Data Envelopment Analysis (DEA)

Melissa Paniagua

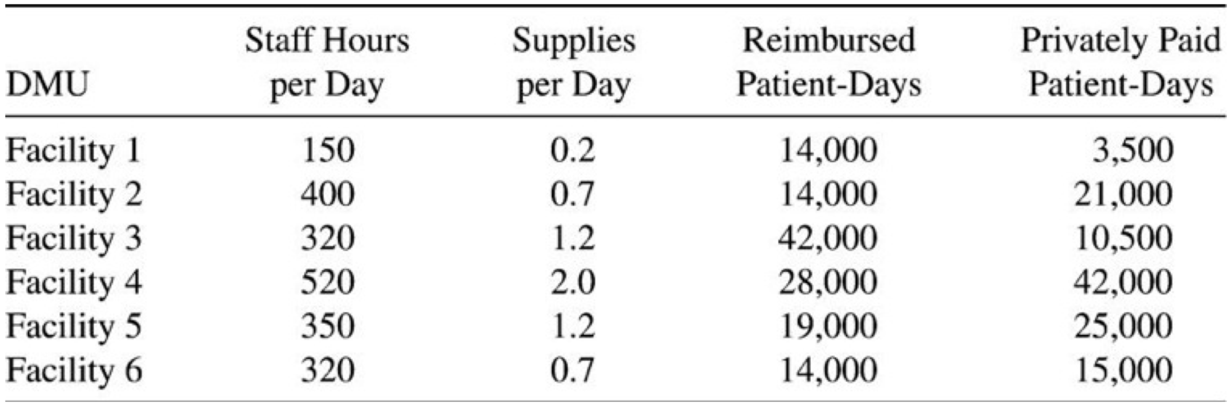
10/26/2020

Assignment 4 | Module 8

The purpose of this assignment is to explore the use of DEA. Throught this assignment, the students will formulate and solve DEA problems under different assumptions.

Hope Valley Health Care Association

The Hope Valley Health Care Association owns and operates six nursing homes in adjoining states. An evaluation of their efficiency has been undertaken using two inputs and two outputs. The inputs are staffing labor (measured in average hours per day) and the cost of supplies (in thousands of dollars per day). The outputs are the number of patient-days reimbursed by third- party sources and the number of patient-days reimbursed privately. A summary of performance data is shown in the table below.



DMU	Staff Hours per Day	Supplies per Day	Reimbursed Patient-Days	Privately Paid Patient-Days
Facility 1	150	0.2	14,000	3,500
Facility 2	400	0.7	14,000	21,000
Facility 3	320	1.2	42,000	10,500
Facility 4	520	2.0	28,000	42,000
Facility 5	350	1.2	19,000	25,000
Facility 6	320	0.7	14,000	15,000

Figure 1: Summary of Performance

```
# Upload libraries needed
library(Benchmarking)
library(tidyverse)
```

Compute the Formulation

Here, we are going to create a matrix and values.

```r
# To create the vectors with our values
input <- matrix(c(150,400,320,520,350, 320, 200, 700, 1200, 2000, 1200, 700),ncol = 2)
output <- matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,
                   25000, 15000),ncol = 2)

# Assign column names
colnames(output) <- c("staff_hours_daily","supplies_daily")
colnames(input) <- c("reimbursed_patient_daily", "privately_paid_patient-daily")

# To see the values of Input
input
```

```
     reimbursed_patient_daily privately_paid_patient-daily
[1,]                      150                          200
[2,]                      400                          700
[3,]                      320                         1200
[4,]                      520                         2000
[5,]                      350                         1200
[6,]                      320                          700
```

```r
# To see the values of Output
output
```

```
     staff_hours_daily supplies_daily
[1,]             14000           3500
[2,]             14000          21000
[3,]             42000          10500
[4,]             28000          42000
[5,]             19000          25000
[6,]             14000          15000
```

As we can see, here we are getting the same values as the performance data table from the six nursing homes owned by Hope Valley Health Care Association.

In the following section, we will perform a Data Envelopment Analysis (DEA), which is an analytical tool that can help organizations to identify and allocate their resources to enhance their efficiency and have better practices.

DEA Analysis using FDH

Now, we are going to formulate and compute the DEA analysis using FDH.

The Free disposability hull (FDH) is the assumption of dispose unwanted inputs and outputs. “Free disposability means that we can always produce fewer outputs with more inputs.” (DEA Slides)

```r
# Provide the input and output
analysis_fdh<- dea(input,output,RTS = "fdh")

# Create a data frame with effciency values
eff_fdh <- as.data.frame(analysis_fdh$eff)

# To assign an appropiate name
colnames(eff_fdh) <- c("efficiency_fdh")

# Identify the peers
peer_fdh <- peers(analysis_fdh)

# To assign an appropiate name
colnames(peer_fdh) <- c("peer1_fdh")

# Identify the relative weights given to the peers using lambda function
lambda_fdh <- lambda(analysis_fdh)

# To assign an appropiate column name for Lambda
colnames(lambda_fdh) <- c("L1_fdh", "L2_fdh", "L3_fdh", "L4_fdh", "L5_fdh", "L6_fdh")

# Create a tabular data with peer, lambda, and efficiency
peer_lamb_eff_fdh <- cbind(peer_fdh, lambda_fdh, eff_fdh)

# Show the summary chart
peer_lamb_eff_fdh
```

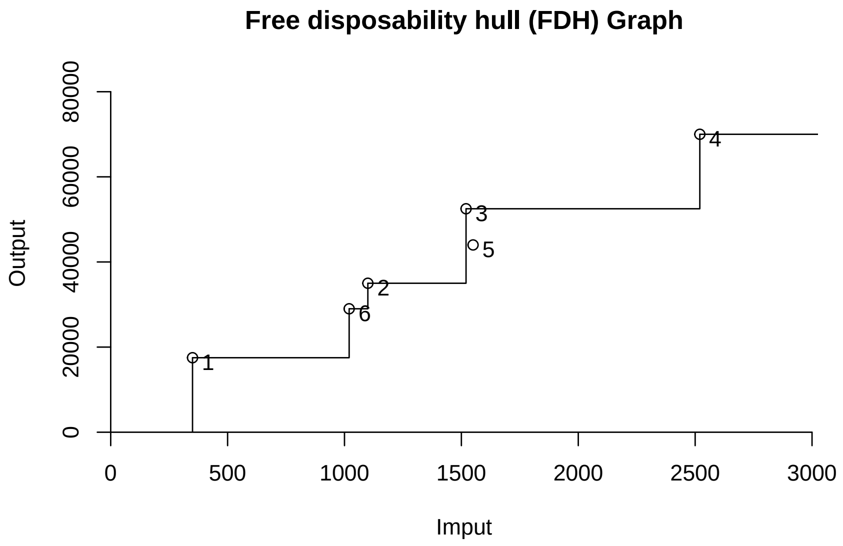
```
  peer1_fdh L1_fdh L2_fdh L3_fdh L4_fdh L5_fdh L6_fdh efficiency_fdh
1         1      1      0      0      0      0      0              1
2         2      0      1      0      0      0      0              1
3         3      0      0      1      0      0      0              1
4         4      0      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 learned during this module, peers are the way we could identify inefficient DMU or units, and Lambda values are the raw weights assigned from the peer units when solving the DEA model.

The summary chart shown above, confirms that every DMU or facility is working using all its capacity and efficiency. Every peer was assigned one unit, for that reason, the Lambda values are 1, and efficiency are 1 as well.

Now, let's see the graph.

```r
# Plot the results
dea.plot(input,output,RTS="fdh",ORIENTATION="in-out",
         txt=TRUE, xlab = 'Imput', ylab= 'Output', main="Free disposability hull (FDH) Graph")
```



DEA Analysis using CRS

Now, we are going to formulate and compute the DEA analysis using Constant Returns to Scale (CRS).

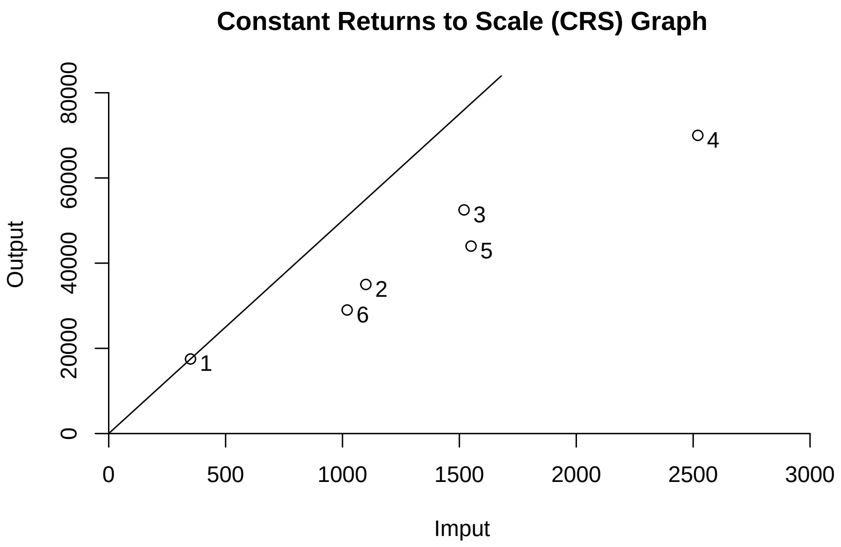
The CRS is part of the scaling assumption, and it allows us to see if there is any possible combination to scale up or down.

```r
# Provide the input and output
analysis_crs <- dea(input,output,RTS = "crs")

# To see the effciency values
eff_crs <- as.data.frame(analysis_crs$eff)

# To assign an appropiate name
colnames(eff_crs) <- c("efficiency_crs")

# Identify the peers
peer_crs <- peers(analysis_crs)

# To assign an appropiate name
colnames(peer_crs) <- c("peer1_crs", "peer2_crs", "peer3_crs")

# Identify the relative weights given to the peers using lambda function
lambda_crs <- lambda(analysis_crs)

# To assign an 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
peer_lamb_eff_crs <- cbind(peer_crs, lambda_crs, eff_crs)

# Show the summary chart
peer_lamb_eff_crs
```

```
  peer1_crs peer2_crs peer3_crs    L1_crs     L2_crs L3_crs    L4_crs efficiency_crs
1         1        NA        NA 1.0000000 0.00000000      0 0.0000000      1.0000000
2         2        NA        NA 0.0000000 1.00000000      0 0.0000000      1.0000000
3         3        NA        NA 0.0000000 0.00000000      1 0.0000000      1.0000000
4         4        NA        NA 0.0000000 0.00000000      0 1.0000000      1.0000000
5         1         2         4 0.2000000 0.08048142      0 0.5383307      0.9774987
6         1         2         4 0.3428571 0.39499264      0 0.1310751      0.8674521
```

Regarding Constant Returns to Scale (CRS), the facilities 1, 2, 3, and 4 are using all its efficiency as the lambdas and peers prove. Facility 5 and 6, on the other hand, need parts of 1, 2, and 4 as the peers and lambdas show above. It means these two facilities (5 and 6) have room to improve because they are getting an efficiency of 97.74% and 86.74% respectively.

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



DEA Analysis using VRS

Now, we are going to formulate and compute the DEA analysis using Variable Returns to Scale (VRS).

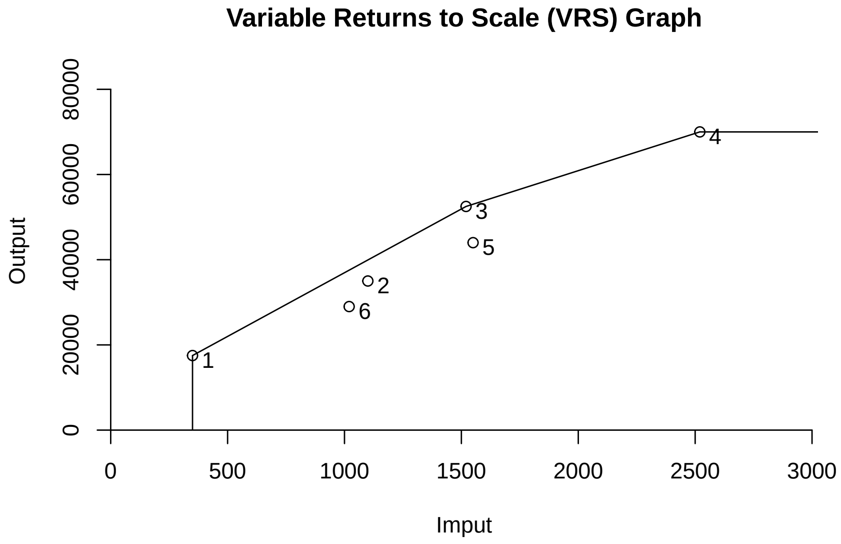
VRS is also part of the scaling assumption, and it helps to estimate the efficiency of the variables whether an increase or decrease is not proportional.

```
# Provide the input and output
analysis_vrs <- dea(input,output,RTS = "vrs")

# To see the effciency values
eff_vrs <- as.data.frame(analysis_vrs$eff)

# To assign an appropiate name
colnames(eff_vrs) <- c("efficiency_vrs")

# Identify the peers
peer_vrs <- peers(analysis_vrs)

# To assign an appropiate name
colnames(peer_vrs) <- c("peer1_vrs", "peer2_vrs", "peer3_vrs")

# Identify the relative weights given to the peers using lambda function
lambda_vrs <- lambda(analysis_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
peer_lamb_eff_vrs <- cbind(peer_vrs, lambda_vrs, eff_vrs)

# Show the summary chart
peer_lamb_eff_vrs
```

```
  peer1_vrs peer2_vrs peer3_vrs    L1_vrs    L2_vrs L3_vrs L4_vrs    L5_vrs efficiency_vrs
1         1        NA        NA 1.0000000 0.0000000      0      0 0.0000000      1.0000000
2         2        NA        NA 0.0000000 1.0000000      0      0 0.0000000      1.0000000
3         3        NA        NA 0.0000000 0.0000000      1      0 0.0000000      1.0000000
4         4        NA        NA 0.0000000 0.0000000      0      1 0.0000000      1.0000000
5         5        NA        NA 0.0000000 0.0000000      0      0 1.0000000      1.0000000
6         1         2         5 0.4014399 0.3422606      0      0 0.2562995      0.8963283
```

Now we run the Variable Returns to Scale (VRS), we can identify that facility 1, 2, 3, 4, and 5 are working in all its capacity or efficiency. However, that does not happen with facility 6, which has an efficiency of 89.63%. As peers and lambdas show, facility 6 needs part of facility 1, 2, and 5 to achieve better efficiency.

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



DEA Analysis using IRS

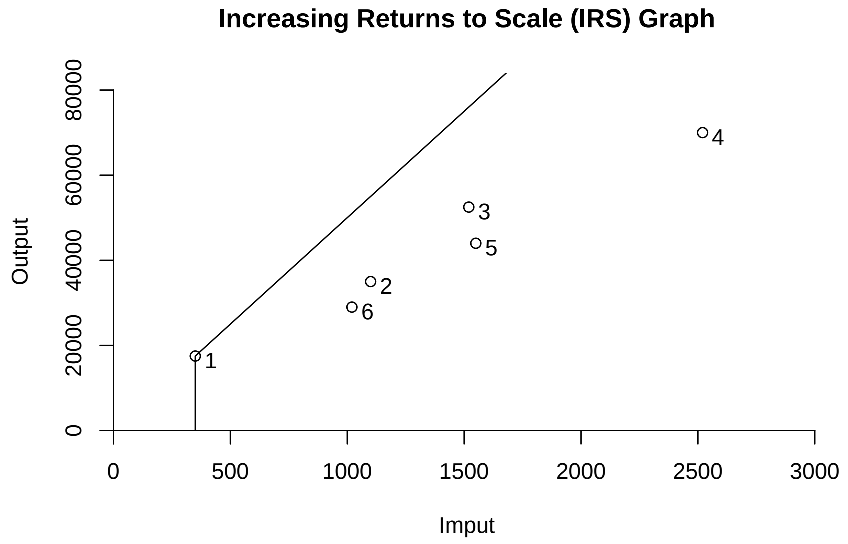
Now, we are going to formulate and compute the DEA analysis using Increasing Returns to Scale (IRS). IRS indicates if it is possible to increate the operation scale.

```
# Provide the input and output
analysis_irs <- dea(input,output,RTS = "irs")

# To see the effciency values
eff_irs <- as.data.frame(analysis_irs$eff)

# To assign an appropiate name
colnames(eff_irs) <- c("efficiency_irs")

# Identify the peers
peer_irs <- peers(analysis_irs)

# To assign an appropiate name
colnames(peer_irs) <- c("peer1_irs", "peer2_irs", "peer3_irs")

# Identify the relative weights given to the peers using lambda function
lambda_irs <- lambda(analysis_irs)

# To assign 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
peer_lamb_eff_irs <- cbind(peer_irs, lambda_irs, eff_irs)

# Show the summary chart
peer_lamb_eff_irs
```

```
  peer1_irs peer2_irs peer3_irs    L1_irs    L2_irs L3_irs L4_irs    L5_irs efficiency_irs
1         1        NA        NA 1.0000000 0.0000000      0      0 0.0000000      1.0000000
2         2        NA        NA 0.0000000 1.0000000      0      0 0.0000000      1.0000000
3         3        NA        NA 0.0000000 0.0000000      1      0 0.0000000      1.0000000
4         4        NA        NA 0.0000000 0.0000000      0      1 0.0000000      1.0000000
5         5        NA        NA 0.0000000 0.0000000      0      0 1.0000000      1.0000000
6         1         2         5 0.4014399 0.3422606      0      0 0.2562995      0.8963283
```

Increasing Returns to Scale (IRS) behives the same as Variable Returns to Scale (VRS) by getting facility 1, 2, 3, 4, and 5 are working all its efficiency, but facility 6 needs to improve needs from units 1, 2, and 5 to improve its efficiency which is 89.63%.

```
# Plot the results
dea.plot(input,output,RTS="irs",ORIENTATION="in-out",
         txt=TRUE, xlab = ’Imput’, ylab= ’Output’, main="Increasing Returns to Scale (IRS) Graph")
```



DEA Analysis using DRS

Now, we are going to formulate and compute the DEA analysis using Decreasing Returns to Scale (DRS).

DRS is the opposite of IRS, which its goal is to decrease the operation scale on any possible production process.

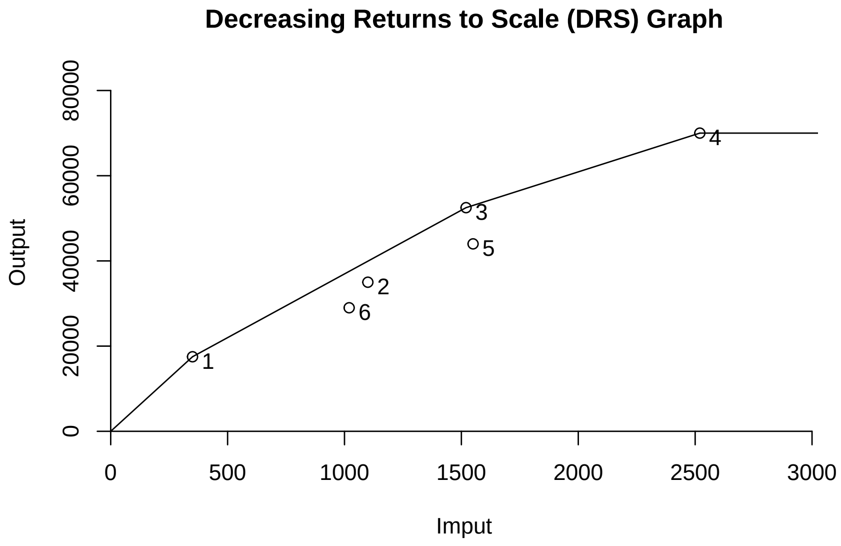
```
# Provide the input and output
analysis_drs <- dea(input,output,RTS = "drs")

# To see the effciency values
eff_drs <- as.data.frame(analysis_drs$eff)

# To assign an appropiate name
colnames(eff_drs) <- c("efficiency_drs")
```

```r
# Identify the peers
peer_drs <- peers(analysis_drs)

# To assign an appropiate name
colnames(peer_drs) <- c("peer1_drs", "peer2_drs", "peer3_drs")

# Identify the relative weights given to the peers using lambda function
lambda_drs <- lambda(analysis_drs)

# To 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
peer_lamb_eff_drs <- cbind(peer_drs, lambda_drs, eff_drs)

# Show the summary chart
peer_lamb_eff_drs
```

```
  peer1_drs peer2_drs peer3_drs    L1_drs     L2_drs L3_drs    L4_drs efficiency_drs
1         1        NA        NA 1.0000000 0.00000000      0 0.0000000      1.0000000
2         2        NA        NA 0.0000000 1.00000000      0 0.0000000      1.0000000
3         3        NA        NA 0.0000000 0.00000000      1 0.0000000      1.0000000
4         4        NA        NA 0.0000000 0.00000000      0 1.0000000      1.0000000
5         1         2         4 0.2000000 0.08048142      0 0.5383307      0.9774987
6         1         2         4 0.3428571 0.39499264      0 0.1310751      0.8674521
```

Decreasing Returns to Scale (DRS) has a good efficiency in facility 1, 2, 3, and 4. Regarding facility 5 and 6, there is room they can improve. Both of them need part of facilities 1, 2, and 4 to be able to achieve their highest efficiency of 1 as we can prove in the previous table.

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



DEA Analysis using FRH

Now, we are going to formulate and compute the DEA analysis using Free Replicability Hull (FRH).

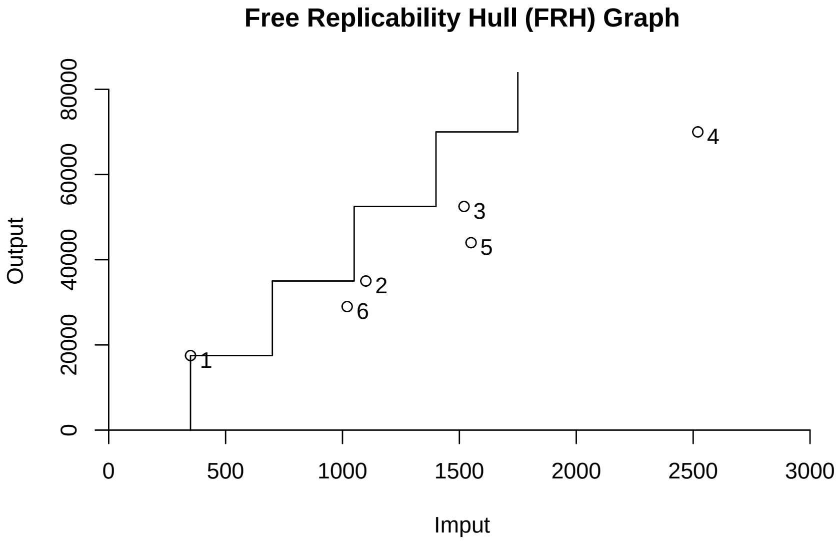
FRH as well as FDH use mixed integer programming, which refers that the variables must be integers to find the optimal solution. The goal of FRH is to replace deterministic data using random variables. (Majid, A. Reza, F., 2012, par. 2)

```
# Provide the input and output
analysis_frh <- dea(input,output,RTS = "add")

# To see the effciency values
eff_frh <- as.data.frame(analysis_frh$eff)

# To assign an appropiate name
colnames(eff_frh) <- c("efficiency_frh")

# Identify the peers
peer_frh <- peers(analysis_frh)

# To assign an appropiate name
colnames(peer_frh) <- c("peer1_frh")

# Identify the relative weights given to the peers using lambda function
lambda_frh <- lambda(analysis_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
peer_lamb_eff_frh <- cbind(peer_frh, lambda_frh, eff_frh)

# Show the summary chart
peer_lamb_eff_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
```

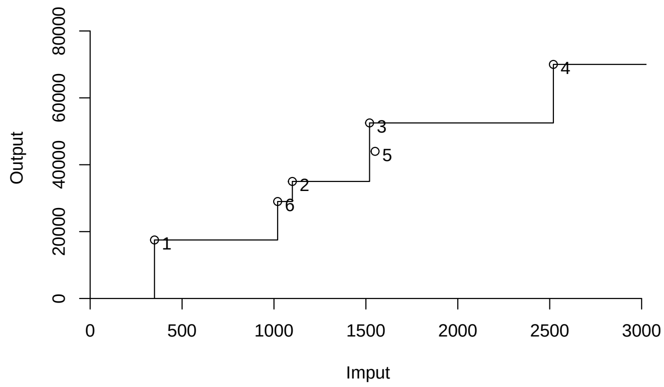
Free Replicability Hull (FRH) has a great efficiency in all its DMU. It behives the same as Free disposability hull (FDH), which all its values have their own peer, lambas and efficiency of 1.

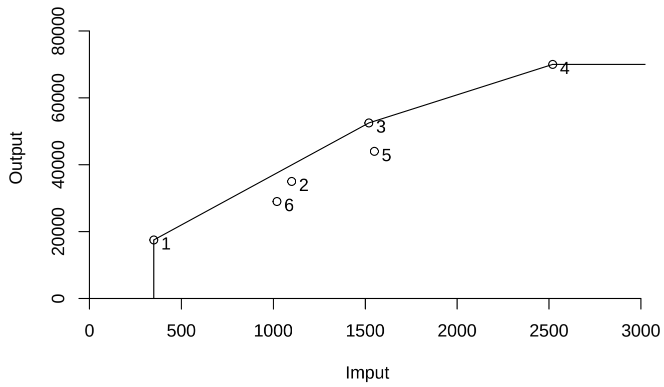
```
# Plot the results
dea.plot(input,output,RTS="add",ORIENTATION="in-out",
         txt=TRUE, xlab = ’Imput’, ylab= ’Output’, main="Free Replicability Hull (FRH) Graph")
```

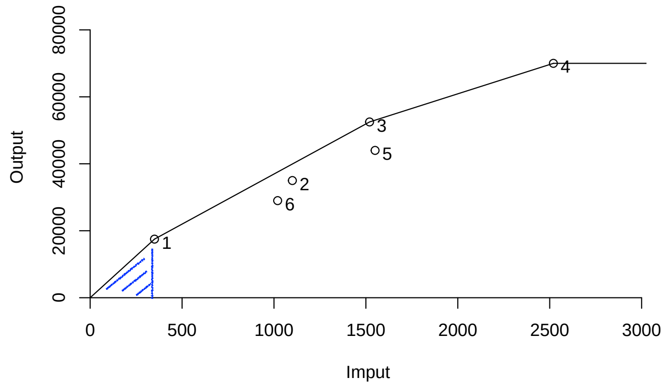


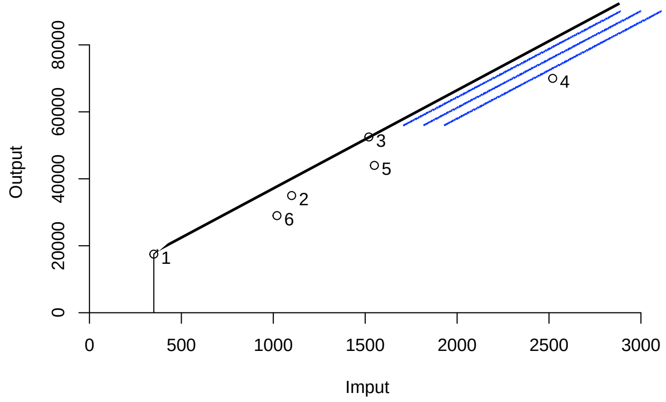
Comparacion between different assumptions

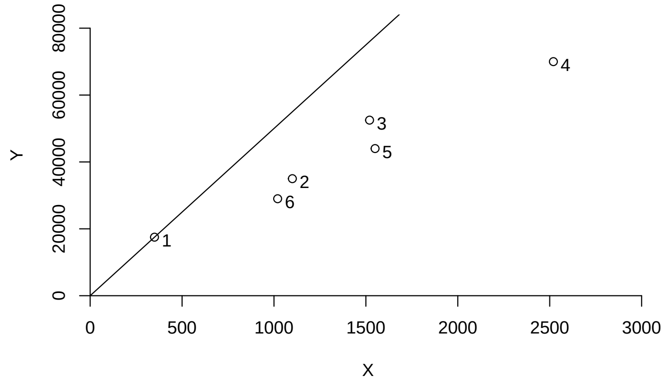
```
dea.plot(input,output,RTS="fdh",ORIENTATION="in-out",
         txt=TRUE, xlab = 'Imput', ylab= 'Output', main="Free disposability hull (FDH) Graph")

dea.plot(input,output,RTS="vrs",ORIENTATION="in-out",
         txt=TRUE, xlab = 'Imput', ylab= 'Output', main="Variable Returns to Scale (VRS) Graph")

dea.plot(input,output,RTS="drs",ORIENTATION="in-out",
         txt=TRUE, xlab = 'Imput', ylab= 'Output', main="Decreasing Returns to Scale (DRS) Graph")

dea.plot(input,output,RTS="irs",ORIENTATION="in-out",
         txt=TRUE, xlab = 'Imput', ylab= 'Output', main="Increasing Returns to Scale (IRS) Graph")

dea.plot(input,output,RTS="crs",ORIENTATION="in-out",
         txt=TRUE, main="Constant Returns to Scale (CRS) Graph")

dea.plot(input,output,RTS="add",ORIENTATION="in-out",
         txt=TRUE, xlab = 'Imput', ylab= 'Output', main="Free Replicability Hull (FRH) Graph")
```

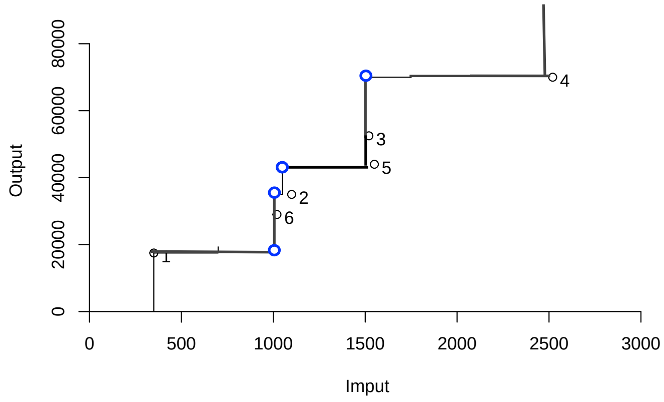

Free disposability hull (FDH) Graph


Variable Returns to Scale (VRS) Graph


Decreasing Returns to Scale (DRS) Graph


Increasing Returns to Scale (IRS) Graph


Constant Returns to Scale (CRS) Graph


Free Replicability Hull (FRH) Graph

These charts allow us to compare the results of each DEA model.

As we learned in this module, “all DEA models share the idea of estimating the technology using a minimal extrapolation approach” (DEA Slides).

As we can see FDH is the smallest technology set. It tries to produce fewer outputs (number of patient-days reimbursed by third-party sources and the number of patient-days reimbursed privately) with more inputs (staffing labor and the cost of supplies). FDH is usually the most wanted model by firms, however, it has some drawbacks due to its assumptions. As we can prove, all the efficiencies in this model are 1, but compared to other models it is not as efficient we think because we find areas/units to improve.

VRS is larger than FDH because it “fills-out” the spaces that FDH reduced. Here we can see that unit 6 can improve its efficiency.

DRS and IRS are larger than VRS as we can see in the charts. DRS tries to increase the set for less input value, while the IRS tries to increase the technology for large input values. DRS indicates that unit 5 and 6 could enhance their efficiency, and IRS shows that facility 6 may improve as well.

CRS is the largest technology set, which allows us to see if there is any possible combination to scale up or down. Based on the efficiency values, units 5 and 6 need to improve.

Regarding FRH, based on the arrow network discussed in class, it is larger than FDH but smaller than CRS, and its goal is to replace deterministic data using random variables. (Bogetoft, P. & Otto, L., 2011, p. 89)

Sources:

Majid, A. Reza, F. (2012). Developing a chance-constrained free replicability hull model for supplier selection. International Journal of Logistics Systems and Management (IJLSM), Vol. 12, No. 4. Retrieved from http://www.inderscience.com/offer.php?id=48365

Bogetoft, P. & Otto, L. (2011). Benchmarking with DEA, SFA, and R. Retrieved from https://books.google.com/books?id=rBiGxrgFk-kC&pg=PA89&lpg=PA89&dq=fdh+is+smallest+and+crs+is+largest&source=bl&ots

Package Benchmarking. (2020). Retrieved from https://cran.r-project.org/web/packages/Benchmarking/Benchmarking.pdf

Frontier Analyst® FAQs. Retrieved from https://banxia.com/frontier/resources/frequent-questions/