Assignment 5

SABIHA MHATARNAIK

11/7/2021

Question 1:

The Hope Valley Health Care Association owns and operates six nursing homes in adjoining states. An evaluation of their efficiency has been undertaken using two inputs and two outputs. 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 third-party sources and the number of patient-days reimbursed privately.

1) Formulate and perform DEA analysis under all DEA assumptions of FDH, CRS, VRS, IRS, DRS, and FRH.
2) Determine the Peers and Lambdas under each of the above assumptions
3) Summarize your results in a tabular format
4) Compare and contrast the above results

Using Bechmarking Libraries for DEA

We will perform DEA analysis using benchmarking library.Install Benchmarking library if we don't have.

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

```
## Loading required package: lpSolveAPI

## Loading required package: ucminf

## Loading required package: quadprog
```

```
library(readxl)
```

Now, we read our input data. We will read the data from an excel file. Our problem had 6 DMUs with two inputs and two outputs.

Inputs: Staffing Labor, Cost of Supplies

Outputs: No of patient-days reimbursed by third party, No of patient-days reimbursed privately

```
#Reading the data from the excel file
data <- read_excel("DEA.xlsx")
#View the
data
```

```
## # A tibble: 6 x 5
##   DMU        `Staff Hours pe~ `Supplies per D~ `Reimbursed Pat~ `Privately Paid~
##   <chr>                 <dbl>            <dbl>            <dbl>            <dbl>
## 1 Facility 1              150              0.2            14000             3500
## 2 Facility 2              400              0.7            14000            21000
## 3 Facility 3              320              1.2            42000            10500
## 4 Facility 4              520              2              28000            42000
## 5 Facility 5              350              1.2            19000            25000
## 6 Facility 6              320              0.7            14000            15000
```

```
#Facility1 to Facility 6 are the DMUs.Extracting only the first column of DMUs
Names_DMU <- data[1]
Names_DMU
```

```
## # A tibble: 6 x 1
##   DMU
##   <chr>
## 1 Facility 1
## 2 Facility 2
## 3 Facility 3
## 4 Facility 4
## 5 Facility 5
## 6 Facility 6
```

```
#Inputs
inputs <- data[c(2,3)]
inputs
```

```
## # A tibble: 6 x 2
##   `Staff Hours per Day` `Supplies per Day`
##                   <dbl>              <dbl>
## 1                   150                0.2
## 2                   400                0.7
## 3                   320                1.2
## 4                   520                2
## 5                   350                1.2
## 6                   320                0.7
```

```r
#Outputs
outputs <- data[c(4,5)]
outputs
```

```
## # A tibble: 6 x 2
##   `Reimbursed Patient-Days` `Privately Paid Patient-Days`
##                       <dbl>                         <dbl>
## 1                     14000                          3500
## 2                     14000                         21000
## 3                     42000                         10500
## 4                     28000                         42000
## 5                     19000                         25000
## 6                     14000                         15000
```

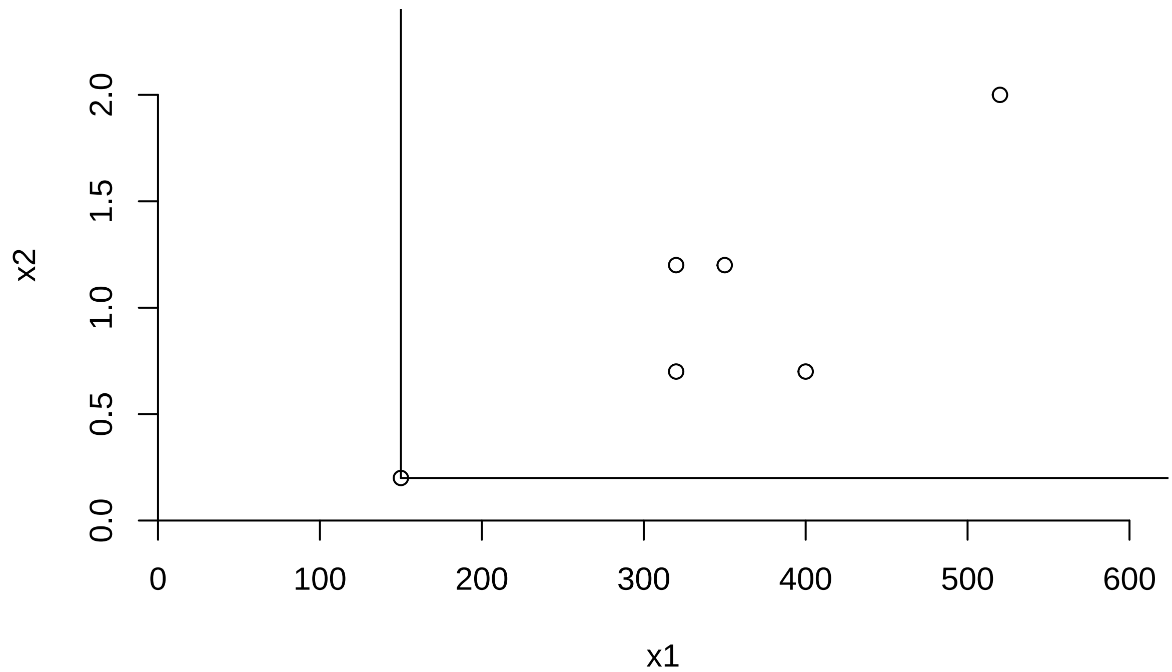
```r
#Creating the input matrix
x <- matrix(c(data$`Staff Hours per Day`,data$`Supplies per Day`),ncol = 2)
#View the input matrix
x
```

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

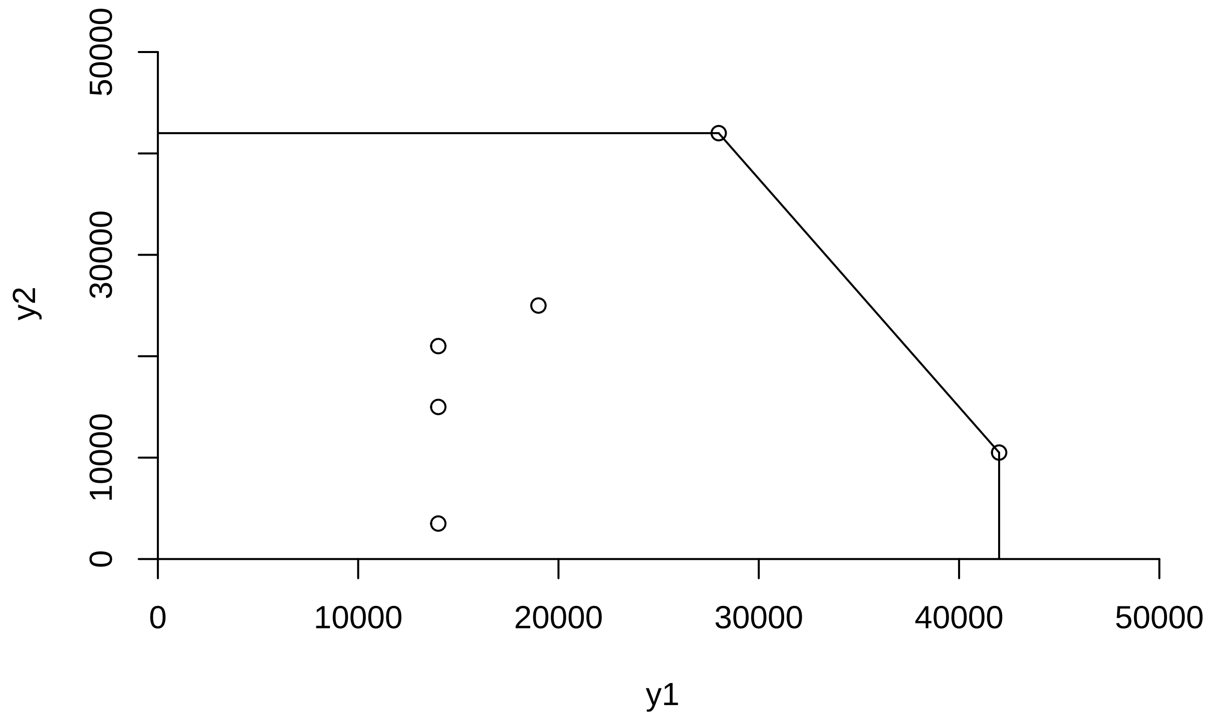
```r
#Creating the output matrix
y <- matrix(c(data$`Reimbursed Patient-Days`,data$`Privately Paid Patient-Days`),ncol = 2)
#View the output matrix
y
```

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

```r
#Plotting the graph for Inputs
dea.plot.isoquant(x[,1],x[,2])
```




```r
#Plotting the graph for Outputs
dea.plot.transform(y[,1],y[,2])
```



Performing the DEA analysis for different assumptions:

We use the option of FDH, Free disposability hull, no convexity assumption

```
#DEA input or output efficiency measures, peers, lambdas and slacks
analysis_fdh <- dea(x,y,RTS = "FDH")
#Show the Efficiency
analysis_fdh
```

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

```
#Show the list of objects calculated
str(analysis_fdh)
```

```
## List of 7
##  $ eff        : num [1:6] 1 1 1 1 1 1
##  $ objval     : num [1:6] 1 1 1 1 1 1
##  $ peers      : int [1:6] 1 2 3 4 5 6
##  $ lambda     : num [1:6, 1:6] 1 0 0 0 0 0 0 1 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "L1" "L2" "L3" "L4" ...
```

```
##  $ RTS        : chr "fdh"
##  $ ORIENTATION: chr "in"
##  $ TRANSPOSE  : logi FALSE
##  - attr(*, "class")= chr "Farrell"
```

```
#Show the peers
peers(analysis_fdh)
```

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

```
#Show the lambda
lambda(analysis_fdh)
```

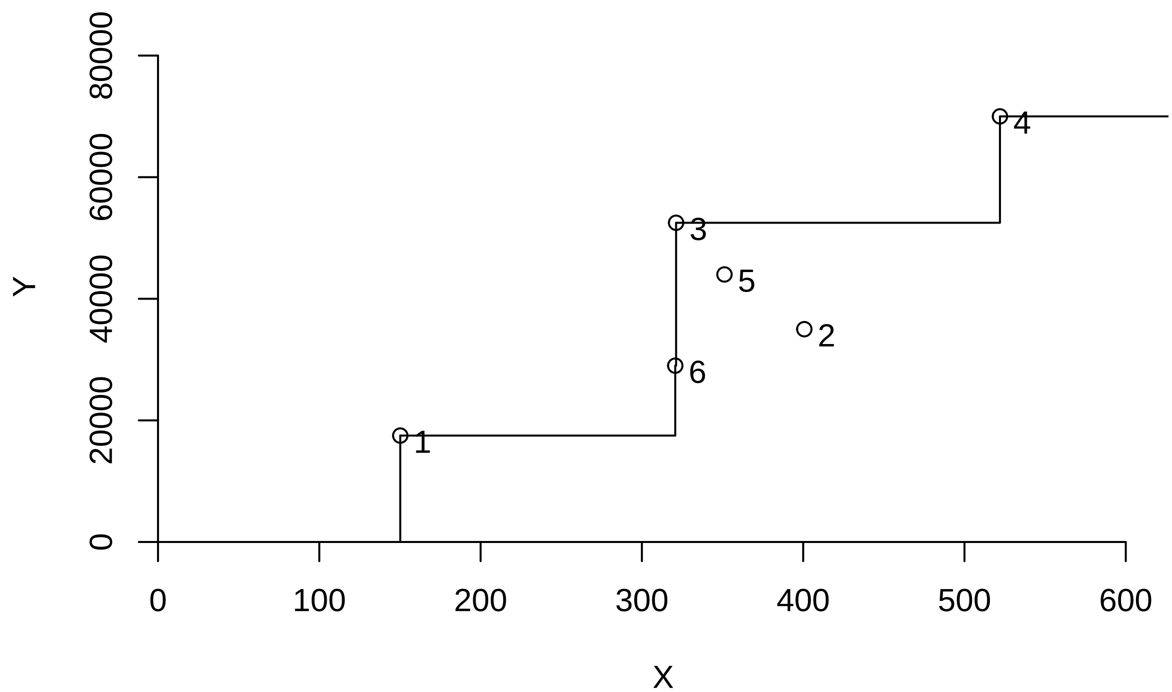
```
##      L1 L2 L3 L4 L5 L6
## [1,]  1  0  0  0  0  0
## [2,]  0  1  0  0  0  0
## [3,]  0  0  1  0  0  0
## [4,]  0  0  0  1  0  0
## [5,]  0  0  0  0  1  0
## [6,]  0  0  0  0  0  1
```

```
#Add the Efficiency, Peers & Lambda values in the table
report1 <- cbind(data, analysis_fdh$eff, analysis_fdh$lambda, analysis_fdh$peers)
#Name the columns of the table
colnames(report1)<- c(names(Names_DMU),names(inputs), names(outputs),'Efficiency','Lambda1','Lambda2','L
#Sow the table
report1
```

```
##          DMU Staff Hours per Day Supplies per Day Reimbursed Patient-Days
## 1 Facility 1                 150              0.2                   14000
## 2 Facility 2                 400              0.7                   14000
## 3 Facility 3                 320              1.2                   42000
## 4 Facility 4                 520              2.0                   28000
## 5 Facility 5                 350              1.2                   19000
## 6 Facility 6                 320              0.7                   14000
##   Privately Paid Patient-Days Efficiency Lambda1 Lambda2 Lambda3 Lambda4
## 1                        3500          1       1       0       0       0
## 2                       21000          1       0       1       0       0
## 3                       10500          1       0       0       1       0
## 4                       42000          1       0       0       0       1
## 5                       25000          1       0       0       0       0
## 6                       15000          1       0       0       0       0
##   Lambda5 Lambda6 Peers
## 1       0       0     1
## 2       0       0     2
## 3       0       0     3
```

```
## 4        0        0      4
## 5        1        0      5
## 6        0        1      6
```

```
#plot the graph for FDH Assumption
dea.plot(x,y,RTS="FDH",txt = rownames(report1))
```



The results indicate that DMUs 1, 2, 3 , 4, 5 and 6 all are efficient.

We use the option of CRS, Constant Return to Scale, convexity and free disposability

```
#DEA input or output efficiency measures, peers, lambdas and slacks
analysis_crs <- dea(x,y,RTS = "CRS")
#Show the Efficiency
analysis_crs
```

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

```
#Show the list of objects calculated
str(analysis_crs)
```

```
## List of 12
##  $ eff        : num [1:6] 1 1 1 1 0.977 ...
##  $ lambda     : num [1:6, 1:6] 1 0 0 0 0.2 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "L1" "L2" "L3" "L4" ...
##  $ objval     : num [1:6] 1 1 1 1 0.977 ...
##  $ RTS        : chr "crs"
##  $ primal     : NULL
##  $ dual       : NULL
##  $ ux         : NULL
##  $ vy         : NULL
##  $ gamma      :function (x)
##  $ ORIENTATION: chr "in"
##  $ TRANSPOSE  : logi FALSE
##  $ param      : NULL
##  - attr(*, "class")= chr "Farrell"
```

```
#Show the peers
peers(analysis_crs)
```

```
##      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
```

```
#Show the lambda
lambda(analysis_crs)
```

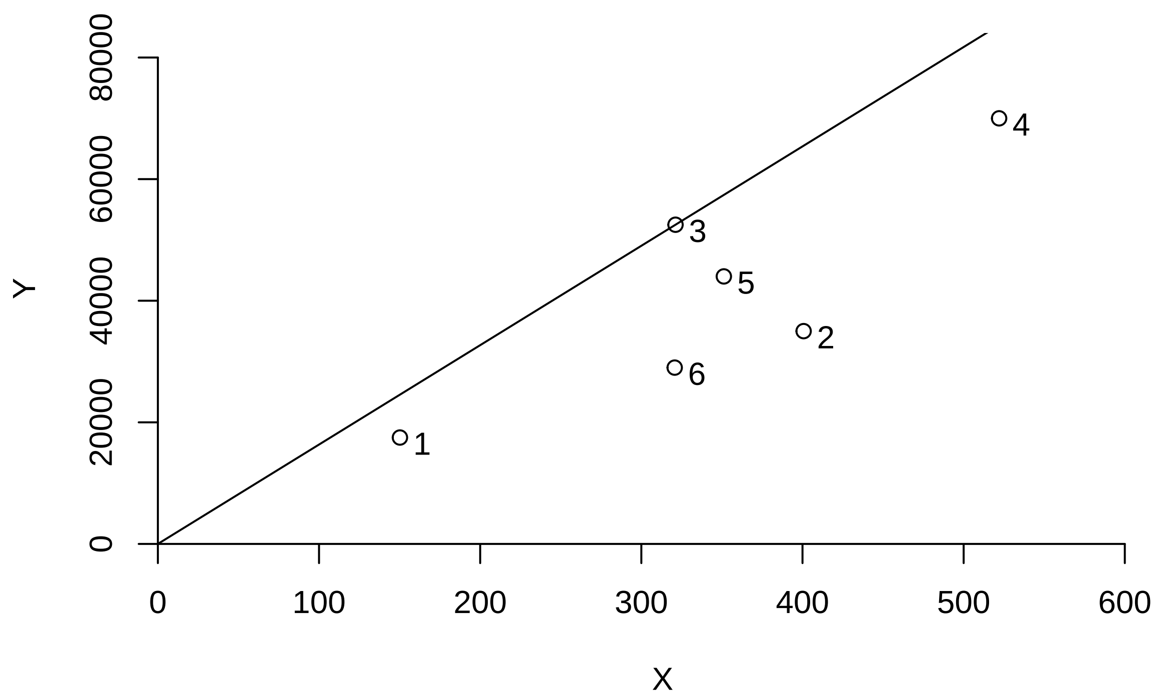
```
##             L1         L2 L3        L4
## [1,] 1.0000000 0.00000000  0 0.0000000
## [2,] 0.0000000 1.00000000  0 0.0000000
## [3,] 0.0000000 0.00000000  1 0.0000000
## [4,] 0.0000000 0.00000000  0 1.0000000
## [5,] 0.2000000 0.08048142  0 0.5383307
## [6,] 0.3428571 0.39499264  0 0.1310751
```

```
#Add the Efficiency & Lambda values in the table
report2 <- cbind(data, analysis_crs$eff, analysis_crs$lambda)
#Name the columns of the table
colnames(report2)<- c(names(Names_DMU),names(inputs), names(outputs),'Efficiency','Lambda1','Lambda2','I
#Sow the table
report2
```

```
##          DMU Staff Hours per Day Supplies per Day Reimbursed Patient-Days
## 1 Facility 1                 150              0.2                   14000
## 2 Facility 2                 400              0.7                   14000
## 3 Facility 3                 320              1.2                   42000
## 4 Facility 4                 520              2.0                   28000
```

```
## 5 Facility 5                   350              1.2                   19000
## 6 Facility 6                   320              0.7                   14000
##   Privately Paid Patient-Days Efficiency   Lambda1    Lambda2 Lambda3   Lambda4
## 1                        3500  1.0000000 1.0000000 0.00000000       0 0.0000000
## 2                       21000  1.0000000 0.0000000 1.00000000       0 0.0000000
## 3                       10500  1.0000000 0.0000000 0.00000000       1 0.0000000
## 4                       42000  1.0000000 0.0000000 0.00000000       0 1.0000000
## 5                       25000  0.9774987 0.2000000 0.08048142       0 0.5383307
## 6                       15000  0.8674521 0.3428571 0.39499264       0 0.1310751
##   Lambda5 Lambda6
## 1       0       0
## 2       0       0
## 3       0       0
## 4       0       0
## 5       0       0
## 6       0       0
```

```
#plot the graph for CRS Assumption
dea.plot(x,y,RTS="CRS",txt = rownames(report2))
```



The results indicate that DMUs 1, 2, 3 and 4 are efficient. DMU(5) is only 97.7% efficient, and DMU(6) is 86.7% efficient.

We use the option of VRS, Variable returns to scale, convexity and free disposability

```r
#DEA input or output efficiency measures, peers, lambdas and slacks
analysis_vrs <- dea(x,y,RTS = "VRS")
#Show the Efficiency
analysis_vrs
```

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

```r
#Show the list of objects calculated
str(analysis_vrs)
```

```
## List of 12
##  $ eff        : num [1:6] 1 1 1 1 1 ...
##  $ lambda     : num [1:6, 1:6] 1 0 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "L1" "L2" "L3" "L4" ...
##  $ objval     : num [1:6] 1 1 1 1 1 ...
##  $ RTS        : chr "vrs"
##  $ primal     : NULL
##  $ dual       : NULL
##  $ ux         : NULL
##  $ vy         : NULL
##  $ gamma      :function (x)
##  $ ORIENTATION: chr "in"
##  $ TRANSPOSE  : logi FALSE
##  $ param      : NULL
##  - attr(*, "class")= chr "Farrell"
```

```r
#Show the peers
peers(analysis_vrs)
```

```
##      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
```

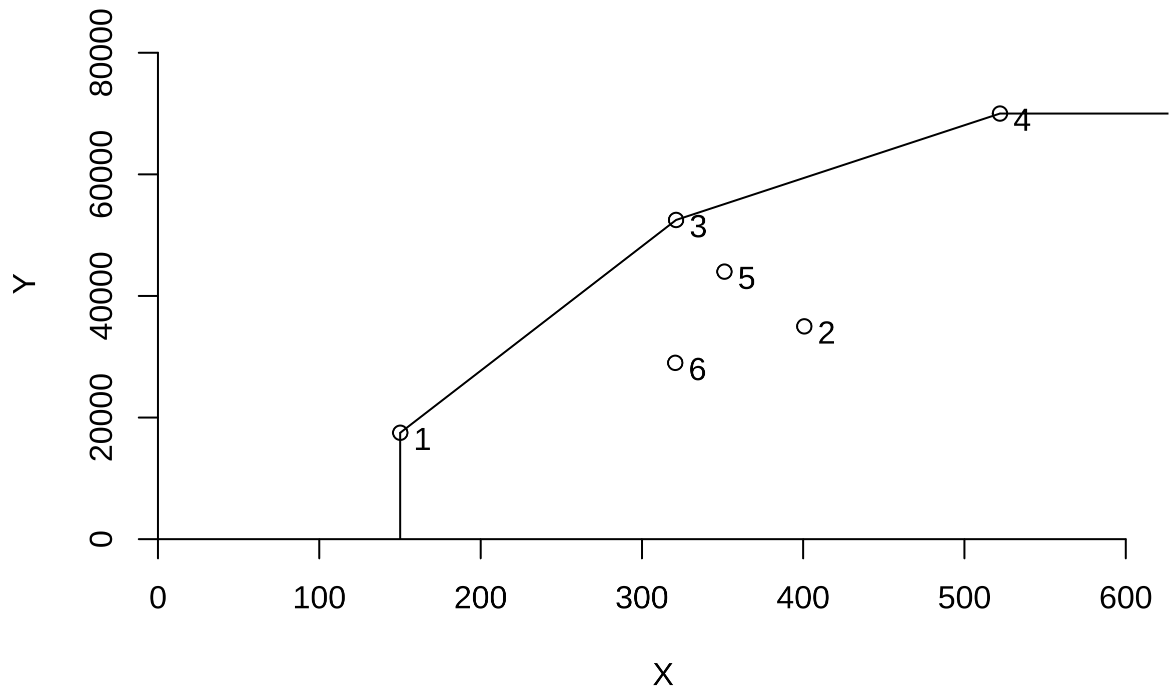
```r
#Show the lambda
lambda(analysis_vrs)
```

```
##             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
#Add the Efficiency & Lambda values in the table
report3 <- cbind(data, analysis_vrs$eff, analysis_vrs$lambda)
#Name the columns of the table
colnames(report3)<- c(names(Names_DMU),names(inputs), names(outputs),'Efficiency','Lambda1','Lambda2','L
#Show the table
report3
```

```
##          DMU Staff Hours per Day Supplies per Day Reimbursed Patient-Days
## 1 Facility 1                 150              0.2                   14000
## 2 Facility 2                 400              0.7                   14000
## 3 Facility 3                 320              1.2                   42000
## 4 Facility 4                 520              2.0                   28000
## 5 Facility 5                 350              1.2                   19000
## 6 Facility 6                 320              0.7                   14000
##   Privately Paid Patient-Days Efficiency   Lambda1   Lambda2 Lambda3 Lambda4
## 1                        3500  1.0000000 1.0000000 0.0000000       0       0
## 2                       21000  1.0000000 0.0000000 1.0000000       0       0
## 3                       10500  1.0000000 0.0000000 0.0000000       1       0
## 4                       42000  1.0000000 0.0000000 0.0000000       0       1
## 5                       25000  1.0000000 0.0000000 0.0000000       0       0
## 6                       15000  0.8963283 0.4014399 0.3422606       0       0
##     Lambda5 Lambda6
## 1 0.0000000       0
## 2 0.0000000       0
## 3 0.0000000       0
## 4 0.0000000       0
## 5 1.0000000       0
## 6 0.2562995       0
```

```r
#plot the graph for VRS Assumption
dea.plot(x,y,RTS="VRS",txt = rownames(report3))
```



The results indicate that DMUs 1, 2, 3, 4 and 5 are efficient. DMU(6) is only 89.6% efficient.

We use the option of IRS, Increasing returns to scale, convexity and free disposability

```
#DEA input or output efficiency measures, peers, lambdas and slacks
analysis_irs <- dea(x,y,RTS = "IRS")
#Show the Efficiency
analysis_irs
```

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

```
#Show the list of objects calculated
str(analysis_irs)
```

```
## List of 12
##  $ eff        : num [1:6] 1 1 1 1 1 ...
##  $ lambda     : num [1:6, 1:6] 1 0 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "L1" "L2" "L3" "L4" ...
```

```
##  $ objval     : num [1:6] 1 1 1 1 1 ...
##  $ RTS        : chr "irs"
##  $ primal     : NULL
##  $ dual       : NULL
##  $ ux         : NULL
##  $ vy         : NULL
##  $ gamma      :function (x)
##  $ ORIENTATION: chr "in"
##  $ TRANSPOSE  : logi FALSE
##  $ param      : NULL
##  - attr(*, "class")= chr "Farrell"
```

```
#Show the peers
peers(analysis_irs)
```

```
##      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
```

```
#Show the lambda
lambda(analysis_irs)
```

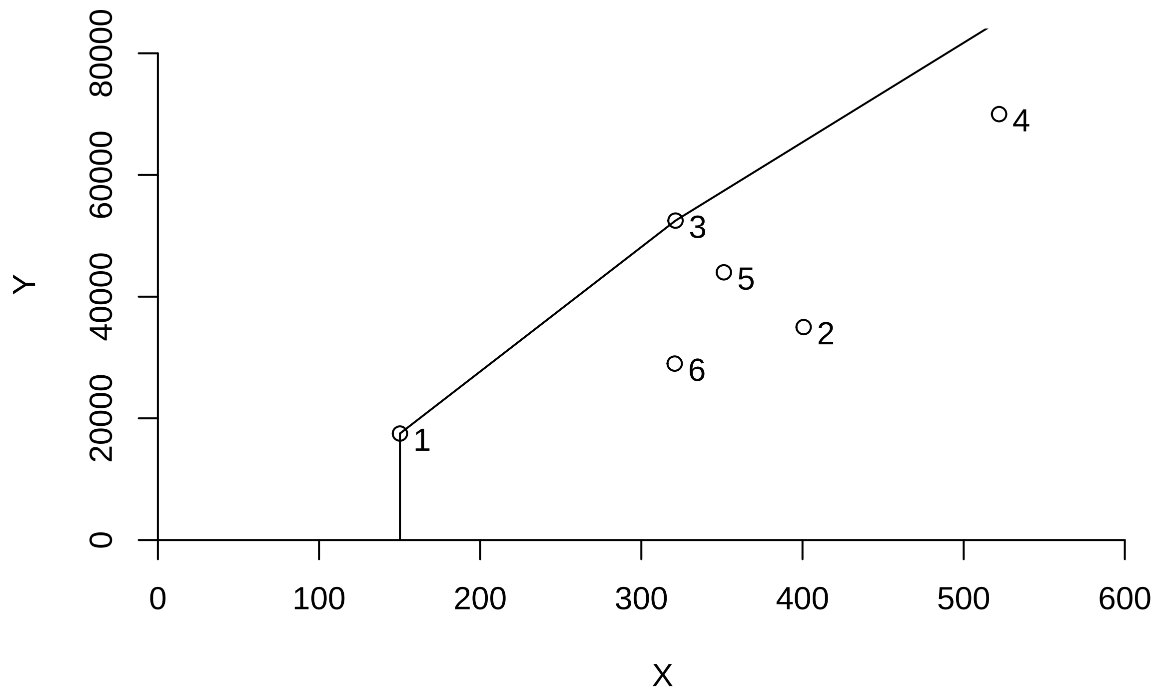
```
##             L1        L2 L3 L4        L5
## [1,] 1.0000000 0.0000000  0  0 0.0000000
## [2,] 0.0000000 1.0000000  0  0 0.0000000
## [3,] 0.0000000 0.0000000  1  0 0.0000000
## [4,] 0.0000000 0.0000000  0  1 0.0000000
## [5,] 0.0000000 0.0000000  0  0 1.0000000
## [6,] 0.4014399 0.3422606  0  0 0.2562995
```

```
#Add the Efficiency & Lambda values in the table
report4 <- cbind(data, analysis_irs$eff, analysis_irs$lambda)
#Name the columns of the table
colnames(report4)<- c(names(Names_DMU),names(inputs), names(outputs),'Efficiency','Lambda1','Lambda2','I
#Show the table
report4
```

```
##          DMU Staff Hours per Day Supplies per Day Reimbursed Patient-Days
## 1 Facility 1                 150              0.2                   14000
## 2 Facility 2                 400              0.7                   14000
## 3 Facility 3                 320              1.2                   42000
## 4 Facility 4                 520              2.0                   28000
## 5 Facility 5                 350              1.2                   19000
## 6 Facility 6                 320              0.7                   14000
##   Privately Paid Patient-Days Efficiency   Lambda1   Lambda2 Lambda3 Lambda4
## 1                        3500  1.0000000 1.0000000 0.0000000       0       0
## 2                       21000  1.0000000 0.0000000 1.0000000       0       0
## 3                       10500  1.0000000 0.0000000 0.0000000       1       0
```

```
## 4                          42000   1.0000000 0.0000000 0.0000000       0       1
## 5                          25000   1.0000000 0.0000000 0.0000000       0       0
## 6                          15000   0.8963283 0.4014399 0.3422606       0       0
##      Lambda5 Lambda6
## 1 0.0000000       0
## 2 0.0000000       0
## 3 0.0000000       0
## 4 0.0000000       0
## 5 1.0000000       0
## 6 0.2562995       0
```

```
#plot the graph for IRS Assumption
dea.plot(x,y,RTS="IRS",txt = rownames(report4))
```



The results indicate that DMUs 1, 2, 3, 4 and 5 are efficient. DMU(6) is only 89.6% efficient.

We use the option of DRS, Decreasing returns to scale, convexity, down-scaling and free disposability

```
#DEA input or output efficiency measures, peers, lambdas and slacks
analysis_drs <- dea(x,y,RTS = "DRS")
```

```r
#Show the Efficiency
analysis_drs
```

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

```r
#Show the list of objects calculated
str(analysis_drs)
```

```
## List of 12
##  $ eff        : num [1:6] 1 1 1 1 0.977 ...
##  $ lambda     : num [1:6, 1:6] 1 0 0 0 0.2 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "L1" "L2" "L3" "L4" ...
##  $ objval     : num [1:6] 1 1 1 1 0.977 ...
##  $ RTS        : chr "drs"
##  $ primal     : NULL
##  $ dual       : NULL
##  $ ux         : NULL
##  $ vy         : NULL
##  $ gamma      :function (x)
##  $ ORIENTATION: chr "in"
##  $ TRANSPOSE  : logi FALSE
##  $ param      : NULL
##  - attr(*, "class")= chr "Farrell"
```

```r
#Show the peers
peers(analysis_drs)
```

```
##      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
#Show the lambda
lambda(analysis_drs)
```

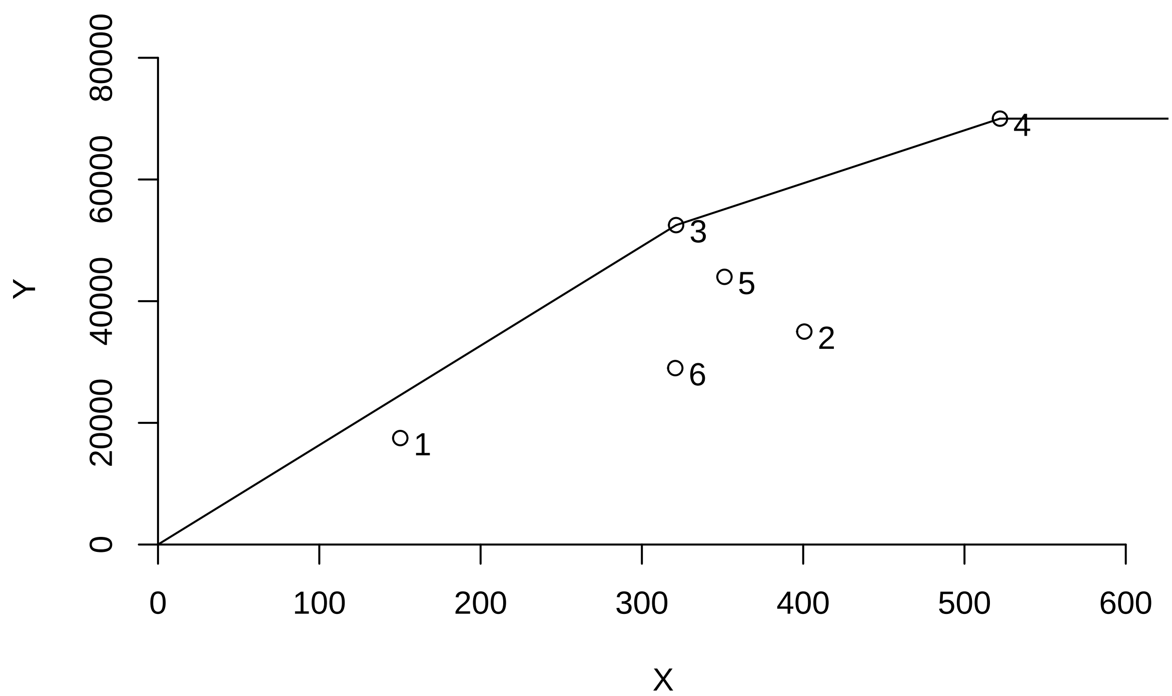
```
##             L1         L2 L3        L4
## [1,] 1.0000000 0.00000000  0 0.0000000
## [2,] 0.0000000 1.00000000  0 0.0000000
## [3,] 0.0000000 0.00000000  1 0.0000000
## [4,] 0.0000000 0.00000000  0 1.0000000
## [5,] 0.2000000 0.08048142  0 0.5383307
## [6,] 0.3428571 0.39499264  0 0.1310751
```

```r
#Add the Efficiency, Peers & Lambda values in the table
report5 <- cbind(data, analysis_drs$eff, analysis_drs$lambda)
#Name the columns of the table
```

```r
colnames(report5)<- c(names(Names_DMU),names(inputs), names(outputs),'Efficiency','Lambda1','Lambda2','
#Show the table
report5
```

```
##          DMU Staff Hours per Day Supplies per Day Reimbursed Patient-Days
## 1 Facility 1                 150              0.2                   14000
## 2 Facility 2                 400              0.7                   14000
## 3 Facility 3                 320              1.2                   42000
## 4 Facility 4                 520              2.0                   28000
## 5 Facility 5                 350              1.2                   19000
## 6 Facility 6                 320              0.7                   14000
##   Privately Paid Patient-Days Efficiency   Lambda1    Lambda2 Lambda3   Lambda4
## 1                        3500  1.0000000 1.0000000 0.00000000       0 0.0000000
## 2                       21000  1.0000000 0.0000000 1.00000000       0 0.0000000
## 3                       10500  1.0000000 0.0000000 0.00000000       1 0.0000000
## 4                       42000  1.0000000 0.0000000 0.00000000       0 1.0000000
## 5                       25000  0.9774987 0.2000000 0.08048142       0 0.5383307
## 6                       15000  0.8674521 0.3428571 0.39499264       0 0.1310751
##   Lambda5 Lambda6
## 1       0       0
## 2       0       0
## 3       0       0
## 4       0       0
## 5       0       0
## 6       0       0
```

```r
#plot the graph for IRS Assumption
dea.plot(x,y,RTS="DRS",txt = rownames(report5))
```



The results indicate that DMUs 1, 2, 3 and 4 are efficient. DMU(5) is only 97.7% efficient, and DMU(6) is 86.7% efficient.

We use the option of FRH, Additivity (scaling up and down, but only with integers), and free disposability

```
#DEA input or output efficiency measures, peers, lambdas and slacks
analysis_frh <- dea(x,y,RTS = "ADD")
#Show the Efficiency
analysis_frh
```

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

```
#Show the list of objects calculated
str(analysis_frh)
```

```
## List of 12
##  $ eff        : num [1:6] 1 1 1 1 1 1
##  $ lambda     : num [1:6, 1:6] 1 0 0 0 0 0 0 1 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:6] "L1" "L2" "L3" "L4" ...
```

```
##  $ objval     : num [1:6] 1 1 1 1 1 1
##  $ RTS        : chr "add"
##  $ primal     : NULL
##  $ dual       : NULL
##  $ ux         : NULL
##  $ vy         : NULL
##  $ gamma      :function (x)
##  $ ORIENTATION: chr "in"
##  $ TRANSPOSE  : logi FALSE
##  $ param      : NULL
##  - attr(*, "class")= chr "Farrell"
```

```
#Show the peers
peers(analysis_frh)
```

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

```
#Show the lambda
lambda(analysis_frh)
```

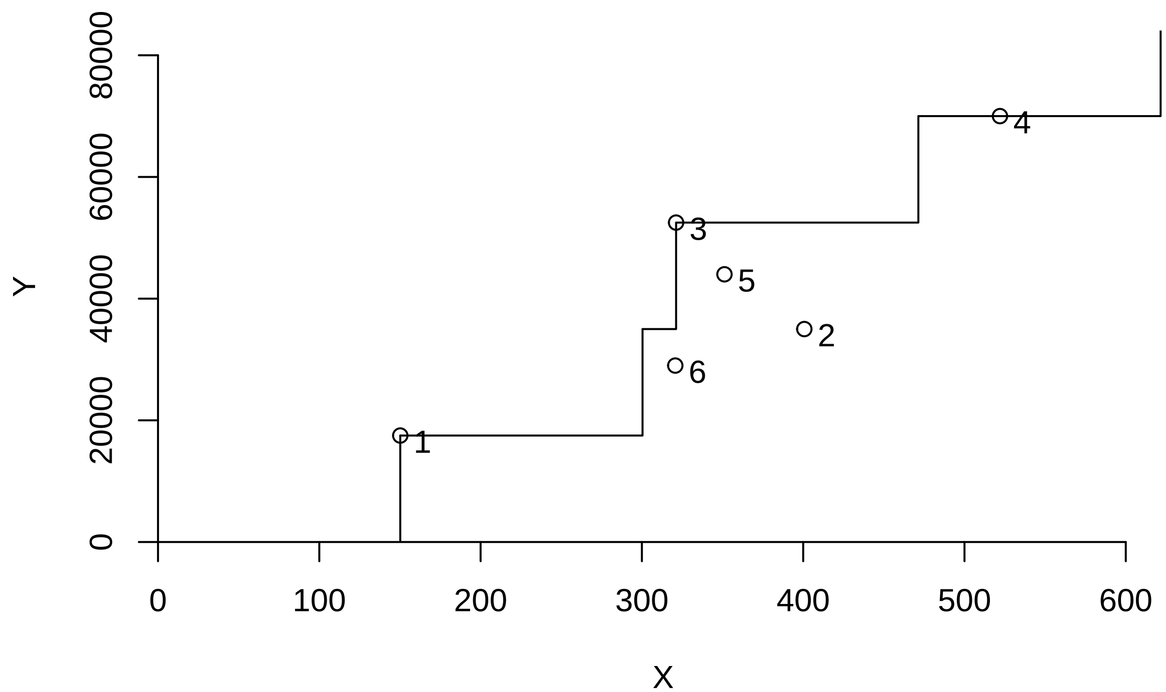
```
##      L1 L2 L3 L4 L5 L6
## [1,]  1  0  0  0  0  0
## [2,]  0  1  0  0  0  0
## [3,]  0  0  1  0  0  0
## [4,]  0  0  0  1  0  0
## [5,]  0  0  0  0  1  0
## [6,]  0  0  0  0  0  1
```

```
#Add the Efficiency, Peers & Lambda values in the table
report6 <- cbind(data, analysis_frh$eff, analysis_frh$lambda)
#Name the columns of the table
colnames(report6)<- c(names(Names_DMU),names(inputs), names(outputs),'Efficiency','Lambda1','Lambda2','L
#Sow the table
report6
```

```
##          DMU Staff Hours per Day Supplies per Day Reimbursed Patient-Days
## 1 Facility 1                 150              0.2                   14000
## 2 Facility 2                 400              0.7                   14000
## 3 Facility 3                 320              1.2                   42000
## 4 Facility 4                 520              2.0                   28000
## 5 Facility 5                 350              1.2                   19000
## 6 Facility 6                 320              0.7                   14000
##   Privately Paid Patient-Days Efficiency Lambda1 Lambda2 Lambda3 Lambda4
## 1                        3500          1       1       0       0       0
## 2                       21000          1       0       1       0       0
## 3                       10500          1       0       0       1       0
```

```
## 4                         42000          1        0        0        0        1
## 5                         25000          1        0        0        0        0
## 6                         15000          1        0        0        0        0
##   Lambda5 Lambda6
## 1       0       0
## 2       0       0
## 3       0       0
## 4       0       0
## 5       1       0
## 6       0       1
```

```
#plot the graph for FDH Assumption
dea.plot(x,y,RTS="ADD",txt = rownames(report5))
```



The results indicate that DMUs 1, 2, 3 , 4, 5 and 6 all are efficient.

Compare and Contrast the above Results

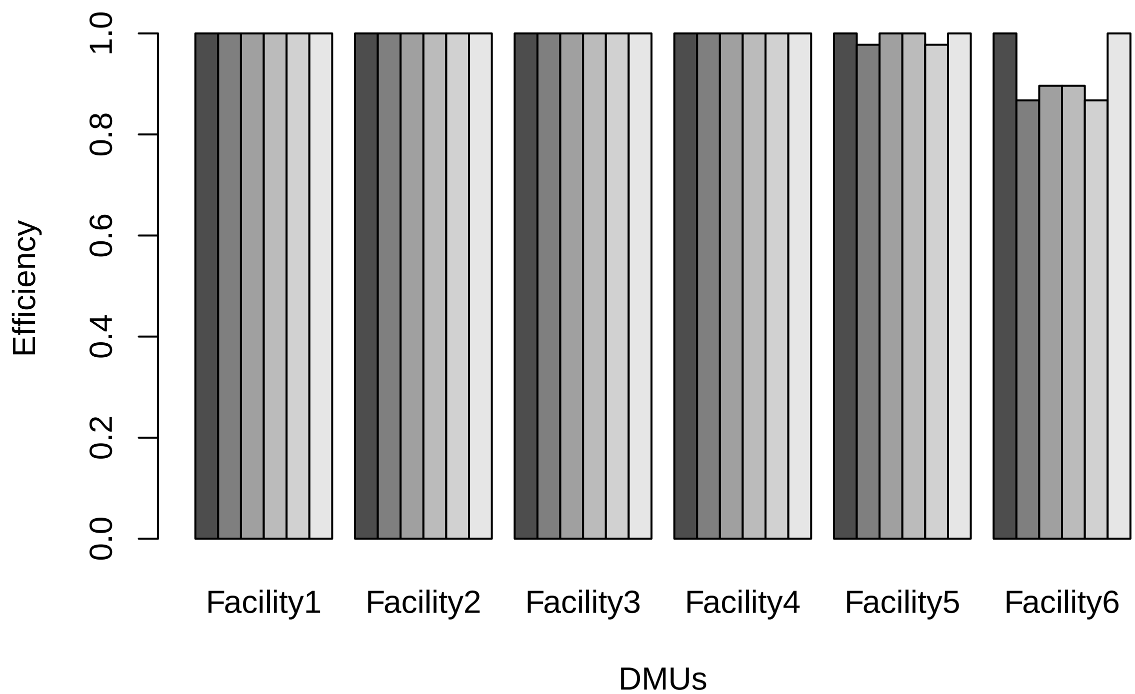
```
#Add the Efficiency of all the DMUs for all the Assumptions in a table
EfficiencyReport <- cbind(data[,1],analysis_fdh$eff,analysis_crs$eff,analysis_vrs$eff,analysis_irs$eff,a
#Name the columns of the table
colnames(EfficiencyReport) <- c(names(Names_DMU),'FDH Efficiency','CRS Efficiency','VRS Efficiency','IRS
#Show the Efficiency table
EfficiencyReport
```

```
##          DMU FDH Efficiency CRS Efficiency VRS Efficiency IRS Efficiency
## 1 Facility 1              1      1.0000000      1.0000000      1.0000000
## 2 Facility 2              1      1.0000000      1.0000000      1.0000000
## 3 Facility 3              1      1.0000000      1.0000000      1.0000000
## 4 Facility 4              1      1.0000000      1.0000000      1.0000000
## 5 Facility 5              1      0.9774987      1.0000000      1.0000000
## 6 Facility 6              1      0.8674521      0.8963283      0.8963283
##   DRS Efficiency FRH Efficiency
## 1      1.0000000              1
## 2      1.0000000              1
## 3      1.0000000              1
## 4      1.0000000              1
## 5      0.9774987              1
## 6      0.8674521              1
```

```
#Let's compare the Efficiency of all the DMUs for all the assumptions using a plot
#Concatenate the Efficiency
spreadsheet <- cbind(analysis_fdh$eff,analysis_crs$eff,analysis_vrs$eff,analysis_irs$eff,analysis_drs$ef
#Name the rows
rownames(spreadsheet) <- c("Facility1","Facility2","Facility3","Facility4","Facility5","Facility6")
#Name the columns
colnames(spreadsheet) <- c ("FDH","CRS","VRS","IRS","DRS","FRH")
#See the result
spreadsheet
```

```
##           FDH       CRS       VRS       IRS       DRS FRH
## Facility1   1 1.0000000 1.0000000 1.0000000 1.0000000   1
## Facility2   1 1.0000000 1.0000000 1.0000000 1.0000000   1
## Facility3   1 1.0000000 1.0000000 1.0000000 1.0000000   1
## Facility4   1 1.0000000 1.0000000 1.0000000 1.0000000   1
## Facility5   1 0.9774987 1.0000000 1.0000000 0.9774987   1
## Facility6   1 0.8674521 0.8963283 0.8963283 0.8674521   1
```

```
#plot the graph
barplot(t(spreadsheet),col=gray.colors(6),xlab = "DMUs", ylab="Efficiency",beside=TRUE)
```



Hence,we can conclude that

1) Facility 1 is fully efficient for all the assumptions.
2) Facility 2 is fully efficient for all the assumptions.
3) Facility 3 is fully efficient for all the assumptions.
4) Facility 4 is fully efficient for all the assumptions.
5) Facility 5 is fully efficient for FDH, VRS, IRS and FRH assumptions. For assumptions DRS and CRS, it is 97.7% efficient.
6) Facility 6 is fully efficient for FDH and FRS assumptions. For CRS and DRS assumptions, it is 86.7% efficient. For IRS and VRS assumptions, it is 89.6% efficient.

Question 2:

The Research and Development Division of the Emax Corporation has developed three new products. A decision now needs to be made on which mix of these products should be produced. Management wants primary consideration given to three factors: total profit, stability in the workforce, and achieving an increase in the company's earnings next year from the $75 million achieved this year. In particular, using the units given in the following table, they want to

Maximize Z = P - 6C - 3D, where

P = total (discounted) profit over the life of the new products, C = change (in either direction) in the current level of employment, D = decrease (if any) in next year's earnings from the current year's level.

The amount of any increase in earnings does not enter into Z, because management is concerned primarily with just achieving some increase to keep the stockholders happy. (It has mixed feelings about a large increase that then would be difficult to surpass in subsequent years.)

1) Define y1+ and y1-, respectively, as the amount over (if any) and the amount under (if any) the employment level goal. Define y2+ and y2- in the same way for the goal regarding earnings next year. Define x1, x2, and x3 as the production rates of Products 1, 2, and 3, respectively. With these definitions, use the goal programming technique to express y1+, y1- , y2+ and y2- algebraically in terms of x1, x2, and x3. Also express P in terms of x1, x2, and x3.

2) Express management's objective function in terms of x1, x2, x3, y1+, y1- , y2+ and y2-.

3) Formulate and solve the linear programming model. What are your findings?

Solution: All of the Goals in this problem are of roughly comparabl importance. As a result, it is a goal programming approach that is not preemptive. The Emax corporation problem includes all three possible types of goals: an upper, one-sided goal (Total profit); a two-sided goal (Employment level); and a lower, one-sided goal (Earnings Next year). Letting the decision variables x1, x2, x3 be the production rates of products 1, 2, and 3, respectively, Total Ptofit (P) can be expressed in terms of x1, x2 and x3 as:

```
Maximize : 20x1+15x2+25x3
```

Also,Employment level and Earnings next year can be expressed as as follows :

```
6x1+4x2+5x3 =50

x1+7x2+5x3 >=75
```

Our goal is to maximize the profit using the constraints i.e. Employment level and Earnings next year,hence they can be written as :

```
Max z: 20x1+15x2+25x3

s.t.:  6x1+4x2+5x3 =50

       8x1+7x2+5x3 >=75
```

The overall objective mathematically can be expressed by introducing some auxiliary variables (extra variables that are helpful for formulating the model) y1 and y2, defined as follows:

```
y1=6x1+4x2+5x3-50    (Employment Level minus the target)

y2=8x1+7x2+5x3-75    (Earnings Next Year minus the Target)
```

Since each yi can be either positive or negative,we replace each one by the difference of two non negative variables:

```
y1=y1p - y1m,  where y1p , y1m >=0
y2=y2p - y2m,  where y2p , y2m >=0
```

y1p represents the penalty for employement level goal exceeding 50 and y1m is the penalty for employment level goal decreasing below 50.

Similarly, y2m represents the penalty for not reaching the next year earnings and y2p for exceeding the next year earnings.

Given these new auxiliary variables, the overall management's objective function can be expressed mathematically as (maximizing the profit and subtracting the penalties)

```
Max z: 20x1+15x2+25x3-6y1p+6y1m-3y2m;

 s.t.: 6x1+4x2+5x3-y1p+y1m =50

       8x1+7x2+5x3-y2p+y2m =75
```

Since there is no penalty for exceeding the earnings next year, so y2p should not appear in the objective function.

Lets formulate and solve the Linear programming model usnig lpSolveAPI.

Install lpSolveAPI package if not already installed

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

Now, load the library

```
library(lpSolveAPI)
```

#We have 7 decision variables, and 2 constraints.

```
lprec <- make.lp(2, 7)
```

Set the maximization objective function

```
set.objfn(lprec, c(20, 15, 25, -6, 6, 0, -3))
lp.control(lprec,sense='max')
```

```
## $anti.degen
## [1] "fixedvars" "stalling"
##
## $basis.crash
## [1] "none"
##
## $bb.depthlimit
## [1] -50
##
```

```
## $bb.floorfirst
## [1] "automatic"
##
## $bb.rule
## [1] "pseudononint" "greedy"       "dynamic"      "rcostfixing"
##
## $break.at.first
## [1] FALSE
##
## $break.at.value
## [1] 1e+30
##
## $epsilon
##      epsb       epsd      epsel     epsint epsperturb   epspivot
##     1e-10      1e-09      1e-12      1e-07      1e-05      2e-07
##
## $improve
## [1] "dualfeas" "thetagap"
##
## $infinite
## [1] 1e+30
##
## $maxpivot
## [1] 250
##
## $mip.gap
## absolute relative
##    1e-11    1e-11
##
## $negrange
## [1] -1e+06
##
## $obj.in.basis
## [1] TRUE
##
## $pivoting
## [1] "devex"    "adaptive"
##
## $presolve
## [1] "none"
##
## $scalelimit
## [1] 5
##
## $scaling
## [1] "geometric"   "equilibrate" "integers"
##
## $sense
## [1] "maximize"
##
## $simplextype
## [1] "dual"   "primal"
##
## $timeout
```

```
## [1] 0
##
## $verbose
## [1] "neutral"
```

Set values for the rows (set the Left hand side constraints)

```
set.row(lprec, 1, c(6, 4, 5, -1 , 1, 0, 0), indices = c(1, 2, 3, 4, 5, 6, 7))
set.row(lprec, 2, c(8, 7, 5,0,0,-1,1), indices = c(1, 2, 3, 4, 5, 6, 7))
```

Set the right hand side values

```
rhs <- c(50, 75)
set.rhs(lprec, rhs)
```

Set constraint type and set variable types and bound

```
set.constr.type(lprec, c("=", "="))
set.bounds(lprec, lower = rep(0, 7))
```

Name the decision variables (column) and constraints (rows)

```
lp.rownames <- c("EmploymentLevelGoal", "NextYearEarningsGoal")
lp.colnames <- c("x1", "x2", "x3","y1p", "y1m", "y2p","y2m")
dimnames(lprec) <- list(lp.rownames, lp.colnames)
```

View the linear program object to make sure it's correct

```
lprec
```

```
## Model name:
##                         x1    x2    x3   y1p   y1m   y2p   y2m
## Maximize                20    15    25    -6     6     0    -3
## EmploymentLevelGoal      6     4     5    -1     1     0     0  =  50
## NextYearEarningsGoal     8     7     5     0     0    -1     1  =  75
## Kind                   Std   Std   Std   Std   Std   Std   Std
## Type                  Real  Real  Real  Real  Real  Real  Real
## Upper                  Inf   Inf   Inf   Inf   Inf   Inf   Inf
## Lower                    0     0     0     0     0     0     0
```

Save this into a file

```r
write.lp(lprec, filename = "emax.lp", type = "lp")
```

##Ssolve the model

```r
solve(lprec)
```

```
## [1] 0
```

Show the value of objective function, variables, constraints and slack

```r
get.objective(lprec)
```

```
## [1] 225
```

```r
get.variables(lprec)
```

```
## [1]  0  0 15 25  0  0  0
```

```r
get.constraints(lprec)
```

```
## [1] 50 75
```

```r
get.constraints(lprec) - rhs
```

```
## [1] 1.421085e-14 0.000000e+00
```

Also,We can now read the lp formulation using an lp file and solve it. I am using the same lp file which I have saved above.

Read from file and solve it

```r
x <- read.lp("emax.lp")     # create an lp object x
x                           # display x
```

```
## Model name:
##                           x1    x2    x3   y1p   y1m   y2m   y2p
## Maximize                  20    15    25    -6     6    -3     0
## EmploymentLevelGoal        6     4     5    -1     1     0     0  =  50
## NextYearEarningsGoal       8     7     5     0     0     1    -1  =  75
## Kind                     Std   Std   Std   Std   Std   Std   Std
## Type                    Real  Real  Real  Real  Real  Real  Real
## Upper                    Inf   Inf   Inf   Inf   Inf   Inf   Inf
## Lower                      0     0     0     0     0     0     0
```

```r
solve(x)                    # Solution
```

```
## [1] 0
```

```r
get.objective(x)            # get objective value
```

```
## [1] 225
```

```r
get.variables(x)            # get values of decision variables
```

```
## [1]  0  0 15 25  0  0  0
```

```r
get.constraints(x)          # get constraints
```

```
## [1] 50 75
```

Applying the simplex method to this formulation yields an optimal solution x1 = 0, x2 = 0, x3 = 15, y1p = 25, y1m = 0, y2p = 0, y2m = 0. Therefore, y1 = 25 and y2 = 0, so the second goal of Next years Earning is fully satisfied, but the employment level goal of 50 is exceeded by 25 (2500 Employees). So the resulting penalty for deviating from the goal is 150. And so the value for the objective function is 225.