Assignment 8

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

We will now run DEA analysis using the benchmarking library.

```
library(lpSolveAPI)
#install.packages("Benchmarking")
library(Benchmarking)
```

```
## Warning: package 'Benchmarking' was built under R version 4.0.3

## Loading required package: ucminf

## Warning: package 'ucminf' was built under R version 4.0.3

## Loading required package: quadprog

## Warning: package 'quadprog' was built under R version 4.0.3
```

Now, we read our input data. We will read the data as input and output as vectors.

```
x <- matrix(c(150,400,320,520,350,320,200,700,1200,2000,1200,700),ncol = 2)
y <- matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,15000),ncol = 2)
colnames(y) <- c("RemPatientDays","PrivatePaidDays")
colnames(x) <- c("StaffHrsPerDay","SuppPerDay")
x
```

```
##      StaffHrsPerDay SuppPerDay
## [1,]            150        200
## [2,]            400        700
## [3,]            320       1200
## [4,]            520       2000
## [5,]            350       1200
## [6,]            320        700
```

```
y
```

```
##      RemPatientDays PrivatePaidDays
## [1,]          14000            3500
## [2,]          14000           21000
## [3,]          42000           10500
## [4,]          28000           42000
## [5,]          19000           25000
## [6,]          14000           15000
```

We now run the DEA analysis. We use the option of CRS, Constant Return to Scale.

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

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

```r
peers(e)                         # 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
```

```r
lambda(e)                        # 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
```

```r
#dea.plot.isoquant(x,y,RTS="crs")     # plot the results
```

The results indicate that DMUs 1, 2, 3 and 4 are efficient. DMU(6) is only 86% efficient, and DMU(5) is 97% efficient. Further, the peer units for DMU(5) are 1, 2 and 4, with relative weights 0.20, 0.08 and 0.53. Similary for DMU(6), the peer units are 1, 2 and 4, with weights 0.34, 0.39 and 0.13, respectively.

We now run the DEA analysis. We use the option of FDH.

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

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

```r
peers(e)                         # identify the peers
```

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

```r
lambda(e)                        # 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
```

```r
#dea.plot.isoquant(x,y,RTS="crs")     # plot the results
```

We now run the DEA analysis. We use the option of VRS.

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

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

```r
peers(e)                        # 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
```

```r
lambda(e)                       # 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
```

```r
#dea.plot.isoquant(x,y,RTS="crs")     # plot the results
```

We now run the DEA analysis. We use the option of IRS

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

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

```r
peers(e)                        # 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
```

```r
lambda(e)                       # 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
```

```r
#dea.plot.isoquant(x,y,RTS="crs")     # plot the results
```

We now run the DEA analysis. We use the option of DRS

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

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

```r
peers(e)                        # 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
```

```r
lambda(e)                       # 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
```

```r
#dea.plot.isoquant(x,y,RTS="crs")     # plot the results
```

We now run the DEA analysis. We use the option of FRH

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

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

```r
peers(e)                        # identify the peers
```

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

```r
lambda(e)                       # 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
```

```r
#dea.plot.isoquant(x,y,RTS="crs")     # plot the results
```