

Assignment 5

Jayasri

11/2/2021

DEA Module

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. A summary of performance data is shown in the table below.

DMU	Staff Hours per Day	Supplies per Day	Reimbursed Patient-Days	Privately Paid Patient-Days
Facility 1	150	0.2	14,000	3,500
Facility 2	400	0.7	14,000	21,000
Facility 3	320	1.2	42,000	10,500
Facility 4	520	2.0	28,000	42,000
Facility 5	350	1.2	19,000	25,000
Facility 6	320	0.7	14,000	15,000

QUESTIONS:

- 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.

SOLUTION:

```
#Loading the required libraries
#install.packages("Benchmarking")
library(Benchmarking)
library(cowplot)
```

Compute the Formulation

As there are Many DMUs we would be using benchmarking Libraries for DEA.

Now, We will read the data as input and output as vectors. Our problem has facilities as 6 DMUs.

- INPUT : staff hours and supplies per day.
- OUTPUT : Reimbursed Patient-Days and Privately paid patient days.

```
input<- matrix(c(150,400,320,520,350,320,0.2,0.7,1.2,2.0,1.2,0.7),ncol = 2)
Output<- matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,15000),ncol=2)
colnames(input)<-c('staff Hours(Per Day)','Supplies(Per Day)')
colnames(Output)<-c('Reimbursed Patient','Privately Paid Patient')
rownames(input)<- c('Facility1','Facility2','Facility3','Facility4','Facility5','Facility6')
rownames(Output)<- c('Facility1','Facility2','Facility3','Facility4','Facility5','Facility6')
input
```

##	staff Hours(Per Day)	Supplies(Per Day)
## Facility1	150	0.2
## Facility2	400	0.7
## Facility3	320	1.2
## Facility4	520	2.0
## Facility5	350	1.2
## Facility6	320	0.7

Output

##	Reimbursed Patient	Privately Paid Patient
## Facility1	14000	3500
## Facility2	14000	21000
## Facility3	42000	10500
## Facility4	28000	42000
## Facility5	19000	25000
## Facility6	14000	15000

We now perform the DEA analysis.

DEA analysis using FDH

Now, we are going to formulate and compute the DEA analysis using FDH.

The free disposability assumption stipulates that we can freely discard unnecessary inputs and unwanted outputs

```
#Provide the input and output
Analysis_fdh <- dea(input,Output,RTS = "fdh")

# Efficiency of the DEA with FDH Assumption
Efficiency_fdh <- as.data.frame(Analysis_fdh$eff)
colnames(Efficiency_fdh) <- c("Efficiency_FDH")
```

```

# Identify the peers
peers_fdh <- as.data.frame(Analysis_fdh$peers)
colnames(peers_fdh) <- c("Peers_FDH")

# Identify the relative weights using Lambda Function
lambda_fdh <- as.data.frame(Analysis_fdh$lambda)
colnames(lambda_fdh) <- c("FDH_Lambda1","FDH_Lambda2","FDH_Lambda3","FDH_Lambda4","FDH_Lambda5","FDH_Lambda6")

#Create a tabular data with peers, lambda and efficiency
dea_fdh <- cbind(Efficiency_fdh,peers_fdh,lambda_fdh)
rownames(dea_fdh)<- c('F1','F2','F3','F4','F5','F6')

#Show the summary of the table
dea_fdh

```

```

##      Efficiency_FDH Peers_FDH FDH_Lambda1 FDH_Lambda2 FDH_Lambda3 FDH_Lambda4
## F1                1         1           1           0           0           0
## F2                1         2           0           1           0           0
## F3                1         3           0           0           1           0
## F4                1         4           0           0           0           1
## F5                1         5           0           0           0           0
## F6                1         6           0           0           0           0
##      FDH_Lambda5 FDH_Lambda6
## F1              0          0
## F2              0          0
## F3              0          0
## F4              0          0
## F5              1          0
## F6              0          1

```

As we are familiar that peers are the way we could identify the inefficient DMUs and Lambda values are the relative weights assigned from the peer units when solving DEA model.

Observations from FDH:

The above summary table of DEA Analysis using FDH represents that every facility is working to the fullest and with full efficiency. Each peer is assigned to a unit which implies $\text{Lambda} = 1$ and $\text{Efficiency} = 1$.

DEA analysis using CRS

Now, we are going to formulate and compute the DEA analysis using CRS

Constant returns to scale (CRS) is one of the scaling assumptions. This assumption stipulates that if any possible production combination can arbitrarily be scaled up or down.

```

#Provide the input and output
Analysis_CRS <- dea(input,Output,RTS = "crs")

# Efficiency of the DEA with CRS Assumption
Efficiency_CRS <- as.data.frame(Analysis_CRS$eff)
colnames(Efficiency_CRS) <- c("Efficiency_CRS")

```

```

# Identify the peers
peers_CRS <- peers(Analysis_CRS)
colnames(peers_CRS) <- c("Peer1_CRS", "Peer2_CRS", "Peer3_CRS")

# Identify the relative weights using Lambda Function
lambda_CRS <- lambda(Analysis_CRS)
colnames(lambda_CRS) <- c("CRS_Lambda1", "CRS_Lambda2", "CRS_Lambda3", "CRS_Lambda4")

dea_CRS <- cbind(Efficiency_CRS, peers_CRS, lambda_CRS)

rownames(dea_CRS) <- c('F1', 'F2', 'F3', 'F4', 'F5', 'F6')

# Summary of the table
dea_CRS

```

##	Efficiency_CRS	Peer1_CRS	Peer2_CRS	Peer3_CRS	CRS_Lambda1	CRS_Lambda2
## F1	1.0000000	1	NA	NA	1.0000000	0.0000000
## F2	1.0000000	2	NA	NA	0.0000000	1.0000000
## F3	1.0000000	3	NA	NA	0.0000000	0.0000000
## F4	1.0000000	4	NA	NA	0.0000000	0.0000000
## F5	0.9774987	1	2	4	0.2000000	0.08048142
## F6	0.8674521	1	2	4	0.3428571	0.39499264

##	CRS_Lambda3	CRS_Lambda4
## F1	0	0.0000000
## F2	0	0.0000000
## F3	1	0.0000000
## F4	0	1.0000000
## F5	0	0.5383307
## F6	0	0.1310751

Observations from CRS:

When we examine the above summary table of CRS assumption, it is clearly evident from the peers and lambda values that the facilities 1,2,3 and 4 are using 100% efficiency. Whereas facilities 5 and 6, need parts of 1,2 and 4 as peers and its corresponding lambdas shown above in the table, which implies that these 2 facilities have to improve its efficiency as they are 97.74% and 86.74% respectively.

DEA analysis using VRS

Now, we are going to formulate and compute the DEA analysis using VRS.

VRS is one of the scaling assumptions. This Assumptions represents that variable return to scale

```

Analysis_VRS <- dea(input, Output, RTS = "vrs")

# Efficiency of the DEA with FDH Assumption
Efficiency_VRS <- as.data.frame(Analysis_VRS$eff)
colnames(Efficiency_VRS) <- c("Efficiency_VRS")

# Identify the peers
peers_VRS <- peers(Analysis_VRS)
colnames(peers_VRS) <- c("Peer1_VRS", "Peer2_VRS", "Peer3_VRS")

```

```

# Identify the relative weights using Lambda Function
lambda_VRS <- lambda(Analysis_VRS)
colnames(lambda_VRS) <- c("VRS_Lambda1","VRS_Lambda2","VRS_Lambda3","VRS_Lambda4","VRS_Lambda5")

dea_VRS <- cbind(Efficiency_VRS,peers_VRS,lambda_VRS)

rownames(dea_VRS)<- c('F1','F2','F3','F4','F5','F6')

#Summary table
dea_VRS

```

```

##      Efficiency_VRS Peer1_VRS Peer2_VRS Peer3_VRS VRS_Lambda1 VRS_Lambda2
## F1      1.0000000      1      NA      NA      1.0000000      0.0000000
## F2      1.0000000      2      NA      NA      0.0000000      1.0000000
## F3      1.0000000      3      NA      NA      0.0000000      0.0000000
## F4      1.0000000      4      NA      NA      0.0000000      0.0000000
## F5      1.0000000      5      NA      NA      0.0000000      0.0000000
## F6      0.8963283      1      2      5      0.4014399      0.3422606
##      VRS_Lambda3 VRS_Lambda4 VRS_Lambda5
## F1      0      0      0.0000000
## F2      0      0      0.0000000
## F3      1      0      0.0000000
## F4      0      1      0.0000000
## F5      0      0      1.0000000
## F6      0      0      0.2562995

```

Observations from VRS:

When we observe the above summary tables of DEA analysis using VRS assumption, it indicates that facilities 1,2,3,4 and 5 are fully efficient whereas facility 6 has its efficiency of 89.63% and need to use the parts of facilities 1,2 and 5 as peers.

DEA analysis with IRS

Now, we are going to formulate and compute the DEA analysis using IRS.

IRS is one of the scaling assumptions It represents that for possible production process we can arbitrarily increase the scale of the operation.

```

Analysis_IRS <- dea(input,Output,RTS = "irs")

# Efficiency of the DEA with FDH Assumption
Efficiency_IRS <- as.data.frame(Analysis_IRS$eff)
colnames(Efficiency_IRS) <- c("Efficiency_IRS")

# Identify the peers
peers_IRS <- peers(Analysis_IRS)
colnames(peers_IRS) <- c("Peer1_IRS","Peer2_IRS","Peer3_IRS")

# Identify the relative weights using Lambda Function
lambda_IRS <- lambda(Analysis_IRS)
colnames(lambda_IRS) <- c("IRS_Lambda1","IRS_Lambda2","IRS_Lambda3","IRS_Lambda4","IRS_Lambda5")

```

```
dea_IRS <- cbind(Efficiency_IRS,peers_IRS,lambda_IRS)
```

```
rownames(dea_IRS)<- c('F1','F2','F3','F4','F5','F6')
```

```
#Summary table
```

```
dea_IRS
```

```
##      Efficiency_IRS Peer1_IRS Peer2_IRS Peer3_IRS IRS_Lambda1 IRS_Lambda2
## F1      1.0000000      1      NA      NA      1.0000000      0.0000000
## F2      1.0000000      2      NA      NA      0.0000000      1.0000000
## F3      1.0000000      3      NA      NA      0.0000000      0.0000000
## F4      1.0000000      4      NA      NA      0.0000000      0.0000000
## F5      1.0000000      5      NA      NA      0.0000000      0.0000000
## F6      0.8963283      1      2      5      0.4014399      0.3422606
##      IRS_Lambda3 IRS_Lambda4 IRS_Lambda5
## F1      0      0      0.0000000
## F2      0      0      0.0000000
## F3      1      0      0.0000000
## F4      0      1      0.0000000
## F5      0      0      1.0000000
## F6      0      0      0.2562995
```

Observations from IRS:

This DEA Analysis of IRS performs same as VRS by utilizing the facilities 1,2,3,4 and 5 to its fullest ie, efficiency of 100% whereas the facility 6 needs to improve its efficiency, which is 89.63% while parts of facilities 1,2 and 5 serving as the peers.

DEA analysis with DRS

Now, we are going to formulate and compute the DEA analysis using DRS.

DRS is one of the scaling assumptions It represents that for possible production process we can arbitrarily decrease the scale of the operation.

```
Analysis_DRS <- dea(input,Output,RTS = "drs")
```

```
# Efficiency of the DEA with FDH Assumption
```

```
Efficiency_DRS <- as.data.frame(Analysis_DRS$eff)
```

```
colnames(Efficiency_DRS) <- c("Efficiency_DRS")
```

```
# Identify the peers
```

```
peers_DRS <- peers(Analysis_DRS)
```

```
colnames(peers_DRS) <- c("Peer1_DRS", "Peer2_DRS", "Peer3_DRS")
```

```
# Identify the relative weights using Lambda Function
```

```
lambda_DRS <- lambda(Analysis_DRS)
```

```
colnames(lambda_DRS) <- c("DRS_Lambda1", "DRS_Lambda2", "DRS_Lambda3", "DRS_Lambda4")
```

```
dea_DRS <- cbind(Efficiency_DRS,peers_DRS,lambda_DRS)
```

```
rownames(dea_DRS)<- c('F1','F2','F3','F4','F5','F6')
```

```
#Summary table
```

```
dea_DRS
```

```
##      Efficiency_DRS Peer1_DRS Peer2_DRS Peer3_DRS DRS_Lambda1 DRS_Lambda2
## F1      1.0000000      1      NA      NA      1.0000000 0.0000000
## F2      1.0000000      2      NA      NA      0.0000000 1.0000000
## F3      1.0000000      3      NA      NA      0.0000000 0.0000000
## F4      1.0000000      4      NA      NA      0.0000000 0.0000000
## F5      0.9774987      1      2      4      0.2000000 0.08048142
## F6      0.8674521      1      2      4      0.3428571 0.39499264
##      DRS_Lambda3 DRS_Lambda4
## F1      0      0.0000000
## F2      0      0.0000000
## F3      1      0.0000000
## F4      0      1.0000000
## F5      0      0.5383307
## F6      0      0.1310751
```

Observations from DRS:

DRS Summary table indicates that facilities 1,2,3 and 4 has 100% efficiency. But facilities 4 and 5 has efficiency of 97.74% and 86.74% respectively. Both of them needs facilities 1,2 and 4 as their peers

DEA analysis with FRH

Now, we are going to formulate and compute the DEA analysis using add.

FRH is a free disposability and replicability hull assumption. It indicates the Additivity (scaling up and down, but only with integers) and free disposability.

```
Analysis_FRH <- dea(input,Output,RTS = "add")
```

```
# Efficiency of the DEA with FDH Assumption
```

```
Efficiency_FRH <- as.data.frame(Analysis_FRH$eff)
```

```
colnames(Efficiency_FRH) <- c("Efficiency_FRH")
```

```
# Identify the peers
```

```
peers_FRH <- peers(Analysis_FRH)
```

```
colnames(peers_FRH) <- c("Peer1_FRH")
```

```
# Identify the relative weights using Lambda Function
```

```
lambda_FRH <- lambda(Analysis_FRH)
```

```
colnames(lambda_FRH) <- c("FRH_Lambda1","FRH_Lambda2","FRH_Lambda3","FRH_Lambda4","FRH_Lambda5","FRH_Lambda6")
```

```
dea_FRH <- cbind(Efficiency_FRH,peers_FRH,lambda_FRH)
```

```
rownames(dea_FRH)<- c('F1','F2','F3','F4','F5','F6')
```

```
#Summary table
```

```
dea_FRH
```

```

##      Efficiency_FRH Peer1_FRH FRH_Lambda1 FRH_Lambda2 FRH_Lambda3 FRH_Lambda4
## F1          1          1          1          0          0          0
## F2          1          2          0          1          0          0
## F3          1          3          0          0          1          0
## F4          1          4          0          0          0          1
## F5          1          5          0          0          0          0
## F6          1          6          0          0          0          0
##      FRH_Lambda5 FRH_Lambda6
## F1          0          0
## F2          0          0
## F3          0          0
## F4          0          0
## F5          1          0
## F6          0          1

```

Observations from FRH:

The summary table of FRH indicates that the efficiency of all the 6 facilities is 100%, with their own peers and lambda values of 1, which is same as FDH assumption.

3) Summarize your results in a tabular format

DMU	FDH		CRS		IRS		DRS		VRS		FRH	
	Eff(%)	Peers	Eff(%)	Peers	Eff(%)	Peers	Eff(%)	Peers	Eff(%)	Peers	Eff(%)	Peers
Facility 1	100	1	100	1	100	1	100	1	100	1	100	1
Facility 2	100	2	100	2	100	2	100	2	100	2	100	2
Facility 3	100	3	100	3	100	3	100	3	100	3	100	3
Facility 4	100	4	100	4	100	4	100	4	100	4	100	4
Facility 5	100	5	97.74	1,2,4	100	5	97.74	1,2,4	100	5	100	5
Facility 6	100	6	86.74	1,2,4	89.63	1,2,5	86.74	1,2,4	89.63	1,2,5	100	6

The above table represents that 6 facilities(DMUs) with their corresponding efficiency and peers for all the assumptions.

4) Compare and Contrast the above results

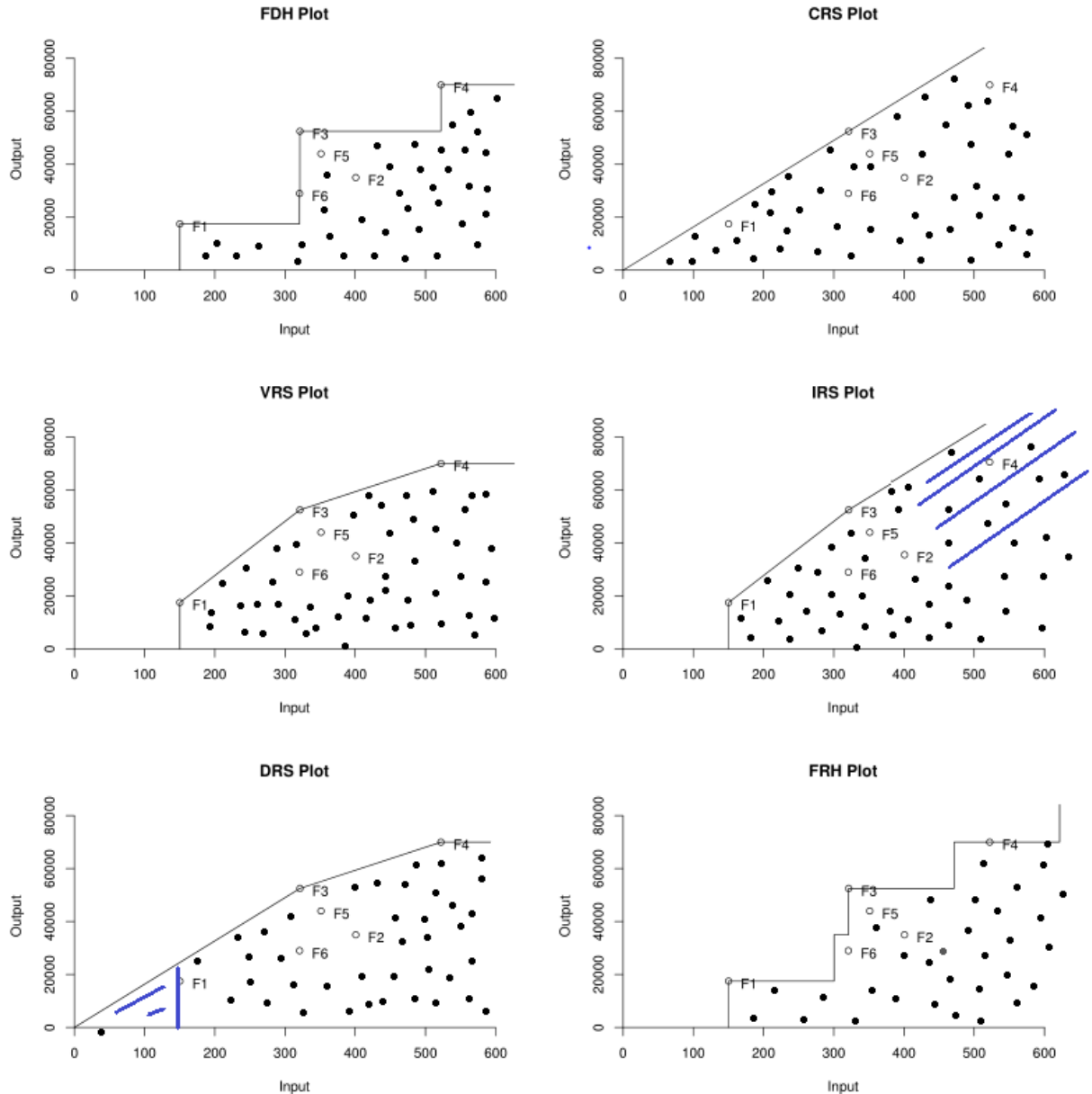
RESULTS FROM SUMMARY TABLE:

Considering,

- FDH and FRH assumptions : All the 6 facilities are 100% efficient. Each facility has their own Peers in both of these assumptions.
- CRS and DRS Assumptions : Facilities 1,2,3 and 4 are 100 % efficient while the efficiency of facilities 5 and 6 is 97.74% and 86.74% respectively and need to use the parts of facilities 1,2 and 4 as their peers.
- VRS and IRS Assumptions : Facilities 1,2,3,4 and 5 are 100 % efficient whereas the 6th facility being inefficient(89.63%) and need to use the parts of facilities 1,2 and 5 as their peers.

Comparison of the different assumptions graphically

```
dea.plot(input,Output,RTS="fdh",ORIENTATION="in-out",txt=rownames(dea_fdh),xlab='Input',ylab='Output',ma
dea.plot(input,Output,RTS="crs",ORIENTATION="in-out",txt=rownames(dea_CRS),xlab='Input',ylab='Output',ma
dea.plot(input,Output,RTS="vrs",ORIENTATION="in-out",txt=rownames(dea_VRS),xlab='Input',ylab='Output',ma
dea.plot(input,Output,RTS="irs",ORIENTATION="in-out",txt=rownames(dea_IRS),xlab='Input',ylab='Output',ma
dea.plot(input,Output,RTS="drs",ORIENTATION="in-out",txt=rownames(dea_DRS),xlab='Input',ylab='Output',ma
dea.plot(input,Output,RTS="add",ORIENTATION="in-out",txt=rownames(dea_FRH),xlab='Input',ylab='Output',ma
```



The above graphs explain the DEA technology sets under different assumptions

The main idea behind the DEA Analysis is :

- DEA approach involves looking for the smallest set that includes or envelopes the input–output observations for all of the firms. This also explains the name: **Data Envelopment Analysis**
- The larger the sets, the larger the estimated technology, i.e. the more optimistic we are in estimating the improvement potential of a firm. The flip-side of this is of course that the firms look less efficient in the larger models.

OBSERVATIONS FROM THE GRAPHS:

- We can see that FDH is the smaller technology set. As we know that FDH means that inputs and outputs can freely be disposed off (or) we can always produce fewer outputs with more inputs. Even though its efficiency is 100% , it has its own drawbacks during theoretical or empirical reference of technology. If the data set is sufficiently large, then free disposability is also sufficiently powerful to create enough relevant comparisons.
- VRS is larger set than FDH because it fills out spaces that FDH reduced. Here we can see that Facility 6 is inefficient.
- Considering IRA and DRS, it is clearly evident that DRS and IRS are larger than VRS.
 1. As we can see that, when input is low, DRS tries to increase the technology set
 2. In contrary, IRS tries to increase the technology for large input values.
 3. DRS analysis indicates that 5 and 6 facilities can enhance their efficiency and IRS shows that facility 6 may improve as well.
- Among all, CRS is largest technology set irrespective of convexity assumptions, which allows us to see if there is any possible combination to scale up or down. Based on the efficiency levels, facilities 5 and 6 should be improved.
- Regarding FRH, it is larger than FDH and smaller than CRS and its efficiency is same as FDH
- As we saw that in the main idea of the DEA analysis that the firms look less efficient in the larger models and it is quite true if examine the graph.

GOAL PROGRAMMING

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.)

The impact of each of the new products (per unit rate of production) on each of these factors is shown in the following table:

Factor	Unit Contribution			Goal	Units
	Product:				
	1	2	3		
Total profit	20	15	25	Maximize	Millions of dollars
Employment level	6	4	5	= 50	Hundreds of employees
Earnings next year	8	7	5	≥ 75	Millions of dollars

QUESTIONS:

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 $x3$.

Let x_i be the production rate of Product i , $i=1,2,3$

- $x1$ = Production rate of Product1
- $x2$ = Production rate of Product2
- $x3$ = Production rate of Product3

Let us define auxiliary variables as :

- $y1+$ and $y1-$, represents the amount over and amount under the employment level goal respectively.
- $y2+$ and $y2-$, represents the amount over and amount under the earnings goal over next year respectively.

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

Maximize $Z: P - 6C - 3D$

$$Z = 20x1 + 15x2 + 25x3 - 6y1+ - 6y2+ - 3y2-$$

S.T)

Employment Level :

$$6x1 + 4x2 + 5x3 - (y1+ - y1-) = 50$$

Earnings Level:

$$8x1 + 7x2 + 5x3 - (y2+ - y2-) = 75$$

Non Negativity constraints :

$$\begin{aligned} x_i &\geq 0 ; \quad i=1,2,3 \\ y_{i+} &\geq 0 ; \quad i=1,2 \\ y_{i-} &\geq 0 ; \quad i=1,2 \end{aligned}$$

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

```
library(lpSolveAPI)

goal<- read.lp("Goal.lp")

solve(goal)
```

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

```
#Print the model
```

```
goal
```

```
## Model name:
##
##      x1      x2      x3      y1p      y1n      y2n      y2p
## Maximize    20     15     25      -6      -6      -3      0
## R1          6      4      5      -1      1      0      0 = 50
## R2          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
```

```
#To identify the Optimal Solution
get.objective(goal)
```

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

The Optimal Value for the model is 225

```
#To Identify the variables
get.variables(goal)
```

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

Analyzing the results:

In our case the values for the variables are as follows: $x_1 = 0$; $x_2 = 0$; $x_3 = 15$; $y_{1+} = 25$, $y_{1-} = 0$, $y_{2-} = 0$

Our Objective Function: $20x_1 + 15x_2 + 25x_3 - 6y_{1+} - 6y_{2+} - 3y_{2-}$: 25×15 (Total profit) - 6×25 (Workforce Penalty) = 225

We can maximize the profit by making production rate of product 3 = 15, getting profits to $(25 \times 15 = 375)$ while we need to hire 25 more employees which will result in a penalty of $(25 \times 6 = 150)$. Hence our objective function would be 225.

which Implies that y_2 (Earnings) goal is fully satisfied. Whereas the Employment goal is projected to be exceeded by 25 and based on the total profit of Product 3 which has negative result on its profit by 150.

Prioritizing the Goals using Streamlined approach

Now Let us consider the workforce stability(Employment) and Earnings goals as a higher priority than the total profit over the life:

In streamlined Approach, we know that * (1). The different penalty weights are assigned within each of the two priority levels

- (2) the individual penalty weights for the first-priority goals have been multiplied by M.

These penalty weights yield the following single linear programming model that incorporates all the goal

PREEMPTIVE GOAL PROGRAMMING FORMULATION:

Let x_i be the production rate of Product $i, 1, 2, 3$

- x_1 = Production rate of Product1
- x_2 = Production rate of Product2
- x_3 = Production rate of Product3

Let us define auxiliary variables as :

- y_1^+ and y_1^- , represents the amount over and amount under the employment level goal respectively.
- y_2^+ and y_2^- , represents the amount over and amount under the earnings goal over next year respectively.

Maximize Z: $P - 6C - 3D$

$$Z = 20x_1 + 15x_2 + 25x_3 - 6My_1^+ - 6My_2^+ - 3My_2^-$$

S.T)

Employment Level :

$$6x_1 + 4x_2 + 5x_3 - (y_1^+ - y_1^-) = 50$$

Earnings Level:

$$8x_1 + 7x_2 + 5x_3 - (y_2^+ - y_2^-) = 75$$

Non Negativity constraints :

$$\begin{aligned} x_i &\geq 0 ; \quad i=1,2,3 \\ y_i^+ &\geq 0 ; \quad i=1,2 \\ y_i^- &\geq 0 ; \quad i=1,2 \end{aligned}$$

```
library(lpSolveAPI)

preemptive_goal<- read.lp("Preemptive_Goal.lp")

solve(preemptive_goal)
```

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

```
#Print the model
```

```
preemptive_goal
```

```
## Model name:
##           x1      x2      x3      y1p      y1n      y2n      y2p
## Maximize   20     15     25    -6000    -6000    -3000      0
## R1         6       4       5       -1       1       0       0 = 50
## R2         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
```

```
#To identify the Optimal Solution
```

```
get.objective(preemptive_goal)
```

```
## [1] 208.3333
```

```
#To identify the Optimal Solution
```

```
get.variables(preemptive_goal)
```

```
## [1] 0.000000 8.333333 3.333333 0.000000 0.000000 0.000000 0.000000
```

In this case, when the workforce and earnings goals are prioritized, we can obtain that the optimal value just by producing product 2 and 3 at the rate of 8.33 and 3.33. While the both workforce and earnings goals are achieved.

FINDINGS OF THE PROBLEM

The Non Preemptive goal programming model represents that objective function can be maximized by hiring 25 more employees which is not feasible solution for the management. Whereas, in the preemptive goal programming model using streamlined approach , we have given the high priority to the Employment and earnings goals over the total profit goal, and the solution indicates that the optimal value has been obtained by achieving all the goals unlike the previous model where Employment goal hasn't been achieved