64018\_Assignment 8

Brian Yim

2025-10-26

rm(list = ls())  
library(Benchmarking)

## Warning: package 'Benchmarking' was built under R version 4.4.3

## Loading required package: lpSolveAPI

## Warning: package 'lpSolveAPI' was built under R version 4.4.3

## Loading required package: ucminf

## Loading required package: quadprog

library(knitr)

## Warning: package 'knitr' was built under R version 4.4.2

library(kableExtra)

## Warning: package 'kableExtra' was built under R version 4.4.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.4.3

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:kableExtra':  
##   
## group\_rows

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.4.3

## Warning: package 'ggplot2' was built under R version 4.4.3

## Warning: package 'forcats' was built under R version 4.4.3

## Warning: package 'lubridate' was built under R version 4.4.2

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ readr 2.1.5  
## ✔ ggplot2 3.5.2 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.4 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::group\_rows() masks kableExtra::group\_rows()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Create the data  
facility\_data <- data.frame(  
DMU <- c("Facility 1", "Facility 2", "Facility 3", "Facility 4", "Facility 5", "Facility 6"),  
Staff\_hours\_per\_day <- c(150, 400, 320, 520, 350, 320),  
Supplies\_per\_day <- c(0.2, 0.7, 1.2, 2, 1.2, 0.7),  
Reimbursed\_patient\_days <- c(14000, 14000, 42000, 28000, 19000, 14000),  
Privately\_paid\_patient\_days <- c(3500, 21000, 10500, 42000, 25000, 15000)  
)  
  
# Inputs: Staff hours per day and Supplies per day  
inputs <- matrix(cbind(c(150, 400, 320, 520, 350, 320), c(0.2, 0.7, 1.2, 2, 1.2, 0.7)), ncol=2)  
colnames(inputs) <- c("Staff hours per day", "Supplies per day (in thousands)")  
  
inputs

## Staff hours per day Supplies per day (in thousands)  
## [1,] 150 0.2  
## [2,] 400 0.7  
## [3,] 320 1.2  
## [4,] 520 2.0  
## [5,] 350 1.2  
## [6,] 320 0.7

# Outputs: Reimbursed patient-days and Privately paid patient-days  
outputs <- matrix(cbind(c(14000, 14000, 42000, 28000, 19000, 14000), c(3500, 21000, 10500, 42000, 25000, 15000)), ncol=2)  
colnames(outputs) <- c("Reimbursed patient-days", "Privately paid patient-days")  
  
outputs

## Reimbursed patient-days Privately paid patient-days  
## [1,] 14000 3500  
## [2,] 14000 21000  
## [3,] 42000 10500  
## [4,] 28000 42000  
## [5,] 19000 25000  
## [6,] 14000 15000

#Q.1. Formulate and perform DEA analysis under all DEA assumptions of FDH, CRS, VRS, IRS, DRS, and FRH.

#Formulate DEA for Facility 1 #This formulation seeks to maximize the efficiency of Facility 1 in comparison to the other facilities. The efficiency score for each Decision Making Unit (DMU) is calculated as the ratio of the weighted sum of outputs to the weighted sum of inputs. Facility 1 achieves maximum efficiency when this ratio reaches its highest possible value. #Given #u1 and u2: Weights for the outputs (reimbursed patient-days and privately paid patient-days respectively). #v1 and v2: Weights for the inputs (staff hours per day and cost of supplies respectively). #Objective Function is to maximize the ratio of weighted outputs to weighted inputs for a specific DMU, Facility 1. #Objective Function: maxZ=u1×14000+u2×3500 #Subject to: #1.Input normalization for Facility 1 u1×150+u2×0.2=1 #2.u1×14000+u2×3500−u1×150−u2×0.2=0 #3.u1×14000+u2×21000−u1×400−u2×0.7≤0 #4.u1×42000+u2×10500−u1×320−u2×1.2≤0 #5.u1×28000+u2×42000−u1×520−u2×2.0≤0 #6.u1×19000+u2×25000−u1×350−u2×1.2≤0 #7.u1×14000+u2×15000−u1×320−u2×0.7≤0

#Formulate DEA for Facility 2~6 #For Facilities 2 through 6, the output values in the constraints are fixed according to each facility’s own #outputs, while the input values vary depending on the facility being evaluated. Once the objective function and input normalization constraint are defined for a facility, the remaining constraints remain unchanged, serving to compare that facility’s efficiency against all others. Therefore

#Facility 2 #Objective Function: maxZ=u1×14000+u2×21000 #Input Normalization Constraint: u1×400+u2×0.7=1

#Facility 3 #Objective Function: maxZ=u1×42000+u2×10500 #Input Normalization Constraint: u1×320+u2×1.2=1

#Facility 4 #Objective Function: maxZ=u1×28000+u2×42000 #Input Normalization Constraint: u1×520+u2×2.0=1

#Facility 5 #Objective Function: maxZ=u1×19000+u2×25000 #Input Normalization Constraint: u1×350+u2×1.2=1

#Facility 6 #Objective Function: maxZ=u1×14000+u2×15000 #Input Normalization Constraint: u1×320+u2×0.7=1

#Q1 and Q2 #It would make more sense to run DEA Analysis and the Peers and Lambdas for different RTS assumptions

#FDH #Results: Under the Free Disposal Hull (FDH) model, all six facilities received an efficiency score of 1.0, indicating that each facility lies on the FDH efficiency frontier. This outcome suggests that no facility is strictly dominated by any other in terms of the combination of inputs and outputs. In other words, each facility is considered fully efficient relative to the observed set of facilities. The peer analysis further supports this interpretation. Each facility’s peer is itself, and the corresponding lambda matrix shows a value of 1 for the facility and 0 for all others. This indicates that the construction of the FDH frontier for each decision-making unit relies solely on the facility itself, with no contribution from other facilities. Such a pattern is common in small datasets or when each facility has a unique combination of inputs and outputs, as FDH does not assume convexity and evaluates efficiency purely based on dominance. The FDH model confirms that all facilities are technically efficient, but it does not provide discrimination among them.

efficiency\_FDH <- dea(inputs, outputs, RTS="fdh")  
peer\_FDH <- peers(efficiency\_FDH)  
lambda\_FDH <- lambda(efficiency\_FDH)  
efficiency\_FDH

## [1] 1 1 1 1 1 1

peer\_FDH

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

lambda\_FDH

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

#CRS #Results: In the CRS (Constant Returns to Scale) DEA model, four out of the six facilities—Facilities 1 through 4—are fully efficient, each with a score of 1.0. This means these facilities are on the efficiency frontier and are producing the maximum output possible for their level of inputs. Facilities 5 and 6, however, have efficiency scores of 0.9775 and 0.8675, which shows they are not fully efficient and could improve by either reducing inputs or increasing outputs. When looking at the peers, Facilities 1 through 4 are their own peers, which makes sense because they are fully efficient. For Facilities 5 and 6, the efficiency frontier is made up of a combination of Facilities 1, 2, and 4. The lambda values show how much each peer contributes: for Facility 5, the largest weight is on Facility 4 (0.538), with smaller weights for Facilities 1 and 2; for Facility 6, the weights are more spread out, showing that it is compared to several efficient peers.The CRS model shows which facilities are fully efficient and which ones could improve. Facilities 5 and 6 can use the top-performing facilities as benchmarks to guide how to reduce inputs or increase outputs. The efficiency scores, peers, and lambda values together give a clear picture of how each facility performs under constant returns to scale.

efficiency\_CRS <- dea(inputs, outputs, RTS="crs")  
peer\_CRS <- peers(efficiency\_CRS)  
lambda\_CRS <- lambda(efficiency\_CRS)  
efficiency\_CRS

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

peer\_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

lambda\_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

#VRS #Results: In the VRS (Variable Returns to Scale) DEA model, five out of the six facilities—Facilities 1 through 5—are fully efficient, each with a score of 1.0. This means these facilities are operating on the efficiency frontier when we account for variable returns to scale. Facility 6, on the other hand, has an efficiency score of 0.8963, which shows it is not fully efficient and could improve by adjusting its inputs or outputs. Looking at the peers, Facilities 1 through 5 are their own peers, which makes sense because they are fully efficient. Facility 6 is benchmarked against Facilities 1, 2, and 5. The lambda values show how much each peer contributes to Facility 6’s efficiency: Facility 1 has the largest weight (about 0.401), Facility 2 contributes 0.342, and Facility 5 contributes 0.256. This tells us that to become efficient, Facility 6 would need to model itself after these three top-performing facilities. The VRS model helps us see that most facilities are efficient, but Facility 6 still has some room to improve. Compared to CRS, VRS gives a more realistic view because it allows for differences in scale among the facilities.

efficiency\_VRS <- dea(inputs, outputs, RTS="vrs")  
peer\_VRS <- peers(efficiency\_VRS)  
lambda\_VRS <- lambda(efficiency\_VRS)  
  
efficiency\_VRS

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

peer\_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

lambda\_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

#IRS #Results: In the Increasing Returns to Scale (IRS) model, five of the six facilities—Facilities 1 through 5—achieve an efficiency score of 1.0, indicating that they are operating on the efficiency frontier under the assumption that output can increase more than proportionally with inputs. Facility 6, however, has a lower efficiency score of 0.8963, suggesting it is not performing as efficiently as the best-practice facilities and could increase output relative to its current level of inputs.Examining the peer relationships, each of the fully efficient facilities (1–5) serves as its own benchmark, reflecting their optimal performance. Facility 6 relies on a combination of Facilities 1, 2, and 5 to form its reference set. The lambda values indicate how much each peer influences Facility 6’s efficiency: Facility 1 contributes approximately 0.401, Facility 2 about 0.342, and Facility 5 around 0.256. These values show that Facility 6 could improve by adopting practices similar to these higher-performing facilities.The IRS analysis emphasizes the potential for efficiency gains when facilities operate at a scale where increasing inputs could lead to disproportionately higher outputs. While most facilities are already efficient, Facility 6 has clear opportunities for improvement, and the combination of peer identification and lambda weights provides guidance on which facilities it should benchmark against.

efficiency\_IRS <- dea(inputs, outputs, RTS="irs")  
peer\_IRS <- peers(efficiency\_IRS)  
lambda\_IRS <- lambda(efficiency\_IRS)  
  
efficiency\_IRS

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

peer\_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

lambda\_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

#DRS #Results: In the Decreasing Returns to Scale (DRS) model, four of the six facilities—Facilities 1 through 4—are fully efficient, each with a score of 1.0. This indicates that these facilities are operating on the efficiency frontier even under the assumption that increasing inputs leads to less than proportional increases in outputs. Facility 5 has a slightly lower efficiency score of 0.9775, while Facility 6 has the lowest score of 0.8675, suggesting that these two facilities are not making optimal use of their inputs and could improve performance. When looking at the peer assignments, the fully efficient facilities (1–4) serve as their own benchmarks. Facilities 5 and 6 rely on a combination of the efficient facilities as references. For Facility 5, the largest weight is on Facility 4 (0.538), with smaller contributions from Facilities 1 (0.200) and 2 (0.080). Facility 6, meanwhile, has a more spread-out set of weights, with the largest contributions from Facilities 2 (0.395) and 1 (0.343) and a smaller weight from Facility 4 (0.131). This suggests that Facilities 5 and 6 could improve efficiency by adopting practices similar to these higher-performing peers.DRS model highlights how some facilities may experience diminishing efficiency when scaling up inputs. While Facilities 1–4 are already operating efficiently, Facilities 5 and 6 have room for improvement, and the peer and lambda information helps identify which top-performing facilities they should emulate to increase their efficiency.

efficiency\_DRS <- dea(inputs, outputs, RTS="drs")  
peer\_DRS <- peers(efficiency\_DRS)  
lambda\_DRS <- lambda(efficiency\_DRS)  
  
efficiency\_DRS

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

peer\_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

lambda\_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

#FRH #Results: The FRH results is identical as the FDH results maybe because of there are only 6 facilities, and the data points are already discrete and not forming complex convex combinations.

efficiency\_FRH <- dea(inputs, outputs, RTS="add")  
peer\_FRH <- peers(efficiency\_FRH)  
lambda\_FRH <- lambda(efficiency\_FRH)  
  
efficiency\_FRH

## [1] 1 1 1 1 1 1

peer\_FRH

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

lambda\_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

#Q3

#Summarize the results in a tabular format\_FDH

# Convert efficiency\_FDH to a dataframe  
df\_efficiency\_FDH <- as.data.frame(efficiency\_FDH$eff)  
colnames(df\_efficiency\_FDH) <- "Efficiency\_FDH"  
# Convert peer\_FDH to a dataframe  
df\_peer\_FDH <- as.data.frame(peer\_FDH)  
colnames(df\_peer\_FDH) <- colnames(peer\_FDH)  
# Convert peer\_FDH to a dataframe  
df\_lambda\_FDH <- as.data.frame(lambda\_FDH)  
colnames(df\_lambda\_FDH) <- colnames(lambda\_FDH)  
# Bind the new columns to your base\_df  
df\_FDH <- cbind(facility\_data, df\_efficiency\_FDH, df\_peer\_FDH, df\_lambda\_FDH)  
df\_FDH

## DMU....c..Facility.1....Facility.2....Facility.3....Facility.4...  
## 1 Facility 1  
## 2 Facility 2  
## 3 Facility 3  
## 4 Facility 4  
## 5 Facility 5  
## 6 Facility 6  
## Staff\_hours\_per\_day....c.150..400..320..520..350..320.  
## 1 150  
## 2 400  
## 3 320  
## 4 520  
## 5 350  
## 6 320  
## Supplies\_per\_day....c.0.2..0.7..1.2..2..1.2..0.7.  
## 1 0.2  
## 2 0.7  
## 3 1.2  
## 4 2.0  
## 5 1.2  
## 6 0.7  
## Reimbursed\_patient\_days....c.14000..14000..42000..28000..19000..  
## 1 14000  
## 2 14000  
## 3 42000  
## 4 28000  
## 5 19000  
## 6 14000  
## Privately\_paid\_patient\_days....c.3500..21000..10500..42000..25000..  
## 1 3500  
## 2 21000  
## 3 10500  
## 4 42000  
## 5 25000  
## 6 15000  
## Efficiency\_FDH peer1 L1 L2 L3 L4 L5 L6  
## 1 1 1 1 0 0 0 0 0  
## 2 1 2 0 1 0 0 0 0  
## 3 1 3 0 0 1 0 0 0  
## 4 1 4 0 0 0 1 0 0  
## 5 1 5 0 0 0 0 1 0  
## 6 1 6 0 0 0 0 0 1

#Summarize the results in a tabular format\_CRS

df\_efficiency\_CRS <- as.data.frame(efficiency\_CRS$eff)  
colnames(df\_efficiency\_CRS) <- "Efficiency\_CRS"  
df\_peer\_CRS <- as.data.frame(peer\_CRS)  
colnames(df\_peer\_CRS) <- colnames(peer\_CRS)  
df\_lambda\_CRS <- as.data.frame(lambda\_CRS)  
colnames(df\_lambda\_CRS) <- colnames(lambda\_CRS)  
df\_CRS <- cbind(facility\_data, df\_efficiency\_CRS, df\_peer\_CRS, df\_lambda\_CRS)  
df\_CRS

## DMU....c..Facility.1....Facility.2....Facility.3....Facility.4...  
## 1 Facility 1  
## 2 Facility 2  
## 3 Facility 3  
## 4 Facility 4  
## 5 Facility 5  
## 6 Facility 6  
## Staff\_hours\_per\_day....c.150..400..320..520..350..320.  
## 1 150  
## 2 400  
## 3 320  
## 4 520  
## 5 350  
## 6 320  
## Supplies\_per\_day....c.0.2..0.7..1.2..2..1.2..0.7.  
## 1 0.2  
## 2 0.7  
## 3 1.2  
## 4 2.0  
## 5 1.2  
## 6 0.7  
## Reimbursed\_patient\_days....c.14000..14000..42000..28000..19000..  
## 1 14000  
## 2 14000  
## 3 42000  
## 4 28000  
## 5 19000  
## 6 14000  
## Privately\_paid\_patient\_days....c.3500..21000..10500..42000..25000..  
## 1 3500  
## 2 21000  
## 3 10500  
## 4 42000  
## 5 25000  
## 6 15000  
## Efficiency\_CRS peer1 peer2 peer3 L1 L2 L3 L4  
## 1 1.0000000 1 NA NA 1.0000000 0.00000000 0 0.0000000  
## 2 1.0000000 2 NA NA 0.0000000 1.00000000 0 0.0000000  
## 3 1.0000000 3 NA NA 0.0000000 0.00000000 1 0.0000000  
## 4 1.0000000 4 NA NA 0.0000000 0.00000000 0 1.0000000  
## 5 0.9774987 1 2 4 0.2000000 0.08048142 0 0.5383307  
## 6 0.8674521 1 2 4 0.3428571 0.39499264 0 0.1310751

#Summarize the results in a tabular format\_VRS

df\_efficiency\_VRS <- as.data.frame(efficiency\_VRS$eff)  
colnames(df\_efficiency\_VRS) <- "Efficiency\_VRS"  
df\_peer\_VRS <- as.data.frame(peer\_VRS)  
colnames(df\_peer\_VRS) <- colnames(peer\_VRS)  
df\_lambda\_VRS <- as.data.frame(lambda\_VRS)  
colnames(df\_lambda\_VRS) <- colnames(lambda\_VRS)  
df\_VRS <- cbind(facility\_data, df\_efficiency\_VRS, df\_peer\_VRS, df\_lambda\_VRS)  
df\_VRS

## DMU....c..Facility.1....Facility.2....Facility.3....Facility.4...  
## 1 Facility 1  
## 2 Facility 2  
## 3 Facility 3  
## 4 Facility 4  
## 5 Facility 5  
## 6 Facility 6  
## Staff\_hours\_per\_day....c.150..400..320..520..350..320.  
## 1 150  
## 2 400  
## 3 320  
## 4 520  
## 5 350  
## 6 320  
## Supplies\_per\_day....c.0.2..0.7..1.2..2..1.2..0.7.  
## 1 0.2  
## 2 0.7  
## 3 1.2  
## 4 2.0  
## 5 1.2  
## 6 0.7  
## Reimbursed\_patient\_days....c.14000..14000..42000..28000..19000..  
## 1 14000  
## 2 14000  
## 3 42000  
## 4 28000  
## 5 19000  
## 6 14000  
## Privately\_paid\_patient\_days....c.3500..21000..10500..42000..25000..  
## 1 3500  
## 2 21000  
## 3 10500  
## 4 42000  
## 5 25000  
## 6 15000  
## Efficiency\_VRS peer1 peer2 peer3 L1 L2 L3 L4 L5  
## 1 1.0000000 1 NA NA 1.0000000 0.0000000 0 0 0.0000000  
## 2 1.0000000 2 NA NA 0.0000000 1.0000000 0 0 0.0000000  
## 3 1.0000000 3 NA NA 0.0000000 0.0000000 1 0 0.0000000  
## 4 1.0000000 4 NA NA 0.0000000 0.0000000 0 1 0.0000000  
## 5 1.0000000 5 NA NA 0.0000000 0.0000000 0 0 1.0000000  
## 6 0.8963283 1 2 5 0.4014399 0.3422606 0 0 0.2562995

#Summarize the results in a tabular format\_IRS

df\_efficiency\_IRS <- as.data.frame(efficiency\_IRS$eff)  
colnames(df\_efficiency\_IRS) <- "Efficiency\_IRS"  
df\_peer\_IRS <- as.data.frame(peer\_IRS)  
colnames(df\_peer\_IRS) <- colnames(peer\_IRS)  
df\_lambda\_IRS <- as.data.frame(lambda\_IRS)  
colnames(df\_lambda\_IRS) <- colnames(lambda\_IRS)  
df\_IRS <- cbind(facility\_data, df\_efficiency\_IRS, df\_peer\_IRS, df\_lambda\_IRS)  
df\_IRS

## DMU....c..Facility.1....Facility.2....Facility.3....Facility.4...  
## 1 Facility 1  
## 2 Facility 2  
## 3 Facility 3  
## 4 Facility 4  
## 5 Facility 5  
## 6 Facility 6  
## Staff\_hours\_per\_day....c.150..400..320..520..350..320.  
## 1 150  
## 2 400  
## 3 320  
## 4 520  
## 5 350  
## 6 320  
## Supplies\_per\_day....c.0.2..0.7..1.2..2..1.2..0.7.  
## 1 0.2  
## 2 0.7  
## 3 1.2  
## 4 2.0  
## 5 1.2  
## 6 0.7  
## Reimbursed\_patient\_days....c.14000..14000..42000..28000..19000..  
## 1 14000  
## 2 14000  
## 3 42000  
## 4 28000  
## 5 19000  
## 6 14000  
## Privately\_paid\_patient\_days....c.3500..21000..10500..42000..25000..  
## 1 3500  
## 2 21000  
## 3 10500  
## 4 42000  
## 5 25000  
## 6 15000  
## Efficiency\_IRS peer1 peer2 peer3 L1 L2 L3 L4 L5  
## 1 1.0000000 1 NA NA 1.0000000 0.0000000 0 0 0.0000000  
## 2 1.0000000 2 NA NA 0.0000000 1.0000000 0 0 0.0000000  
## 3 1.0000000 3 NA NA 0.0000000 0.0000000 1 0 0.0000000  
## 4 1.0000000 4 NA NA 0.0000000 0.0000000 0 1 0.0000000  
## 5 1.0000000 5 NA NA 0.0000000 0.0000000 0 0 1.0000000  
## 6 0.8963283 1 2 5 0.4014399 0.3422606 0 0 0.2562995

#Summarize the results in a tabular format\_DRS

df\_efficiency\_DRS <- as.data.frame(efficiency\_DRS$eff)  
colnames(df\_efficiency\_DRS) <- "Efficiency\_DRS"  
df\_peer\_DRS <- as.data.frame(peer\_DRS)  
colnames(df\_peer\_DRS) <- colnames(peer\_DRS)  
df\_lambda\_DRS <- as.data.frame(lambda\_DRS)  
colnames(df\_lambda\_DRS) <- colnames(lambda\_DRS)  
df\_DRS <- cbind(facility\_data, df\_efficiency\_DRS, df\_peer\_DRS, df\_lambda\_DRS)  
df\_DRS

## DMU....c..Facility.1....Facility.2....Facility.3....Facility.4...  
## 1 Facility 1  
## 2 Facility 2  
## 3 Facility 3  
## 4 Facility 4  
## 5 Facility 5  
## 6 Facility 6  
## Staff\_hours\_per\_day....c.150..400..320..520..350..320.  
## 1 150  
## 2 400  
## 3 320  
## 4 520  
## 5 350  
## 6 320  
## Supplies\_per\_day....c.0.2..0.7..1.2..2..1.2..0.7.  
## 1 0.2  
## 2 0.7  
## 3 1.2  
## 4 2.0  
## 5 1.2  
## 6 0.7  
## Reimbursed\_patient\_days....c.14000..14000..42000..28000..19000..  
## 1 14000  
## 2 14000  
## 3 42000  
## 4 28000  
## 5 19000  
## 6 14000  
## Privately\_paid\_patient\_days....c.3500..21000..10500..42000..25000..  
## 1 3500  
## 2 21000  
## 3 10500  
## 4 42000  
## 5 25000  
## 6 15000  
## Efficiency\_DRS peer1 peer2 peer3 L1 L2 L3 L4  
## 1 1.0000000 1 NA NA 1.0000000 0.00000000 0 0.0000000  
## 2 1.0000000 2 NA NA 0.0000000 1.00000000 0 0.0000000  
## 3 1.0000000 3 NA NA 0.0000000 0.00000000 1 0.0000000  
## 4 1.0000000 4 NA NA 0.0000000 0.00000000 0 1.0000000  
## 5 0.9774987 1 2 4 0.2000000 0.08048142 0 0.5383307  
## 6 0.8674521 1 2 4 0.3428571 0.39499264 0 0.1310751

#Q.4. Compare and Contrast the above Results

#Under CRS, Facilities 1–4 are fully efficient, while Facilities 5 and 6 score below 1, showing they could improve by adjusting inputs or outputs proportionally. VRS and IRS relax the scale assumption, making Facility 5 fully efficient and leaving only Facility 6 below the frontier. DRS, assuming diminishing returns, shows Facilities 1–4 fully efficient, with Facilities 5 and 6 less efficient, reflecting inefficiencies at larger scales.FDH and FRH treat each facility individually and non-convexly, resulting in all six facilities being fully efficient with each as its own peer.