Assignment 4

Balaji Ravi Kumar

30/10/2022

# Upload libraries needed  
library(Benchmarking)

## Warning: package 'Benchmarking' was built under R version 4.1.3

## Loading required package: lpSolveAPI

## Warning: package 'lpSolveAPI' was built under R version 4.1.3

## Loading required package: ucminf

## Warning: package 'ucminf' was built under R version 4.1.3

## 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)

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.1.1 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

**Compute the Formulation**

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

# 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

# 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**

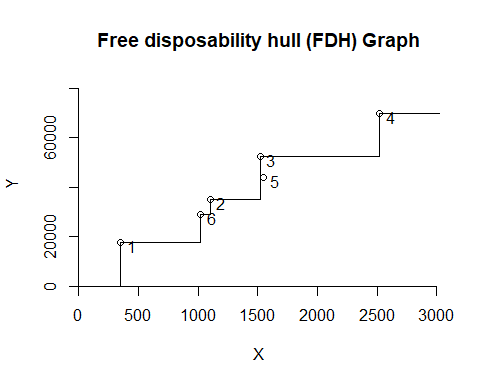
Now, we are going to formulate and compute the DEA analysis using FDH. The Free disposability hull (FDH) is the assumption of dispose unwanted inputs and outputs. “Free disposability means that we can always produce fewer outputs with more inputs.” (DEA Slides)

# 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.

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



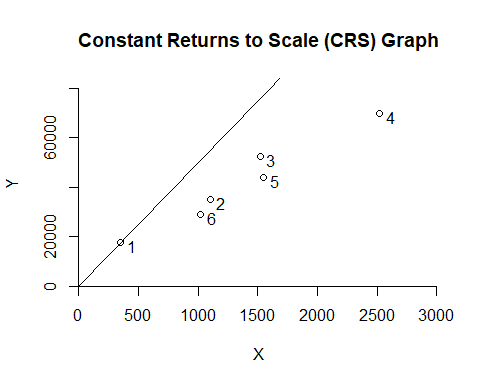
**DEA Analysis using CRS** Now, we are going to formulate and compute the DEA analysis using Constant Returns to Scale (CRS). The CRS is part of the scaling assumption, and it allows us to see if there is any possible combination to scale up or down.

# 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  
## 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

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, main="Constant Returns to Scale (CRS) Graph")



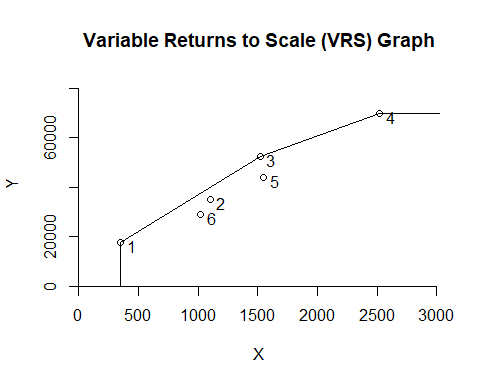
**DEA Analysis using VRS** Now, we are going to formulate and compute the DEA analysis using Variable Returns to Scale (VRS). VRS is also part of the scaling assumption, and it helps to estimate the efficiency of the variables whether an increase or decrease is not proportional.

# 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  
## 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

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, main="Variable Returns to Scale (VRS) Graph")



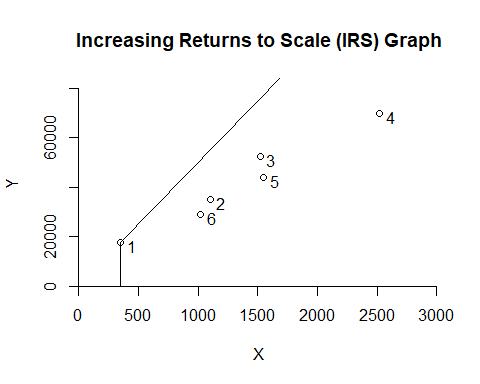
**DEA Analysis using IRS** Now, we are going to formulate and compute the DEA analysis using Increasing Returns to Scale (IRS). IRS indicates if it is possible to increate the operation scale.

# 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  
## 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

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 , main="Increasing Returns to Scale (IRS) Graph")



**DEA Analysis using DRS**

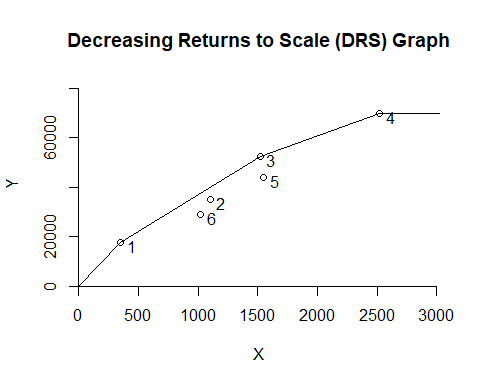
Now, we are going to formulate and compute the DEA analysis using Decreasing Returns to Scale (DRS). DRS is the opposite of IRS, which its goal is to decrease the operation scale on any possible production process.

# 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")  
  
# 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  
## 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) 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.

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



**DEA Analysis using FRH**

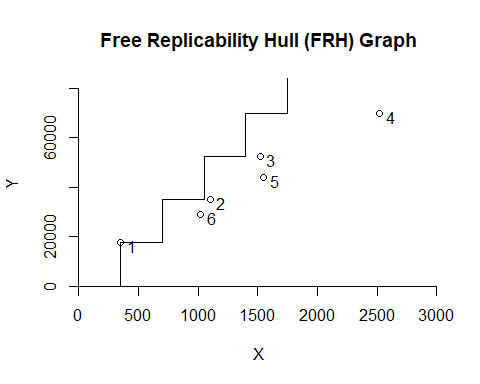
Now, we are going to formulate and compute the DEA analysis using Free Replicability Hull (FRH). FRH as well as FDH use mixed integer programming, which refers that the variables must be integers tofind the optimal solution. The goal of FRH is to replace deterministic data using random variables.

# 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

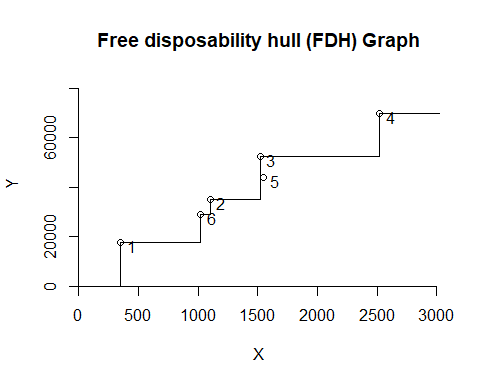
Free Replicability Hull (FRH) has a great efficiency in all its DMU. It behives the same as Free disposability hull (FDH), which all its values have their own peer, lambas and efficiency of 1.

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

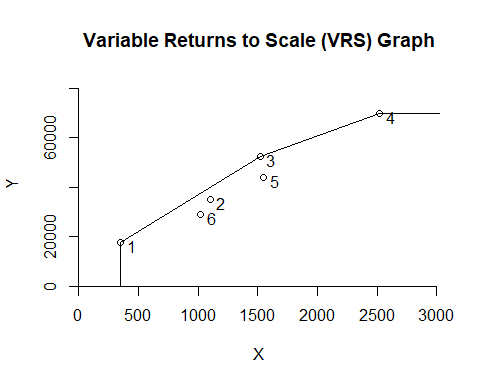


**Comparacion between different assumptions**

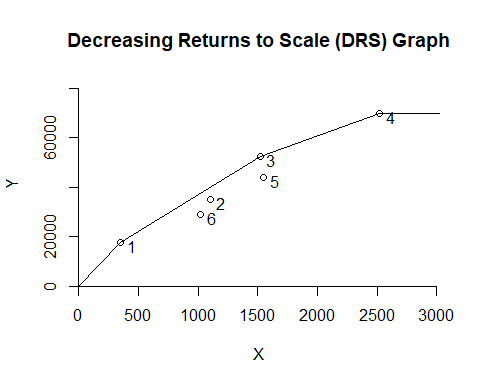
dea.plot(input,output,RTS="fdh",ORIENTATION="in-out",txt=TRUE, main="Free disposability hull (FDH) Graph")



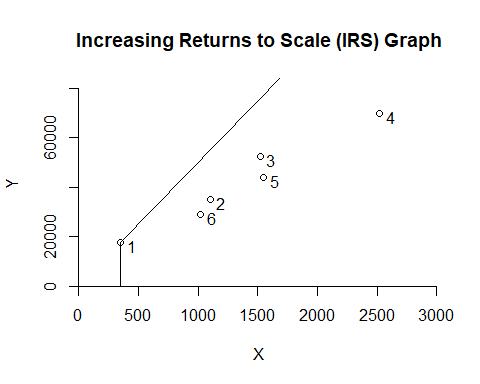
dea.plot(input,output,RTS="vrs",ORIENTATION="in-out",txt=TRUE, main="Variable Returns to Scale (VRS) Graph")



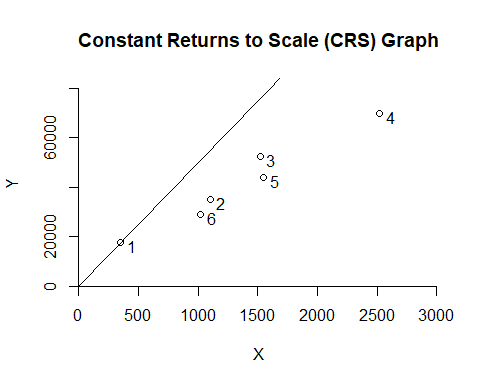
dea.plot(input,output,RTS="drs",ORIENTATION="in-out",txt=TRUE, main="Decreasing Returns to Scale (DRS) Graph")



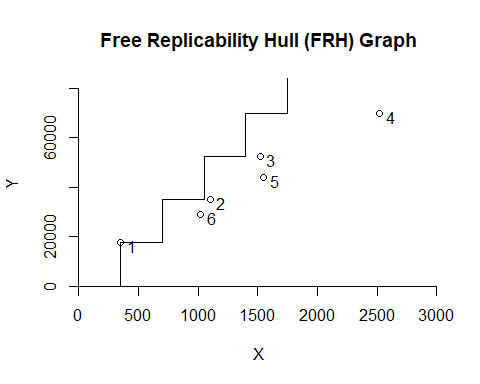
dea.plot(input,output,RTS="irs",ORIENTATION="in-out",txt=TRUE, main="Increasing Returns to Scale (IRS) Graph")



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, main="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 patientdays 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.