DEA Assignment

2022-10-29

```
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
```

```
library("Benchmarking")
x <- matrix(c("Facility 1","Facility 2","Facility 3","Facility 4","Facility 5", "Facility 6",
              150,400,320,520,350,320,
              0.2,0.7,1.2,2.0,1.2,0.7,
              14000,14000,42000,28000,19000,14000,
              3500,21000,10500,42000,25000,15000), ncol=5, byrow=F)

colnames(x) <- c("DMU", "Staff_Hours_Per_Day","Supplies_Per_Day","Reimbursed_Patient_Days","Privately_Pa

table.df <- as.table(x)
table.df
```

```
##   DMU        Staff_Hours_Per_Day Supplies_Per_Day Reimbursed_Patient_Days
## A Facility 1 150                 0.2              14000
## B Facility 2 400                 0.7              14000
## C Facility 3 320                 1.2              42000
## D Facility 4 520                 2                28000
## E Facility 5 350                 1.2              19000
## F Facility 6 320                 0.7              14000
##   Privately_Paid_Patient_Days
## A 3500
## B 21000
## C 10500
## D 42000
## E 25000
## F 15000
```

#Calculating Constant Returns to Scale (CRS)

```
x <- matrix(c(150,400,320,520,350,320,
              0.2,0.7,1.2,2.0,1.2,0.7),ncol=2)

y <- matrix(c(14000,14000,42000,28000,19000,14000,
              3500,21000,10500,42000,25000,15000),ncol=2)

colnames(y) <- c("Reimbursed_Patient_Days","Privately_Paid_Patient_Days")

colnames(x) <- c("Staff_Hours_Per_Day","Supplies_Per_Day")

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

```
## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675
```

```
peers(DEA_CRS)
```

```
##      peer1 peer2 peer3
## [1,]     1    NA    NA
## [2,]     2    NA    NA
## [3,]     3    NA    NA
## [4,]     4    NA    NA
## [5,]     1     2     4
## [6,]     1     2     4
```

```
lambda(DEA_CRS)
```

```
##             L1         L2 L3        L4
## [1,] 1.0000000 0.00000000  0 0.0000000
## [2,] 0.0000000 1.00000000  0 0.0000000
## [3,] 0.0000000 0.00000000  1 0.0000000
## [4,] 0.0000000 0.00000000  0 1.0000000
## [5,] 0.2000000 0.08048142  0 0.5383307
## [6,] 0.3428571 0.39499264  0 0.1310751
```

#CRS Observations *1. We observe how well Facility 1, Facility 2, Facility 3, and Facility 4 operate. 2. Also, we get to see that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities. 3. Facility 5 is 97% efficient, leaving 2.25 percent of inefficiency, and Facility 6 is 86.75% efficient, leaving 13.25 percent of inefficiency.*

#Calculating Decreasing Returns to Scale (DRS)

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

```
## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675
```

```
peers(DEA_DRS)
```

```
##      peer1 peer2 peer3
## [1,]     1    NA    NA
## [2,]     2    NA    NA
## [3,]     3    NA    NA
## [4,]     4    NA    NA
## [5,]     1     2     4
## [6,]     1     2     4
```

```
lambda(DEA_DRS)
```

```
##             L1         L2 L3        L4
## [1,] 1.0000000 0.00000000  0 0.0000000
## [2,] 0.0000000 1.00000000  0 0.0000000
## [3,] 0.0000000 0.00000000  1 0.0000000
## [4,] 0.0000000 0.00000000  0 1.0000000
## [5,] 0.2000000 0.08048142  0 0.5383307
## [6,] 0.3428571 0.39499264  0 0.1310751
```

#DRS Observations -
1.We observe how well Facilities 1, Facility 2, Facility 3, and Facility 4 operate.

2. Also, we get to see that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities.

3.Facilities 5 and 6 are both 96.75% efficient, leaving 2.25% and 13.25% of inefficiency, respectively..

#Calculating Increasing Returns to Scale (IRS)

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

```
## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963
```

```
peers(DEA_IRS)
```

```
##      peer1 peer2 peer3
## [1,]     1    NA    NA
## [2,]     2    NA    NA
## [3,]     3    NA    NA
## [4,]     4    NA    NA
## [5,]     5    NA    NA
## [6,]     1     2     5
```

```
lambda(DEA_IRS)
```

```
##             L1        L2 L3 L4        L5
## [1,] 1.0000000 0.0000000  0  0 0.0000000
## [2,] 0.0000000 1.0000000  0  0 0.0000000
## [3,] 0.0000000 0.0000000  1  0 0.0000000
## [4,] 0.0000000 0.0000000  0  1 0.0000000
## [5,] 0.0000000 0.0000000  0  0 1.0000000
## [6,] 0.4014399 0.3422606  0  0 0.2562995
```

#IRS Observations - *1. We observe how well Facilities 1, Facility 2, Facility 3, and Facility 4 operate. 2. Also, we get to see that Facility 1, Facility 2 and Facility 5 are the peer members for Facility 6 which is the only inefficient facility. 3.Facility 6 has an efficiency of 89.63%, leaving a 10.37% inefficiency.*

#Calculating Variable Returns to Scale (VRS)

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

```
## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963
```

```
peers(DEA_VRS)
```

```
##      peer1 peer2 peer3
## [1,]     1    NA    NA
## [2,]     2    NA    NA
## [3,]     3    NA    NA
## [4,]     4    NA    NA
## [5,]     5    NA    NA
## [6,]     1     2     5
```

```
lambda(DEA_VRS)
```

```
##             L1        L2 L3 L4        L5
## [1,] 1.0000000 0.0000000  0  0 0.0000000
## [2,] 0.0000000 1.0000000  0  0 0.0000000
## [3,] 0.0000000 0.0000000  1  0 0.0000000
## [4,] 0.0000000 0.0000000  0  1 0.0000000
## [5,] 0.0000000 0.0000000  0  0 1.0000000
## [6,] 0.4014399 0.3422606  0  0 0.2562995
```

VRS Observations -
1. We observe how well Facilities 1, Facility 2, Facility 3, and Facility 4 operate. 2. Also, we get to see that Facility 1, Facility 2 and Facility 5 are the peer members for Facility 6 which is the only inefficient facility. 3.89.63% of Facility 6 is efficient, leaving 10.37% inefficient.

#Calculating Free Disposability Hull (FDH)

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

```
## [1] 1 1 1 1 1 1
```

```
peers(DEA_FDH)
```

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

```
lambda(DEA_FDH)
```

```
##      L1 L2 L3 L4 L5 L6
## [1,]  1  0  0  0  0  0
## [2,]  0  1  0  0  0  0
## [3,]  0  0  1  0  0  0
## [4,]  0  0  0  1  0  0
## [5,]  0  0  0  0  1  0
## [6,]  0  0  0  0  0  1
```

FDH Observations -

The DMUs are all effective. This is primarily because the FDH technique thereby to the concept, which allows it to identify even a very low level of efficiency..

#Calculating Free Replicability Hull (FRH)

```r
#here FRH is calculated by specifying RTS = "add"
DEA_FRH <- dea(x, y, RTS = "add")
DEA_FRH
```

```
## [1] 1 1 1 1 1 1
```

```r
peers(DEA_FRH)
```

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

```r
lambda(DEA_FRH)
```

```
##      L1 L2 L3 L4 L5 L6
## [1,]  1  0  0  0  0  0
## [2,]  0  1  0  0  0  0
## [3,]  0  0  1  0  0  0
## [4,]  0  0  0  1  0  0
## [5,]  0  0  0  0  1  0
## [6,]  0  0  0  0  0  1
```

#FRH Observations - *The DMUs are all effective. It thereby to the no convexity assumption, which guarantees that the output is not intended for reuse or duplication..*

#Summary of Results (Inefficient DMUs)

```r
data.df.summarise.inefficient <- matrix(c("CRS","DRS","IRS","VRS","FDH","FRH",
2,2,1,1,0,0,
"Facility 5 & 6", "Facility 5 & 6","Facility 6", "Facility 6", "-","-",
"97.75% & 86.7%","97.75% & 86.7%","89.63%","89.63%","-","-",
"Facility 1, 2 & 4","Facility 1, 2 & 4","Facility 1, 2 & 5","Facility 1, 2 & 5","-","-",
"0.2, 0.08, 0.54 and 0.34, 0.4, 0.13", "0.2, 0.08, 0.54 and 0.34, 0.4, 0.13", "0.4, 0.34 and 0.26", "0.4


colnames(data.df.summarise.inefficient) <- c("RTS","Count_Inefficient_DMUs","Name_DMUs","%_Inefficiency

as.table(data.df.summarise.inefficient)
```

```
##   RTS Count_Inefficient_DMUs Name_DMUs      %_Inefficiency Peers
## A CRS 2                     Facility 5 & 6 97.75% & 86.7% Facility 1, 2 & 4
## B DRS 2                     Facility 5 & 6 97.75% & 86.7% Facility 1, 2 & 4
## C IRS 1                     Facility 6     89.63%         Facility 1, 2 & 5
## D VRS 1                     Facility 6     89.63%         Facility 1, 2 & 5
## E FDH 0                     -              -              -
## F FRH 0                     -              -              -
##   Lambda
```

```
## A 0.2, 0.08, 0.54 and 0.34, 0.4, 0.13
## B 0.2, 0.08, 0.54 and 0.34, 0.4, 0.13
## C 0.4, 0.34 and 0.26
## D 0.4, 0.34 and 0.26
## E -
## F -
```

#Summary of Results (Efficient DMUs)

```
data.df.summarise.efficient <- matrix(c("CRS","DRS","IRS","VRS","FDH","FRH",
"Facility 1, 2, 3 & 4","Facility 1, 2, 3 & 4","Facility 1, 2, 3, 4 & 5", "Facility 1, 2, 3, 4 & 5", "Al

colnames(data.df.summarise.efficient) <- c("RTS", "Efficient_DMUs")

as.table(data.df.summarise.efficient)
```

```
##   RTS Efficient_DMUs
## A CRS Facility 1, 2, 3 & 4
## B DRS Facility 1, 2, 3 & 4
## C IRS Facility 1, 2, 3, 4 & 5
## D VRS Facility 1, 2, 3, 4 & 5
## E FDH All DMUs
## F FRH All DMUs
```

#Interpretation of the DEA Analysis - *It's crucial to understand the variations between the scales before interpreting the results (RTS),*

Constant Returns to Scale (CRS) is considered as the original scale which is widely used by most of the firms.

The dispersion scales known as Decreasing, Increasing and Varying Returns to Scale (DRS, IRS, and VRS) aid us in determining what to increase and what to decrease based on the deployment of information..

The Free Disposability and Free Replicability Hull (FDH & FRH), which does not make use of the convexity assumption, is regarded as the non-parametric way to evaluate the effectiveness of DMUs..

#CRS - Constant Returns to Scale***

The findings show that DMUs 1, 2, 3, and 4 are effective. Only 97.75% of DMU(5) and 86.7% of DMU(6) are effectively used. On the basis of our initial analysis, we discovered this. In addition, DMU(4peer)'s units are 1, 2, and 4, with respective weights of 0.2, 0.08, and 0.54. The peer units for DMU(6) are 1, 2, and 4, and their weights are 0.34, 0.4, and 0.13, respectively..

In essence, CRS enables us to determine whether any potential DMUs can be scaled up or down; in this instance, DMUs 1, 2, 3 and 4 can be scaled up.

#DRS - Decreasing Returns to Scale***

The findings show that DMUs 1, 2, 3, and 4 are effective. Only 97.75% of DMU(5) and 86.7% of DMU(6) are efficient. On the basis of our initial analysis, we discovered this. In addition, DMU(4peer)'s units are 1, 2, and 4, with respective weights of 0.2, 0.08, and 0.54. The peer units for DMU(6) are 1, 2, and 4, and their weights are 0.34, 0.4, and 0.13, respectively.

By looking at the inefficient DMUs, in this case DMUs 5 and 6, we may determine whether there are any alternative DMUs where we can scale the processes. As the base original scale, the CRS values can also be used to obtain this information.

#IRS - Increasing Returns to Scale***

The outcomes show that DMUs 1, 2, 3, 4, and 5 are productive. Only 89.63% of the DMU(6) is effective. On the basis of our initial analysis, we discovered this. Additionally, DMU(6peer)'s units are 1, 2, and 5, with respective relative weights of 0.4, 0.34, and 0.26..

By examining the efficiency scores, it enables any organization to determine if it can arbitrarily expand its operational footprint. (See table data.df.summarize.efficient)

#VRS - Variable Returns to Scale***

The outcomes show that DMUs 1, 2, 3, 4, and 5 are productive. Only 89.63% of the DMU(6) is effective. On the basis of our initial analysis, we discovered this. Additionally, DMU(6peer)'s units are 1, 2, and 5, with respective relative weights of 0.4, 0.34, and 0.26.

Understanding the scale of operations with changes to the input and output component, either by increasing or decreasing or by using both, is made possible by varying or variable returns to scale.

#FDH - Free Disposability Hull***

The outcomes show that every DMU is effective. This is primarily because the scale is able to measure even the lowest degree of efficiency because there is no convexity assumption.

#FRH - Free Replicability Hull***

All DMUs are effective, according to the FRH data. This is mainly because there isn't a convexity assumption used, and most of the time, this technique enables the scale to capture even the tiniest amount of efficiency that is not subject to replication or disposal.

Note -Only the DMUs that are inefficient would be able to retrieve the peer values, also known as neighbors and lambda values, or weights of the peers. Lambda weights and peers are absent from efficient DMUs.

Conclusion

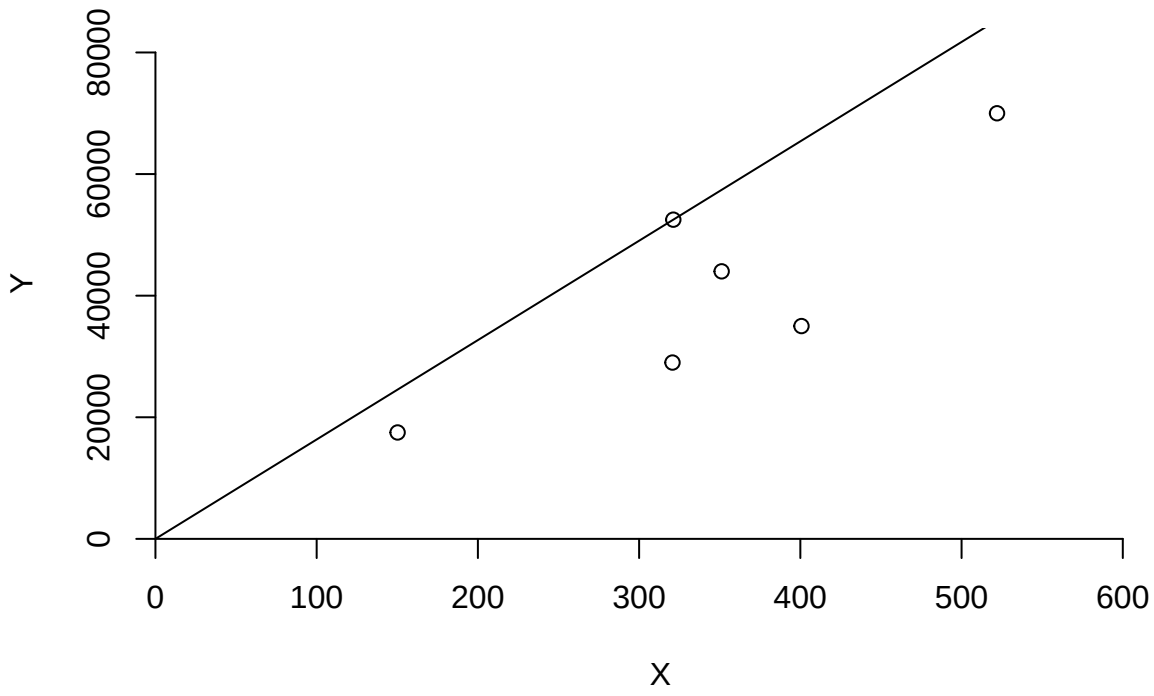
It is crucial to keep in mind that DEA is an extremely helpful tool for any company in determining which of the Decision Making Units (DMUs) has to be maximized so that there would be an increase, decrease, or any other modifications to the output by feeding input into it.

Additionally, a corporation can choose which RTS, or Returns to Scale, it wants to use based on its needs; each of these scales has a unique significance.

#Plotting the Graphs***

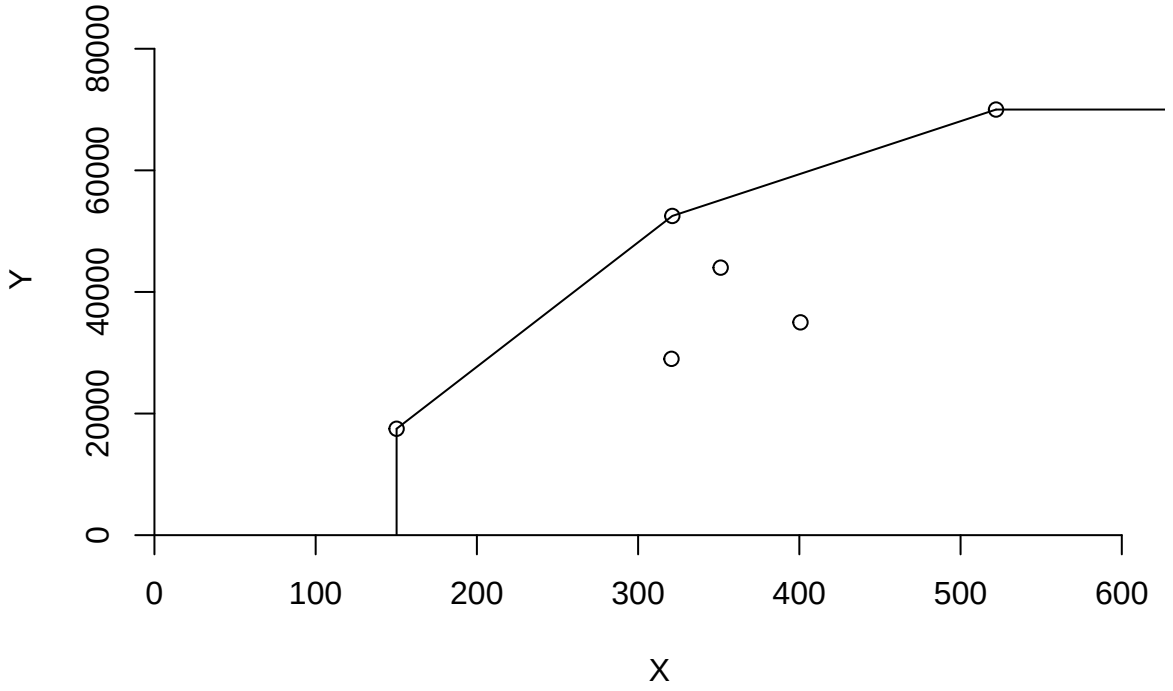
CRS Plot***

```
dea.plot(x, y, RTS='crs')
```



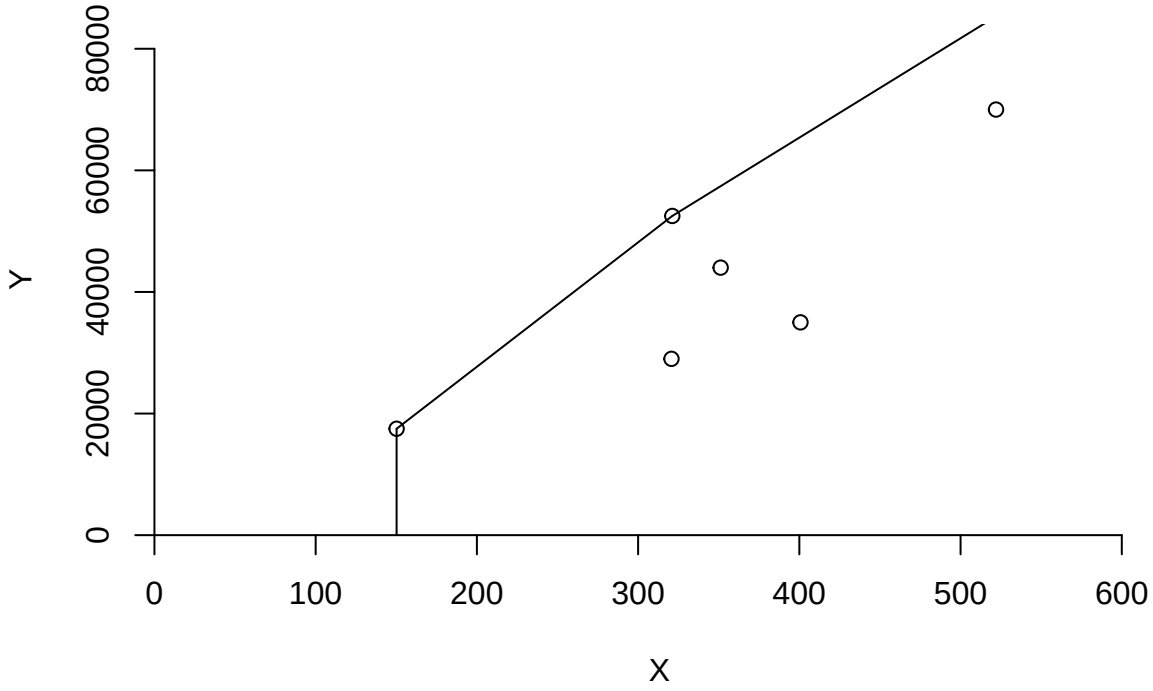
DRS Plot

```
dea.plot(x,y,RTS="vrs")
```



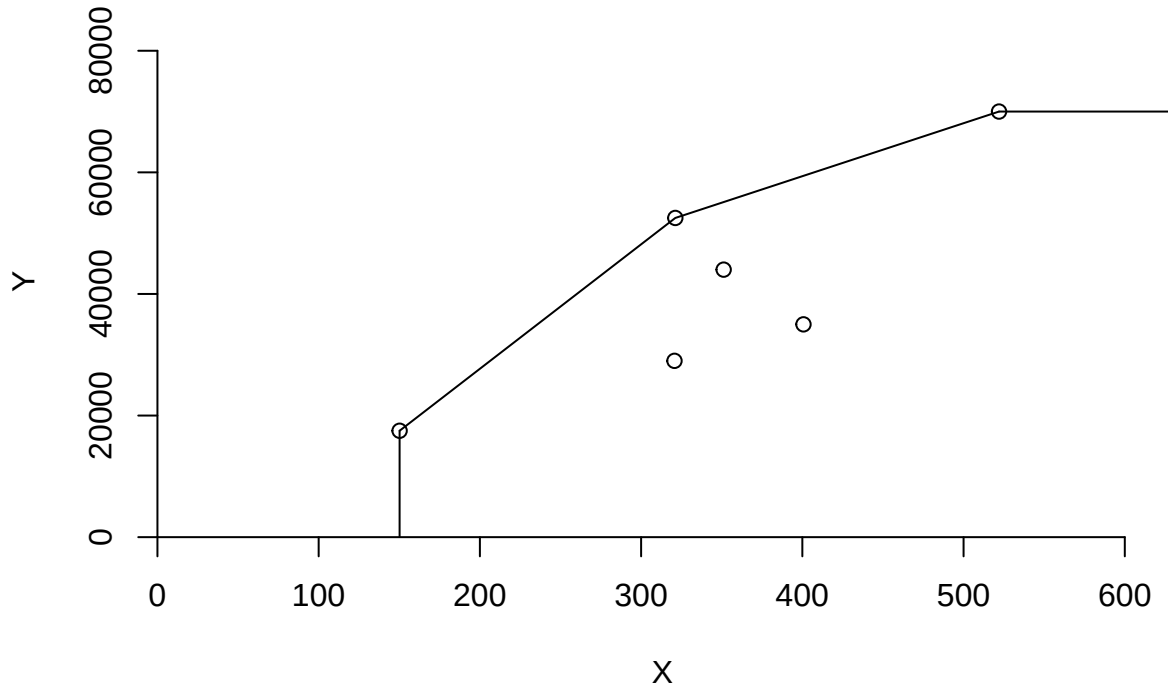
IRS Plot

```
dea.plot(x,y,RTS="irs")
```



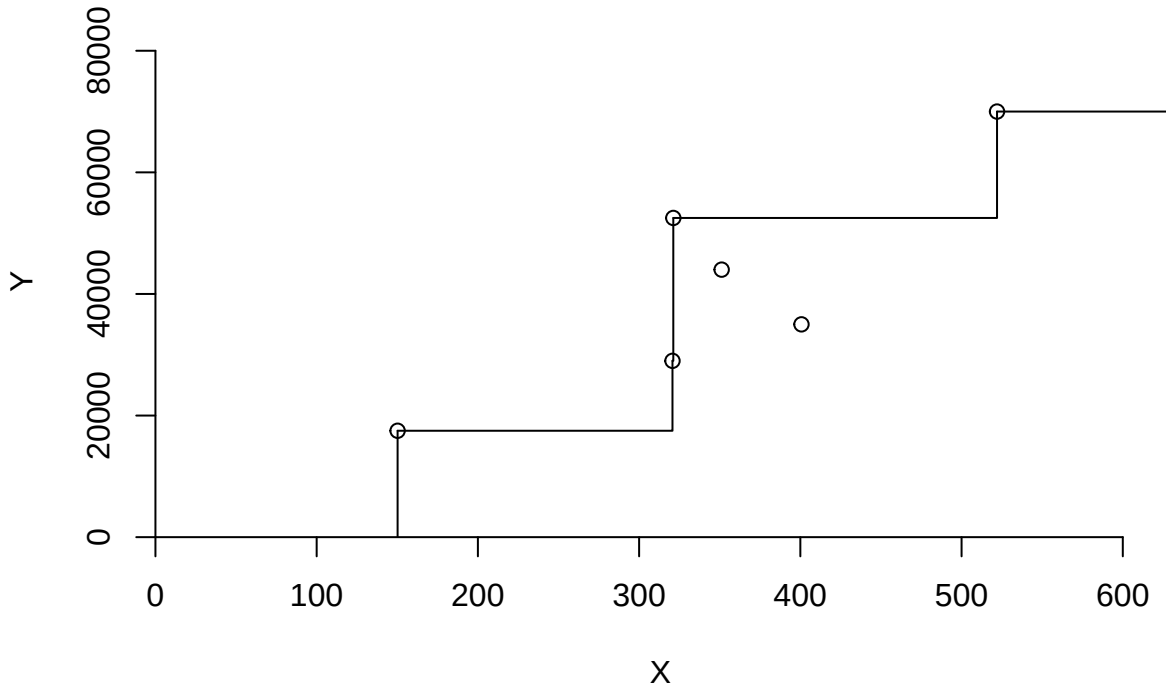
VRS Plot

```
dea.plot(x,y,RTS="vrs")
```



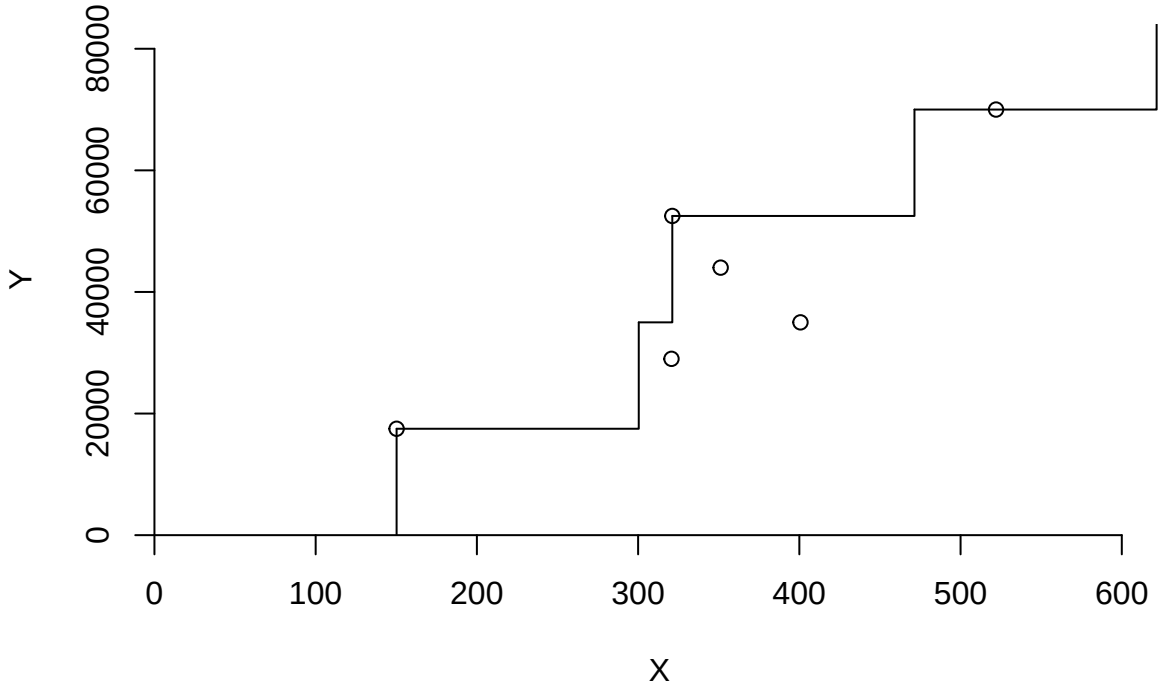
FDH Plot

```
dea.plot(x,y,RTS="fdh")
```



FRH Plot

```
dea.plot(x,y,RTS="add")
```


Y
0
20000
40000
60000
80000
0
100
200
300
400
500
600
X