Assignment-4-QMM

ADITYA MUNAGALA

2022-10-30

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

```
library("Benchmarking")
```

```
data.df.values <- 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(data.df.values) <- c("DMU", "Staff_Hours_Per_Day","Supplies_Per_Day","Reimbursed_Patient_Days"

table.df <- as.table(data.df.values)
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
```

Here we are calculating Constant 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(y) <- c("Reimbursed_Patient_Days","Privately_Paid_Patient_Days")
```

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

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
```

```r
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
```

```r
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)*

```r
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
```

```r
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
```

```r
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)

```r
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
```

```r
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
```

```r
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)

```
#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
```

```
peers(D_E_A_frh)
```

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

```
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)

```
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

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

1.Before interpreting, it is essential to understand how the scales vary (RTS).*

2.Additionally, we learn that the ineffective Facility 6 has as peer members Facility 1, Facility 2, and Facility 5.

3.A non-parametric method to assess the efficacy of DMUs is the Free Disposability and Free Replicability Hull (FDH & FRH), which makes no assumptions regarding convexity.*

4.Based on the deployment of information, the Decreasing, Increasing and Varying Returns to Scale (DRS, IRS, and VRS) dispersion scales let us decide what to increase and what to decrease.**

DRS - Decreasing Returns to Scale

1.The outcomes demonstrate the efficiency of DMUs 1, 2, 3, and 4. DMU(6) has an efficiency of 86.7%, while DMU(5) has a 97.75% efficiency.

2.Based on our early investigation, we found this. Additionally, the units of DMU(4peer) are 1, 2, and 4, with 0.2, 0.08, and 0.54 as their respective relative weights.

3.The peer units for DMU(6) are 1, 2, and 4, and their corresponding weights are 0.34, 0.4, and 0.13.

4.This scale identifies any potential DMUs where we might be able to scale the processes, for instance by examining the ineffective DMUs in this case, DMUs 5 and 6. Since this is the original base scale, the CRS values can also be used to determine it.

CRS - Constant Returns to Scale

1.The results demonstrate the productivity of DMUs 1, 2, 3, and 4. Only 86.7% of DMU(6) and 97.75% of DMU(5) are utilized efficiently. We found this based on our preliminary analysis.

2.Furthermore, the units of DMU(4peer) are 1, 2, and 4, with corresponding weights of 0.2, 0.08, and 0.54. Peer units 1, 2, and 4 for DMU(6) have weights of 0.34, 0.4, and 0.13, respectively.

3. In summary, CRS helps us to assess whether any prospective DMUs, in this example DMUs 1, 2, 3, and 4, may be scaled up or down.

IRS - Increasing Returns to Scale

1.The outcomes demonstrate the effectiveness of DMUs 1, 2, 3, and 4. While DMU(6) is 86.7% efficient, DMU(5) is only 97.75% efficient. We conducted some preliminary study and found this. Additionally, DMU(4) has three peer units: 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 corresponding weights are 0.34, 0.4, and 0.13.

2.By looking at the ineffective DMUs in this scenario, DMUs 5 and 6, this scale identifies any potential DMUs where we might scale the activities. Since this is the base original scale, the CRS values can also be used to determine it.

VRS - Variable Returns to Scale

1.The outcomes demonstrate the efficiency of DMUs 1, 2, 3, 4, and 5. The effectiveness of DMU(6) is just 89.63%. Based on our early investigation, we found this.

2.Additionally, DMU(6) contains three peer units with relative weights of 0.4, 0.34, and 0.26: 1, 2, and 5. 3.Understanding the scale of processes with variations in the input and output component, either rising or decreasing or utilizing both, is made easier by varying or variable returns to scale. ***FRH - Free Replicability Hull***

1.The FRH findings demonstrate the effectiveness of every DMU. In general, this technique enables the scale to capture even the smallest degree of efficiency that is devoid of replication and disposal. This is primarily owing to the assumption of no convexity.

2.Only the ineffective DMUs would receive the peer values, or neighbors, and lambda values, or weights of the peers. Lambda weights and peers are absent from efficient DMUs.

FDH - Free Disposability Hull

*The outcomes demonstrate the effectiveness of every DMU. This is partly because no convexity is assumed, and this method enables the scale to measure even the tiniest amount of efficiency.

Conclusion

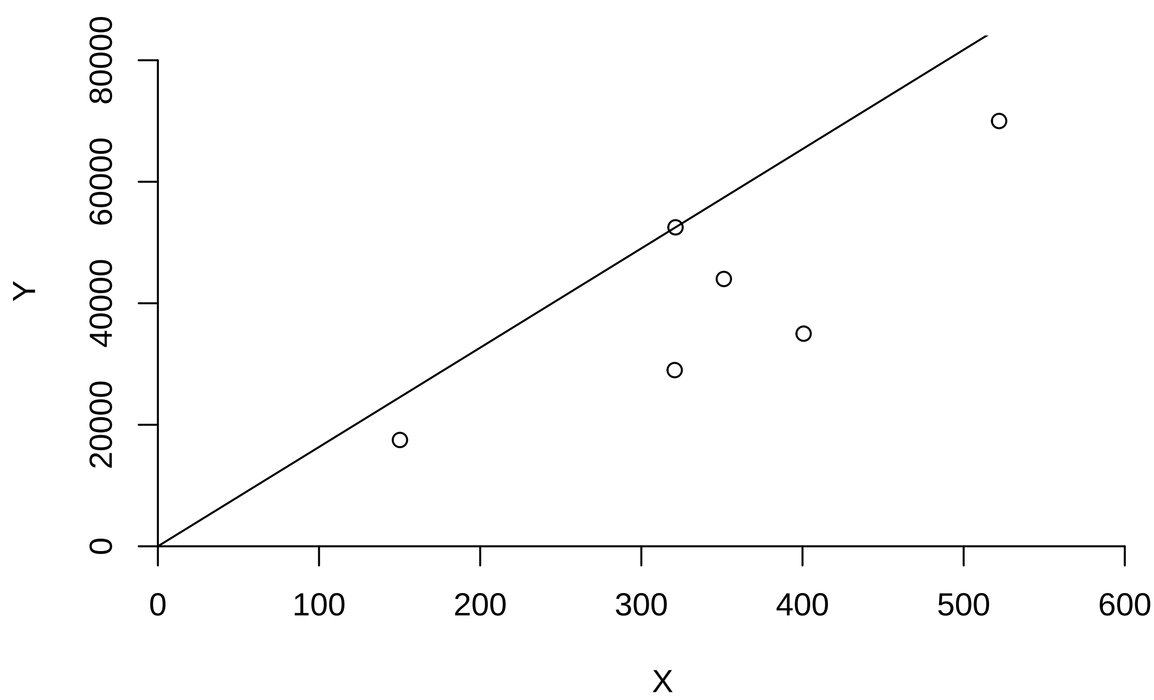
1.It's crucial to remember that DEA is a very helpful tool for any company in figuring out which DMU is the best, i.e. which of the Decision Making Units should be maximized so that there is an increase, decrease, or any other kind of variation in the output by feeding input into it.

2.Additionally, a business can decide which Returns to Scale (RTS) to use based on their needs; each of these scales has a different level of significance.*

Plotting the Graphs

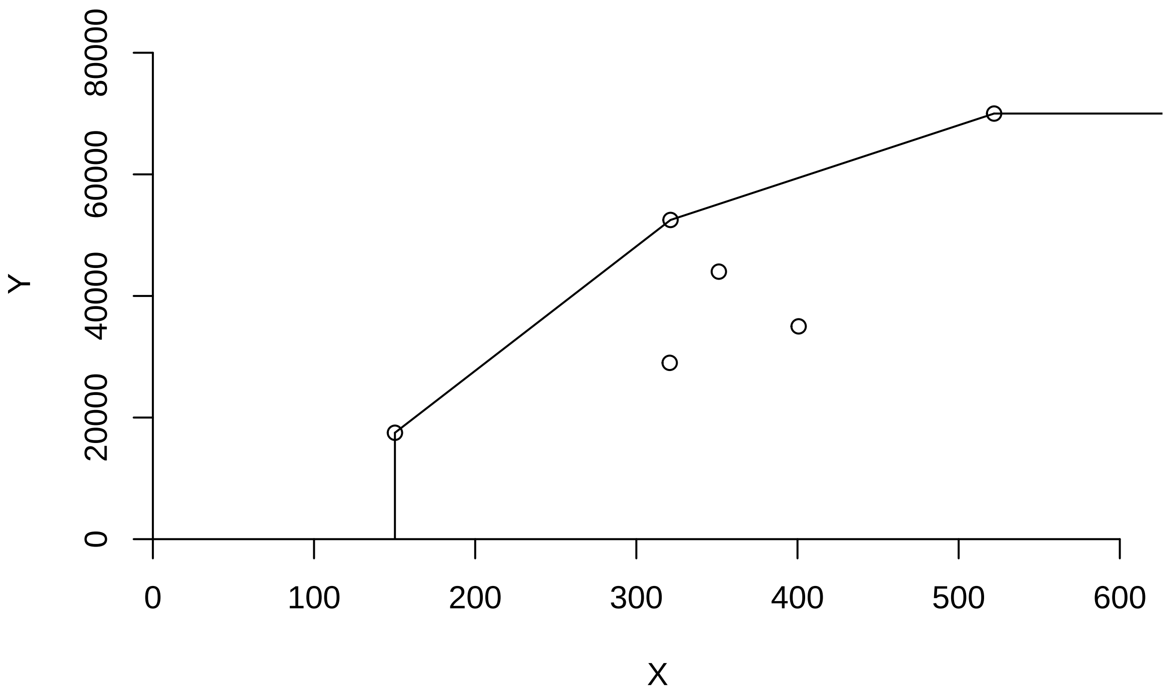
CRS Plot

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



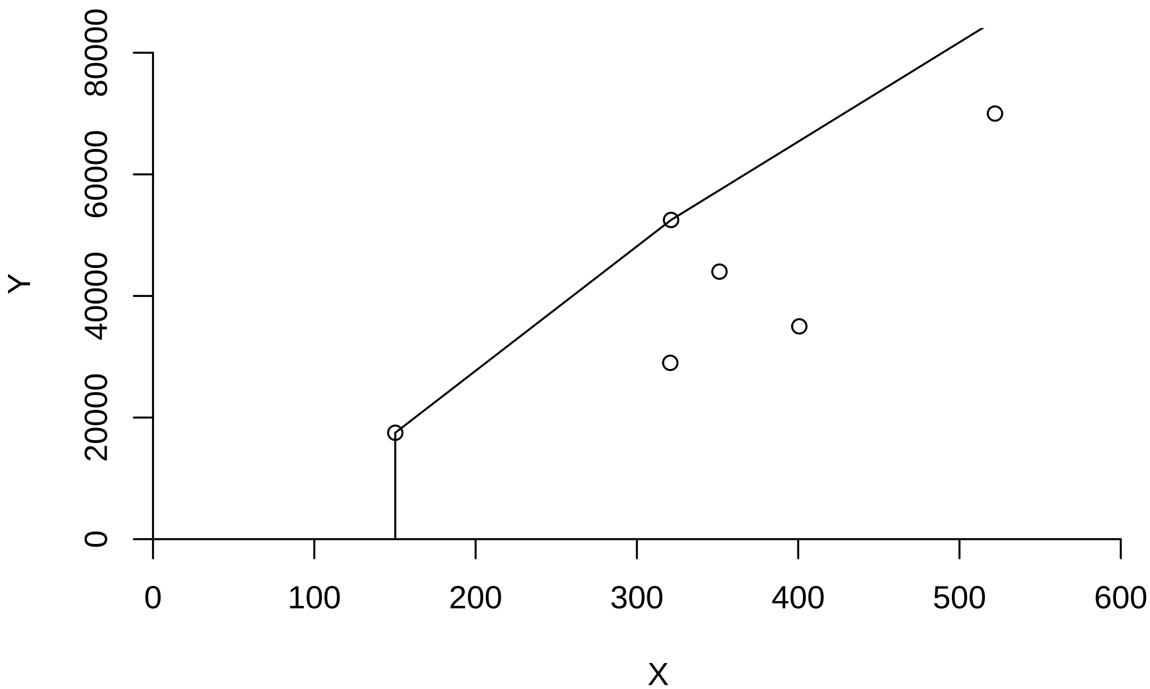
DRS Plot

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



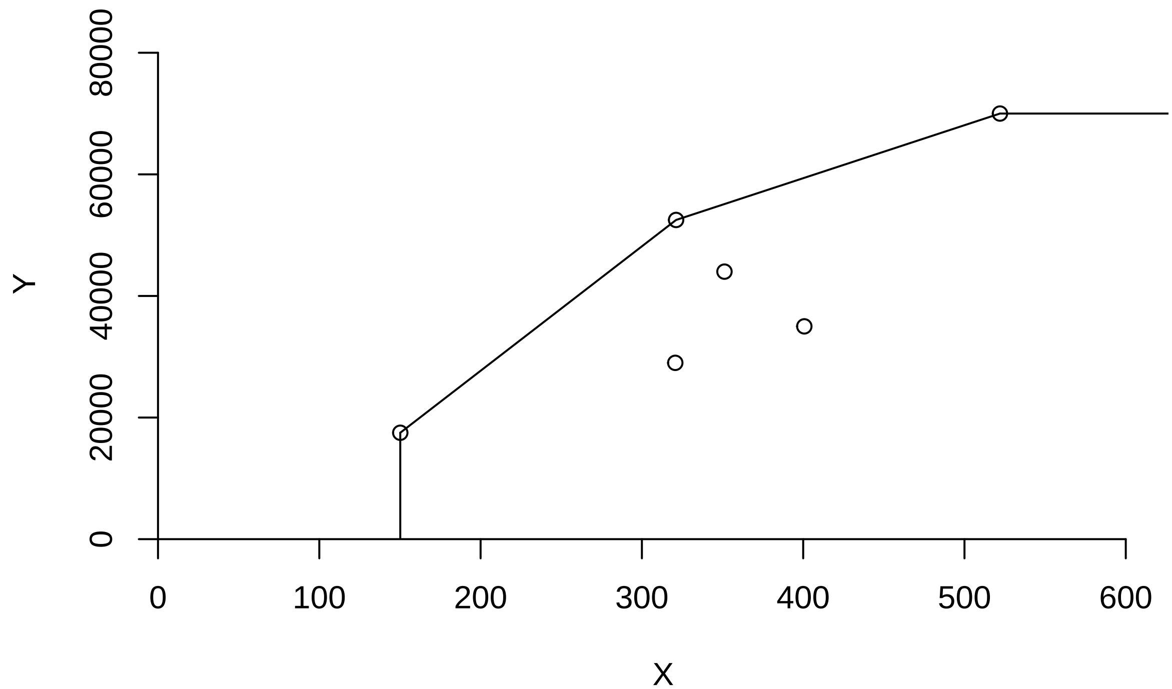
IRS Plot

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



VRS Plot

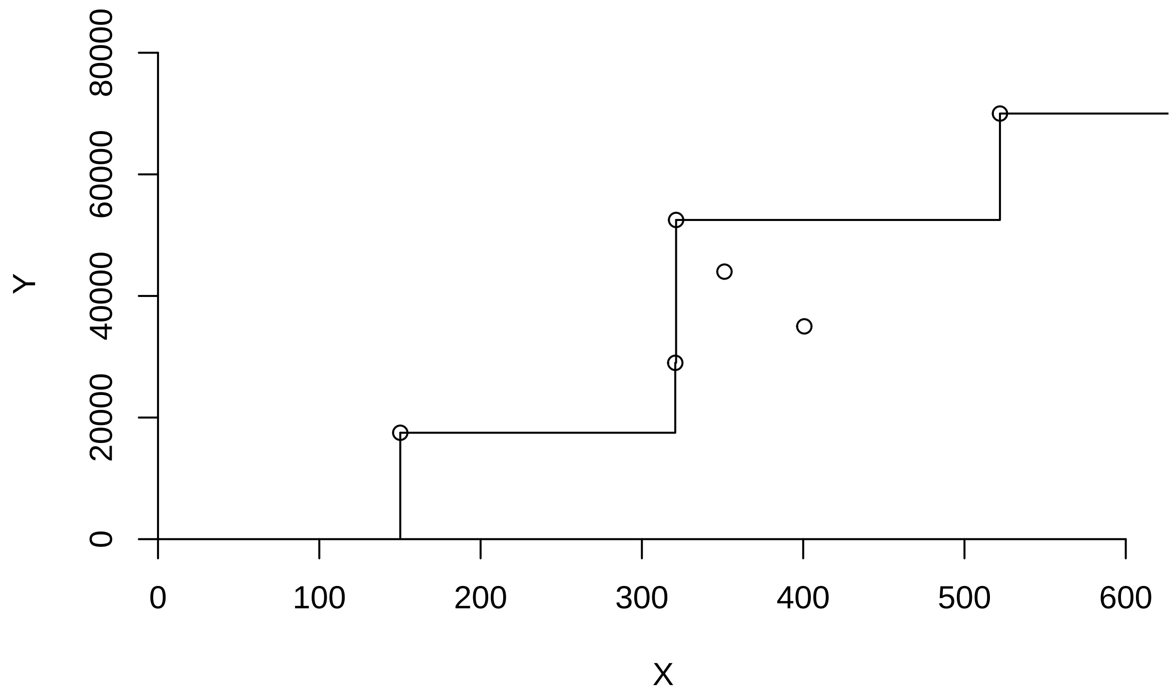
```
dea.plot(x,y,RTS="vrs")
```




```r
#tinytex::install_tinytex()
```

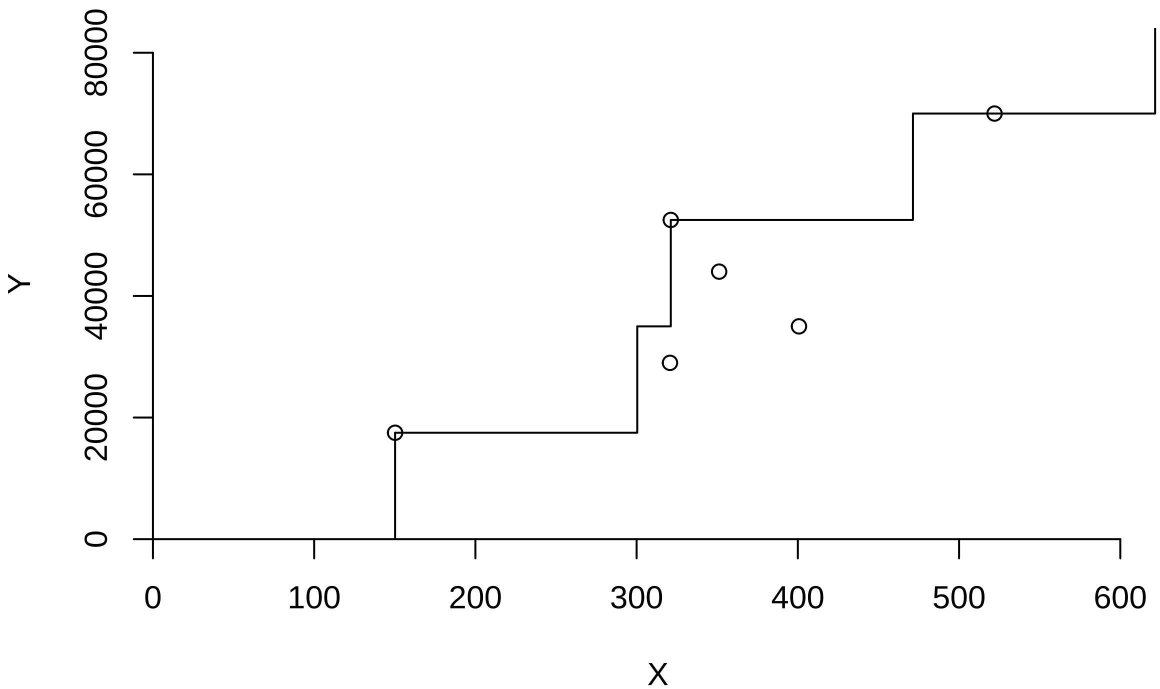
FDH Plot

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



FRH Plot

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



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