Assignment 4

MADHUKAR HEEREKAR

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 
                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
```

The constant that returns to scale is being calculated here (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. It is clear that Facilities 1, 2, 3, and 4 are productive.*

B. In addition, we can see that the ineffective facilities Facility 5 and Facility 6 are peers of Facilities 1, 2, and 4, respectively.*

C. Facility 5 has an efficiency rate of 97.75%, leaving 2.25% inefficient.*

D. Facility 6 has an efficiency rate of 86.75%, leaving a 13.25% inefficiency.*

Decreasing that Returns to Scale (DRS) Calculation

```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
```

```
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. In addition, we note that Facilities 5 and 6, which are ineffective facilities, are peers of Facilities 1, 2, and 4.*
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)

```
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 are given the opportunity to observe the effectiveness of Facilities 1, 2, 3, 4, and 5.*

2. we learn that the ineffective Facility 6, which is a peer facility of Facilities 1, 2, and 5, are these facilities.*

3. Facility 6 has an efficiency rate of 89.63%, leaving a 10.37% inefficiency.*

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 are given the opportunity to observe the effectiveness of Facilities 1, 2, 3, 4, and 5.

2. we learn that Facility 6—the lone inefficient 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 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

The DMUs are all effective. Due to the principle that the FDH technique uses, it can typically identify even a very low degree 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. It ensures that the o/p is free from disposal and replication because it adheres to the no convexity assumption.

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, understanding the scale variations (RTS) is essential..*

2.The majority of firms use Constant Returns to Scale (CRS), which is regarded as the original scale..**

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 about convexity..*

4.We can choose what to increase and what to reduce based on the distribution of information by using the dispersion scales known as Decreasing, Increasing and Varying Returns to Scale (DRS, IRS, and VRS).*

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 relative weights of 0.2, 0.08, and 0.54..

3.The peer units for DMU(6) are 1, 2, and 4, respectively, with weights of 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. It can also be found by looking at the CRS values because this is the base original scale..

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. Based on our preliminary study, we learned this.

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 for DMU(6) are 1, 2, and 4, and their weights are 0.34, 0.4, and 0.13, respectively.*

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

IRS - Increasing Returns to Scale

1.DMUs 1, 2, 3, and 4 are effective, according to the results. DMU(6) is more effective than DMU(5), which has an efficiency of 86.7%. This is what we found from our early investigation. Moreover, DMU(4peer)'s units are 1, 2, and 4, with 0.2, 0.08, and 0.54 as their respective relative weights. Similar to this, peer units 1, 2, and 4 for DMU(6) have weights of 0.34, 0.4, and 0.13, respectively.

2.By, for instance, examining the ineffective DMUs in this scenario, DMUs 5 and 6, this scale reveals any potential DMUs where we might scale the processes. This being the base original scale, it can also be obtained by examining the CRS values..

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.peer units 1, 2, and 5 have relative weights of 0.4, 0.34, and 0.26 for DMU(6), respectively..

3.Understanding the scale of processes with changes to the input and output factors, either increasing or decreasing or employing both, is made easier by varying or variable returns to scale..*

FRH - Free Replicability Hull

1.The assumption of no convexity and, more generally, this technique allow the scale to capture even the smallest level of efficiency that is devoid of replication and disposal, as shown by the FRH results, which indicate that all of the DMUs are efficient.

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 in 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

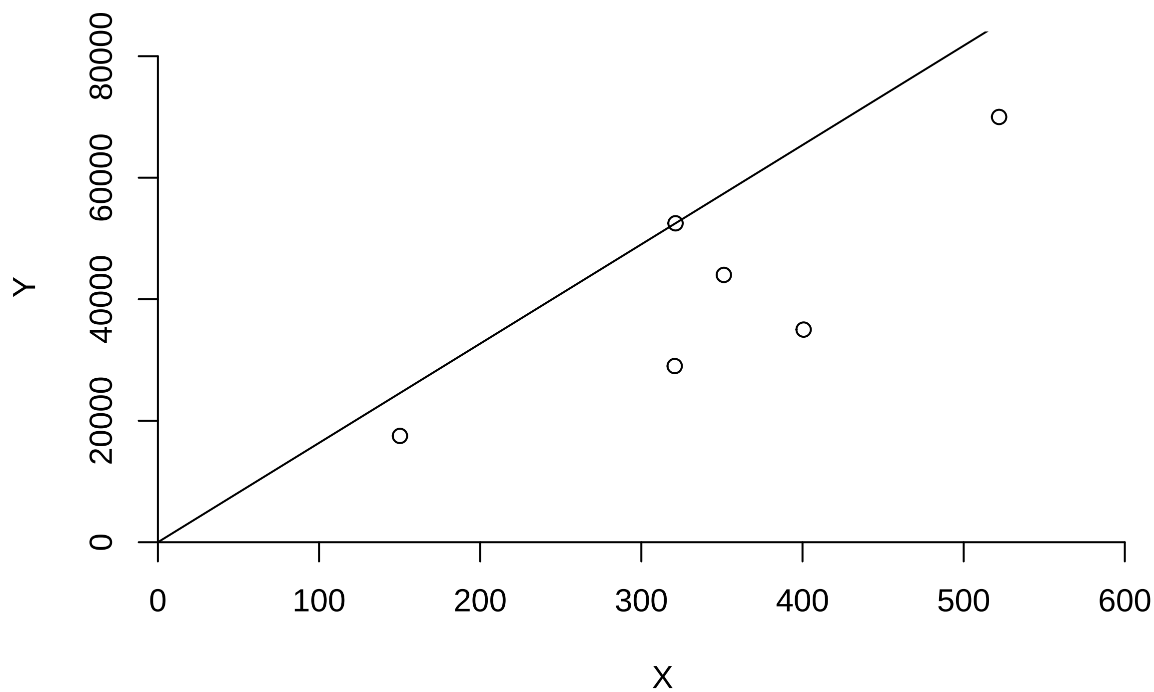
1. It is crucial to remember that DEA is an excellent tool for any company to use when deciding which Decision Making Unit (DMU) is the best, i.e. which of the Decision Making Units should be maximized so that there is an increase, decrease, or any other variation in the output by feeding input into it.

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

Plotting the Graphs

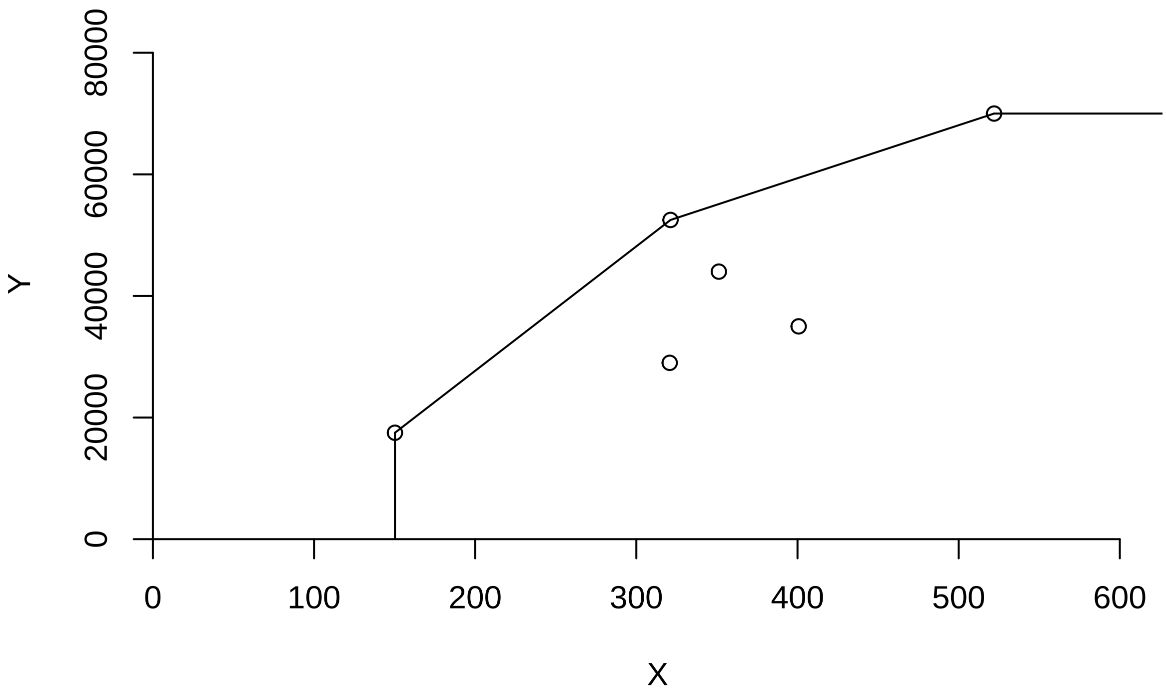
CRS Plot

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



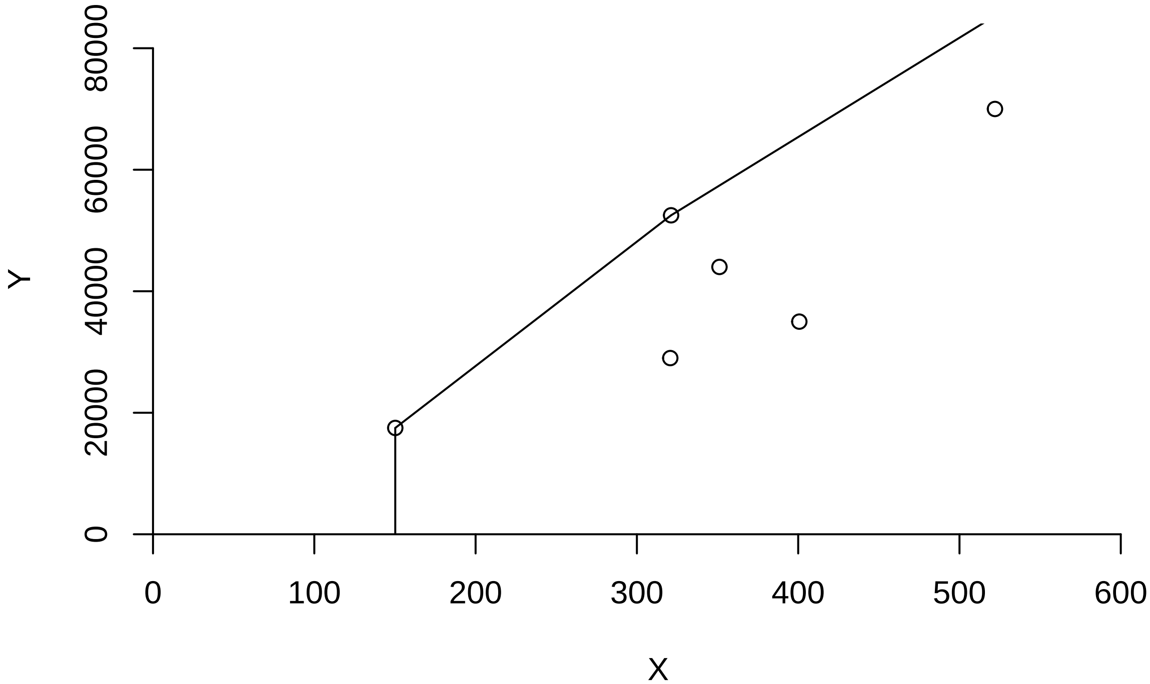
DRS Plot

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



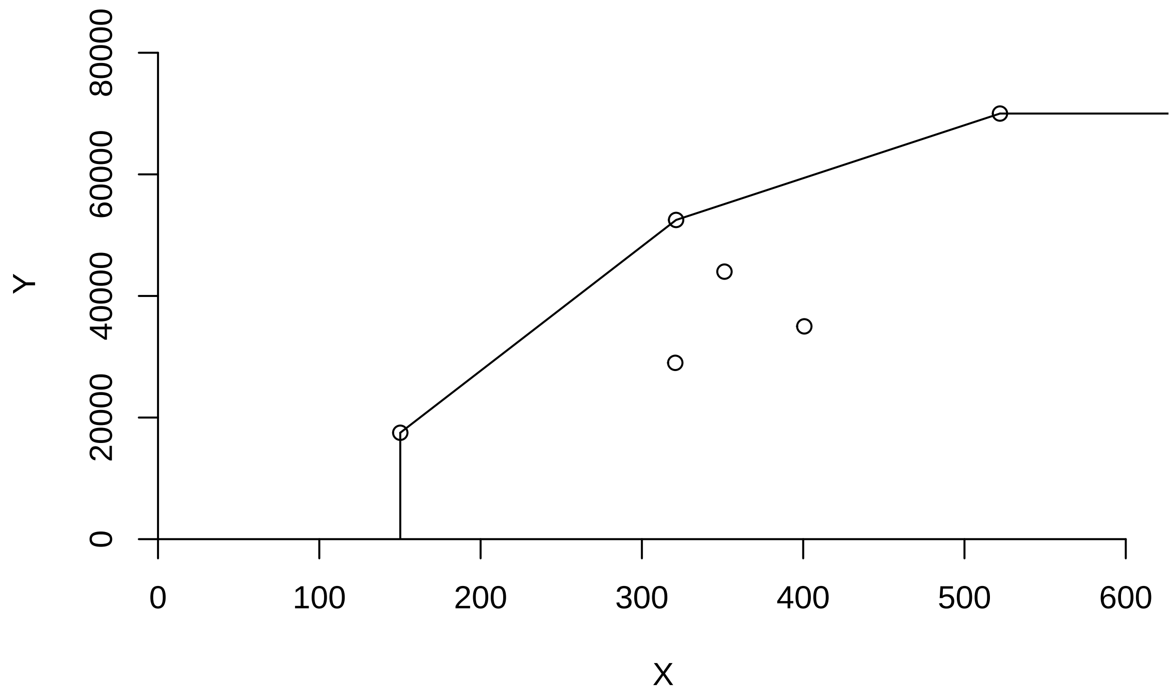
IRS Plot

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



VRS Plot

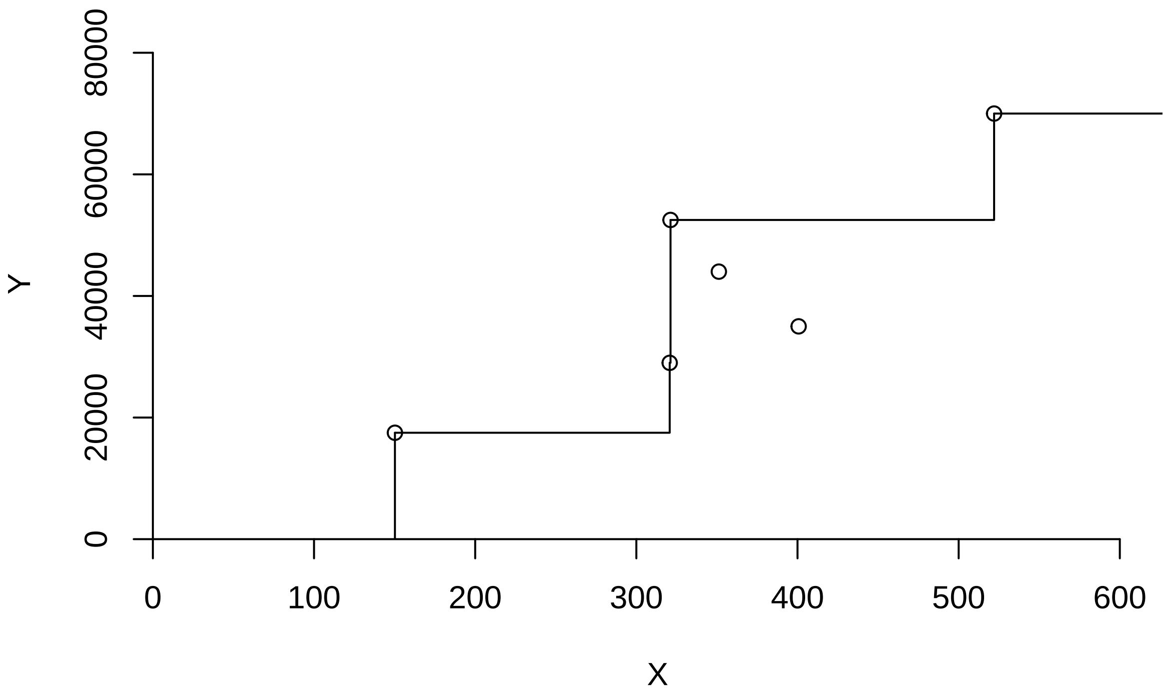
```
dea.plot(x,y,RTS="vrs")
```




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

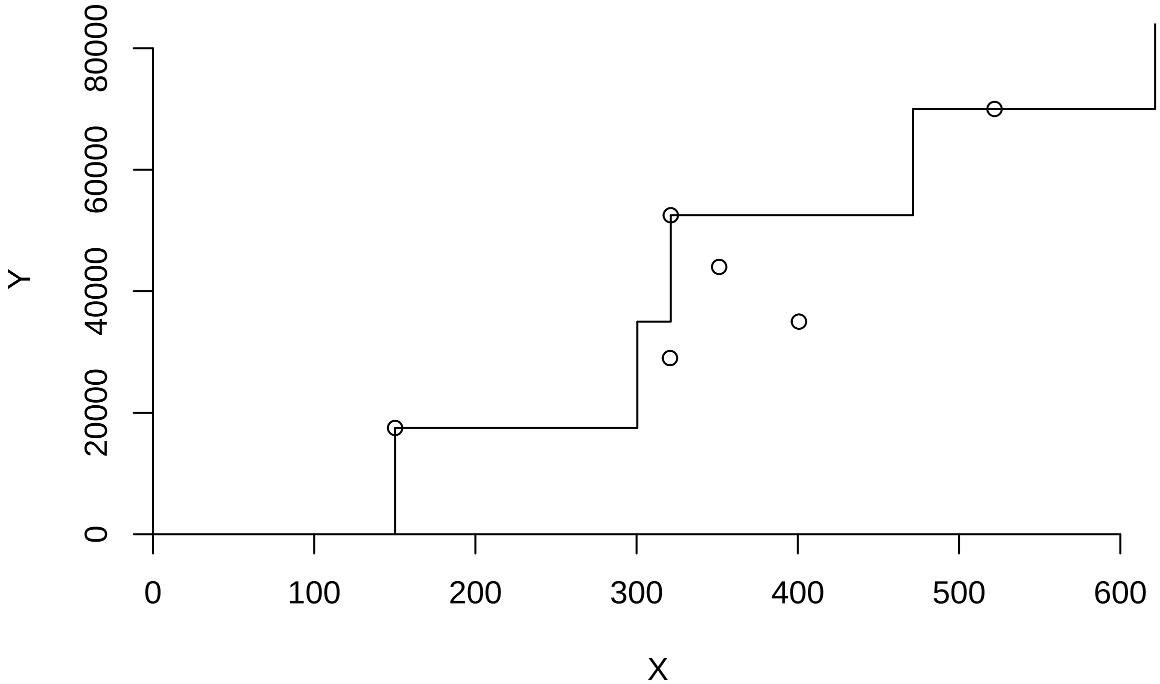
FDH Plot

```
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