mbruner3_8

Mark Bruner

10/19/2020

```
rm(list=ls())
```

```
library(Benchmarking)
```

```
## Loading required package: lpSolveAPI

## Loading required package: ucminf

## Loading required package: quadprog
```

```
library(lpSolveAPI)
library(ucminf)
```

HOPE VALLEY HEALTH CARE ASSOC.

DEA Formulation

Our DMU's are the 6 different nursing home facilities. The inputs are staffing labor (measured in average hours per day) and the cost of supplies (in thousands of dollars per day). The outputs are the number of patient-days reimbursed by thirdparty sources and the number of patient-days reimbursed privately.

QUESTIONS 1 & 2: Formulate and Model Analysis

Summary of Performance Data

```
y <- as.data.frame(matrix(c(14000, 3500, 14000, 21000, 42000, 10500, 28000, 42000, 19000, 25000, 14000,
x <- as.data.frame(matrix(c(150, .2, 400, .7, 320, 1.2, 520, 2.0, 350, 1.2, 320, .7),ncol = 2, byrow = 

hope_valley <- cbind(x, y)
rownames(hope_valley) <- c("F1","F2","F3","F4","F5","F6")
colnames(hope_valley) <- c("reimbur_patient_day", "priv_patient_day", "st_hour_day","supplies_day")

hope_valley
```

```
##    reimbur_patient_day priv_patient_day st_hour_day supplies_day
## F1                 150              0.2       14000         3500
## F2                 400              0.7       14000        21000
## F3                 320              1.2       42000        10500
## F4                 520              2.0       28000        42000
## F5                 350              1.2       19000        25000
## F6                 320              0.7       14000        15000
```

FDH Model

```
fdh <- dea(x, y, RTS = "fdh")            # provide the input and output
fdh
```

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

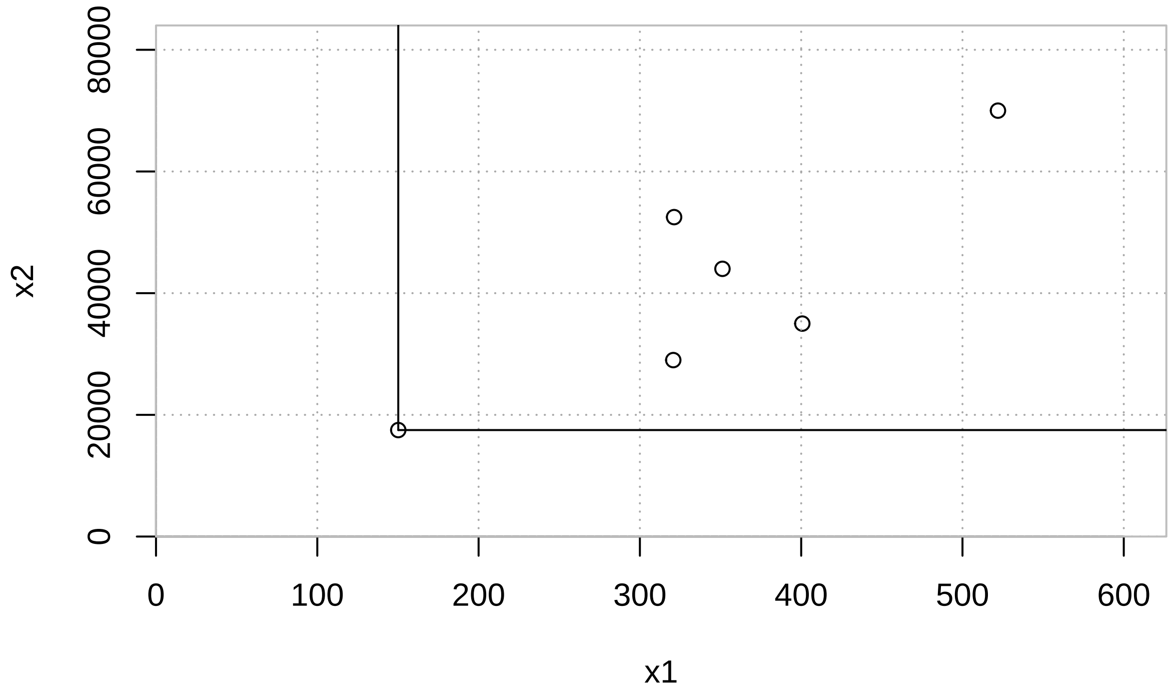
```
peers(fdh)                             # identify the peers
```

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

```
lambda(fdh)                            # identify the relative weights given to the peers
```

```
##      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
```

```
dea.plot.isoquant(x,y,RTS="fdh", GRID = TRUE)     # plot the results
```



```
e_fdh <- fdh$eff
```

The results indicate that DMUs 1 through 6 are all efficient. Their peers are themselves.

VRS Model

```
vrs <- dea(x,y,RTS = "vrs")            # provide the input and output
vrs
```

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

```
peers(vrs)                             # identify the peers
```

```
##      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(vrs)                           # identify the relative weights given to the peers
```

```
##              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
```

```
e_vrs <- vrs$eff
```

The results indicate that DMUs 1 through 5 are efficient. DMU(6) is only 89.63% efficient. Further, the peer units for DMU(6) are 1, 2, and 5, with relative weights 0.401, .342 and 0.256.

DRS Model

```
drs <- dea(x,y,RTS = "drs")           # provide the input and output
drs
```

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

```
peers(drs)                            # identify the peers
```

```
##      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(drs)                           # identify the relative weights given to the peers
```

```
##              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
```

```
e_drs <- drs$eff
```

The results indicate that DMUs 1 through 4 are efficient. DMU(5) is only 97.75% efficient, and DMU(6) is 86.75% efficient. For DMU(6), the peer units are 1, 2, and 5, with relative weights 0.401, .342 and 0.256. **What is interesting is that for DMU(5), it is not fully efficient but it's peer is itself and has not additional weight.**

CRS model

```
crs <- dea(x,y,RTS = "crs")            # provide the input and output
crs
```

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

```
peers(crs)                             # identify the peers
```

```
##      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(crs)                            # identify the relative weights given to the lamdba.
```

```
##             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
```

```
e_crs <- crs$eff
```

The results indicate that DMUs 1 through 4 are efficient. DMU(5) is only 97.75% efficient, and DMU(6) is 86.75% efficient. Further, the peer units for DMU(5) are 1, 2, and 4, with relative weights 0.200, .080 and 0.538. Similarly for DMU(6), the peer units are 1, 2, and 4, with weights 0.342, .395, and 0.131, respectively.

IRS model

```
irs <- dea(x,y,RTS = "irs")            # provide the input and output
irs
```

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

```
peers(irs)                             # identify the peers
```

```
##      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(irs)                               # identify the relative weights given to the peers
```

```
##              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
```

```
e_irs <- irs$eff
```

The results indicate that DMUs 1 through 5 are efficient. DMU(6) is only 89.63% efficient. Further, the peer units for DMU(6) are 1, 2, and 5, with relative weights 0.401, .342 and 0.256.

FRH model

```
frh <- dea(x,y,RTS = "add")               # provide the input and output
frh
```

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

```
peers(frh)                                # identify the peers
```

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

```
lambda(frh)                               # identify the relative weights given to the peers
```

```
##      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
```

```
e_frh <- frh$eff
```

The results indicate that DMUs 1 through 6 are all efficient. Their peers are themselves.

QUESTION 3: Summary of Results

```
hope_valley <- cbind(hope_valley, e_crs, e_drs, e_fdh, e_frh, e_irs, e_vrs)
hope_valley
```

```
##    reimbur_patient_day priv_patient_day st_hour_day supplies_day     e_crs
## F1                 150              0.2       14000         3500 1.0000000
## F2                 400              0.7       14000        21000 1.0000000
## F3                 320              1.2       42000        10500 1.0000000
## F4                 520              2.0       28000        42000 1.0000000
## F5                 350              1.2       19000        25000 0.9774987
## F6                 320              0.7       14000        15000 0.8674521
##        e_drs e_fdh e_frh     e_irs     e_vrs
## F1 1.0000000     1     1 1.0000000 1.0000000
## F2 1.0000000     1     1 1.0000000 1.0000000
## F3 1.0000000     1     1 1.0000000 1.0000000
## F4 1.0000000     1     1 1.0000000 1.0000000
## F5 0.9774987     1     1 1.0000000 1.0000000
## F6 0.8674521     1     1 0.8963283 0.8963283
```

QUESTION 4

Compare/Contrast

All six models agree that DMU's 1 to 4 are efficient. DMU 5 and 6 are the facilities where they differ in results. Although DMU 5 may be sufficiently high enough at 97% to also classify it as efficient since the other models classified it as so. DMU(6), however, has similiar efficiency of 4 out of the 6 models. This tells me that the performance at this facility can be improved by looking at the operations at 1 and 2. Two of the models include DMU(5) as a peer but not DMU(4), while the other two have DMU(4) as the peer and not DMU(5). I would say that DMU(4) would be the better peer to use since all the models include it as efficient while 5 is may not be fully efficient.