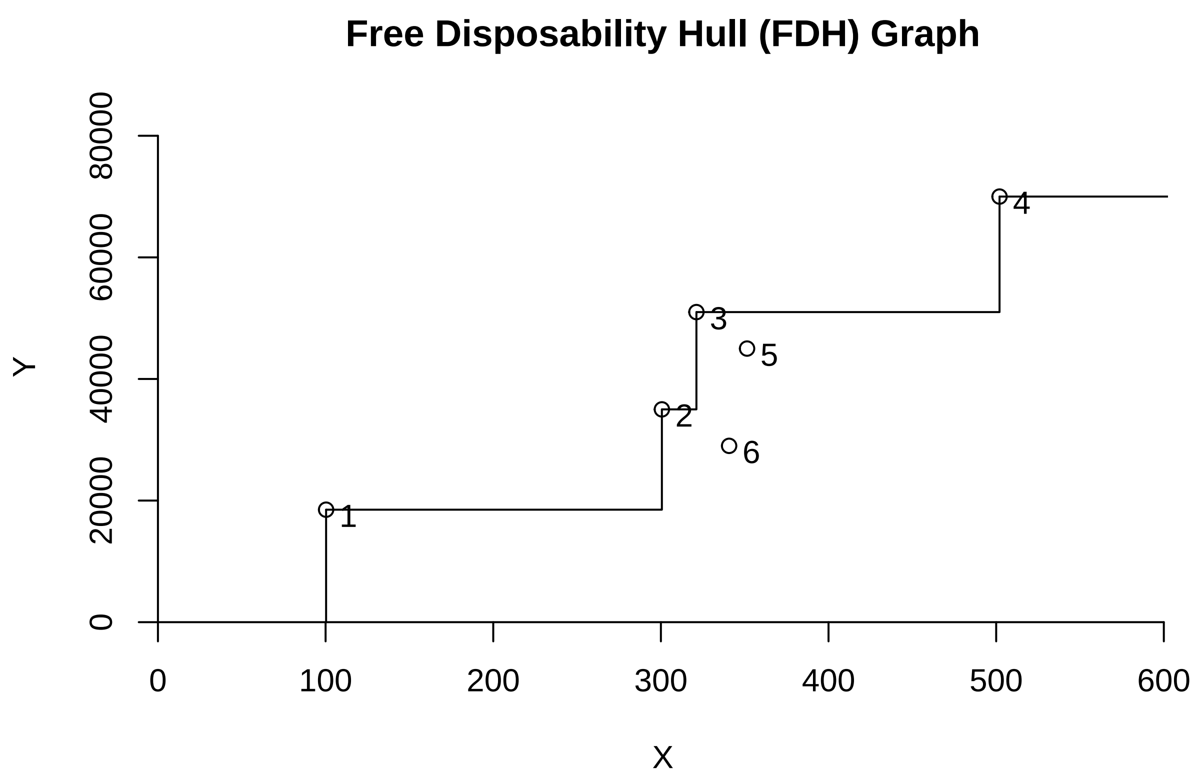
```
##   peer_fdh L1 L2 L3 L4 L5 efficiency_fdh
## 1        1  1  0  0  0  0      1.0000000
## 2        2  0  1  0  0  0      1.0000000
## 3        3  0  0  1  0  0      1.0000000
## 4        4  0  0  0  1  0      1.0000000
## 5        5  0  0  0  0  1      1.0000000
## 6        2  0  1  0  0  0      0.8823529
```

Interpretation of lambda

Lambda values in DEA indicate how much a DMU (Decision Making Unit) borrows or learns from its efficient peers. Larger lambda values mean a DMU learns more from its peers to become efficient.According to the tabular data, only facility 6 is inefficient. So inefficiency of DMU_6= $1 - 0.88 = 0.12$. DMU_6 can reduce this 12% inefficiency learning from its peer of DMU_2.

Plot the result

```r
dea.plot(x,y,RTS = "fdh", ORIENTATION = "in-out", txt = TRUE, main= "Free Disposability Hull (FDH) Grap
```



DEA Analysis using CRS

Now, we will formulate and compute the DEA analysis using the Constant Returns to Scale (CRS) assumption. CRS is a part of the scaling assumption, and it helps us determine if there are any feasible combinations for scaling up or down, without changing the overall efficiency

input and output values

```
crs <- dea(x,y,RTS = "crs")
```

calculate efficiency and name the column

```
eff_crs <- as.data.frame(crs$eff)
colnames(eff_crs)<- c("efficiency_crs")
eff_crs
```

```
##   efficiency_crs
## 1      1.0000000
## 2      1.0000000
## 3      0.8793468
## 4      1.0000000
## 5      0.8941998
## 6      0.7047619
```

As per the above results, Facility 1, 2 and 4 are operating at optimal efficiency under the CRS assumption. Facilities 3, 5, and 6 have room for improvement in their resource utilization and output generation. You can use these efficiency scores to rank and compare the facilities in terms of their efficiency and identify which ones have the most significant efficiency gaps.

Find the peers

```
peer_crs<-peers(crs)
peer_crs
```

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

The above matrix represent potential peers for each of the six entities (or DMUs). DMU1,DMU2 and DMU4 are their own peers.Therefore those three DMUs can be considered as efficient. DMU3 and 5 have two peers for each of them,DMU1 and DMU4. DMU6 also has two peers, DMU1 and DMU2.Due to those three DMUs (DMU3,5 and 6) have two peers those are considered as inefficient.

Find the lambda

```
lambda_crs <- lambda(crs)
lambda_crs
```

```
##              L1        L2         L4
## [1,] 1.0000000 0.0000000 0.00000000
## [2,] 0.0000000 1.0000000 0.00000000
## [3,] 2.5789474 0.0000000 0.04699248
## [4,] 0.0000000 0.0000000 1.00000000
## [5,] 0.2631579 0.0000000 0.57330827
## [6,] 0.2222222 0.7111111 0.00000000
```

Rows 1 to 6 correspond to Facilities 1 to 6.

Column "L1" represents the lambda value for Facility 1. It is 1.0000000, indicating that Facility 1 can achieve efficiency by itself (it doesn't need to rely on other facilities).

Column "L2" represents the lambda value for Facility 2. Similar to Facility 1, it is 1.0000000, indicating that Facility 2 is also efficient on its own.

Column "L4" represents the lambda value for Facility 4. It is 1.0000000, indicating that Facility 4 can achieve efficiency independently.

For the other facilities (Facilities 3, 5, and 6):

Facility 3 has a lambda value of 2.5789474 for "L1" and 0.04699248 for "L4." This means that Facility 3 can achieve efficiency by a combination of itself (2.5789474 times) and Facility 1, as well as to a lesser extent by itself and Facility 4 (0.04699248 times).

Facility 5 has a lambda value of 0.2631579 for "L1" and 0.57330827 for "L4." Facility 5 can achieve efficiency by combining itself and Facility 1 (0.2631579 times) and itself and Facility 4 (0.57330827 times).

Facility 6 has a lambda value of 0.2222222 for "L1" and 0.7111111 for "L2." Facility 6 can achieve efficiency by combining itself and Facility 1 (0.2222222 times) and itself and Facility 2 (0.7111111 times).

Tabular data for CRS

```
tabular_crs <- cbind(peer_crs,lambda_crs,eff_crs)
tabular_crs
```

```
##   peer1 peer2        L1        L2         L4 efficiency_crs
## 1     1    NA 1.0000000 0.0000000 0.00000000      1.0000000
## 2     2    NA 0.0000000 1.0000000 0.00000000      1.0000000
## 3     1     4 2.5789474 0.0000000 0.04699248      0.8793468
## 4     4    NA 0.0000000 0.0000000 1.00000000      1.0000000
## 5     1     4 0.2631579 0.0000000 0.57330827      0.8941998
## 6     1     2 0.2222222 0.7111111 0.00000000      0.7047619
```

According to the tabular data of CRS, DMU_3 is learns from its peers, $DMU_1(L1)$ and $DMU_4(L4)$. The lambda value of (L1) and (L4) in this row indicate the degree to which DMU_3 learns from its peers. The maximum of this inefficiency of DMU_3 can learn from its peer DMU_2 because it's lambda value is more than 1. (it is 2.5789474) and the rest of the inefficiency can learn from DMU_4.

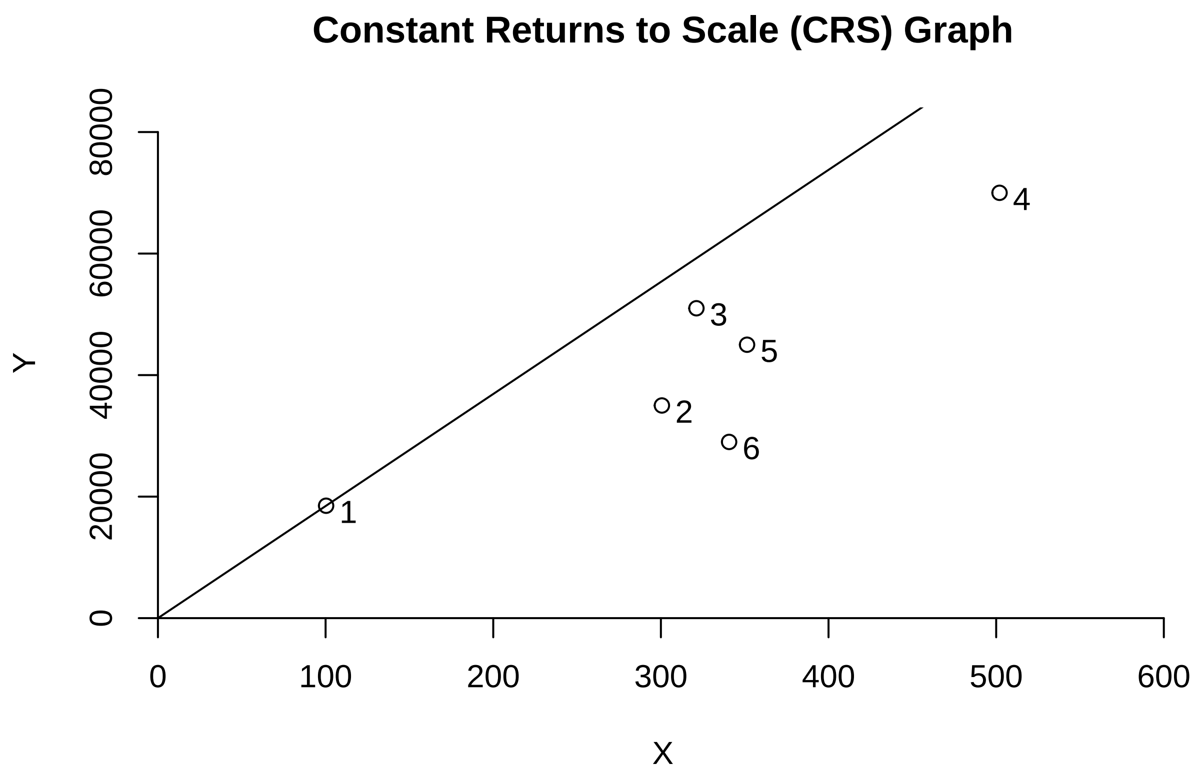
As in the above,for DMU_5) also inefficient and it is 1-(0.2631579+0.57330827)=0.16353383) So DMU_5 can reduce this 16.35% inefficiency learning from its peers $DMU_1(L1)$ and $DMU_4(L4)$. The ratio of $DMU_1(L1)$ to $DMU_4(L4)$ is 4:5. Hence that DMU_5 can learn 0.1058002*(4/9)=0.047022311= 4.7% from peer DMU_1 and 0.1058002*(5/9)=0.058777889=5.88%$ from peer DMU_4.

As well as DMU_6 is inefficient by 1-(0.2222222+0.7111111)= 0.0666667= 6.67%. DMU_6 can reduce this inefficiency learning from its peers $DMU_1(L1)$ and $DMU_2(L2)$.The ratio of $DMU_1(L1)$ to $DMU_2(L2)$ is 14:15. So DMU_6 learns 0.0666667 *(14/29)=0.032183924= 3.22% from peer $DMU_1(L1)$ and 0.0666667*(15/29)=0.034482776=3.45% from peer $DMU_2(L2)$.

All the other facilities, DMU_1, DMU_2 and DMU_4 are operated in full their full efficiency of 1.0000.

Plot the result

```
dea.plot(x,y,RTS = "crs", ORIENTATION = "in-out", txt = TRUE, main= "Constant Returns to Scale (CRS) Gra
```



DEA Analysis using VRS

We will now conduct a DEA analysis with Variable Returns to Scale (VRS). VRS, a key element of the scaling assumption, enables us to assess the efficiency of operations when changes in inputs and outputs are not proportionate.

input and output values

```
vrs <- dea(x,y,RTS = "vrs")
```

calculate efficiency and name the column

```
eff_vrs <- as.data.frame(vrs$eff)
colnames(eff_vrs)<- c("efficiency_vrs")
eff_vrs
```

```
##   efficiency_vrs
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
```

```
## 5      0.9239332
## 6      0.7272727
```

Facilities 1, 2, 3, and 4 all have an efficiency score of 1.000, meaning they are efficient under the VRS assumption. These facilities are getting the most out of their inputs, given their specific circumstances. Facility 5 has an efficiency score of approximately 0.924. This suggests that it is somewhat less efficient than the fully efficient facilities. It may have room for improvement in resource utilization. Facility 6 has an efficiency score of approximately 0.727. This facility is the least efficient among the group, indicating that it may need significant improvements in resource allocation and management.

Find the peers

```
peer_vrs<-peers(vrs)
peer_vrs
```

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

facility 1,2,3 and 4 are their own peers and no any secondary peer. But for facility 5 has two peers, facility 1 and 4.For facility 6 also has two peers as facility 1 and facility 2.

Find the lambda

```
lambda_vrs <- lambda(vrs)
lambda_vrs
```

```
##             L1        L2 L3        L4
## [1,] 1.0000000 0.0000000  0 0.0000000
## [2,] 0.0000000 1.0000000  0 0.0000000
## [3,] 0.0000000 0.0000000  1 0.0000000
## [4,] 0.0000000 0.0000000  0 1.0000000
## [5,] 0.4415584 0.0000000  0 0.5584416
## [6,] 0.3030303 0.6969697  0 0.0000000
```

As per the above table Facility 1,2,3 and 4 are efficient and indicate the efficiency as 1.000. But Facility 5 is efficient with a weight of 0.4415584. It relies on Facility 1 to achieve efficiency and weight of 0.5584416 and it relies on Facility 4 to achieve efficiency.

Facility 6 is efficient with a weight of 0.3030303. It relies on Facility 1 to achieve efficiency and it is efficient with a weight of 0.6969697. It relies on Facility 2 to achieve efficiency.

Tabular data for VRS

```
tabular_vrs <- cbind(peer_vrs,lambda_vrs,eff_vrs)
tabular_vrs
```

```
##   peer1 peer2        L1        L2 L3        L4 efficiency_vrs
## 1     1    NA 1.0000000 0.0000000  0 0.0000000      1.0000000
```

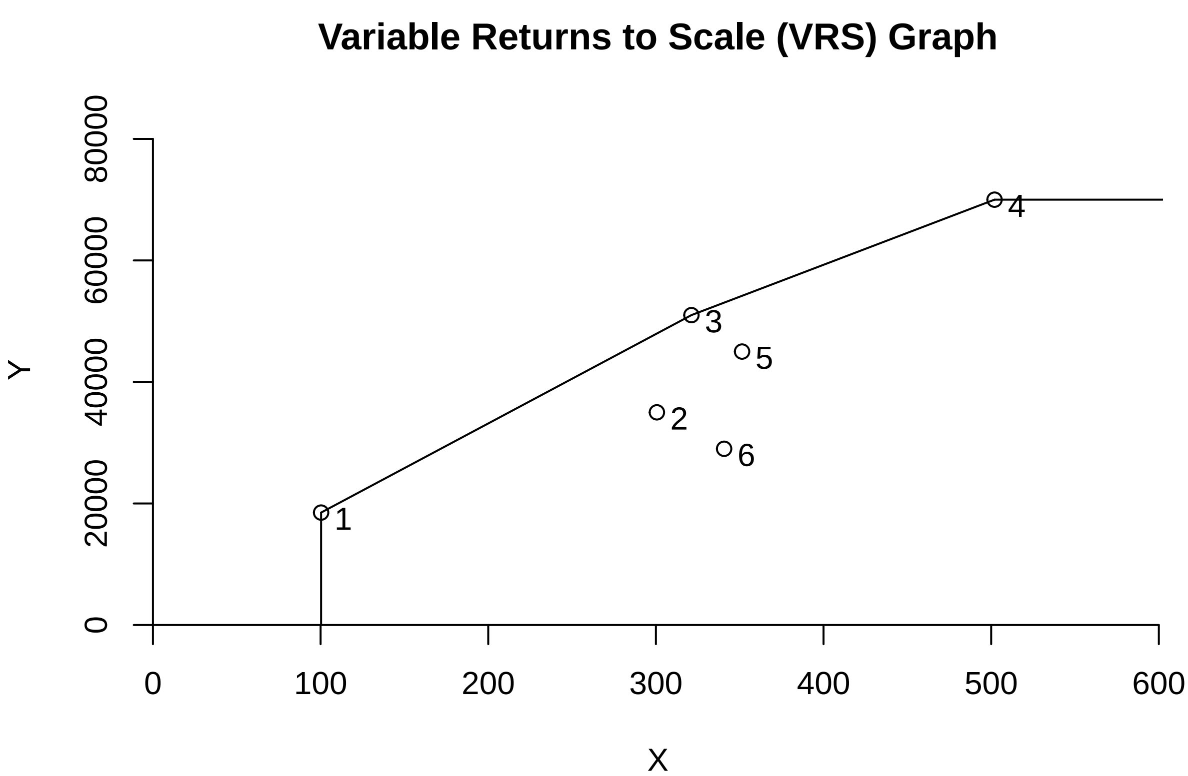
```
## 2      2     NA 0.0000000 1.0000000  0 0.0000000      1.0000000
## 3      3     NA 0.0000000 0.0000000  1 0.0000000      1.0000000
## 4      4     NA 0.0000000 0.0000000  0 1.0000000      1.0000000
## 5      1      4 0.4415584 0.0000000  0 0.5584416      0.9239332
## 6      1      2 0.3030303 0.6969697  0 0.0000000      0.7272727
```

As per the above tabular data, $DMU_1, DMU_2, DMU_3, DMU_4$ are identified as efficient. DMU_5 is inefficient and it is 1-(0.44 +0.55)= 0.01. But DMU_5 can reduce this 1% inefficiency learning from its peers $DMU_1(L1)$ and $DMU_4(L4)$.The ratio of $DMU_1(L1)$ to $DMU_4(L4)$ is 4:5. Therefore, DMU_5 can learn 0.01*(4/9)=0.004444444= 0.4% from its peer DMU_1 and 0.01*(5/9)= 0.005555556= 0.56% from its peer DMU_4.

Not only that, DMU_6) also inefficient by 1-(0.3+0.69)=0.01. This inefficiency can reduce from learning its peers of $DMU_1(L1)$ to $DMU_2(L2)$.The ratio between those peers is 14:15. Hence that $DMU_6)canlearn0.01*$ (14/29) = 0.004827586 = 0.48 and 0.01*(15/29)=0.0052= 0.52% from peer $DMU_2(L2)$.

Plot the result

```
dea.plot(x,y,RTS = "vrs", ORIENTATION = "in-out", txt = TRUE, main= "Variable Returns to Scale (VRS) Gra
```



DEA Analysis using IRS

We will now perform a DEA analysis under Increasing Returns to Scale (IRS) to determine if scaling up operations is feasible.

input and output values

```r
irs <- dea(x,y,RTS = "irs")
```

calculate efficiency and name the column

```r
eff_irs <- as.data.frame(irs$eff)
colnames(eff_irs)<- c("efficiency_irs")
eff_irs
```

```
##   efficiency_irs
## 1      1.0000000
## 2      1.0000000
## 3      0.8793468
## 4      1.0000000
## 5      0.9239332
## 6      0.7272727
```

According to the above results, under the IRS assumption Facility1,2 and 4 are identified in their full efficiency. But Facility4, 5 and 6 are not full efficient and their efficient score is below 1.

Find the peers

```r
peer_irs<-peers(irs)
peer_irs
```

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

According to the above table results, DMU_1, DMU_2 and DMU_4 don't have other peers and they have become their own peers. So they can be identified as in their full efficiency. But DMU_3 and DMU_5 have 2 peers for each of them, peerDMU_1 and peerDMU_4.DMU_6 also have two peers DMU_1 and DMU_2.

Find the lambda

```r
lambda_irs <- lambda(irs)
lambda_irs
```

```
##             L1        L2         L4
## [1,] 1.0000000 0.0000000 0.00000000
## [2,] 0.0000000 1.0000000 0.00000000
## [3,] 2.5789474 0.0000000 0.04699248
## [4,] 0.0000000 0.0000000 1.00000000
## [5,] 0.4415584 0.0000000 0.55844156
## [6,] 0.3030303 0.6969697 0.00000000
```

These lambda values represent the efficiency scores for each facility:

Facility 1: L1 = 1.000 (Fully efficient under IRS) Facility 2: L2 = 1.000 (Fully efficient under IRS) Facility 3: L1 = 2.579, L4 = 0.047 (Inefficient under IRS, but it can improve by reallocating resources) Facility 4: L4

= 1.000 (Fully efficient under IRS) Facility 5: L1 = 0.442, L4 = 0.558 (Inefficient under IRS, and it needs to improve resource allocation) Facility 6: L1 = 0.303, L2 = 0.697 (Inefficient under IRS, and it needs to improve resource allocation)

Tabular data for IRS

```
tabular_irs <- cbind(peer_irs,lambda_irs,eff_irs)
tabular_irs
```

```
##   peer1 peer2        L1        L2         L4 efficiency_irs
## 1     1    NA 1.0000000 0.0000000 0.00000000      1.0000000
## 2     2    NA 0.0000000 1.0000000 0.00000000      1.0000000
## 3     1     4 2.5789474 0.0000000 0.04699248      0.8793468
## 4     4    NA 0.0000000 0.0000000 1.00000000      1.0000000
## 5     1     4 0.4415584 0.0000000 0.55844156      0.9239332
## 6     1     2 0.3030303 0.6969697 0.00000000      0.7272727
```

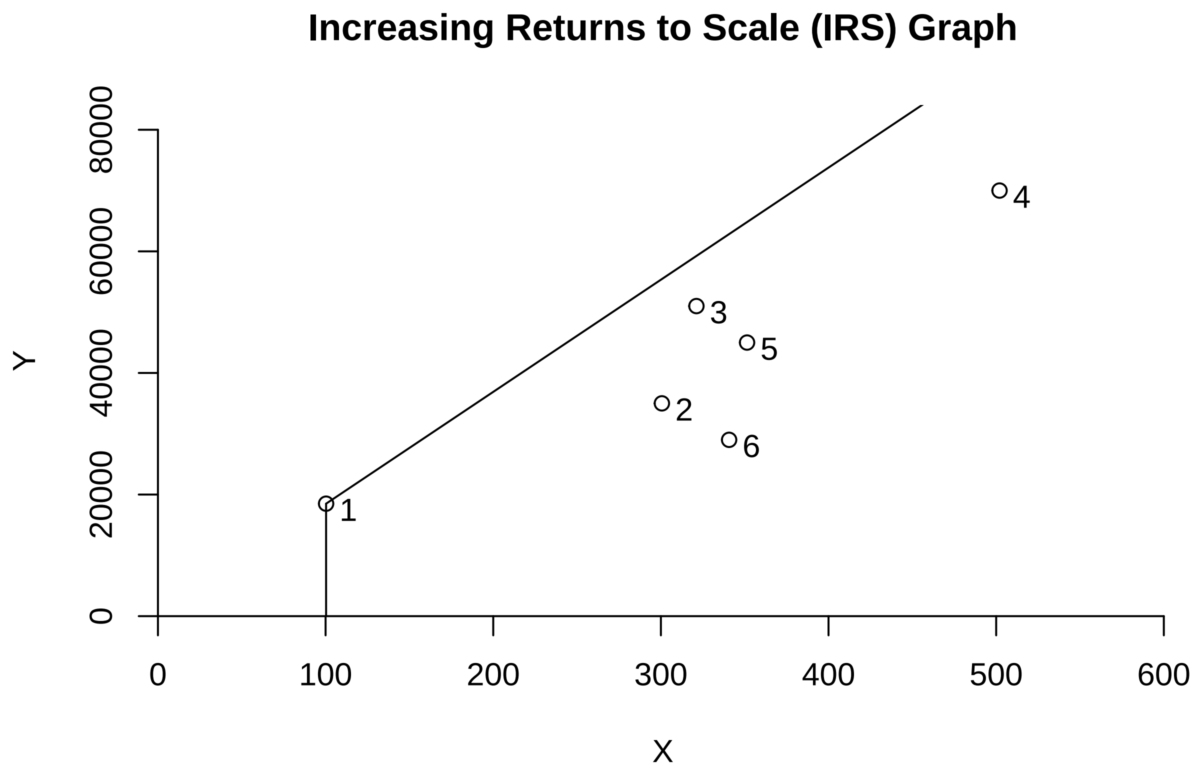
As per the above tabular data of IRS, DMU_1, DMU_2, DMU_4 are identified as efficient. DMU_3 is inefficient and it can reduce the inefficiency learning the majority from peer DMU_1 because its' lambda value is higher than 1. As well as the rest of the inefficiency can lean from its' other peer DMU_4.

DMU_5 also inefficient and it is 1-(0.44+0.55)=0.01. $DMU_1(L1)$ and $DMU_4(L4)$ are the peers of DMU_5. The ratio between those peers is 4:5. So DMU_5 can learn 0.01*(4/9)= 0.0044=0.44% from peer $DMU_1(L1)$ and 0.01*(5/9)= 0.005555556= 0.56% from peer $DMU_4(L4)$.

The DMU_6 also inefficient. It is 1-(0.3+0.69) = 0.01. Although it is inefficient it can learn from its peers of $DMU_1(L1)$ and $DMU_2(L2)$. The ratio between the peer L1 and L2 is 14:15. So DMU_6 can learn 0.01*(14/29)=0.004827586 =0.48% from peer $DMU_1(L1)$ and 0.01*(15/29)=0.005172414= 0.52% from peer $DMU_2(L2)$.

Plot the result

```
dea.plot(x,y,RTS = "irs", ORIENTATION = "in-out", txt = TRUE, main= "Increasing Returns to Scale (IRS) 
```



DEA Analysis using DRS

We will now conduct a DEA analysis with Decreasing Returns to Scale (DRS), which examines the potential for reducing the scale of operations in various production processes, in contrast to IRS.

input and output values

```
drs <- dea(x,y,RTS = "drs")
```

calculate efficiency and name the column

```
eff_drs <- as.data.frame(drs$eff)
colnames(eff_drs)<- c("efficiency_drs")
eff_drs
```

```
##   efficiency_drs
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
## 5      0.8941998
## 6      0.7047619
```

According to the above table results, $DMU_1, DMU_2, DMU_3, DMU_4$ are in their full efficiency. Due to DMU_5 and DMU_6 efficiencies are below 1, those two identified as inefficient DMUs under the DRS assumption.

Find the peers

```
peer_drs<-peers(drs)
peer_drs
```

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

According to the above matrix, Facility1,facility2, facility3 and facility 4 become their own peers and they don't have any other peers. So those 4 DMUs are efficient. But Dmu_5 has two peers DMU_1 and and DMU_4. As well as DMU_6 has two peers as DMU_1 and DMU_2. Therefore DMU_5 and DMU_6 are inefficient.

Find the lambda

```
lambda_drs <- lambda(drs)
lambda_drs
```

```
##             L1        L2 L3        L4
## [1,] 1.0000000 0.0000000  0 0.0000000
## [2,] 0.0000000 1.0000000  0 0.0000000
## [3,] 0.0000000 0.0000000  1 0.0000000
## [4,] 0.0000000 0.0000000  0 1.0000000
## [5,] 0.2631579 0.0000000  0 0.5733083
## [6,] 0.2222222 0.7111111  0 0.0000000
```

According to the above lambda values facility1, facility 2, facility 3 and facility 4 are in their full efficiency and they are indicating by the lambda value of 1.Row 5: This row represents the transformation of the data. It indicates that the first dimension (L1) is transformed with a lambda value of approximately 0.2631579, and the fourth dimension (L4) is transformed with a lambda value of approximately 0.5733083.

Row 6: Similarly, this row represents a transformation where the first dimension (L1) is transformed with a lambda value of approximately 0.2222222, and the second dimension (L2) is transformed with a lambda value of approximately 0.7111111. So Facility 5 and 6 are inefficient.

Tabular data for DRS

```
tabular_drs <- cbind(peer_drs,lambda_drs,eff_drs)
tabular_drs
```

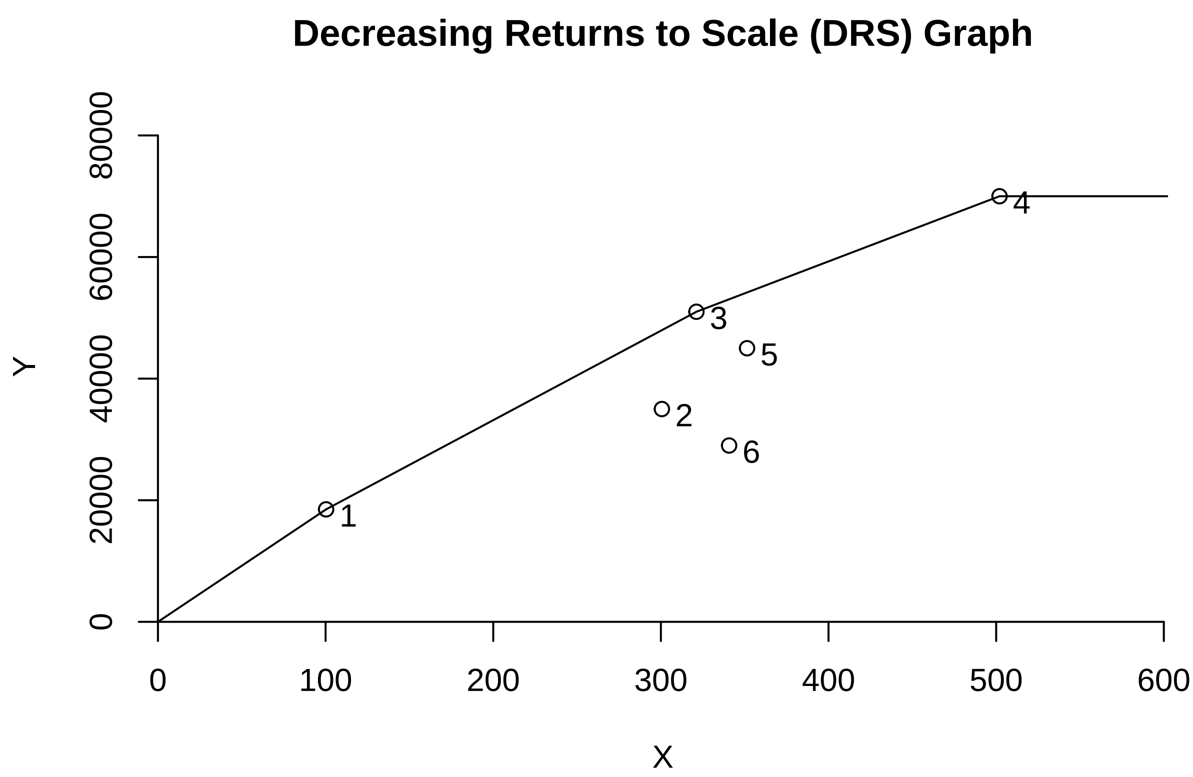
```
##   peer1 peer2        L1        L2 L3        L4 efficiency_drs
## 1     1    NA 1.0000000 0.0000000  0 0.0000000      1.0000000
## 2     2    NA 0.0000000 1.0000000  0 0.0000000      1.0000000
## 3     3    NA 0.0000000 0.0000000  1 0.0000000      1.0000000
## 4     4    NA 0.0000000 0.0000000  0 1.0000000      1.0000000
## 5     1     4 0.2631579 0.0000000  0 0.5733083      0.8941998
## 6     1     2 0.2222222 0.7111111  0 0.0000000      0.7047619
```

As per the above tabular data of DRS, $DMU_1, DMU_2, DMU_3, DMU_4$ are identified as efficient. DMU_5 is inefficient and it is 1-(0.2631579+0.5733083)=0.1635338= 16.35% . But DMU_5 can reduce this 16.35% inefficiency learning from its peers $DMU_1(L1)$ and $DMU_4(L4)$.The ratio of $DMU_1(L1)$ to $DMU_4(L4)$ is 14:15. Therefore, DMU_5 can learn 0.1635338*(14/29)=0.078947352 = 7.89% from peer* $DMU_1(L1)$ *and 0.1635338*(15/29)= 0.084586448=8.46%. from peer $DMU_4(L4)$.

On the other hand, DMU_6also inefficient and it is 1- (0.2222222+0.7111111)=0.0666667=6.67%. But DMU_6 can reduce its' inefficiency learning from its peers $DMU_1(L1)$ and $DMU_2(L2)$. the ratio between $DMU_1(L1)$ and $DMU_2(L2)$ is 14:15 So DMU_6 can learn 0.0666667*(14/29)= 0.032183924 from peer* $DMU_1(L1)$ *and 0.0666667*(15/29)= 0.034482776 from peer $DMU_1(L1)$.

Plot the result

```
dea.plot(x,y,RTS = "drs", ORIENTATION = "in-out", txt = TRUE, main= "Decreasing Returns to Scale (DRS) (
```



DEA Analysis using FRH

We will now perform a DEA analysis using the Free Replicability Hull (FRH). Like FDH, FRH also utilizes mixed integer programming, where variables are treated as integers to find optimal solutions. FRH aims to replace deterministic data with random variables.

input and output values

```
frh <- dea(x,y,RTS = "add")
```

calculate efficiency and name the column

```r
eff_frh<- as.data.frame(frh$eff)
colnames(eff_frh)<- c("efficiency_frh")
eff_frh
```

```
##   efficiency_frh
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
## 5      1.0000000
## 6      0.8823529
```

According to the above efficiency values, DMU_6 is inefficient because its' efficient value is below 1 and all the other DMUs are identified as efficient under the assumption of FRH.

Find the peers

```r
peer_frh<-peers(frh)
peer_frh
```

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

According to the above peer identification, all the DMUs except DMU_6 become their own peers and only DMU_6 consider DMU_2 as its' peer. Therefore DMU_6 seems to be inefficient.

Find the lambdas

```r
lambda_frh <- lambda(frh)
lambda_frh
```

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

These lambda values are exactly same as the lambda values of FDH. Facility 1,2,3,4 and 5 are 100% efficient, and its efficiency score is calculated by comparing it to itself.Facility 6 is less efficient than the other facilities, and its efficiency score is calculated by comparing it to Facility 2. This means that Facility 6 could improve its efficiency by using its inputs and outputs more efficiently, or by finding ways to reduce its inputs or increase its outputs.

Tabular data for FRH

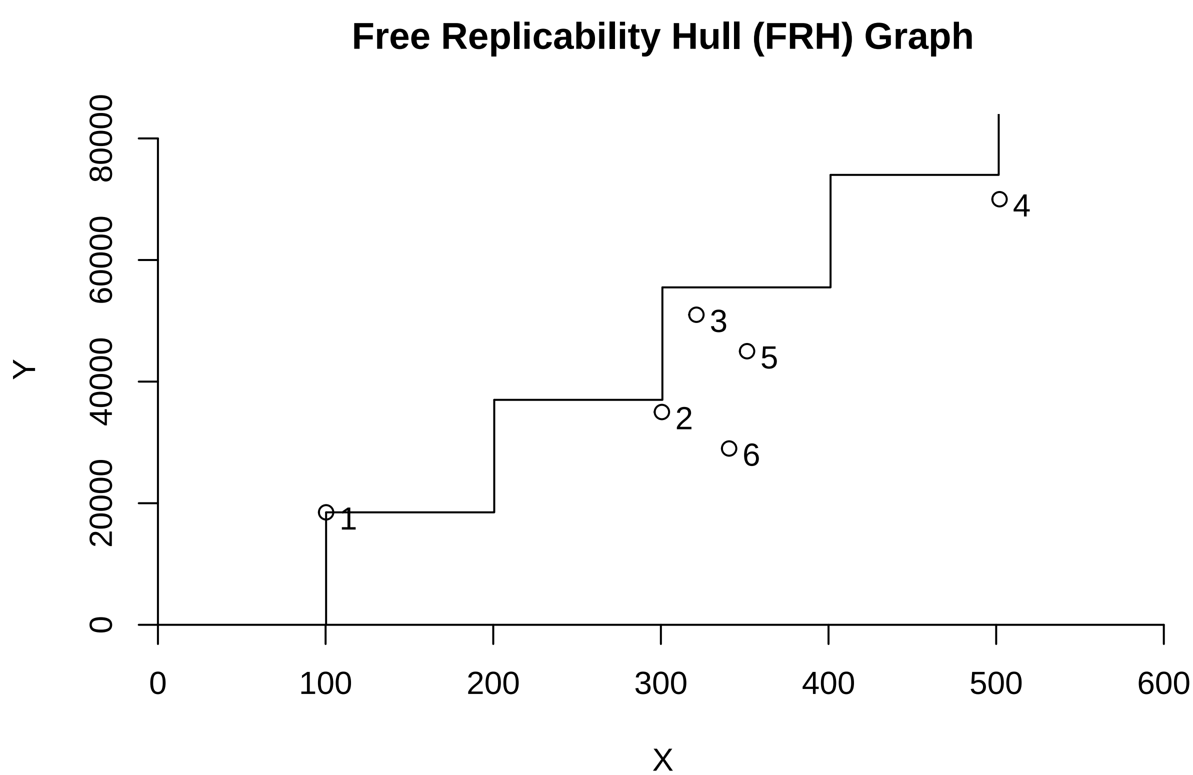
```r
tabular_frh <- cbind(peer_frh,lambda_frh,eff_frh)
tabular_frh
```

```
##   peer1 L1 L2 L3 L4 L5 efficiency_frh
## 1     1  1  0  0  0  0      1.0000000
## 2     2  0  1  0  0  0      1.0000000
## 3     3  0  0  1  0  0      1.0000000
## 4     4  0  0  0  1  0      1.0000000
## 5     5  0  0  0  0  1      1.0000000
## 6     2  0  1  0  0  0      0.8823529
```

According to the tabular data of FRH, DMU_6 is inefficient and it can reduce its' inefficiency from learning its peer of $DMU_2(L2)$.

Plot the result

```r
dea.plot(x,y,RTS = "add", ORIENTATION = "in-out", txt = TRUE, main= "Free Replicability Hull (FRH) Grap
```



Summary

According to the DEA analysis under all DEA assumptions of FDH, CRS, VRS,IRS, DRS, and FRH all the inefficient DMUs can be summarize as follow. So Under all the assumptions DMU_6 is identified as inefficient. Under four assumptions DMU_5 is identified as inefficient. As well as DMU_3 is inefficient according to CRS and IRS assumptions.

```r
df2 <- data.frame(
  FDH = c("DMU_6", " ", " "),
  CRS = c("DMU_3", "DMU_5", "DMU_6"),
  VRS = c("DMU_5", "DMU_6", " "),
  IRS = c("DMU_3", "DMU_5", "DMU_6"),
  DRS = c("DMU_5", "DMU_6", " "),
  FRH = c("DMU_6"," ", " ")
)
df2
```

```
##     FDH   CRS   VRS   IRS   DRS   FRH
## 1 DMU_6 DMU_3 DMU_5 DMU_3 DMU_5 DMU_6
## 2       DMU_5 DMU_6 DMU_5 DMU_6      
## 3       DMU_6       DMU_6            
```