QMM DEA

Gloria Stephen

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
```

```
ddf_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(ddf_values) <- c("DMU","Staff_Hours_Per_Day","Supplies_Per_Day","Reimbursed_Patient_Days","Pri
table.dataframe <- as.table(ddf_values)
table.dataframe
```

```
##   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 the Constant Returns to Scale (CRS)

```r
x1 <- matrix(c(150,400,320,520,350,320,
            0.2,0.7,1.2,2.0,1.2,0.7),ncol=2)
y1 <- matrix(c(14000,14000,42000,28000,19000,14000,
                3500,21000,10500,42000,25000,15000),ncol=2)
colnames(y1) <- c("Reimbursed_Patient_Days","Privately_Paid_Patient_Days")
colnames(x1) <- c("Staff_Hours_Per_Day","Supplies_Per_Day")
DEA_CRS <-dea(x1, y1, RTS = "crs")
DEA_CRS
```

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

```r
peers(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(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

It is observed that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient.*

That Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 that are the inefficient facilities.*

Facility 5 is 97.75 % is efficient leaving 2.25 % as inefficient and Facility 6 is 86.75 % efficient leaving 13.25 % as inefficient

#Calculating the Decreasing Returns to Scale (DRS)

```
DEA_CRS <- dea(x1, y1, RTS = "drs")
DEA_CRS
```

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

```
peers(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
```

```
lambda(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
```

#DRS Observations # It is observed that simiarly that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient # It is also observed that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities # Facility 5 is 97.75 % efficient leaving 2.25 % as inefficient and Facility 6 is 86.75 % efficient leaving 13.25 % as inefficient

Calculating the Increasing Returns to Scale (IRS)

```
DEA_IRS <- dea(x1, y1, RTS = "irs")
DEA_IRS
```

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

```
peers(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(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
```

The IRS Observations

It is observed that simiarly that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient

It is also observed that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities

Facility 6 is 89.63 % efficient leaving 10.37 % as inefficient

Calculating the Variable Returns to Scale (VRS)

```
DEA_VRS <- dea(x1, y1, RTS = "vrs")
DEA_VRS
```

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

```
peers(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
```

```
lambda(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
```

The VRS Observations

It is observed that simiarly that Facility 1, Facility 2, Facility 3 and Facility 4 are efficient

It is also observed that Facility 1, Facility 2 and Facility 4 are the peer members for Facility 5 and Facility 6 which are the inefficient facilities

Facility 6 is 89.63 % efficient leaving 10.37 % as inefficient.*

Calculating the Free Disposability Hull (FDH)

```
DEA_FDH <- dea(x1, y1, RTS = "fdh")
DEA_FDH
```

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

```
peers(DEA_FDH)
```

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

```
lambda(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
```

The FDH Observations

The DMUs are efficient, this is FDH follows this principle such that it detects a small level of efficiency

Calculating the Free Replicability Hull (FRH)

```
# The FRH is calculated by specifying RTS = "add"
DEA_FRH <- dea(x1, y1, RTS = "add")
DEA_FRH
```

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

```
peers(DEA_FRH)
```

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

```
lambda(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

All the DMUs are efficient and it follows the no convexity assumption ensuring that the output is free from disposal and replication.*

Summary of Results - Inefficient DMU's

```
ddf.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",
colnames(ddf.summarise.inefficient) <- c("RTS","Count_Inefficient_DMUs","Name_DMUs","%_Inefficiency","Pe
as.table(ddf.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 DMU's

```
ddf.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(ddf.summarise.efficient) <- c("RTS", "Efficient_DMUs")
as.table(ddf.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: # A brief gist of the scales are as follows

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

The Decreasing, Increasing and Varying Returns to Scale (DRS, IRS, and VRS) dispersion scales aid in our understanding of what to increase and what to decrease 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.

Constant Returns to Scale (CRS)

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, with respective weights of 0.34, 0.4, and 0.13.

In general, CRS aids us in determining whether any potential DMUs may be scaled up or down; in this example, DMUs 1, 2, 3, and 4 can be scaled up.

Decreasing Returns to Scale (DRS)

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, with respective weights of 0.34, 0.4, and 0.13.

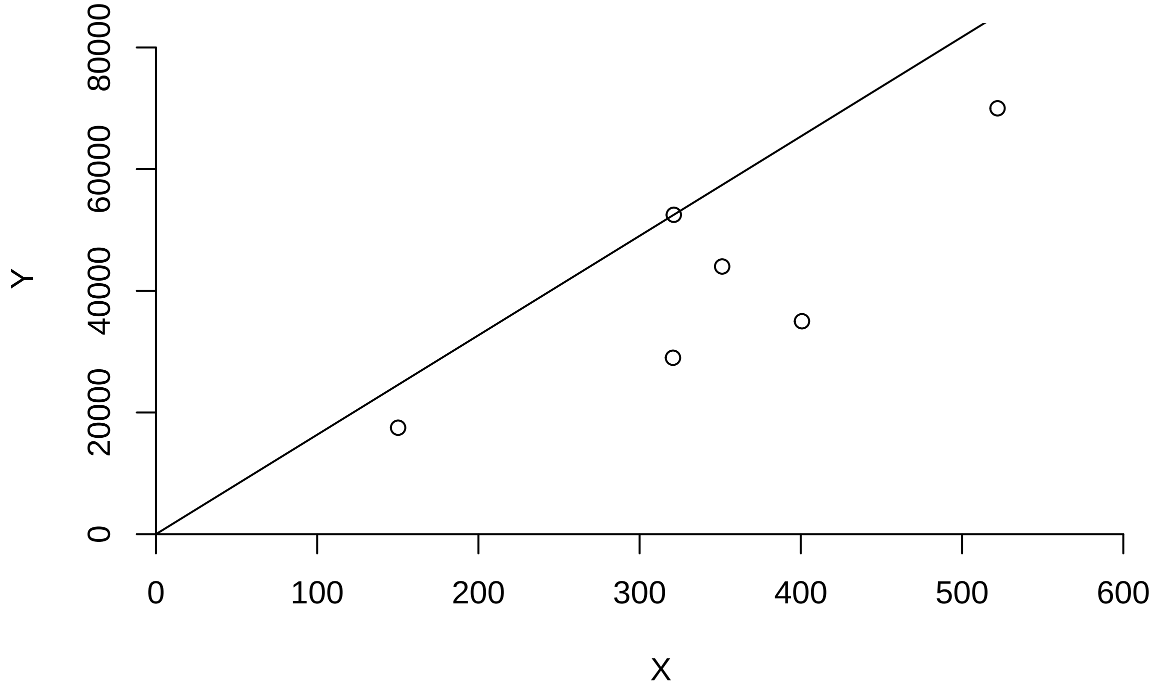
By looking at the inefficient DMUs, in this case DMUs 5 and 6, we may determine whether there are any alternative DMUs where we can scale the processes. Since the CRS values are the base original, they can also be used to retrieve this item.

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

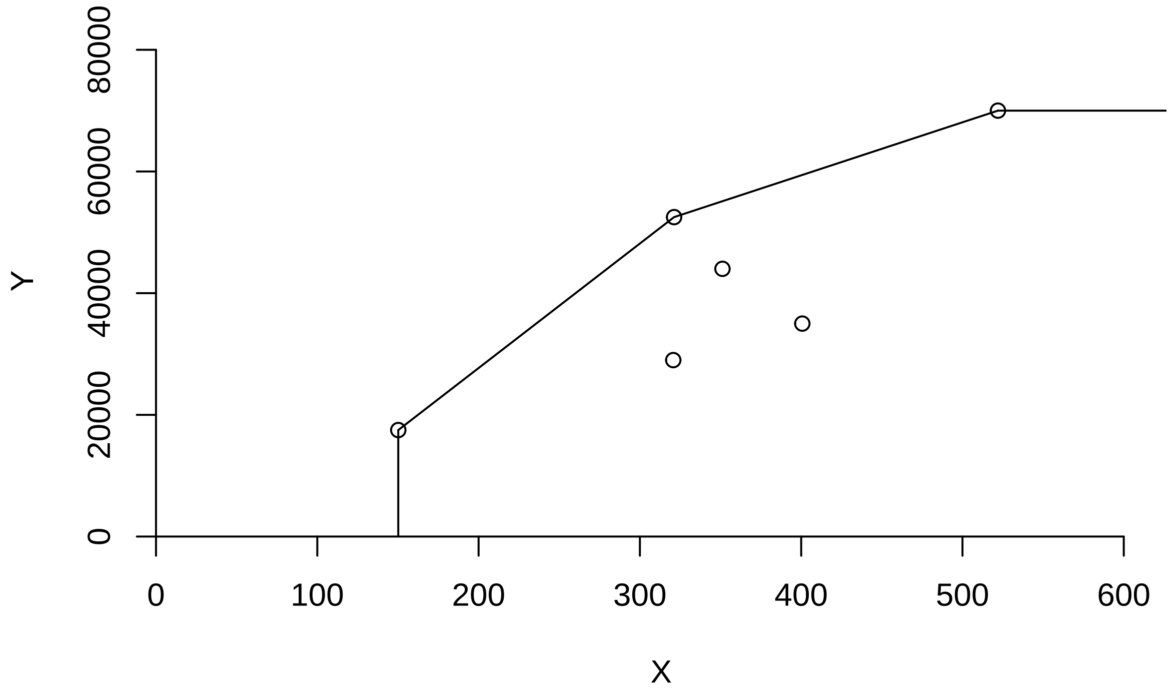
CRS Plot

```
dea.plot(x1, y1, RTS='crs')
```



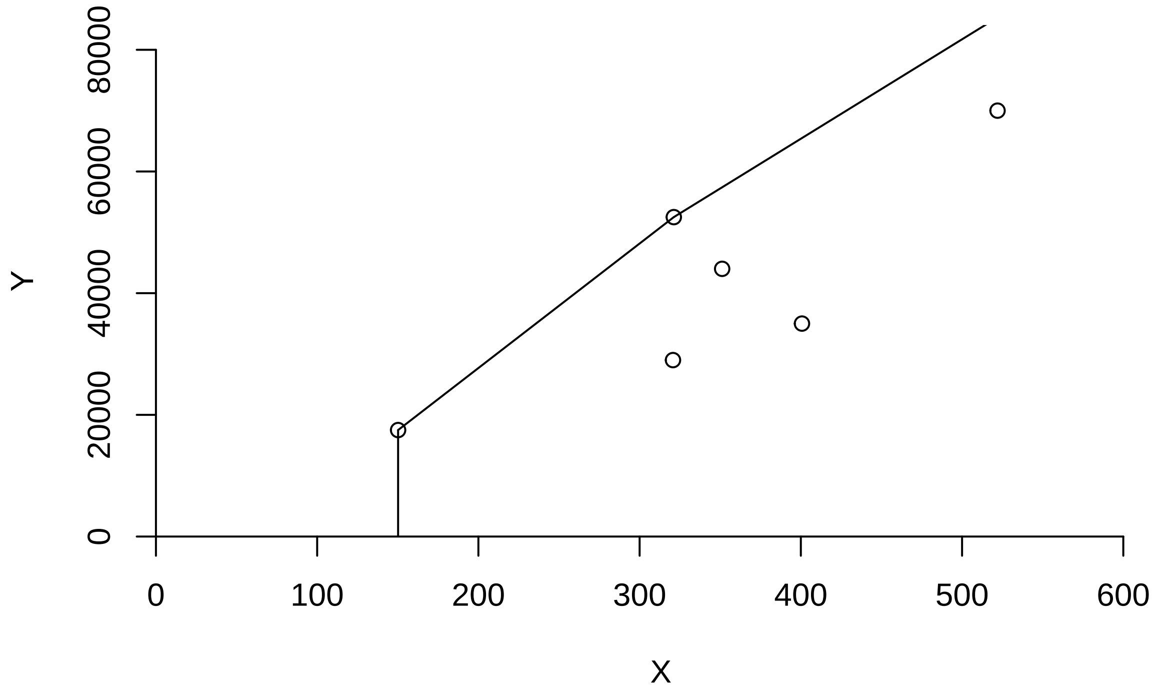
DRS Plot

```
dea.plot(x1,y1,RTS="vrs")
```



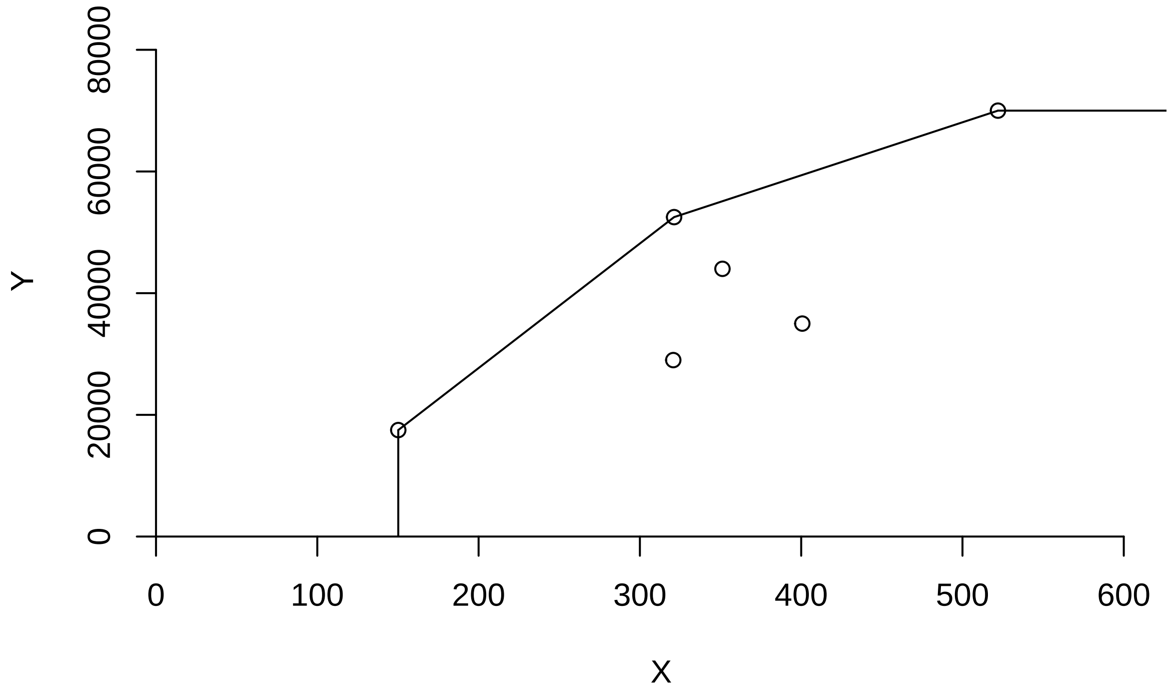
IRS PLOT

```
dea.plot(x1,y1,RTS="irs")
```



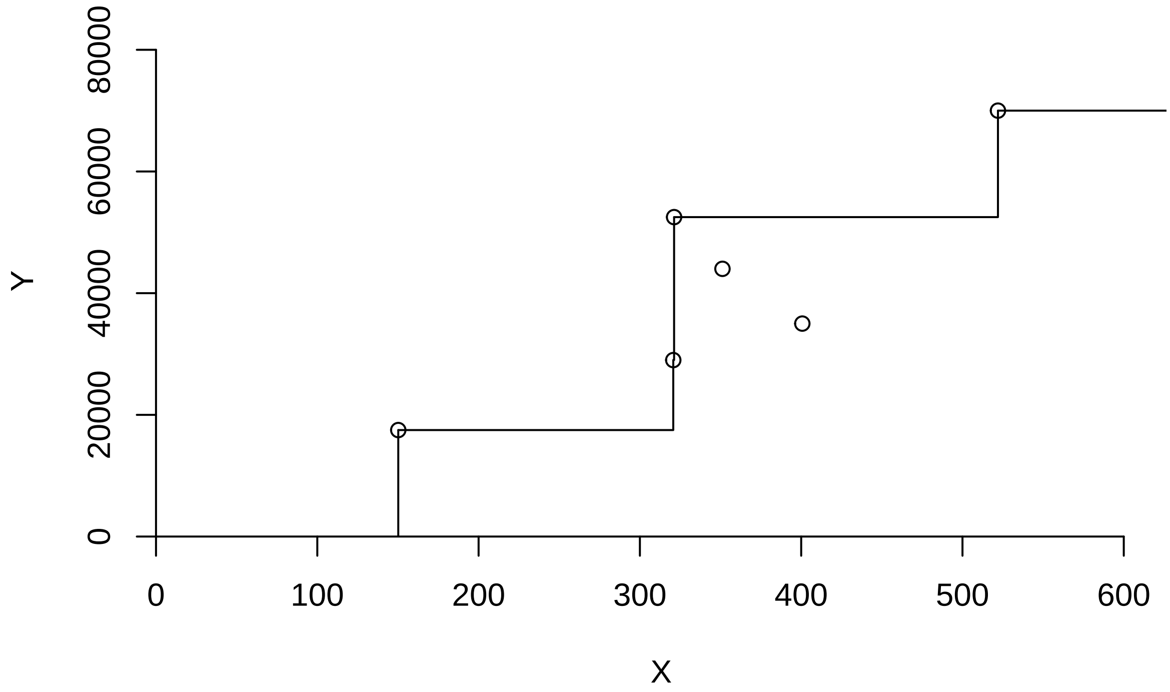
#VRS PLOT

```
dea.plot(x1,y1,RTS="vrs")
```



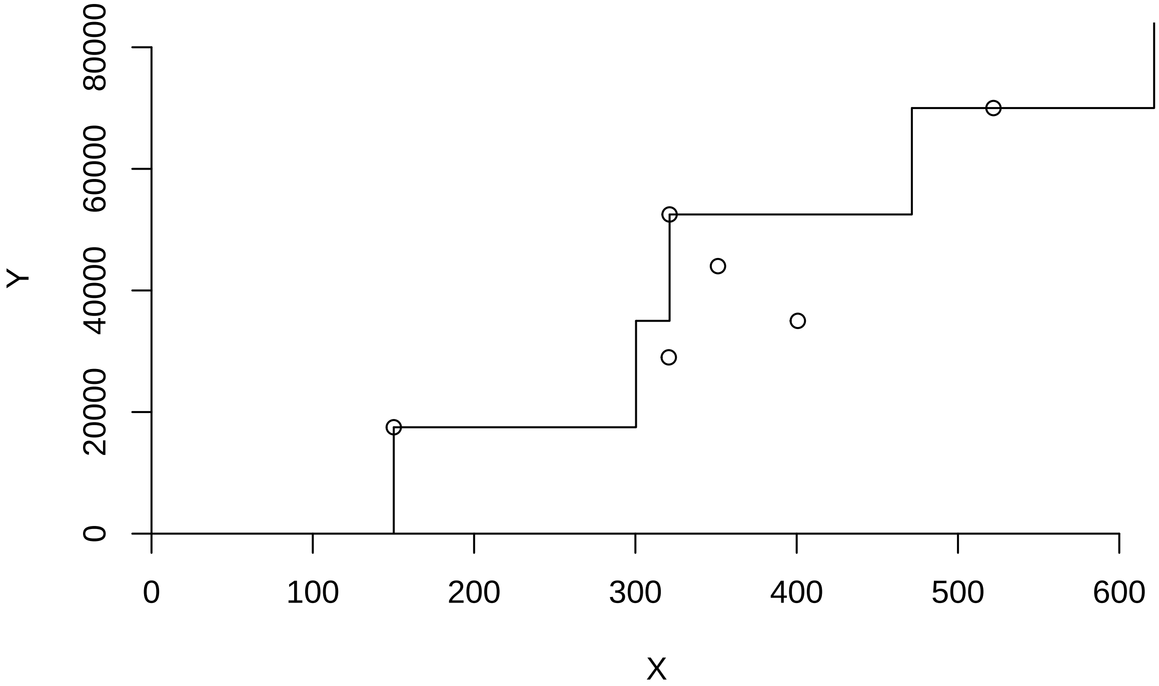
FDH Plot

```
dea.plot(x1,y1,RTS="fdh")
```



# FRH Plot

```
dea.plot(x1,y1,RTS="add")
```


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