Assignment 3

Prajwal C N

11/7/2021

Question 1: Hope Valley Health Care Association

Part A: Formulating and performing DEA analysis under all DEA assumptions of FDH, CRS, VRS, IRS, DRS, and FRH.

Formulation:

(X_1, X_2, Y_1, Y_2) are weighed output and input variables respectively.

Efficiency = (X_1 * Reimbursed patients per day + X_2 * Privately paid patients per day) / (Y_1 * Staff hours per day + Y_2 * Cost of supplies per day)

Objective function for all the Facilities is mentioned below:

Facility 1:

Max, Z= 14000 X1+ 3500 X2

Constraints:

14000 X1+ 3500 X2– 150 Y1– 0.2 Y2<= 0

14000 X1+ 21000 X2– 400 Y1– 0.7 Y2<= 0

42000 X1+ 10500 X2– 320 Y1– 1.2 Y2<= 0

28000 X1+ 42000 X2– 520 Y1– 2.0 Y2<= 0

19000 X1+ 25000 X2– 350 Y1– 1.2 Y2<= 0

14000 X1+ 15000 X2– 320 Y1– 0.7 Y2<= 0

150 Y1+ 0.2 Y2= 1

Y1, Y2, X1, X2, >=0

Facility 2:

Max, Z= 14000 X1+ 21000 X2

Constraints:

14000 X1+ 3500 X2– 150 Y1– 0.2 Y2<= 0

14000 X1+ 21000 X2– 400 Y1– 0.7 Y2<= 0

42000 X1+ 10500 X2– 320 Y1– 1.2 Y2<= 0

28000 X1+ 42000 X2– 520 Y1– 2.0 Y2<= 0

19000 X1+ 25000 X2– 350 Y1– 1.2 Y2<= 0

14000 X1+ 15000 X2– 320 Y1– 0.7 Y2<= 0

400 Y1+ 0.7 Y2= 1

Y1, Y2, X1, X2, >=0

Facility 3:

Max, Z= 42000 X1+ 10500 X2

Constraints:

14000 X1+ 3500 X2– 150 Y1– 0.2 Y2<= 0

14000 X1+ 21000 X2– 400 Y1– 0.7 Y2<= 0

42000 X1+ 10500 X2– 320 Y1– 1.2 Y2<= 0

28000 X1+ 42000 X2– 520 Y1– 2.0 Y2<= 0

19000 X1+ 25000 X2– 350 Y1– 1.2 Y2<= 0

14000 X1+ 15000 X2– 320 Y1– 0.7 Y2<= 0

320 Y1+ 1.2 Y2= 1

Y1, Y2, X1, X2, >=0

Facility 4:

Max, Z= 28000 X1+ 42000 X2

Constraints:

14000 X1+ 3500 X2– 150 Y1– 0.2 Y2<= 0

14000 X1+ 21000 X2– 400 Y1– 0.7 Y2<= 0

42000 X1+ 10500 X2– 320 Y1– 1.2 Y2<= 0

28000 X1+ 42000 X2– 520 Y1– 2.0 Y2<= 0

19000 X1+ 25000 X2– 350 Y1– 1.2 Y2<= 0

14000 X1+ 15000 X2– 320 Y1– 0.7 Y2<= 0

520 Y1+ 2.0 Y2= 1

Y1, Y2, X1, X2, >=0

Facility 5:

Max, Z= 19000 X1+ 25000 X2

Constraints:

14000 X1+ 3500 X2– 150 Y1– 0.2 Y2<= 0

14000 X1+ 21000 X2– 400 Y1– 0.7 Y2<= 0

42000 X1+ 10500 X2– 320 Y1– 1.2 Y2<= 0

28000 X1+ 42000 X2– 520 Y1– 2.0 Y2<= 0

19000 X1+ 25000 X2– 350 Y1– 1.2 Y2<= 0

14000 X1+ 15000 X2– 320 Y1– 0.7 Y2<= 0

350 Y1+ 1.2 Y2= 1

Y1, Y2, X1, X2, >=0

Facility 6:

Max, Z= 14000 X1+ 15000 X2

Constraints:

14000 X1+ 3500 X2– 150 Y1– 0.2 Y2<= 0

14000 X1+ 21000 X2– 400 Y1– 0.7 Y2<= 0

42000 X1+ 10500 X2– 320 Y1– 1.2 Y2<= 0

28000 X1+ 42000 X2– 520 Y1– 2.0 Y2<= 0

19000 X1+ 25000 X2– 350 Y1– 1.2 Y2<= 0

14000 X1+ 15000 X2– 320 Y1– 0.7 Y2<= 0

320 Y1+ 0.7 Y2= 1

Y1, Y2, X1, X2, >=0

R Code:

```
library(lpSolveAPI)
library(Benchmarking)
library(ucminf)
```

Facility 1

```
lprec1<-make.lp(0,4)
lp.control(lprec1,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.objfn(lprec1,c(0,0,14000,3500))
# Facility 1 Constraints:
add.constraint(lprec1,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec1,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec1,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec1,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec1,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec1,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec1,c(150,0.2),"=",1,indices = c(1,2))
solve(lprec1)
```

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

```
get.objective(lprec1)
```

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

```
get.variables(lprec1)
```

```
## [1] 5.172414e-03 1.120690e+00 7.142857e-05 0.000000e+00
```

The result shows that the objective value is 1, indicating that we are able to maximize Facility1's efficiency. This occurs when the output weights are 0 and 0.00714 and the input weights are 0.5172414 and 1.12069.

Facility 2

```
lprec2<-make.lp(0,4)
lp.control(lprec2,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.objfn(lprec2,c(0,0,14000,21000))
# Facility 2 Constraints:
add.constraint(lprec2,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec2,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec2,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec2,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec2,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec2,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec2,c(400,0.7),"=",1,indices = c(1,2))
solve(lprec2)
```

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

```
get.objective(lprec2)
```

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

```
get.variables(lprec2)
```

```
## [1] 1.376147e-03 6.422018e-01 0.000000e+00 4.761905e-05
```

The result shows that the objective value is 1, indicating that we are able to maximize Facility2's efficiency. This occurs when the output weights are 0 and 0.00476 and the input weights are 0.1376147 and 0.6422.

Facility 3

```
lprec3<-make.lp(0,4)
lp.control(lprec3,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.objfn(lprec3,c(0,0,42000,10500))
# Facility 3 Constraints:
add.constraint(lprec3,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec3,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec3,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec3,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec3,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec3,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec3,c(320,1.2),"=",1,indices = c(1,2))
solve(lprec3)
```

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

```
get.objective(lprec3)
```

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

```
get.variables(lprec3)
```

```
## [1] 1.724138e-03 3.735632e-01 2.380952e-05 0.000000e+00
```

The result shows that the objective value is 1, indicating that we are able to maximize Facility3's efficiency. This occurs when the output weights are 0 and 0.00238 and the input weights are 0.001724 and 0.3735.

Facility 4

```
lprec4<-make.lp(0,4)
lp.control(lprec4,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.objfn(lprec4,c(0,0,28000,42000))
# Facility 4 Constraints:
add.constraint(lprec4,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec4,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec4,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec4,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec4,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec4,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec4,c(520,2.0),"=",1,indices = c(1,2))
solve(lprec4)
```

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

```
get.objective(lprec4)
```

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

```
get.variables(lprec4)
```

```
## [1] 6.880734e-04 3.211009e-01 0.000000e+00 2.380952e-05
```

The result shows that the objective value is 1, indicating that we are able to maximize Facility4's efficiency. This occurs when the output weights are 0 and 0.00238 and the input weights are 0.0688 and 0.3211.

Facility 5

```
lprec5<-make.lp(0,4)
lp.control(lprec5,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.objfn(lprec5,c(0,0,19000,25000))
# Facility 5 Constraints:
add.constraint(lprec5,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec5,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec5,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec5,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec5,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec5,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec5,c(350,1.2),"=",1,indices = c(1,2))
solve(lprec5)
```

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

```
get.objective(lprec5)
```

```
## [1] 0.9774987
```

```
get.variables(lprec5)
```

```
## [1] 0.0010989011 0.5128205128 0.0000115123 0.0000303506
```

According to the solution, the objective value for Facility5 is 0.9775, indicating that we are unable to reach maximum efficiency. This occurs when the output and input weights are 0.00001151 and 0.000030350, respectively, and the input weights are 0.001098 and 0.5128.

Facility 6

```
lprec6<-make.lp(0,4)
lp.control(lprec6,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.objfn(lprec6,c(0,0,14000,15000))
# Facility 6 Constraints:
add.constraint(lprec6,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec6,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec6,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec6,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec6,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec6,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))
add.constraint(lprec6,c(320,0.7),"=",1,indices = c(1,2))
solve(lprec6)
```

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

```
get.objective(lprec6)
```

```
## [1] 0.8674521
```

```
get.variables(lprec6)
```

```
## [1] 1.546392e-03 7.216495e-01 1.620029e-05 4.270987e-05
```

The result shows that the objective value for Facility6 is 0.08674521, indicating that we are unable to reach maximum efficiency. This occurs when the output weights are 0.00162 and 0.00427, and the input weights are 0.1546 and 0.7216.

DEA Analysis

```
x<-matrix(c(150,400,320,520,350,320,0.2,0.7,1.2,2.0,1.2,0.7),ncol=2)
y<-
matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,150
00),ncol=2)
colnames(x)<-c("Staff Hours PerDay","Supplies PerDay")
colnames(y)<-c("Reimbursed Patients PerDayDay","Privately Paid Patients
PerDayDay")
rownames(x)<-paste0(rep("Facility",6),seq(1,6,1))
rownames(y)<-paste0(rep("Facility",6),seq(1,6,1))
row1 <-dea(x,y,RTS = "crs")
```

```
row2 <-dea(x,y,RTS = "fdh")
row3 <-dea(x,y,RTS = "vrs")
row4 <-dea(x,y,RTS = "irs")
row5 <-dea(x,y,RTS = "drs")
row6 <-dea(x,y,RTS = "add")
row7 <-dea(x,y,RTS = "irs2")
row8 <-dea(x,y,RTS = "fdh+")
row9 <-dea(x,y,RTS = "vrs+")
```

Part B: Determining the Peers and Lambdas for each assumptions

```
peers(row1)

##           peer1 peer2 peer3
## Facility1     1    NA    NA
## Facility2     2    NA    NA
## Facility3     3    NA    NA
## Facility4     4    NA    NA
## Facility5     1     2     4
## Facility6     1     2     4

lambda(row1)

##           L_Facility1 L_Facility2 L_Facility3 L_Facility4
## Facility1   1.0000000  0.00000000           0   0.0000000
## Facility2   0.0000000  1.00000000           0   0.0000000
## Facility3   0.0000000  0.00000000           1   0.0000000
## Facility4   0.0000000  0.00000000           0   1.0000000
## Facility5   0.2000000  0.08048142           0   0.5383307
## Facility6   0.3428571  0.39499264           0   0.1310751
```

The peer units for Facility(5) are [1,2,4], with relative weights [0.2000000,0.08048142,0.5383307]. Similary for Facility(6), the peer units are [1,2,4], with weights [0.3428571,0.39499264,0.1310751], respectively.

```
peers(row2)

##           peer1
## Facility1     1
## Facility2     2
## Facility3     3
## Facility4     4
## Facility5     5
## Facility6     6

lambda(row2)

##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1           1           0           0           0           0
## Facility2           0           1           0           0           0
## Facility3           0           0           1           0           0
```

```
## Facility4           0           0           0           1           0
## Facility5           0           0           0           0           1
## Facility6           0           0           0           0           0
##           L_Facility6
## Facility1           0
## Facility2           0
## Facility3           0
## Facility4           0
## Facility5           0
## Facility6           1
```

```
peers(row3)
```

```
##           peer1 peer2 peer3
## Facility1     1    NA    NA
## Facility2     2    NA    NA
## Facility3     3    NA    NA
## Facility4     4    NA    NA
## Facility5     5    NA    NA
## Facility6     1     2     5
```

```
lambda(row3)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1   1.0000000   0.0000000           0           0   0.0000000
## Facility2   0.0000000   1.0000000           0           0   0.0000000
## Facility3   0.0000000   0.0000000           1           0   0.0000000
## Facility4   0.0000000   0.0000000           0           1   0.0000000
## Facility5   0.0000000   0.0000000           0           0   1.0000000
## Facility6   0.4014399   0.3422606           0           0   0.2562995
```

```
peers(row4)
```

```
##           peer1 peer2 peer3
## Facility1     1    NA    NA
## Facility2     2    NA    NA
## Facility3     3    NA    NA
## Facility4     4    NA    NA
## Facility5     5    NA    NA
## Facility6     1     2     5
```

```
lambda(row4)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1   1.0000000   0.0000000           0           0   0.0000000
## Facility2   0.0000000   1.0000000           0           0   0.0000000
## Facility3   0.0000000   0.0000000           1           0   0.0000000
## Facility4   0.0000000   0.0000000           0           1   0.0000000
## Facility5   0.0000000   0.0000000           0           0   1.0000000
## Facility6   0.4014399   0.3422606           0           0   0.2562995
```

```
peers(row5)
```

```
##           peer1 peer2 peer3
## Facility1     1    NA    NA
## Facility2     2    NA    NA
## Facility3     3    NA    NA
## Facility4     4    NA    NA
## Facility5     1     2     4
## Facility6     1     2     4
```

```
lambda(row5)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4
## Facility1   1.0000000  0.00000000           0   0.0000000
## Facility2   0.0000000  1.00000000           0   0.0000000
## Facility3   0.0000000  0.00000000           1   0.0000000
## Facility4   0.0000000  0.00000000           0   1.0000000
## Facility5   0.2000000  0.08048142           0   0.5383307
## Facility6   0.3428571  0.39499264           0   0.1310751
```

```
peers(row6)
```

```
##           peer1
## Facility1     1
## Facility2     2
## Facility3     3
## Facility4     4
## Facility5     5
## Facility6     6
```

```
lambda(row6)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1           1           0           0           0           0
## Facility2           0           1           0           0           0
## Facility3           0           0           1           0           0
## Facility4           0           0           0           1           0
## Facility5           0           0           0           0           1
## Facility6           0           0           0           0           0
##           L_Facility6
## Facility1           0
## Facility2           0
## Facility3           0
## Facility4           0
## Facility5           0
## Facility6           1
```

```
peers(row7)
```

```
##           peer1
## Facility1     1
## Facility2     2
## Facility3     3
## Facility4     4
```

```
## Facility5     5
## Facility6     6

lambda(row7)

##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1           1           0           0           0           0
## Facility2           0           1           0           0           0
## Facility3           0           0           1           0           0
## Facility4           0           0           0           1           0
## Facility5           0           0           0           0           1
## Facility6           0           0           0           0           0
##           L_Facility6
## Facility1           0
## Facility2           0
## Facility3           0
## Facility4           0
## Facility5           0
## Facility6           1

peers(row8)

##           peer1
## Facility1     1
## Facility2     2
## Facility3     3
## Facility4     4
## Facility5     5
## Facility6     6

lambda(row8)

##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1           1           0           0           0           0
## Facility2           0           1           0           0           0
## Facility3           0           0           1           0           0
## Facility4           0           0           0           1           0
## Facility5           0           0           0           0           1
## Facility6           0           0           0           0           0
##           L_Facility6
## Facility1           0
## Facility2           0
## Facility3           0
## Facility4           0
## Facility5           0
## Facility6           1

peers(row9)

##           peer1
## Facility1     1
## Facility2     2
```

```
## Facility3     3
## Facility4     4
## Facility5     5
## Facility6     6
```

```
lambda(row9)
```

```
##           L_Facility1 L_Facility2 L_Facility3 L_Facility4 L_Facility5
## Facility1           1           0           0           0           0
## Facility2           0           1           0           0           0
## Facility3           0           0           1           0           0
## Facility4           0           0           0           1           0
## Facility5           0           0           0           0           1
## Facility6           0           0           0           0           0
##           L_Facility6
## Facility1           0
## Facility2           0
## Facility3           0
## Facility4           0
## Facility5           0
## Facility6           1
```

Part C: Summarizing the results in a tabular format

```
results <-cbind(round(row1$eff,4),round(row1$lambda,4))
colnames(results)<-c("efficieny",rownames(results))
results
```

```
##           efficieny Facility1 Facility2 Facility3 Facility4 Facility5
Facility6
## Facility1    1.0000    1.0000    0.0000         0    0.0000         0
0
## Facility2    1.0000    0.0000    1.0000         0    0.0000         0
0
## Facility3    1.0000    0.0000    0.0000         1    0.0000         0
0
## Facility4    1.0000    0.0000    0.0000         0    1.0000         0
0
## Facility5    0.9775    0.2000    0.0805         0    0.5383         0
0
## Facility6    0.8675    0.3429    0.3950         0    0.1311         0
0
```

The remaining nursing homes, with the exception of Facility5 and Facility6, are efficient, as shown in the table above.

lambda values from the above table, they are the variables related to the constraints limiting the efficiency of each unit to be no greater than 1 and the efficiency of these nursing homes can be achieved by using shadow prices.

Part D: Comparing the above results

```
c1<-
cbind(row1$eff,row2$eff,row3$eff,row4$eff,row5$eff,row6$eff,row7$eff,row8$eff
,row9$eff)
colnames(c1)<-c(paste0(rep("row",9),seq(1,9,1)))
c1
```

```
##                row1 row2      row3      row4      row5 row6 row7 row8 row9
## Facility1 1.0000000    1 1.0000000 1.0000000 1.0000000    1    1    1    1
## Facility2 1.0000000    1 1.0000000 1.0000000 1.0000000    1    1    1    1
## Facility3 1.0000000    1 1.0000000 1.0000000 1.0000000    1    1    1    1
## Facility4 1.0000000    1 1.0000000 1.0000000 1.0000000    1    1    1    1
## Facility5 0.9774987    1 1.0000000 1.0000000 0.9774987    1    1    1    1
## Facility6 0.8674521    1 0.8963283 0.8963283 0.8674521    1    1    1    1
```

The efficiency of the unit has differed from each assumption, as seen in the table above.

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.

Objective Function:

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.

R Solution:

```
library(lpSolveAPI)


lprec <- read.lp("prob_2.lp")

# Solution
solve(lprec)
```

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

```
# Getting objective value
get.objective(lprec)
```

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

```
# Get constraints
get.constraints(lprec)

## [1] 50 75

# Getting decision variable values
get.variables(lprec)

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

The **optimal solution** obtained from the simplex method is : x1=0, x2=0, x3=15, y1m=0, y1p=25, y2m=0, y2p=0. And the objective function in 225.