Ecurley1_Assignment 8

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

```
## Warning: package 'Benchmarking' was built under R version 4.0.5
```

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

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

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

```
x <- matrix(c(150,400,320,520,350,320,.2,.7,1.2,2.0,1.2,.7),ncol = 2)
y <- matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,15000),ncol = 2)
colnames(y) <- c("3rd Party","Private")
colnames(x) <- c("Staff Hours","Supplies")
x
```

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

```
y
```

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

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

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

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

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

```
lambda(e)
```

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

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

```
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
```

```
lambda(e)
```

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

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

```
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
```

```
lambda(e)
```

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

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

```
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
```

```
lambda(e)
```

```
##             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 <- 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)
```

```
##             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
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)
```

```
##      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
x <- matrix(c(1,1,1,1,1,1,
         1.0000,1.0000,1.0000,1.0000,0.9775,0.8675,
         1.0000,1.0000,1.0000,1.0000,1.0000,0.8963,
         1.0000,1.0000,1.0000,1.0000,1.0000,0.8963,
         1.0000,1.0000,1.0000,1.0000,0.9775,0.8675,
```

```
         1,1,1,1,1,1
          ),ncol = 6)
colnames(x) <- c("FDH","CRS","VRS","IRS","DRS","FRH/FDH+")
x
```

```
##      FDH    CRS    VRS    IRS    DRS FRH/FDH+
## [1,]   1 1.0000 1.0000 1.0000 1.0000        1
## [2,]   1 1.0000 1.0000 1.0000 1.0000        1
## [3,]   1 1.0000 1.0000 1.0000 1.0000        1
## [4,]   1 1.0000 1.0000 1.0000 1.0000        1
## [5,]   1 0.9775 1.0000 1.0000 0.9775        1
## [6,]   1 0.8675 0.8963 0.8963 0.8675        1
```

We can see here that under the assumption of FDH and FHR all sites are preforming at peek efficiency However when we consider other methods such as CRS VRS IRS and DRS we see that site 5 & 6 perform under peek efficiency.

For example under CRS site 5 is %97.75 efficient, thus they should be able to improve their efficiency by mimicing the effort of their closest peers (Sites 1,2,&4) This same process can be applied to the other methods and peers.

Under the assumption of CRS and DRS site 5 should mimic sites 1,2, & 4 Under the assumption of CRS and DRS site 6 should mimic sites 1,2, & 4 Under the assumption of VRS and IRS site 6 should mimic sites 1,2, & 5

The weights as indicated by the lambdas will indicate how important each of those sites should be weighted.