assignment_4

sai prasad desineni/811191807

2022-10-31

```
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
```

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

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

dataframe.table <- as.table(values_table)
dataframe.table
```

```
##   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
```

##now lets calculate Constants that 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(x) <- c("Staff_Hours_Per_Day","Supplies_Per_Day")

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


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

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

```
peers(D_E_A_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(D_E_A_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:-*

a. We see that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient.

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

c. Facility 5 is 97.75 % efficient leaving 2.25 % as inefficient

d.And Facility 6 is 86.75 % efficient leaving 13.25 % as inefficient.

**Calculating the Decreasing that returns to Scale (DRS)*

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

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

```
peers(D_E_A_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(D_E_A_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 get to see that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient.*
2. Also, we see that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are inefficient facilities.*
3. Facility 5 is 97.75 % efficient leaving 2.25 % as inefficient and Facility 6 is 86.75 % efficient i.e., leaving 13.25 % as inefficient.*

##Calculating Increasing Returns to Scale (IRS)

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

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

```
peers(D_E_A_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(D_E_A_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 get to see that Facility 1, Facility 2, Facility 3, Facility 4 and Facility 5 are efficient.*
2. Also, we get to see that Facility 1, Facility 2 and Facility 5 are the peer members for Facility 6 which is inefficient facility.*
3. Facility 6 is 89.63 % efficient leaving 10.37 % as inefficient.*

##Calculating Variable Returns to Scale (VRS)

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

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

```
peers(D_E_A_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(D_E_A_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 get to see that Facility 1, Facility 2, Facility 3, Facility 4 and Facility 5 are efficient.
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 is 89.63 % efficient leaving 10.37 % as inefficient.

##Calculating the Free Disposability Hull (FDH)

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

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

```
peers(D_E_A_fdh)
```

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

```
lambda(D_E_A_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

All the DMUs are efficient. Usually due to the principal which FDH technique follows there by detecting even a small level of efficiency.

##Calculating Free Replicability Hull (FRH)

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

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

```r
peers(D_E_A_frh)
```

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

```r
lambda(D_E_A_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

All the DMUs are efficient. It follows the no convexity assumption it ensures that the o/p is free from disposal and replication.

##Summary of Results (Inefficient DMUs)

```r
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(df.summarise.inefficient) <- c("RTS","Count_Inefficient_DMUs","Name_DMUs","%_Inefficiency","Pee

as.table(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)

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

as.table(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
```

***4.interpretation of the DEA analysis

***Understanding the varieties between the scales is vital. FDH and FRH are viewed as non-parametric methods for assessing the adequacy of DMUs. The scattering scales, DRS, VRS, and IRS, help us in figuring out what to raise and diminish in light of the information organization.

#CRS 1.Steady re-visitations of scale happen while expanding the quantity of sources of info leads to an identical expansion in the result. 2.It assists us with knowing whether the DMUs would be able be increased or down. DMUs 1,2,3 and 4 are plainly productive from the outcomes though DMU 5 and 6 are 97.75% and 86.7% productive separately. 3.The companion units for DMU 5 are 1,2 and 4 and their general loads are 0.2, 0.08 and 0.54 separately. 4.The friend units for DMU 6 are 1,2 and 4 and their relative loads are 0.34,0.39 and 0.131. From the above perceptions, it very well may be deciphered that the DMUs 1,2,3 and 4 can be increased.

#VRS

In the event that an increment or reduction in data sources or results doesn't result in a relative change in the data sources or results, separately, factor returns to scale can be utilized to gauge productivity. As indicated by the VRS measurements, DMUs 1, 2, 3, 4, and 5 are effective, though DMU 6 is 89.63% productive. For DMU 6, the companion units are 1, 2, and 5, and their comparing relative loads are 0, 0, and 0.25.

#IRS

1.At the point when the ascent in yield during the creation cycle is more than the expansion in input, this is known as a rising re-visitations of scale. By inspecting the productivity scores, the organization can decide if they can immediately grow their activity's scale. 2.The IRS results show unequivocally that DMUs 1, 2, 3, 4, and 5 are effective, yet DMU 6 just has a productivity of 89.63%. For DMU 6, the companion units are 1, 2, and 5, and their individual loads are 0, 0, and 0.25.

#DRS

1.At the point when the extent of result is lower than the normal expanded input during the creation cycle, a declining gets back to scale happens. 2.In the event that there are any potential DMUs where we can scale

the activities, DRS will let us know. This is achieved by looking at the inadequate DMUs. 3.The wasteful DMUs in this situation are 5 and 6, whose efficiencies are 97.75% and 86.75%, individually. For DMU 5, the companion units are 1, 2, and 4, and their particular relative loads are 0.2, 0.08, and 0.54. Peer units for DMU 6 are 1, 2, and 4, with respective relative weights of 0.34, 0.40, and 0.13.

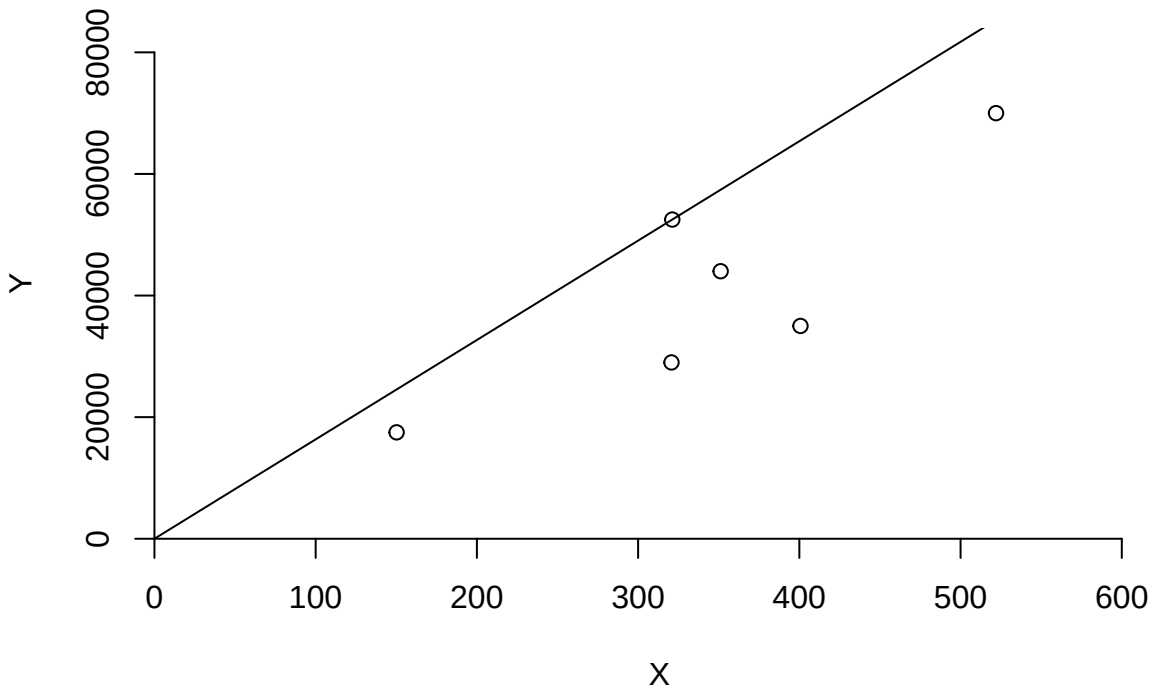
#FDH The FDH results plainly show that all DMUs are powerful. Because of the reality that there is no convexity supposition, FDH empowers the scale to catch even the most moment measure of effectiveness.

#FRH The FRH results obviously show that all DMUs are proficient. FRH permits us to catch the most reduced degree of proficiency at scale without duplication or disposing of.

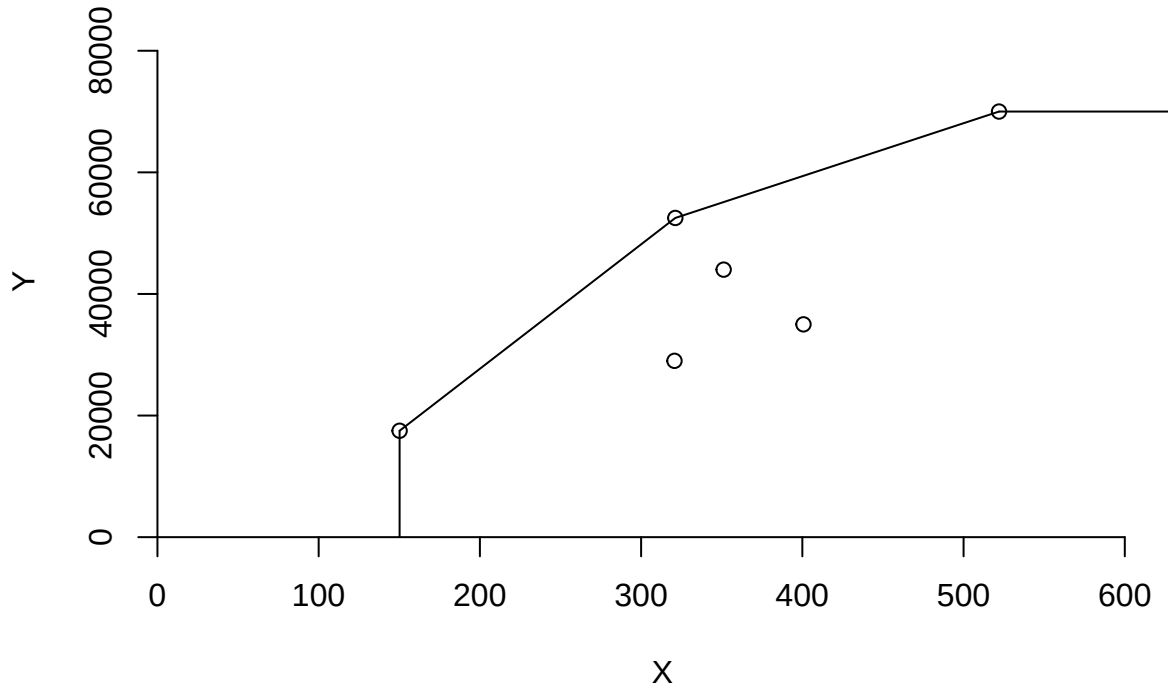
#conclusion: facilities 1,2,3 and 4 are completely proficient for every one of the suspicions and facility 5 and 6 are not effective. 1.facility 5 is completely effective for FDH,VRS,IRS and FRH presumptions. 2.For Office 6, CRS and DRS suppositions are 86.7% productive 3.We can see that there is 97.7% effectiveness for the DRS presumption. 4.At long last, for facility 6, IRS and VRS suppositions are 89.63% effective

##plotting the graphs

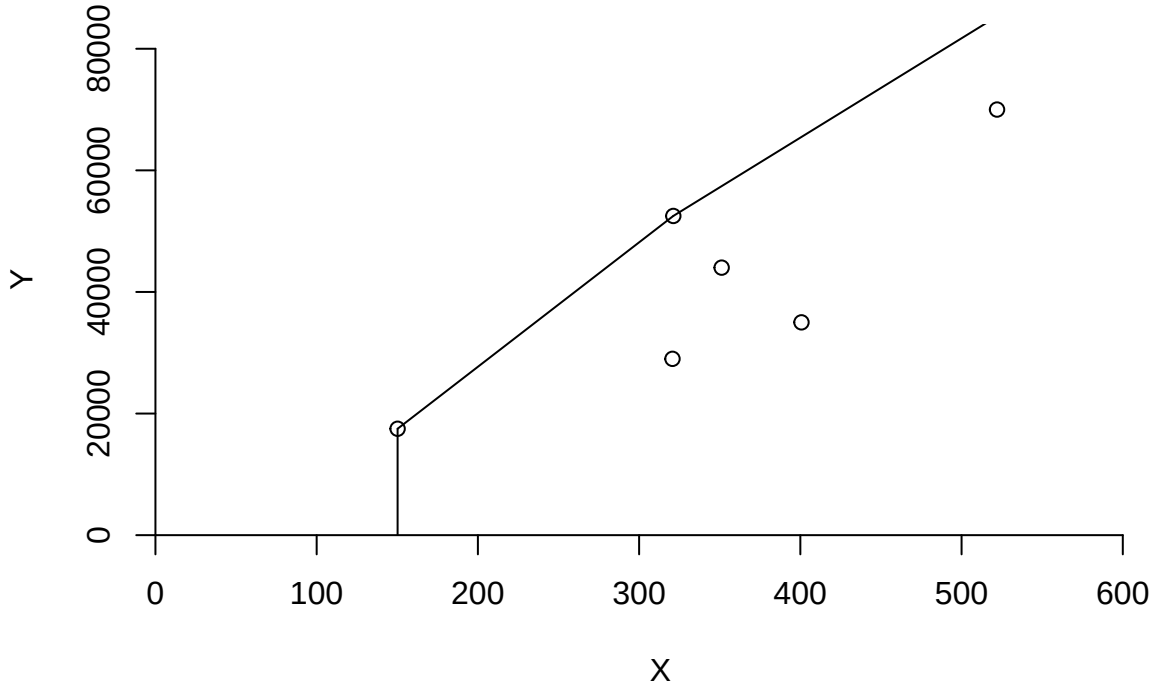
```
##crs plot
dea.plot(x, y, RTS='crs')
```



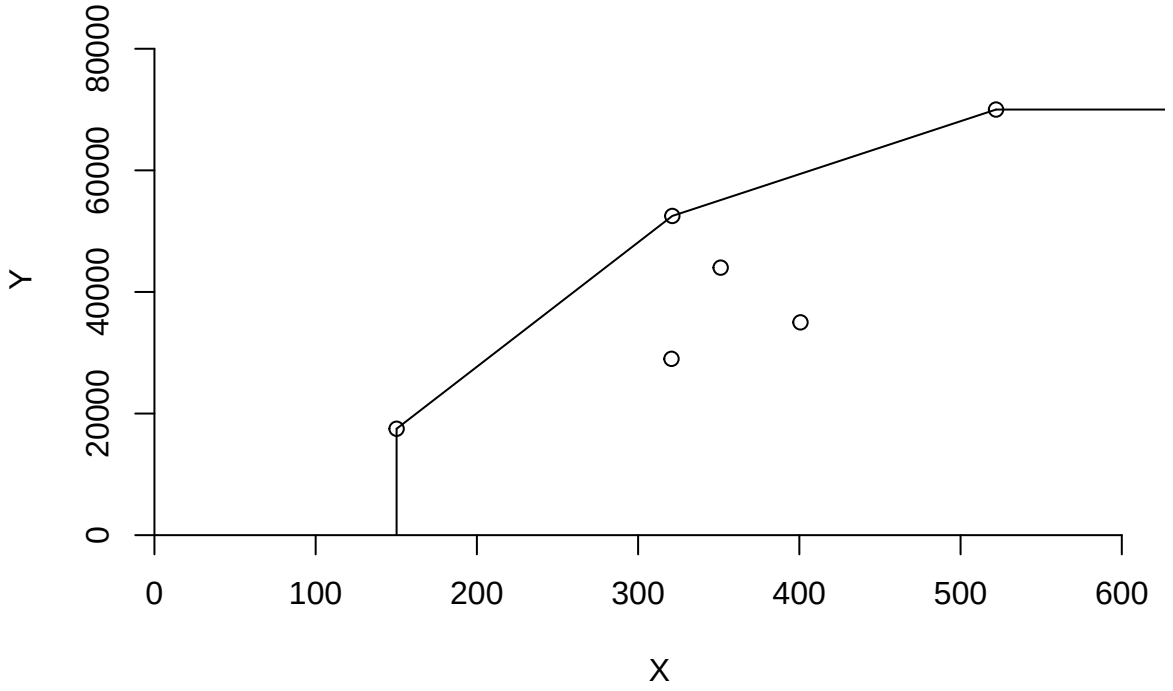

```
##Drs plot
dea.plot(x,y,RTS="vrs")
```



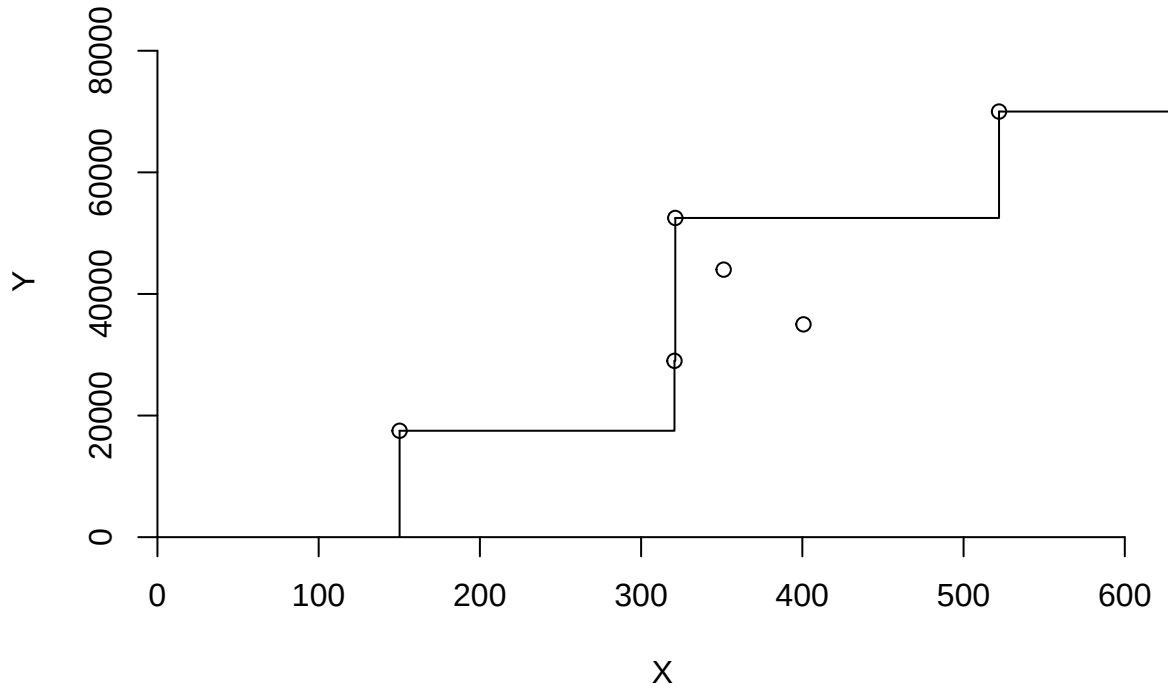

```
##Irs plot
dea.plot(x,y,RTS="irs")
```



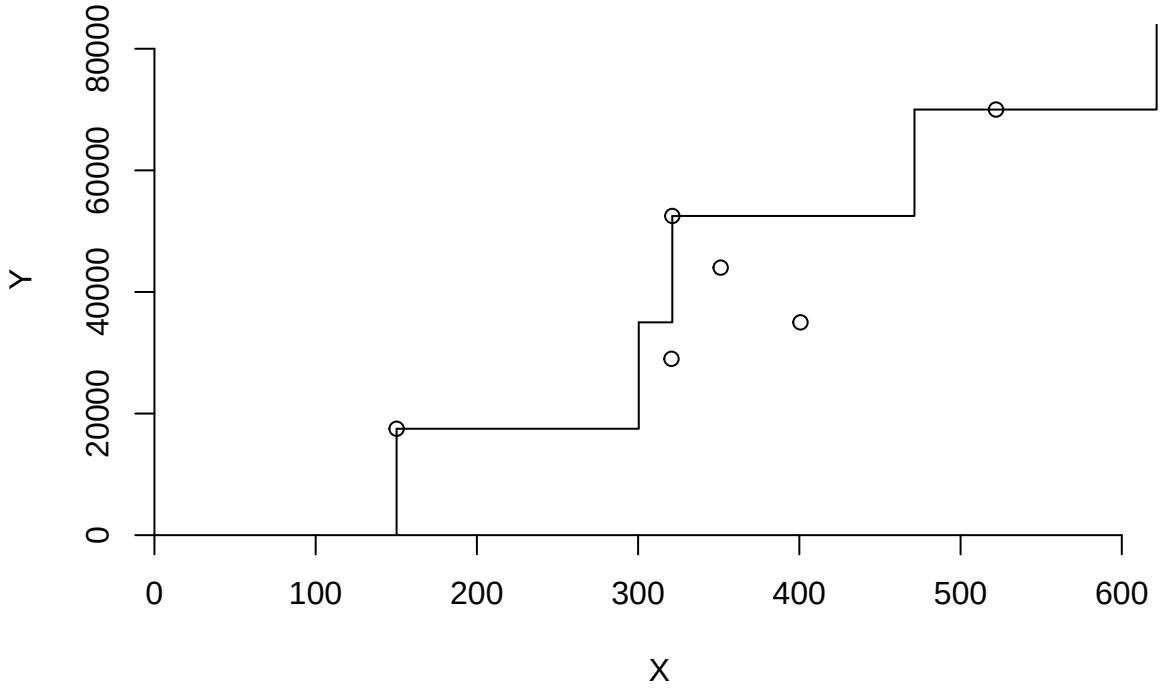

```
##vrs plot
dea.plot(x,y,RTS="vrs")
```




```
##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