assignment module 8

vineeth goud maddi

10/30/2022

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

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

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

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

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

```r
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 -
1. We learn how effective Facilities 1, 2, 3, and 4 are.
2. Additionally, we learn that the ineffective facilities Facility 5 and Facility 6 have Facility 1, Facility 2, and Facility 4 as peer members.
3. Facilities 5 and 6 are both 96.75% efficient, leaving 2.25% and 13.25% of inefficiencies, respectively.

Calculating Decreasing 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 observe how well Facilities 1, Facility 2, Facility 3, and Facility 4 operate.

2. Additionally, we learn that the ineffective facilities Facility 5 and Facility 6 are peers of Facilities 1, 2, and 4, which are themselves
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)

```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 learn that Facilities 1, 2, 3, 4, and 5 are effective.
2. Additionally, we learn that Facility 6—the only inefficient facility—has as peer members Facility 1, Facility 2, and Facility 5.
3. Facility 6 has an efficiency rate of 89.63%, leaving a 10.37% inefficiency.

Calculating Variable Returns to Scale (VRS)

```r
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
```

```r
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
```

```r
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.It is possible for us to observe the effectiveness of Facilities 1, 2, 3, 4, and 5.
2. Additionally, we learn that Facility 6—the lone inefficient facility—has as peer members Facilities 1, 2, and 5.
3. Facility 6 has an efficiency of 89.63%, leaving a 10.37% inefficiency..

Calculating 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 -
The DMUs are all effective. This is mainly because the FDH technique adheres to a certain principal, which allows it to identify even a very low 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 -
The DMUs are all 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.


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 -
Knowing the differences between the scales is crucial before interpretation (RTS),
The original scale, Constant Returns to Scale (CRS), is still utilized by the majority of businesses.
The Decreasing, Increasing and Varying Returns to Scale (DRS, IRS and VRS) dispersion scales aid us in determining which inputs to increase and which to reduce dependent on input deployment.
The Free Disposability and Free Replicability Hull (FDH & FRH), which makes no assumptions about convexity, is regarded as a 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 aids in determining whether any potential DMUs can be scaled up or down; in this example, DMUs 1, 2, 3 and 4 can be.

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, the peer units for DMU(6) are 1, 2, and 5, with corresponding relative weights of 0.4, 0.34, and 0.26.
By examining the efficiency rankings, it allows any company to determine whether they can arbitrarily expand their operation.(Refer data.df.summarise.efficient table)

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, the peer units for DMU(6) are 1, 2, and 5, with corresponding relative weights of 0.4, 0.34, and 0.26.
Understanding the scale of operations with changes to the input and output factors—either by increasing or decreasing, or by using both—is made easier 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 inefficient DMUs will receive the peer values, or neighbors, and lambda values, or weights, of the peers. Peers and lambda weights are not present in efficient DMUs.

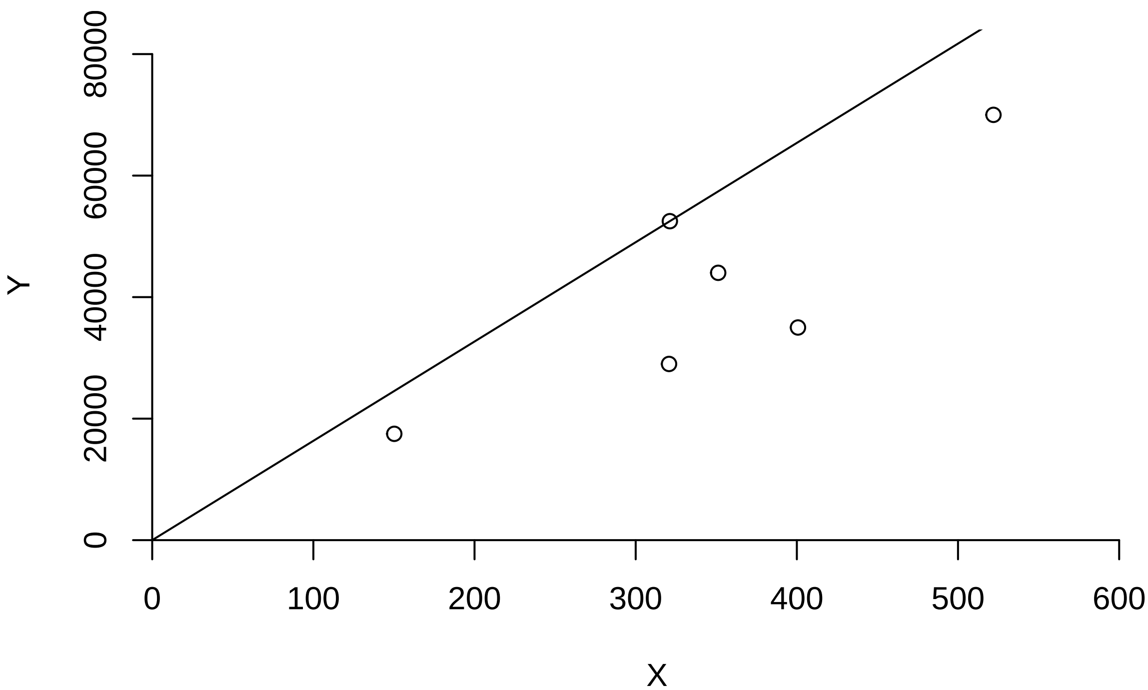
Conclusion

It is crucial to keep in mind that DEA is a very helpful tool for any company in determining which DMU is the best, i.e., which of the Decision Making Units 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 business can choose which Returns to Scale (RTS) scale to use based on its needs; each of these scales has a distinct significance.

Plotting the Graphs

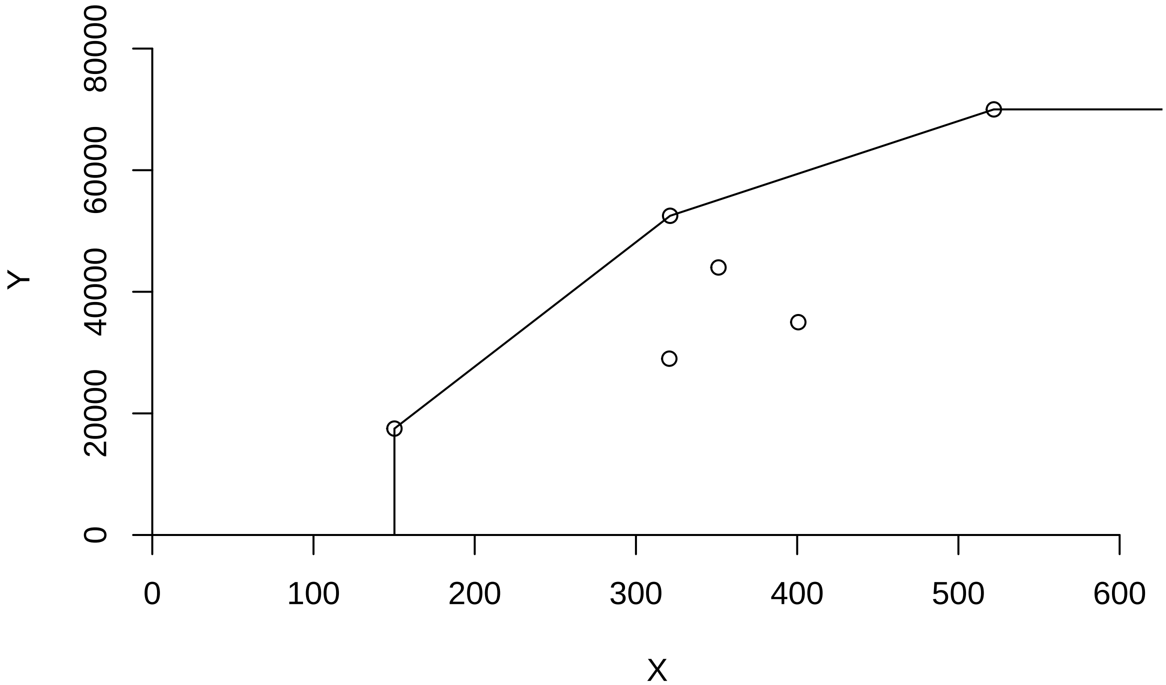
CRS Plot

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



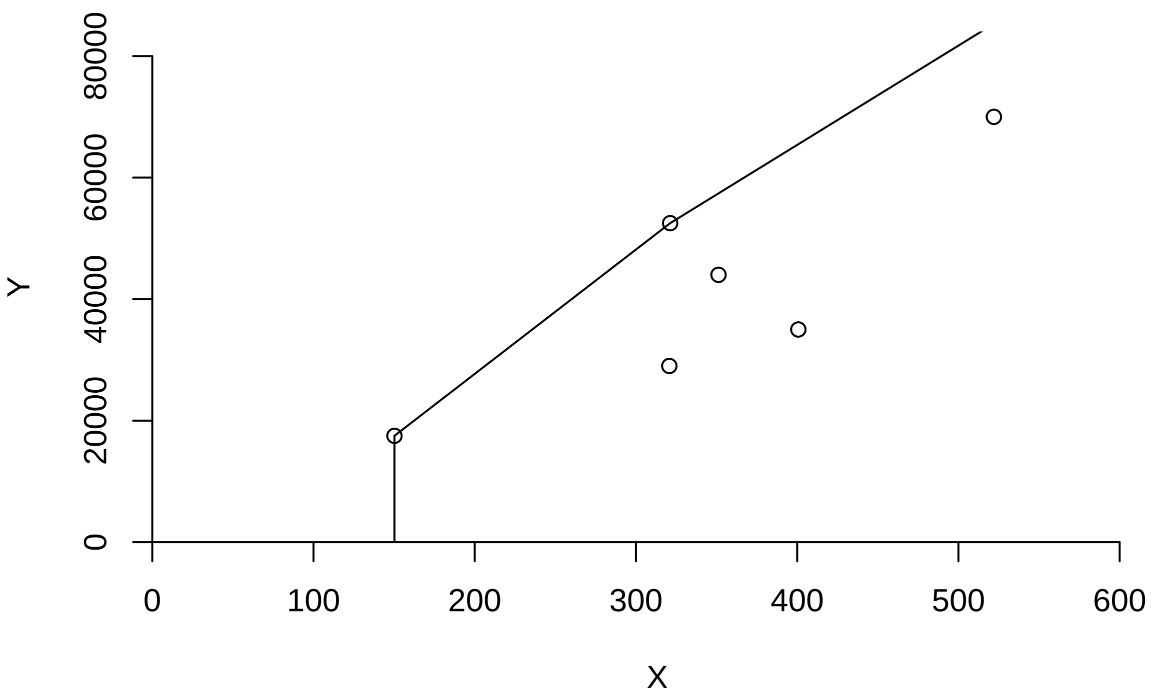
DRS Plot

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



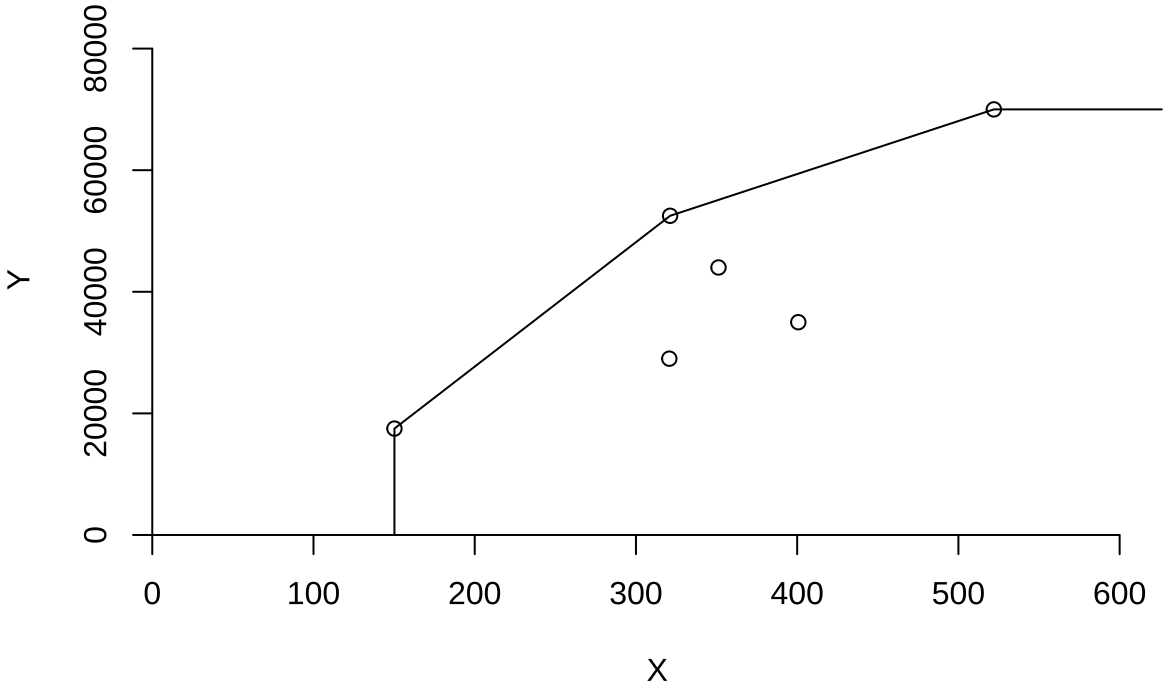
IRS Plot

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



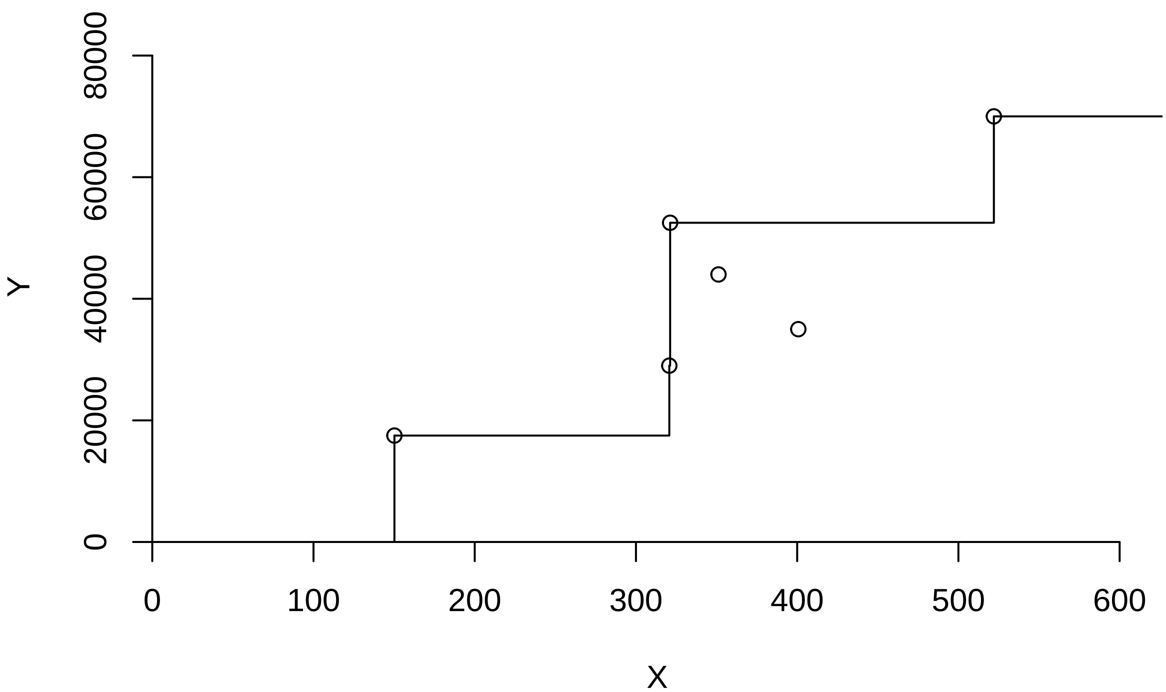
VRS Plot

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



FDH Plot

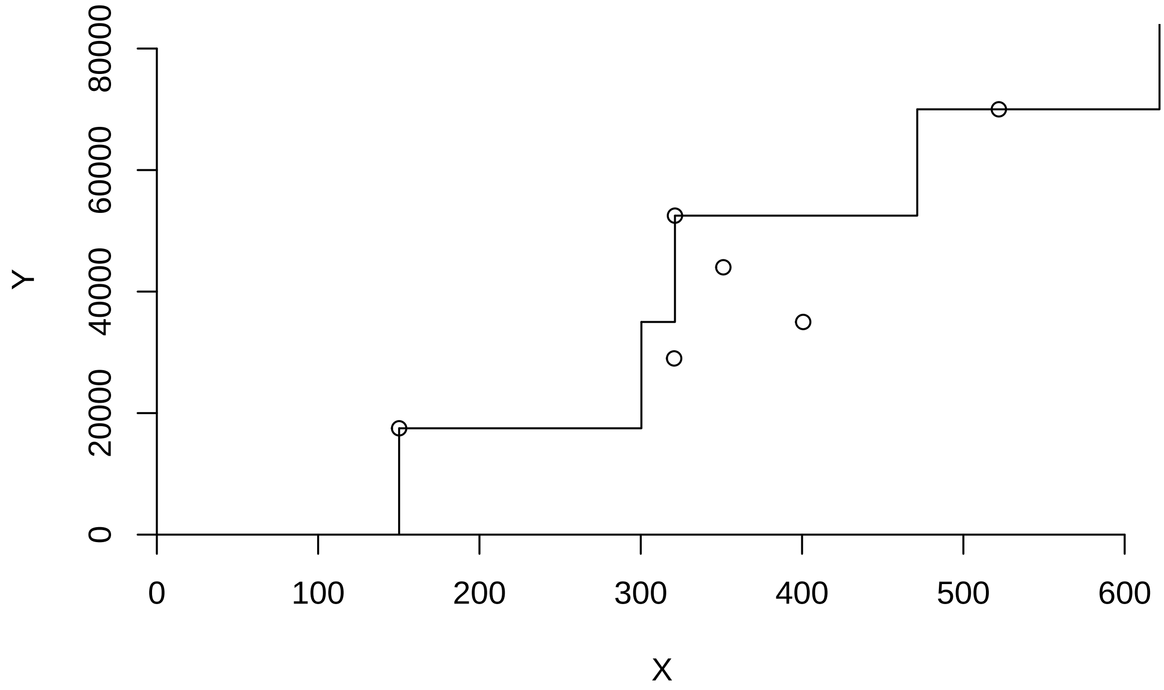
```
dea.plot(x,y,RTS="fdh")
```



FRH

Plot

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


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