Data Envelopment Analysis(DEA)

Shashidhar Reddy Boreddy

2022-10-29

```r
# 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
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
#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
```

```r
# 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
```

```r
#As we can see, we are obtaining the identical numbers as in the performance data table from the six nu
#In the next part, we will conduct a Data Envelopment Analysis (DEA), which is an analytical method tha
```

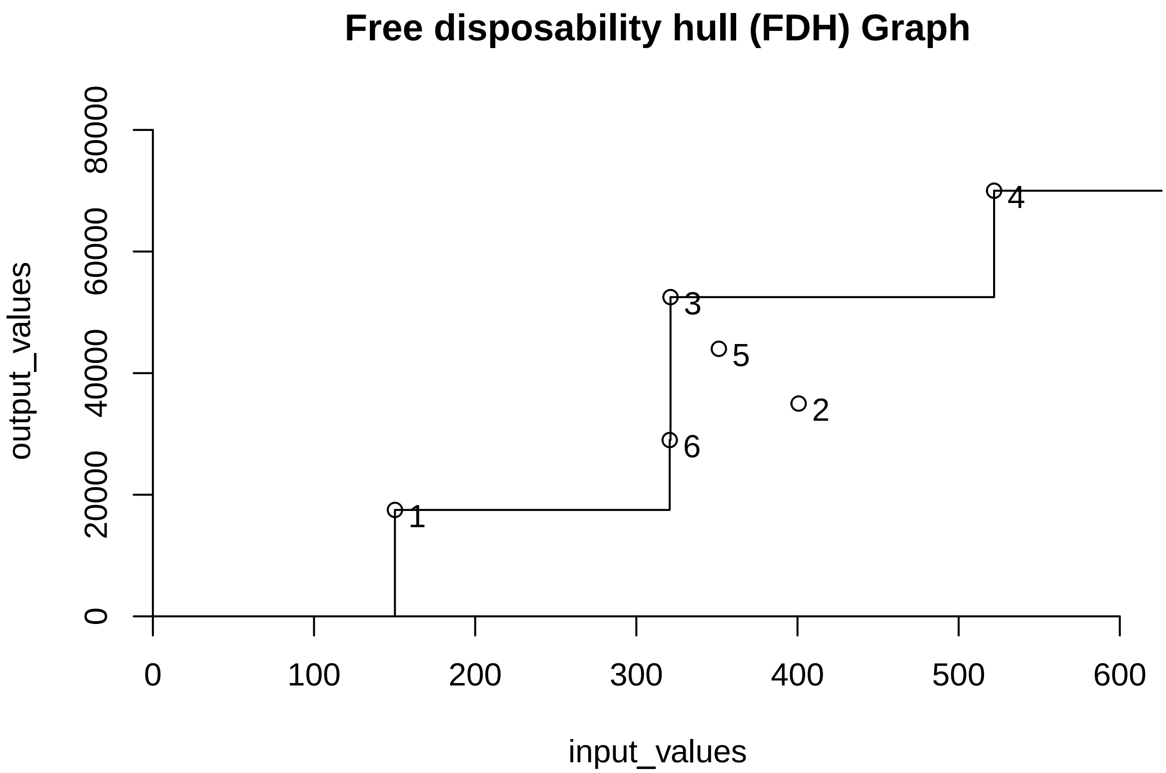
```r
#FDH-based DEA Analysis
#Now we'll use FDH to define and compute the DEA analysis.The Free disposability hull (FDH) is the assu
#Giving the input_values and output_values.
Anal.FDH<- dea(input_values,output_values,RTS = "FDH")
#Making a data frame with efficiency values.
EV.FDH <- as.data.frame(Anal.FDH$eff)
#  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", ylal
```



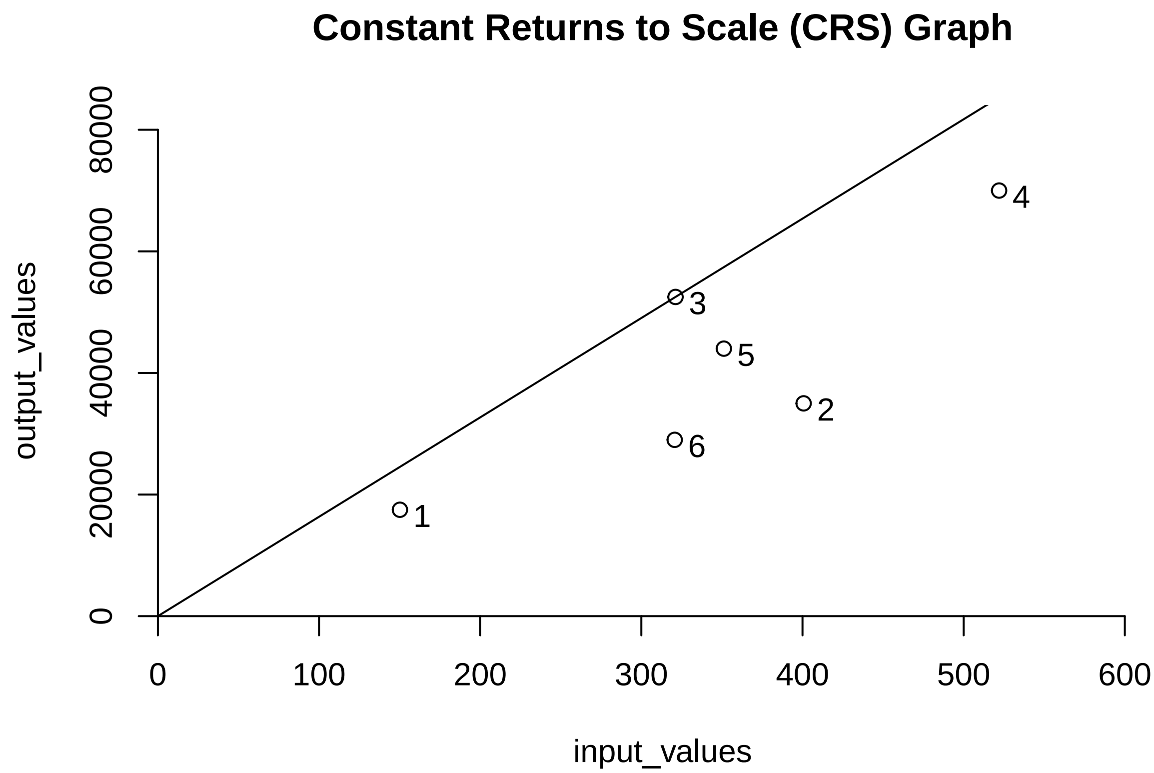

```
#CRS-based DEA Analysis
#Now we'll formulate and calculate the DEA analysis with Constant Returns to Scale (CRS).The CRS is par
# Giving the input_values and output_values
Anal.CRS <- dea(input_values,output_values,RTS = "CRS")
# observing the effciency values
EV.CRS <- as.data.frame(Anal.CRS$eff)
#  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
```

```r
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.

```r
# 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")
```



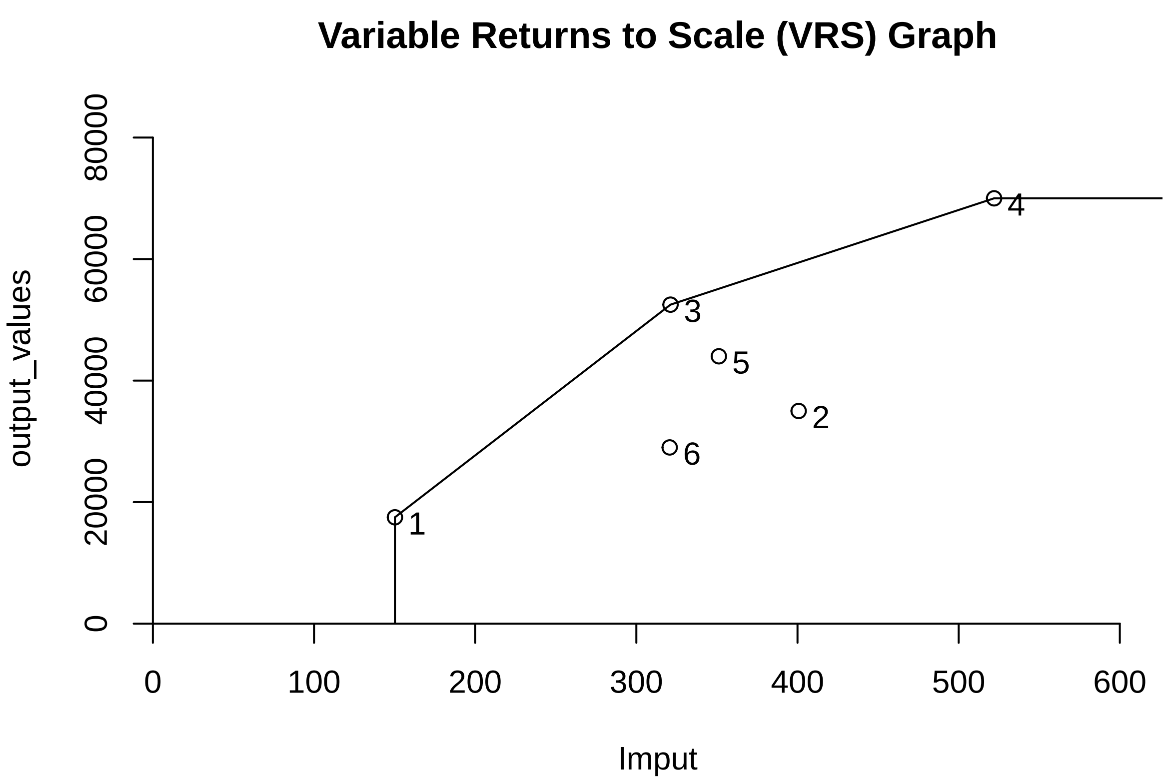

```r
#VRS DEA Analysis
#Now we will construct and calculate the DEA analysis utilizing Variable Returns to Scale (VRS). VRS is
# Giving the input_values and output_values
Anal.VRS <- dea(input_values,output_values,RTS = "VRS")
# To observe effciency values
EV.VRS <- as.data.frame(Anal.VRS$eff)
#  assigning an appropriate 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.

```r
# 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")
```



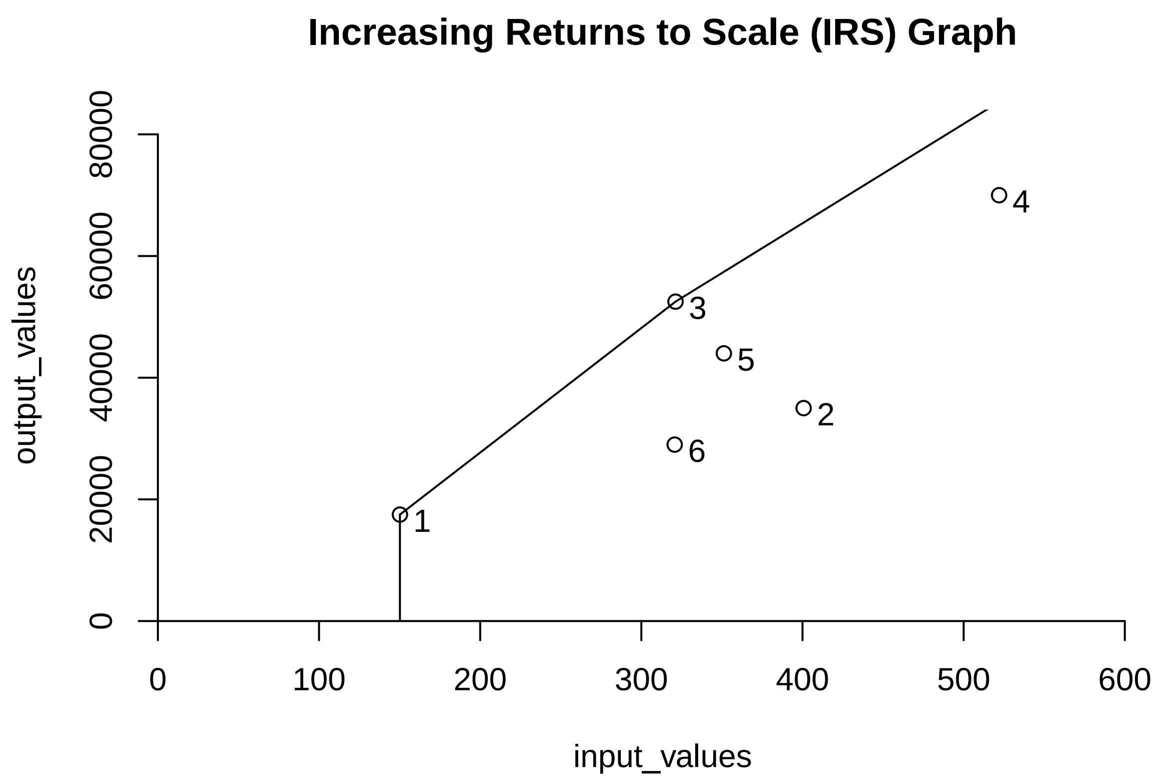

```r
#IRS Analysis for DEA
#Now we'll formulate and compute the DEA analysis with Increasing Returns to Scale (IRS).IRS shows whet
# Giving the input_values and output_values
Anal.IRS <- dea(input_values,output_values,RTS = "irs")
# observe effciency values
EV.IRS <- as.data.frame(Anal.IRS$eff)
#  assigning an appropiate name
```

```r
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%.

```r
# 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")
```



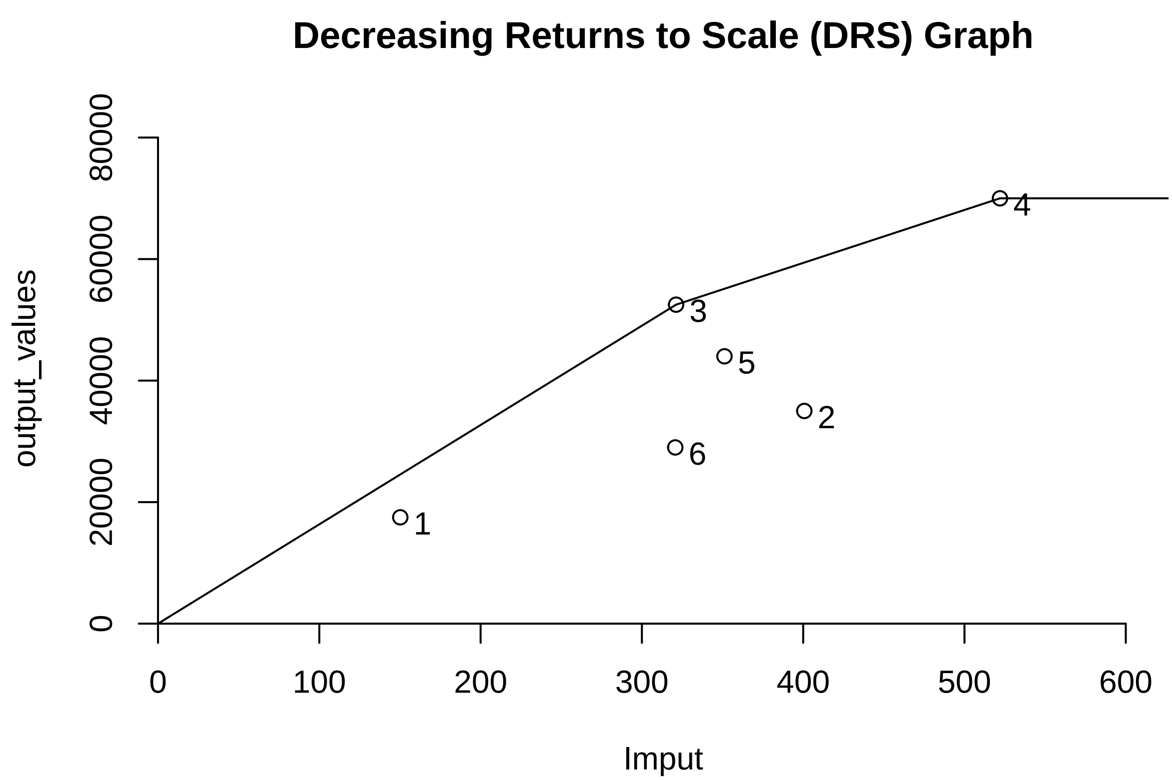

```r
#DRS for DEA Analysis
#Now we will develop and perform the DEA analysis based on Decreasing Returns to Scale (DRS).DRS is the
# Giving the input_values and output_values
Anal.DRS <- dea(input_values,output_values,RTS = "DRS")
# To observe the effciency values
EV.DRS <- as.data.frame(Anal.DRS$eff)
#  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")
```



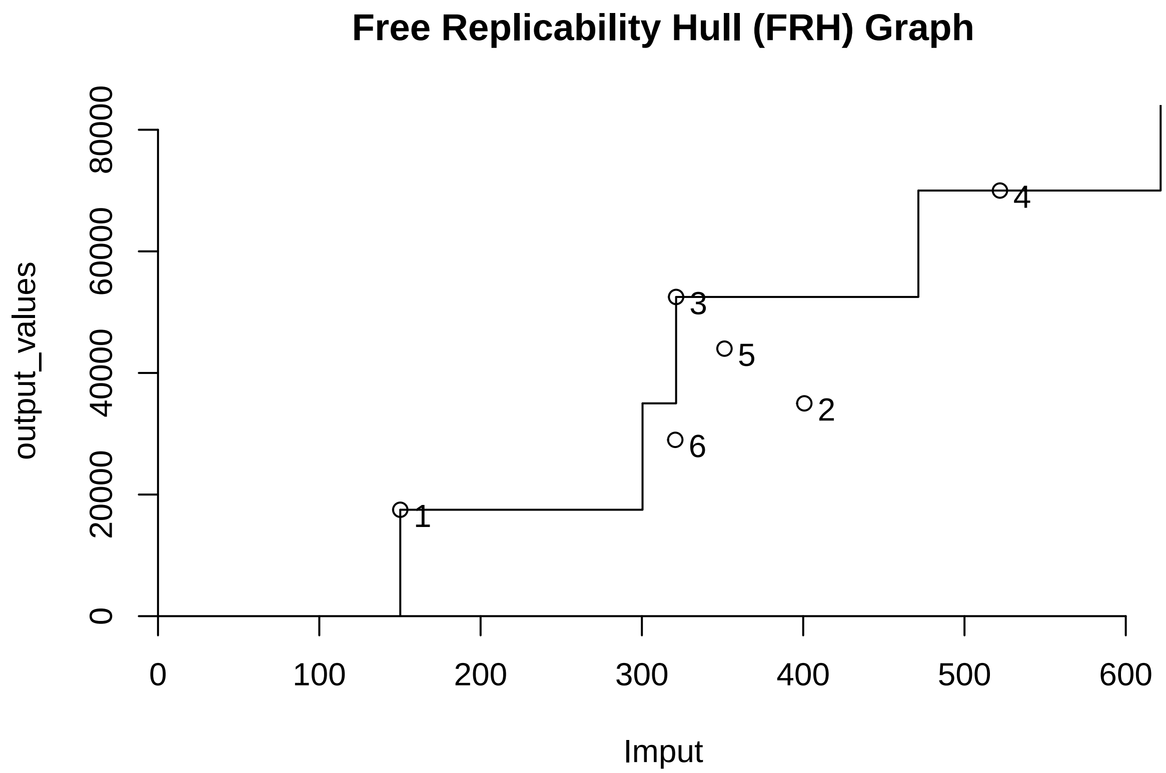

```
#FRH DEA Analysis
#Now, we'll use Free Replicability Hull to define and perform the DEA analysis (FRH).Both FRH and FDH e
# Giving the input_values and output_values
Anal.FRH <- dea(input_values,output_values,RTS = "add")
# To observe effciency values
EV.FRH <- as.data.frame(Anal.FRH$eff)
#  assigning an appropiate name
```

```r
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.

```r
# 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
```