DEA

Ryan McGuinness

10/29/2021

Problem Description

The Hope Valley Health Care Association operates six nursing homes. Each of these facilities are evaluated on the efficiency using average hours per day and cost of supplies per day as inputs and number of patient-days reimbursed by third party sources and number of patient-days reimbursed privately as outputs. A summary of performance is shown in the table below.

Nursing Home	Staff Hours per Day	Supply Cost per Day (thousands of dollars)	Reimbursed Patient-Days	Privately Paid Patient-Days
Facility 1	150	0.2	14000	3500
Facility 2	400	0.7	14000	21500
Facility 3	320	1.2	42000	10500
Facility 4	520	2.0	28000	42000
Facility 5	350	1.2	19000	25000
Facility 6	320	0.7	14000	15000

Preparing DEA Analysis

Importing Required Packages

```
library(Benchmarking)
```

```
## Loading required package: lpSolveAPI
```

```
## Loading required package: ucminf
```

```
## Loading required package: quadprog
```

Reading Input Data

The `dea` function requires the input and output data to be in matrix format. So, we must start by putting the data into matrix form and give descriptive column names to those matrices.

```r
inputs  <- matrix(c(150, 400, 320, 520, 350, 320,
                    0.2, 0.7, 1.2, 2.0, 1.2, 0.7), ncol=2) # Defining inputs
outputs <- matrix(c(14000, 14000, 42000, 28000, 19000, 14000,
                    3500, 21000, 10500, 42000, 25000, 15000), ncol=2) # Defining outputs

# Defining column names
colnames(inputs)  <- c("Daily Staff Hours", "Daily Supplies Cost")
colnames(outputs) <- c("Reimbursed Patient-Days", "Privately-Paid Patient Days")

inputs
```

```
##      Daily Staff Hours Daily Supplies Cost
## [1,]               150                 0.2
## [2,]               400                 0.7
## [3,]               320                 1.2
## [4,]               520                 2.0
## [5,]               350                 1.2
## [6,]               320                 0.7
```

```r
outputs
```

```
##      Reimbursed Patient-Days Privately-Paid Patient Days
## [1,]                   14000                        3500
## [2,]                   14000                       21000
## [3,]                   42000                       10500
## [4,]                   28000                       42000
## [5,]                   19000                       25000
## [6,]                   14000                       15000
```

Creating DEA Analysis Function

To save ourselves from repetitive code, let's define a function that will perform the DEA under the given assumption.

```r
analyze_dea <- function(assumption) {
  analysis <- dea(inputs, outputs, RTS=assumption)
  print(eff(analysis))
  print(peers(analysis))
  print(lambda(analysis))
}
```

In order to use the function, we just supply the DEA assumption to use and the function will print the efficiencies, peers, and lambdas.

Performing DEA Analysis

Now, we use the function created above to perform DEA on each of the six assumptions in the order of FDH, CRS, VRS, IRS, DRS, and FRH.

Free Disposability Hull

```r
analyze_dea("FDH")
```

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

Constant Return to Scale

```r
analyze_dea("CRS")
```

```
## [1] 1.0000000 1.0000000 1.0000000 1.0000000 0.9774987 0.8674521
##      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
##             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
```

Varying Return to Scale

```r
analyze_dea("VRS")
```

```
## [1] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.8963283
##      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
##             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
```

Increasing Return to Scale

```
analyze_dea("IRS")
```

```
## [1] 1.0000000 1.0000000 1.0000000 1.0000000 1.0000000 0.8963283
##      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
##             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
```

Decreasing Return to Scale

```
analyze_dea("DRS")
```

```
## [1] 1.0000000 1.0000000 1.0000000 1.0000000 0.9774987 0.8674521
##      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
##             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
```

Free Replicability Hull

```
analyze_dea("ADD")
```

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

Summary of Results

The efficiency of each firm under the given assumption is given below.

DMU	E^{FDH}	E^{CRS}	E^{VRS}	E^{IRS}	E^{DRS}	E^{FRH}
Facility 1	1	1	1	1	1	1
Facility 2	1	1	1	1	1	1
Facility 3	1	1	1	1	1	1
Facility 4	1	1	1	1	1	1
Facility 5	1	0.977	1	1	0.977	1
Facility 6	1	0.867	0.896	0.896	0.867	1

Under all six assumptions, facilities 1, 2, 3, and 4 are considered efficient, and without the assumption of convexity all six facilities are efficient.

Under the VRS and IRS assumptions, facility 6 is inefficient. Its peers are facilities 1, 2, and 5 with $\lambda_1^6 = 0.401$, $\lambda_2^6 = 0.342$, and $\lambda_5^6 = 0.256$.

Under the CRS and DRS assumptions, facilities 5 and 6 are both inefficient. The peers of both facilities 5 and 6 are facilities 1, 2, and 4, with $\lambda_1^5 = 0.2$, $\lambda_2^5 = 0.080$, $\lambda_3^5 = 0.538$, $\lambda_1^6 = 0.343$, $\lambda_2^6 = 0.395$, and $\lambda_3^6 = 0.131$.