811129289\_5

Purna Sai Kiran, G

11/8/2021

Loading required libraries

library(lpSolveAPI)  
library(Benchmarking)

## Loading required package: ucminf

## Loading required package: quadprog

library(ucminf)

Question 1: DMU1 Formulating a linear model and setting the objective function. This is a maximization problem

LP\_1<-make.lp(0,4)  
lp.control(LP\_1,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(LP\_1,c(0,0,14000,3500))

Setting constraints

add.constraint(LP\_1,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_1,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_1,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_1,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_1,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_1,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_1,c(150,0.2),"=",1,indices = c(1,2))

Solving the LP problem

solve(LP\_1)

## [1] 0

get.objective(LP\_1)

## [1] 1

get.variables(LP\_1)

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

The objective value is 1 where we will get maximum efficiency.

DMU2 Formulating an LP and setting the function to maximization

LP\_2<-make.lp(0,4)  
lp.control(LP\_2,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(LP\_2,c(0,0,14000,21000))

Loading the constraints

add.constraint(LP\_2,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_2,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_2,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_2,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_2,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_2,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_2,c(400,0.7),"=",1,indices = c(1,2))

Solving the LP

solve(LP\_2)

## [1] 0

get.objective(LP\_2)

## [1] 1

get.variables(LP\_2)

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

For DMU 2 also, the maximum efficiency is attained at objective value 1

DMU 3 Formulating an LP and setting the function to maximization

LP\_3<-make.lp(0,4)  
lp.control(LP\_3,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(LP\_3,c(0,0,42000,10500))

Adding constraints

add.constraint(LP\_3,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_3,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_3,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_3,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_3,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_3,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_3,c(320,1.2),"=",1,indices = c(1,2))

Solving the LP

solve(LP\_3)

## [1] 0

get.objective(LP\_3)

## [1] 1

get.variables(LP\_3)

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

The maximum efficiency is at objective value 1. Inputs are 0.0017 and 0.373 where output is 0.00238 and 0

DMU 4 Formulating the LP with maximization problem

LP\_4<-make.lp(0,4)  
lp.control(LP\_4,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(LP\_4,c(0,0,28000,42000))

Adding constraints

add.constraint(LP\_4,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_4,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_4,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_4,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_4,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_4,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_4,c(520,2.0),"=",1,indices = c(1,2))

Solving the LP

solve(LP\_4)

## [1] 0

get.objective(LP\_4)

## [1] 1

get.variables(LP\_4)

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

Maximum efficiency is at 1 where the outputs are 0 and 0.0000238 and inputs are 0.000688 and 0.321

DMU 5 Formulating the lp with maximization problem

LP\_5<-make.lp(0,4)  
lp.control(LP\_5,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(LP\_5,c(0,0,19000,25000))

Adding constraints

add.constraint(LP\_5,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_5,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_5,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_5,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_5,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_5,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_5,c(350,1.2),"=",1,indices = c(1,2))

Solving the LP problem

solve(LP\_5)

## [1] 0

get.objective(LP\_5)

## [1] 0.9774987

get.variables(LP\_5)

## [1] 0.0010989011 0.5128205128 0.0000115123 0.0000303506

Maximum efficiency is at 1 when the input weights are 0.001 and 0.5 and output weights are 0.00, 0.00

DMU 6 Formulating the LP problem with maximization function

LP\_6<-make.lp(0,4)  
lp.control(LP\_6,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(LP\_6,c(0,0,14000,15000))

Adding the constraints

add.constraint(LP\_6,c(-400,-0.7,14000,21000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_6,c(-150,-0.2,14000,3500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_6,c(-320,-1.2,42000,10500),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_6,c(-520,-2.0,28000,42000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_6,c(-350,-1.2,19000,25000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_6,c(-320,-0.7,14000,15000),"<=",0,indices = c(1,2,3,4))  
add.constraint(LP\_6,c(320,0.7),"=",1,indices = c(1,2))

Solving the LP

solve(LP\_6)

## [1] 0

get.objective(LP\_6)

## [1] 0.8674521

get.variables(LP\_6)

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

The effieincy is found at objective value 0.8 where the inpu weights are 0.0015 and 0.721 and outputs weights are set to 0.000016 and 0.000042

DAE 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(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,15000),ncol=2)  
colnames(X)<-c("Staff","Supplies")  
colnames(Y)<-c("Reimbursed\_Patients","Paid\_Patients")  
rownames(X)<-paste0(rep("Facility",6),seq(1,6,1))  
rownames(Y)<-paste0(rep("Facility",6),seq(1,6,1))  
A <-dea(X,Y,RTS = "crs")  
A

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

B <-dea(X,Y,RTS = "fdh")  
B

## [1] 1 1 1 1 1 1

C <-dea(X,Y,RTS = "vrs")  
C

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

D <-dea(X,Y,RTS = "irs")  
D

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

E <-dea(X,Y,RTS = "drs")  
E

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

F <-dea(X,Y,RTS = "add")  
F

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1 1 1 1 1 1

G <-dea(X,Y,RTS = "irs2")  
G

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1 1 1 1 1 1

H <-dea(X,Y,RTS = "fdh+")  
H

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1 1 1 1 1 1

I <-dea(X,Y,RTS = "vrs+")  
I

## Facility1 Facility2 Facility3 Facility4 Facility5 Facility6   
## 1 1 1 1 1 1

Question 2: Peers and Lambda Values

peers(A)

## 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(A)

## 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(B)

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

lambda(B)

## 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(C)

## 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(C)

## 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(D)

## 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(D)

## 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(E)

## 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(E)

## 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(F)

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

lambda(F)

## 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(G)

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

lambda(G)

## 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(H)

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

lambda(H)

## 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(I)

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

lambda(I)

## 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

Question 3:

M <- cbind(A$eff, B$eff, C$eff, D$eff, E$eff, F$eff)  
colnames(M) <- c('CRS Efficiency','FDH Efficiency','VRS Efficiency',  
'IRS Efficiency', 'DRS Efficiency', 'FRH Efficiency')  
M

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

Question 4:

N<-cbind(A$eff,B$eff,C$eff,D$eff,E$eff,F$eff,G$eff,H$eff,I$eff)  
colnames(N)<-c(paste0(rep("Q",9),seq(1,7,1)))  
N

## Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q1 Q2  
## 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

We can see that all facilities are efficient but facility 5 and 6 are inefficient Facility 5 is efficiency for FDH, VRS, IRS and FRH and for CRS and DRS assumptions it is 97% efficient Facility 6 is fully efficient for FDH and FRS assumptions. For CRS and DRS it is 86.7% efficient and for IRS and VRS it is 89.6% efficient