Assignment 7 - DEA

Gordon Wall (gwall2)

```
library(lpSolve)
library(lpSolveAPI)
library(Benchmarking)
## Loading required package: ucminf
```

##DMU1 Formulation

```
dmu1 <- read.lp("DMU1.lp")
dmu1
## Model name:
##             u1    u2    v1    v2
## Maximize    14   3.5     0     0
## R1          14   3.5  -150  -0.2  <=  0
## R2          14    21  -400  -0.7  <=  0
## R3          42  10.5  -320  -1.2  <=  0
## R4          28    42  -520    -2  <=  0
## R5          19    25  -350  -1.2  <=  0
## R6          14    15  -320  -0.7  <=  0
## R7           0     0   150   0.2   =  1
## Kind       Std   Std   Std   Std
## Type      Real  Real  Real  Real
## Upper      Inf   Inf   Inf   Inf
## Lower        0     0     0     0
solve(dmu1)
## [1] 0
get.objective(dmu1)
## [1] 1
get.variables(dmu1)
## [1] 0.071428571 0.000000000 0.005172414 1.120689655
```

##DMU2 Formulation

```
dmu2 <- read.lp("DMU2.lp")
dmu2
## Model name:
##             u1    u2    v1    v2
## Maximize    14    21     0     0
## R1          14   3.5  -150  -0.2  <=  0
## R2          14    21  -400  -0.7  <=  0
## R3          42  10.5  -320  -1.2  <=  0
## R4          28    42  -520    -2  <=  0
## R5          19    25  -350  -1.2  <=  0
## R6          14    15  -320  -0.7  <=  0
## R7           0     0   400   0.7   =  1
## Kind       Std   Std   Std   Std
## Type      Real  Real  Real  Real
## Upper      Inf   Inf   Inf   Inf
## Lower        0     0     0     0
solve(dmu2)
```

```
## [1] 0
get.objective(dmu2)
## [1] 1
get.variables(dmu2)
## [1] 0.000000000 0.047619048 0.001376147 0.642201835
```

##DMU3 Formulation

```
dmu3 <- read.lp("DMU3.lp")
dmu3
## Model name:
##             u1    u2    v1    v2
## Maximize    42  10.5     0     0
## R1          14   3.5  -150  -0.2  <=  0
## R2          14    21  -400  -0.7  <=  0
## R3          42  10.5  -320  -1.2  <=  0
## R4          28    42  -520    -2  <=  0
## R5          19    25  -350  -1.2  <=  0
## R6          14    15  -320  -0.7  <=  0
## R7           0     0   320   1.2   =  1
## Kind       Std   Std   Std   Std
## Type      Real  Real  Real  Real
## Upper      Inf   Inf   Inf   Inf
## Lower        0     0     0     0
solve(dmu3)
## [1] 0
get.objective(dmu3)
## [1] 1
get.variables(dmu3)
## [1] 0.023809524 0.000000000 0.001724138 0.373563218
```

##DMU4 Formulation

```
dmu4 <- read.lp("DMU4.lp")
dmu4
## Model name:
##             u1    u2    v1    v2
## Maximize    28    42     0     0
## R1          14   3.5  -150  -0.2  <=  0
## R2          14    21  -400  -0.7  <=  0
## R3          42  10.5  -320  -1.2  <=  0
## R4          28    42  -520    -2  <=  0
## R5          19    25  -350  -1.2  <=  0
## R6          14    15  -320  -0.7  <=  0
## R7           0     0   520     2   =  1
## Kind       Std   Std   Std   Std
## Type      Real  Real  Real  Real
## Upper      Inf   Inf   Inf   Inf
## Lower        0     0     0     0
solve(dmu4)
## [1] 0
get.objective(dmu4)
## [1] 1
get.variables(dmu4)
## [1] 0.0000000000 0.0238095238 0.0006880734 0.3211009174
```

##DMU5 Formulation

```
dmu5 <- read.lp("DMU5.lp")
dmu5
## Model name:
##             u1    u2    v1    v2
## Maximize    19    25     0     0
## R1          14   3.5  -150  -0.2  <=  0
## R2          14    21  -400  -0.7  <=  0
## R3          42  10.5  -320  -1.2  <=  0
## R4          28    42  -520    -2  <=  0
## R5          19    25  -350  -1.2  <=  0
## R6          14    15  -320  -0.7  <=  0
## R7           0     0   350   1.2   =  1
## Kind       Std   Std   Std   Std
## Type      Real  Real  Real  Real
## Upper      Inf   Inf   Inf   Inf
## Lower        0     0     0     0
solve(dmu5)
## [1] 0
get.objective(dmu5)
## [1] 0.9774987
get.variables(dmu5)
## [1] 0.011512297 0.030350602 0.001098901 0.512820513
```

##DMU6 Formulation

```
dmu6 <- read.lp("DMU6.lp")
dmu6
## Model name:
##             u1    u2    v1    v2
## Maximize    14    15     0     0
## R1          14   3.5  -150  -0.2  <=  0
## R2          14    21  -400  -0.7  <=  0
## R3          42  10.5  -320  -1.2  <=  0
## R4          28    42  -520    -2  <=  0
## R5          19    25  -350  -1.2  <=  0
## R6          14    15  -320  -0.7  <=  0
## R7           0     0   320   0.7   =  1
## Kind       Std   Std   Std   Std
## Type      Real  Real  Real  Real
## Upper      Inf   Inf   Inf   Inf
## Lower        0     0     0     0
solve(dmu6)
## [1] 0
get.objective(dmu6)
## [1] 0.8674521
get.variables(dmu6)
## [1] 0.016200295 0.042709867 0.001546392 0.721649485
```

##DEA Section

```
x <- matrix(c(14,14,42,28,19,14,3.5,21,10.5,42,25,15),ncol = 2)
y <- matrix(c(150,400,320,520,350,320,0.2,0.7,1.2,2.0,1.2,0.7),ncol = 2)
colnames(y) <- c("Staff Hours/Day","Supplies/Day")
colnames(x) <- c("Reimbursed","Private")
x
##      Reimbursed Private
```

```
## [1,]         14     3.5
## [2,]         14    21.0
## [3,]         42    10.5
## [4,]         28    42.0
## [5,]         19    25.0
## [6,]         14    15.0
y
##      Staff Hours/Day Supplies/Day
## [1,]             150          0.2
## [2,]             400          0.7
## [3,]             320          1.2
## [4,]             520          2.0
## [5,]             350          1.2
## [6,]             320          0.7
```

#CRS Analysis

```
crs <- dea(x,y,RTS = "crs")
crs
## [1] 1.0000 1.0000 1.0000 1.0000 0.9908 1.0000
peers(crs)
##      peer1 peer2 peer3
## [1,]     1    NA    NA
## [2,]     2    NA    NA
## [3,]     3    NA    NA
## [4,]     4    NA    NA
## [5,]     3     4     6
## [6,]     6    NA    NA
lambda(crs)
##      L1 L2         L3        L4        L6
## [1,]  1  0 0.00000000 0.0000000 0.0000000
## [2,]  0  1 0.00000000 0.0000000 0.0000000
## [3,]  0  0 1.00000000 0.0000000 0.0000000
## [4,]  0  0 0.00000000 1.0000000 0.0000000
## [5,]  0  0 0.03481462 0.4834408 0.2733441
## [6,]  0  0 0.00000000 0.0000000 1.0000000
```

#IRS Analysis

```
irs <- dea(x,y,RTS = "irs")
irs
## [1] 1 1 1 1 1 1
peers(irs)
##      peer1
## [1,]     1
## [2,]     2
## [3,]     3
## [4,]     4
## [5,]     5
## [6,]     6
lambda(irs)
##      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
```

#DRS Analysis

```
drs <- dea(x,y,RTS = "drs")
drs
## [1] 1.0000 1.0000 1.0000 1.0000 0.9908 1.0000
peers(drs)
##      peer1 peer2 peer3
## [1,]     1    NA    NA
## [2,]     2    NA    NA
## [3,]     3    NA    NA
## [4,]     4    NA    NA
## [5,]     3     4     6
## [6,]     6    NA    NA
lambda(drs)
##      L1 L2         L3        L4        L6
## [1,]  1  0 0.00000000 0.0000000 0.0000000
## [2,]  0  1 0.00000000 0.0000000 0.0000000
## [3,]  0  0 1.00000000 0.0000000 0.0000000
## [4,]  0  0 0.00000000 1.0000000 0.0000000
## [5,]  0  0 0.03481462 0.4834408 0.2733441
## [6,]  0  0 0.00000000 0.0000000 1.0000000
```

#VRS Analysis

```
vrs <- dea(x,y,RTS = "vrs")
vrs
## [1] 1 1 1 1 1 1
peers(vrs)
##      peer1
## [1,]     1
## [2,]     2
## [3,]     3
## [4,]     4
## [5,]     5
## [6,]     6
lambda(vrs)
##      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
```

#FDH Analysis

```
fdh <- dea(x,y,RTS = "fdh")
fdh
## [1] 1 1 1 1 1 1
peers(fdh)
##      peer1
## [1,]     1
## [2,]     2
## [3,]     3
```

```
## [4,]     4
## [5,]     5
## [6,]     6
lambda(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
```

#FRH Analysis

```
add <- dea(x,y,RTS = "add")
add
## [1] 1 1 1 1 1 1
peers(add)
##      peer1
## [1,]     1
## [2,]     2
## [3,]     3
## [4,]     4
## [5,]     5
## [6,]     6
lambda(add)
##      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
```

##RESULTS:

Formulation and solving of the 6 DMU's reveals that DMU 1-4 are efficient, while DMU 5 and 6 are inefficient at 97.7% and 86.7% respectively. It would seem that both CRS and DRS analysis produced the same subset of peer units for DMU(5), with peer units being 3, 4, and 6 and relative weights being 0.035, 0.48, and 0.27 respectively.Essentially, these results show us that DMU(5)'s potential efficient performance could be emulated from DMU (3,4 and 6). However, DMU(6) results indicate that while it may not be operating at maximum efficiency, that no possible combination of weights and emulation of other DMU's exists for it to reach full efficiency potential.

///NOTE: I could not, for the life of me, figure out how to condense all this information into tabular format. Forgive me. Instead, I knitted the file to a github document and uploaded that to my git repository as well, so you can view the results that way. (all .lp files necessary are uploaded too) Best, Gordon///