QMM Assignment DEA

Pavan Chaitanya

2022-10-26

```
# Upload libraries needed
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

library(tidyverse)

## ── Attaching packages
## ─────────────────────────────────────
## tidyverse 1.3.2 ──

## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.0      ✓ stringr 1.4.1
## ✓ readr   2.1.2      ✓ forcats 0.5.2
## ── Conflicts ──────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ 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, 0.2, 0.7, 1.2, 2.0, 1.2,
0.7),ncol = 2)
output <-
matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,
25000, 15000),ncol = 2)

# Assign column names
```

```
colnames(input) <- c("staff_hours_daily","supplies_daily")
colnames(output) <- c("reimbursed_patient_daily", "privately_paid_patient-
daily")

# To see the values of Input
input

##      staff_hours_daily supplies_daily
## [1,]               150            0.2
## [2,]               400            0.7
## [3,]               320            1.2
## [4,]               520            2.0
## [5,]               350            1.2
## [6,]               320            0.7

# To see the values of Output
output

##      reimbursed_patient_daily privately_paid_patient-daily
## [1,]                    14000                          3500
## [2,]                    14000                         21000
## [3,]                    42000                         10500
## [4,]                    28000                         42000
## [5,]                    19000                         25000
## [6,]                    14000                         15000

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

#We will conduct a Data Envelopment Analysis (DEA) in the part that follows.
The DEA is an analytical method that can assist businesses in identifying and
allocating their resources to improve their efficiency and have better
practices.
```

1. Formulate and perform DEA analysis under all DEA assumptions of FDH, CRS, VRS, IRS, DRS, and FRH.

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
```

```r
eff_fdh <- as.data.frame(analysis_fdh$eff)

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

DEA Analysis using CRS

```r
# 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")
```

DEA Analysis using VRS

```r
# 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")
```

DEA Analysis using IRS

```r
# 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
```

```r
eff_irs <- as.data.frame(analysis_irs$eff)

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

DEA Analysis using DRS

```r
# 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")
```

DEA Analysis using FRH

```r
# 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.

# 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")
```

2. Determine the Peers and Lambdas under each of the above assumptions

Peers and Lambdas for FDH

```r
# 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")
```

Peers and Lambdas for 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")
```

Peers and Lambdas for 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")
```

Peers and Lambdas for 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")
```

Peers and Lambdas for 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")
```

Peers and Lambdas for 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")
```

3. Summarize your results in a tabular format

FDH Results in Tabular form

```
# 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

# 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.
```

CRS Results in Tabular form

```
# 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.
```

VRS Results in Tabular form

```
# 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.
```

IRS Results in Tabular form

```
# 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%.
```

DRS Results in Tabular form

```
# 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.
```

FRH Results in Tabular form

```
# 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
```

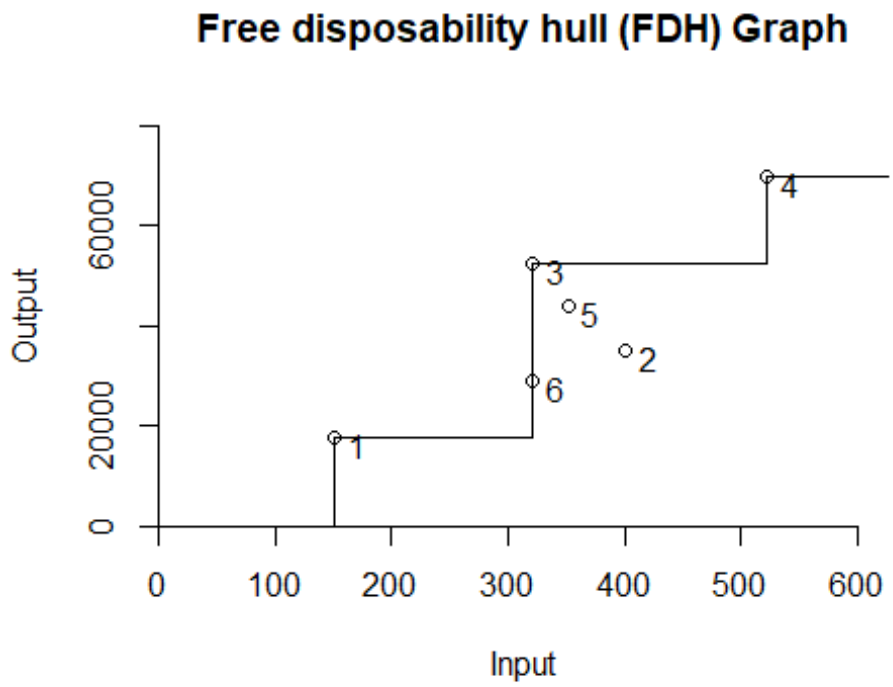
```
## 5         5      0      0      0      0      1      0              1
## 6         6      0      0      0      0      0      1              1

# 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.
```

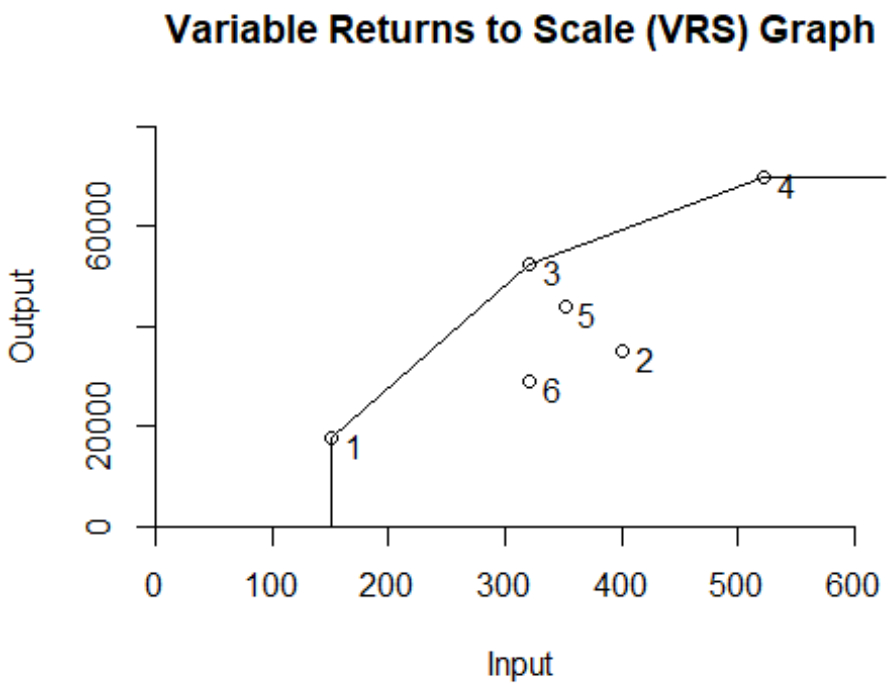
4. Compare and contrast the above results

Graphs for all FDH,CRS,VRS,IRS,DRS,FRH.

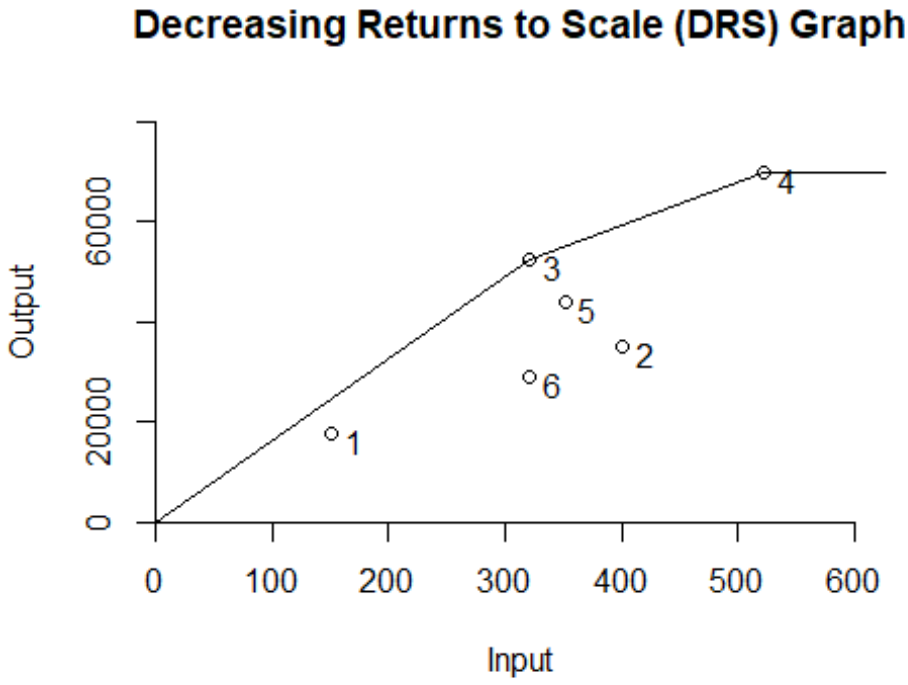
```
dea.plot(input,output,RTS="fdh",ORIENTATION="in-out",txt=TRUE, xlab ="Input",
ylab= "Output", main="Free disposability hull (FDH) Graph")
```



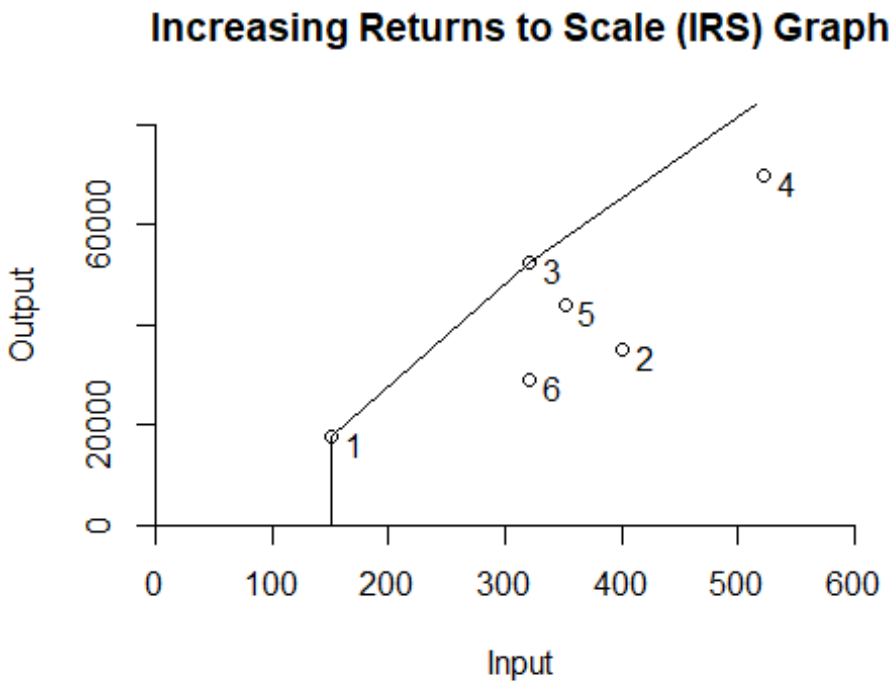

```
dea.plot(input,output,RTS="vrs",ORIENTATION="in-out",
txt=TRUE, xlab = "Input", ylab= "Output", main="Variable Returns to Scale
(VRS) Graph")
```



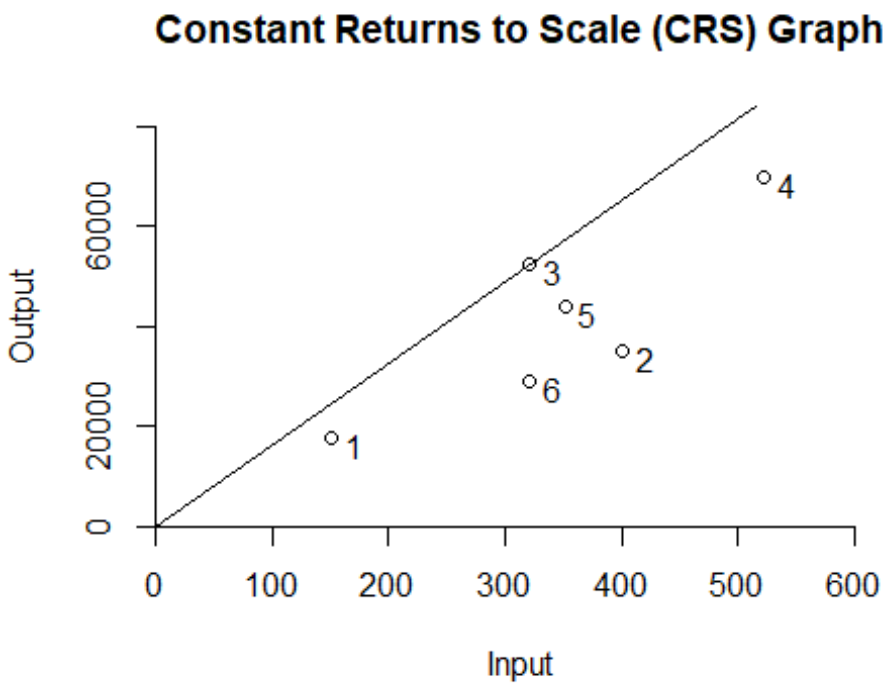

```
dea.plot(input,output,RTS="drs",ORIENTATION="in-out",
txt=TRUE, xlab = "Input", ylab= "Output", main="Decreasing Returns to Scale
(DRS) Graph")
```



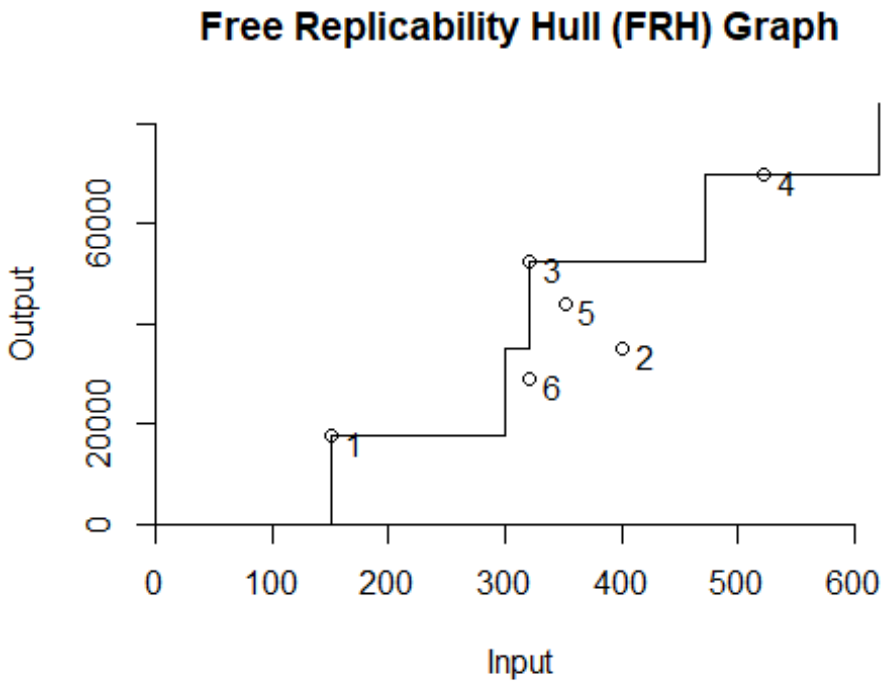
```
dea.plot(input,output,RTS="irs",ORIENTATION="in-out",
txt=TRUE, xlab = "Input", ylab= "Output", main="Increasing Returns to Scale
(IRS) Graph")
```




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




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



```
#We may compare the outcomes of each DEA model using these charts.

# The principle of estimating the technology via a minimal extrapolation
strategy is one that all DEA models share, as we learnt in this session (DEA
Slides).

# FDH is the smallest technology set, as can be shown. It aims to generate
more inputs from fewer outputs (the number of patient days funded privately
and the number of patient days reimbursed by third parties) (staffing labor
and the cost of supplies). FDH is typically the model that businesses want
the most, however because of its assumptions, it has several limitations. All
of the efficiencies in this model are 1, as we can demonstrate, but when
compared to other models, it is not as efficient as we believe it to be since
we identify areas/units for improvement.

# Because VRS "fills-out" the gaps that FDH eliminated, it is larger than
FDH. We can observe that unit 6 can increase its effectiveness in this area.

# The charts show that DRS and IRS are bigger than VRS. While the IRS seeks
to boost technology for high input values, DRS tries to increase the set for
lower input values. DRS suggests that units 5 and 6 might become more
efficient, while IRS suggests that facility 6 might as well.

# The largest technology set is CRS, which enables us to determine whether
any combinations could be used to scale up or down. The efficiency results
indicate that units 5 and 6 require improvement.

# FRH aims to replace deterministic data with random variables, and is larger
than FDH but less than CRS based on the arrow network mentioned in class.
```