DEA(Data Envelopment Analysis)

Nikhil Reddy Addula

2022-10-31

```
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
```

```
NR <- 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(NR) <- c("DMU", "Staff_Hours_Per_Day","Supplies_Per_Day","Reimbursed_Patient_Days","Privately_I

table.df <- as.table(NR)
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)

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

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

```r
colnames(R) <- c("Reimbursed_Patient_Days","Privately_Paid_Patient_Days")

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

MY_DEA_CRS<-dea(N,R, RTS = "crs")
MY_DEA_CRS
```

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

```r
peers(MY_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
```

```r
lambda(MY_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. Our ability to observe the effectiveness of Facilities 1, 2, 3, 4, and 5 is demonstrated.
2. Additionally, we learn that Facility 6—the lone ineffective facility—has as peer members Facility 1, Facility 2, and Facility 5.
3. 10.37% of Facility 6 is inefficient, leaving 89.63% efficient.

#Calculating Decreasing Returns to Scale (DRS)

```r
MY_DEA_DRS <- dea(N, R, RTS = "drs")
MY_DEA_DRS
```

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

```r
peers(MY_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
```

```r
lambda(MY_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. Our ability to observe the effectiveness of Facilities 1, 2, 3, 4, and 5 is demonstrated.
2. Additionally, we learn that Facility 6—the lone ineffective facility—has as peer members Facility 1, Facility 2, and Facility 5.
3. 10.37% of Facility 6 is inefficient, leaving 89.63% efficient.

#Calculating Increasing Returns to Scale (IRS)

```
MY_DEA_IRS <- dea(N, R, RTS = "irs")
MY_DEA_IRS
```

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

```
peers(MY_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(MY_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. Our ability to observe the effectiveness of Facilities 1, 2, 3, 4, and 5 is demonstrated.
2. Additionally, we learn that Facility 6—the lone ineffective facility—has as peer members Facility 1, Facility 2, and Facility 5.
3. 10.37% of Facility 6 is inefficient, leaving 89.63% efficient.

#Calculating Variable Returns to Scale (VRS)

```
MY_DEA_VRS <- dea(N, R, RTS = "vrs")
MY_DEA_VRS
```

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

```
peers(MY_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
```

```r
lambda(MY_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
```

#Observations made by VRS
1. Our ability to observe the effectiveness of Facilities 1, 2, 3, 4, and 5 is demonstrated.
2. Additionally, we learn that Facility 6—the lone ineffective facility—has as peer members Facility 1, Facility 2, and Facility 5.
3. 10.37% of Facility 6 is inefficient, leaving 89.63% efficient.

#Calculating Free Disposability Hull (FDH)

```r
MY_DEA_FDH <- dea(N, R, RTS = "fdh")
MY_DEA_FDH
```

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

```r
peers(MY_DEA_FDH)
```

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

```r
lambda(MY_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 -

Every DMU is effective. This is mostly caused by the principle that the FDH approach follows, which allows it to identify even a very low degree of efficiency.

#Calculating Free Replicability Hull (FRH)

```r
#here FRH is calculated by specifying RTS = "add"
MY_DEA_FRH <- dea(N, R, RTS = "add")
MY_DEA_FRH
```

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

```
peers(MY_DEA_FRH)
```

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

```
lambda(MY_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- *Every DMU is effective. Because it adheres to the no convexity assumption, the output is protected against disposal and reproduction.*

#Summary of Results (Inefficient DMUs)

```
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", "All
```

```r
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: *Before the interpretation, it's critical to understand the RTS scale differences.*

The original scale, Constant Returns to Scale (CRS), is regarded as being utilized by the majority of businesses.

The Decreasing, Increasing and Varying Returns to Scale (DRS, IRS and VRS) are the dispersion scales that assist us in determining which inputs to increase and which to reduce based on input deployment.

The Free Disposability and Free Replicability Hull (FDH & FRH) are regarded as the non-parametric technique to quantify the effectiveness of the DMUs, i.e., there is no convexity assumption.

#Constant Returns to Scale (#CRS)***

The findings show that DMUs 1, 2, 3, and 4 are productive. Only 97.75% of DMU(5) and 86.7% of DMU(6) are effectively used. On the basis of our initial study, 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, with corresponding weights of 0.34, 0.4, and 0.13.

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

#Decreasing Returns to Scale (#DRS)***

The findings show that DMUs 1, 2, 3, and 4 are productive. Only 97.75% of DMU(5) and 86.7% of DMU(6) are efficient. On the basis of our initial study, 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, with corresponding weights of 0.34, 0.4, and 0.13.

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

#Increasing Returns to Scale (IRS)***

The findings 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 study, 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.

As the name implies, by examining the efficiency ratings, it informs any company if it may arbitrarily expand its operations.

(See table in data.df.summarise.efficient)

#Variable Returns to Scale (VRS)***

The findings 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 study, 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 processes with modifications toward the input and output factor—by increasing or lowering, or by using both—is made possible by the concept of varying or variable returns to scale.

#Free Disposability Hull(FDH)***

The findings show that all DMUs are effective. This is primarily because there is no convexity assumption, and it is mainly because of this method that the scale is able to measure even the lowest degree of efficiency.

#Free Replicability Hull (FRH)***

The FRH results show that all the DMUs are effective. This is mostly because there is no assumption regarding convexity, and this method enables the scale to capture even the tiniest amount of efficiency that is not subject to duplication or disposal.

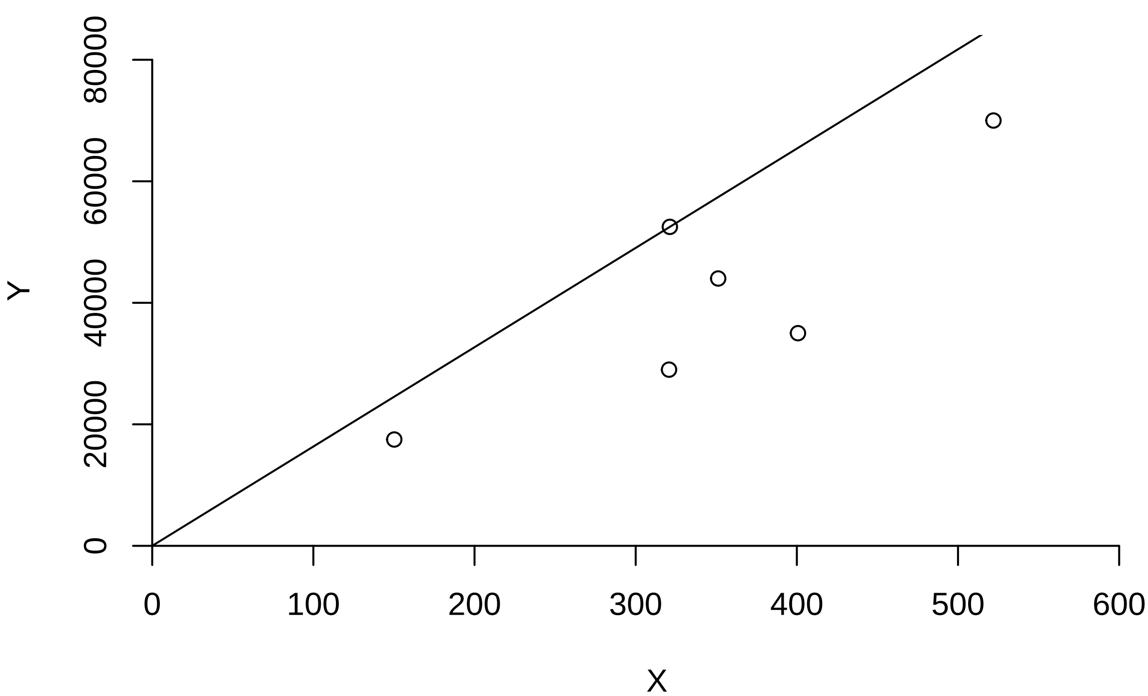
Note - Only inefficient DMUs will get the peer values, also known as neighbors and lambda values, or weights of the peers. Peers and lambda weights are not present in efficient DMUs.

Conclusion
It is important to keep in mind that DEA is a very helpful tool for any organization in determining which of the Decision Making Units (DMUs) has to be maximized so that there would be an increase, reduction, or any other sort of changes to the output by feeding input into it.

Additionally, a corporation may choose which RTS, or Returns to Scale, it wishes to use based on its needs; each of these scales has a different level of significance.

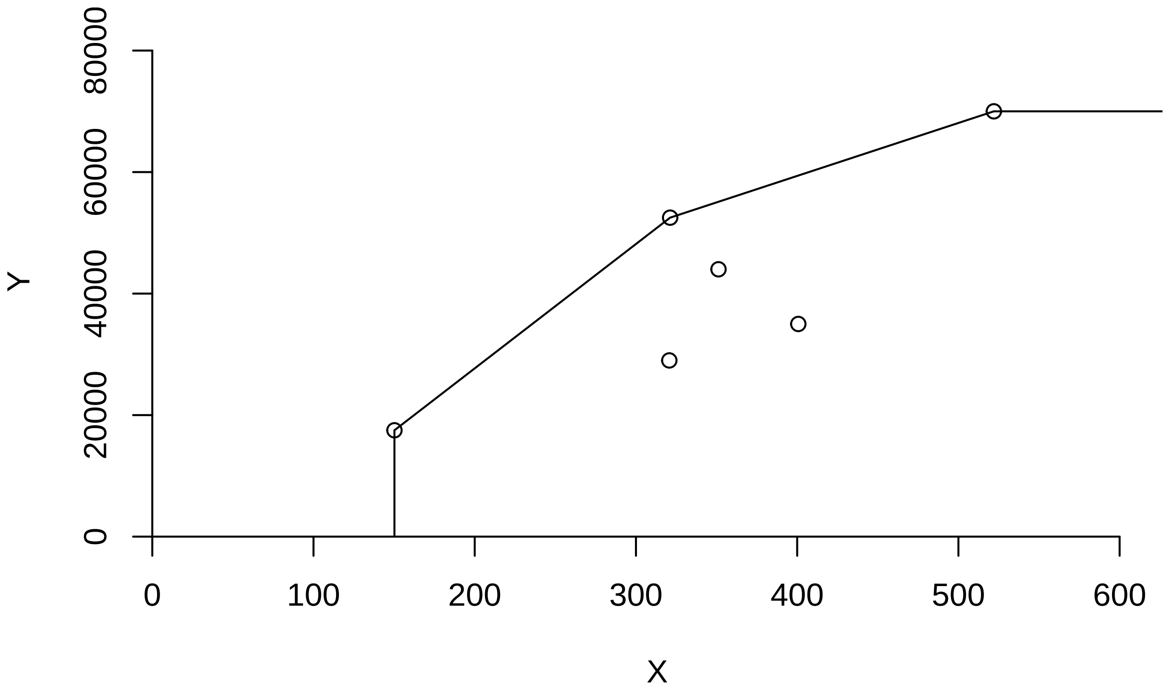
#Plotting the Graphs
 # CRS Plot

```
dea.plot(N, R, RTS='crs')
```



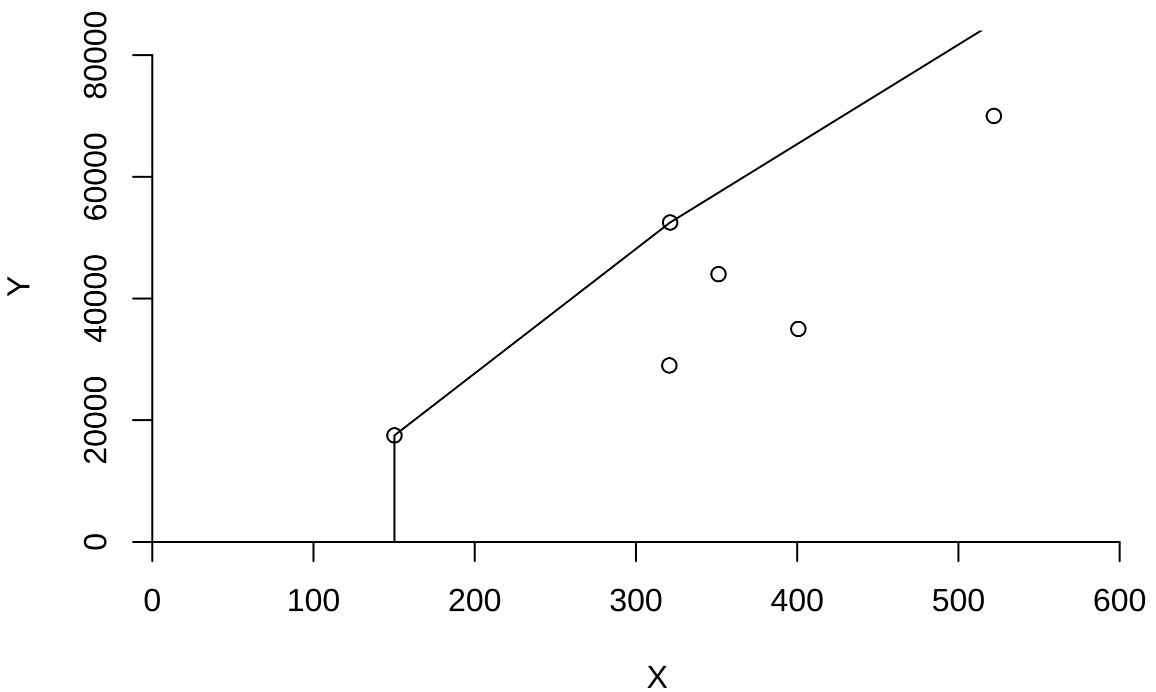
DRS Plot

```
dea.plot(N,R,RTS="vrs")
```



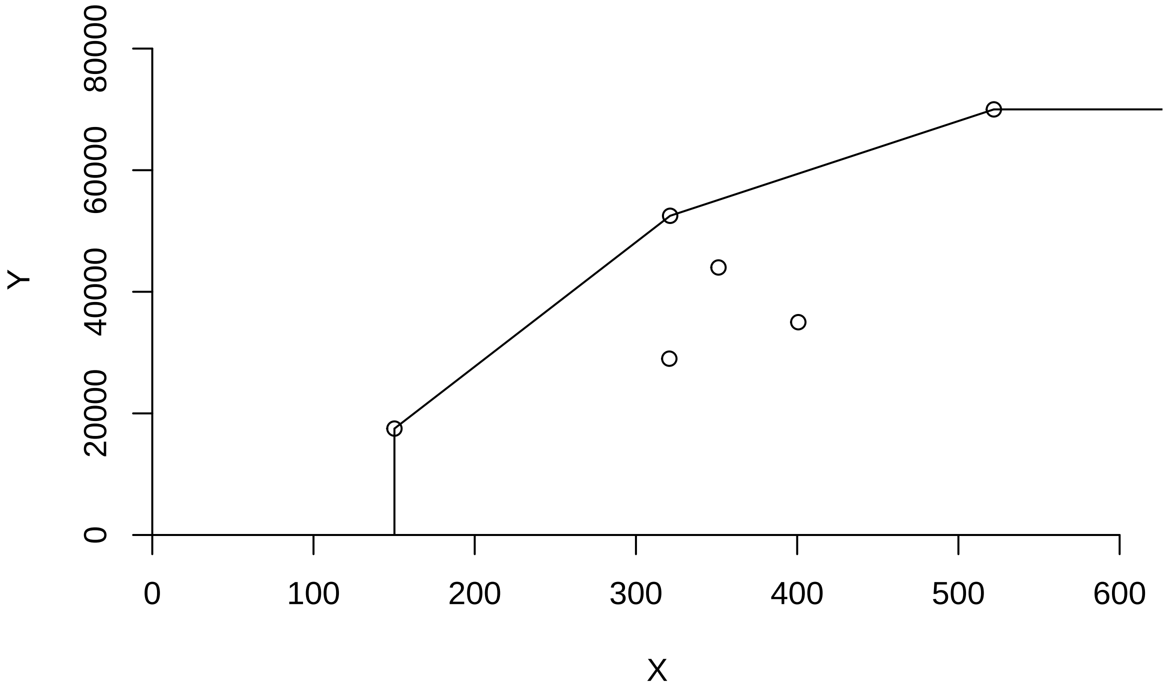
IRS Plot

```
dea.plot(N,R,RTS="irs")
```



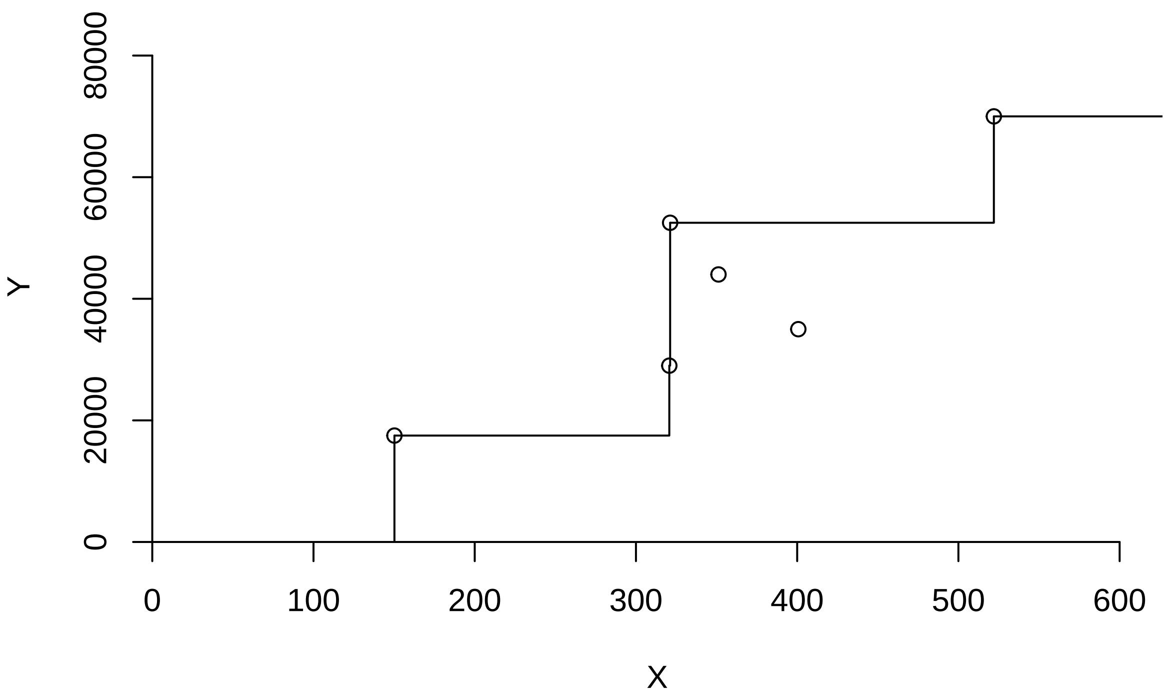
VRS Plot

```
dea.plot(N,R,RTS="vrs")
```



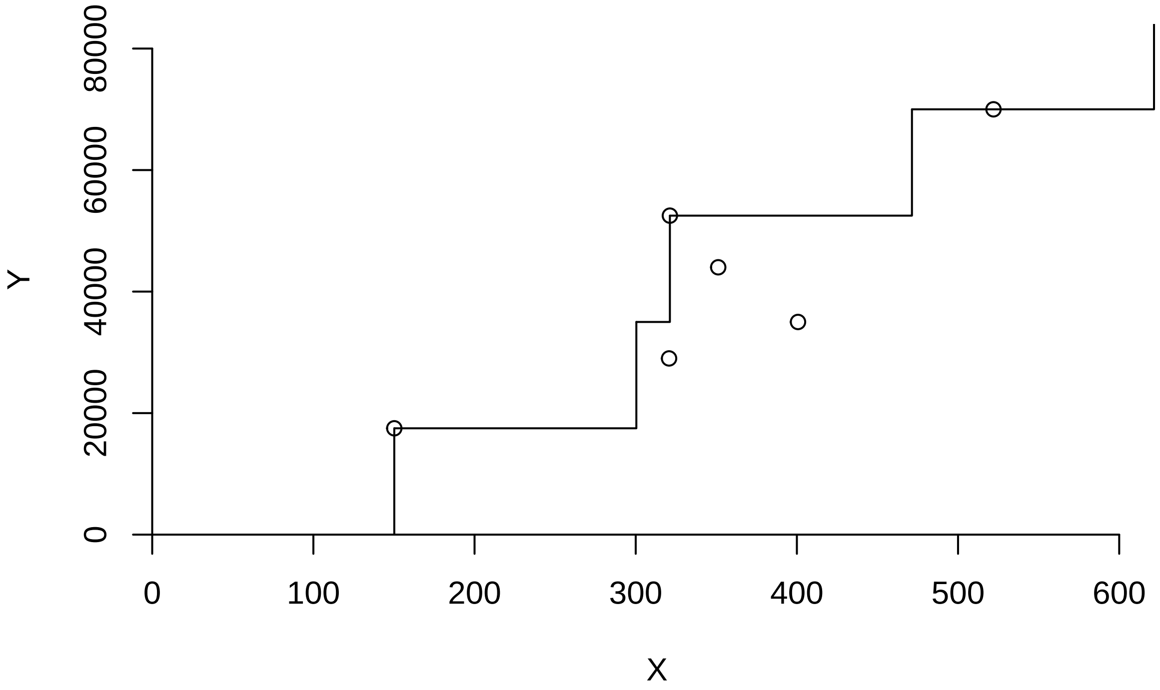
FDH Plot

```
dea.plot(N,R,RTS="fdh")
```



FRH Plot

```
dea.plot(N,R,RTS="add")
```



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