Mderangu\_DEA

2024-10-23

library("Benchmarking")

## Warning: package 'Benchmarking' was built under R version 4.2.3

## Loading required package: lpSolveAPI

## Warning: package 'lpSolveAPI' was built under R version 4.2.3

## Loading required package: ucminf

## Warning: package 'ucminf' was built under R version 4.2.3

## Loading required package: quadprog

##   
## Loading Benchmarking version 0.32h (Revision 263, 2024/03/13 15:04:04) ...

## Build 2024/03/13 15:05:00

# Define facility names as a character vector  
facility\_names <- c("Facility 1", "Facility 2", "Facility 3", "Facility 4", "Facility 5", "Facility 6")  
  
# Input Data for the Six Facilities  
staff\_hours <- c(150, 400, 320, 520, 350, 320) # Staffing Labor (hours per day)  
supplies <- c(0.2, 0.7, 1.2, 2.0, 1.2, 0.7) # Cost of Supplies (thousands of dollars per day)  
  
# Output Data  
reimbursed\_patient\_days <- c(14000, 14000, 42000, 28000, 19000, 14000) # Reimbursed by third-party  
privately\_paid\_days <- c(3500, 21000, 10500, 42000, 25000, 15000) # Privately paid patient-days  
  
# Prepare input and output matrices  
x <- matrix(c(staff\_hours, supplies), ncol=2) # Inputs: Staff hours and supplies  
y <- matrix(c(reimbursed\_patient\_days, privately\_paid\_days), ncol=2) # Outputs: Patient-days  
  
# Set column names for clarity  
colnames(x) <- c("Staff\_Hours\_Per\_Day", "Supplies\_Per\_Day")  
colnames(y) <- c("Reimbursed\_Patient\_Days", "Privately\_Paid\_Patient\_Days")  
  
combined\_data <- data.frame(DMU = facility\_names, x, y)  
combined\_data

## DMU Staff\_Hours\_Per\_Day Supplies\_Per\_Day Reimbursed\_Patient\_Days  
## 1 Facility 1 150 0.2 14000  
## 2 Facility 2 400 0.7 14000  
## 3 Facility 3 320 1.2 42000  
## 4 Facility 4 520 2.0 28000  
## 5 Facility 5 350 1.2 19000  
## 6 Facility 6 320 0.7 14000  
## Privately\_Paid\_Patient\_Days  
## 1 3500  
## 2 21000  
## 3 10500  
## 4 42000  
## 5 25000  
## 6 15000

# Perform DEA under Constant Returns to Scale (CRS)  
DEA\_CRS <- dea(x, y, RTS = "crs")  
  
# Print DEA CRS Results  
print("DEA CRS Results:")

## [1] "DEA CRS Results:"

print(DEA\_CRS)

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

# Print peers for CRS  
print("Peers for CRS:")

## [1] "Peers for CRS:"

print(peers(DEA\_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

# Print lambdas for CRS  
print("Lambdas for CRS:")

## [1] "Lambdas for CRS:"

print(lambda(DEA\_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

# DEA under Variable Returns to Scale (VRS)  
DEA\_VRS <- dea(x, y, RTS = "vrs") # Perform DEA with VRS assumption  
print("DEA VRS Results:") # Print VRS results

## [1] "DEA VRS Results:"

print(DEA\_VRS)

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

print("Peers for VRS:") # Print peers for VRS

## [1] "Peers for VRS:"

print(peers(DEA\_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

print("Lambdas for VRS:") # Print lambda values for VRS

## [1] "Lambdas for VRS:"

print(lambda(DEA\_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

# DEA under Decreasing Returns to Scale (DRS)  
DEA\_DRS <- dea(x, y, RTS = "drs") # Perform DEA with DRS assumption  
print("DEA DRS Results:") # Print DRS results

## [1] "DEA DRS Results:"

print(DEA\_DRS)

## [1] 1.0000 1.0000 1.0000 1.0000 0.9775 0.8675

print("Peers for DRS:") # Print peers for DRS

## [1] "Peers for DRS:"

print(peers(DEA\_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

print("Lambdas for DRS:") # Print lambda values for DRS

## [1] "Lambdas for DRS:"

print(lambda(DEA\_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

# DEA under Increasing Returns to Scale (IRS)  
DEA\_IRS <- dea(x, y, RTS = "irs") # Perform DEA with IRS assumption  
print("DEA IRS Results:") # Print IRS results

## [1] "DEA IRS Results:"

print(DEA\_IRS)

## [1] 1.0000 1.0000 1.0000 1.0000 1.0000 0.8963

print("Peers for IRS:") # Print peers for IRS

## [1] "Peers for IRS:"

print(peers(DEA\_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

print("Lambdas for IRS:") # Print lambda values for IRS

## [1] "Lambdas for IRS:"

print(lambda(DEA\_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

# DEA under Free Disposability Hull (FDH)  
DEA\_FDH <- dea(x, y, RTS = "fdh") # Perform DEA with FDH assumption  
print("DEA FDH Results:") # Print FDH results

## [1] "DEA FDH Results:"

print(DEA\_FDH)

## [1] 1 1 1 1 1 1

print("Peers for FDH:") # Print peers for FDH

## [1] "Peers for FDH:"

print(peers(DEA\_FDH))

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

print("Lambdas for FDH:") # Print lambda values for FDH

## [1] "Lambdas for FDH:"

print(lambda(DEA\_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

# DEA under Free Replicability Hull (FRH)  
DEA\_FRH <- dea(x, y, RTS = "add") # Perform DEA with FRH assumption  
print("DEA FRH Results:") # Print FRH results

## [1] "DEA FRH Results:"

print(DEA\_FRH)

## [1] 1 1 1 1 1 1

print("Peers for FRH:") # Print peers for FRH

## [1] "Peers for FRH:"

print(peers(DEA\_FRH))

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

print("Lambdas for FRH:") # Print lambda values for FRH

## [1] "Lambdas for FRH:"

print(lambda(DEA\_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

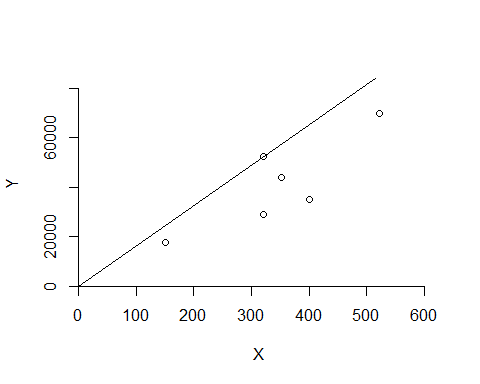
##Peers and lambdas: Peers refer to the other DMUs that serve as benchmarks for a given facility's performance. When a facility is identified as inefficient (i.e., its efficiency score is less than 1), its peers are the facilities that demonstrate similar inputs and outputs but achieve higher efficiency scores. By examining peers, a facility can identify best practices, strategies, and operational adjustments that could improve its own performance. Lambdas are the weights assigned to the peers of a DMU in the DEA model. They quantify how much each peer contributes to the efficiency score of the DMU in question. A lambda value greater than zero indicates that the peer is contributing to the efficiency assessment of the inefficient DMU.  
  
# Summarizing the efficiency results in a data frame  
results <- data.frame(  
 Facility = c("Facility 1", "Facility 2", "Facility 3", "Facility 4", "Facility 5", "Facility 6"),  
 CRS\_Efficiency = DEA\_CRS$eff, # Efficiency results from CRS DEA  
 VRS\_Efficiency = DEA\_VRS$eff, # Efficiency results from VRS DEA  
 DRS\_Efficiency = DEA\_DRS$eff, # Efficiency results from DRS DEA  
 IRS\_Efficiency = DEA\_IRS$eff, # Efficiency results from IRS DEA  
 FDH\_Efficiency = DEA\_FDH$eff, # Efficiency results from FDH DEA  
 FRH\_Efficiency = DEA\_FRH$eff # Efficiency results from FRH DEA  
)  
  
# Print summary of efficiency results  
print("Summary of Efficiency Results:")

## [1] "Summary of Efficiency Results:"

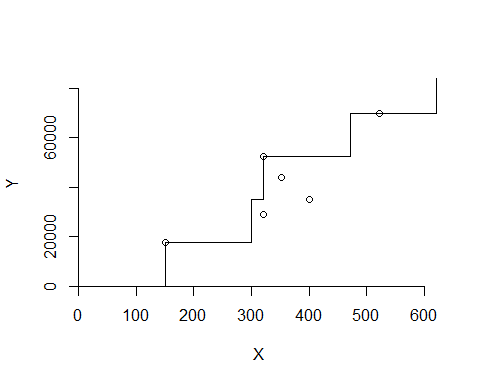
print(results)

## Facility CRS\_Efficiency VRS\_Efficiency DRS\_Efficiency IRS\_Efficiency  
## 1 Facility 1 1.0000000 1.0000000 1.0000000 1.0000000  
## 2 Facility 2 1.0000000 1.0000000 1.0000000 1.0000000  
## 3 Facility 3 1.0000000 1.0000000 1.0000000 1.0000000  
## 4 Facility 4 1.0000000 1.0000000 1.0000000 1.0000000  
## 5 Facility 5 0.9774987 1.0000000 0.9774987 1.0000000  
## 6 Facility 6 0.8674521 0.8963283 0.8674521 0.8963283  
## FDH\_Efficiency FRH\_Efficiency  
## 1 1 1  
## 2 1 1  
## 3 1 1  
## 4 1 1  
## 5 1 1  
## 6 1 1

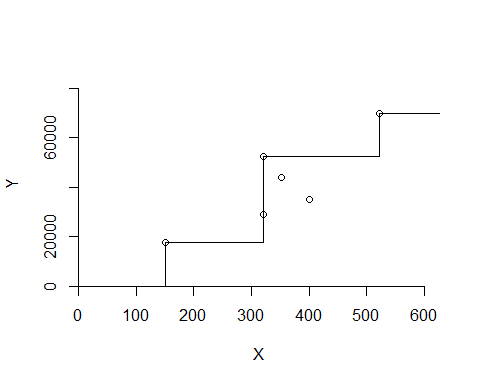
##High Efficiency Across Most Facilities: The efficiency scores indicate that Facilities 1, 2, 3, and 4 achieve perfect efficiency (1.0000) across all DEA models. This suggests that these facilities are operating at optimal levels, effectively utilizing their inputs (staff hours and supply costs) to maximize patient-days, both reimbursed and privately paid.  
  
##Moderate Efficiency for Facility 5: Facility 5 exhibits high efficiency under CRS (0.9775) and DRS (0.9775) but achieves perfect efficiency under VRS and IRS. This discrepancy may indicate that while the facility operates efficiently within its current scale, it may benefit from adjustments in scale or operational practices to enhance its efficiency further.  
  
##Lowest Efficiency for Facility 6: Facility 6 demonstrates the lowest efficiency scores across several models, particularly in CRS (0.8675) and VRS (0.8963). This indicates potential operational inefficiencies and suggests that the facility may need to investigate areas such as staffing levels, supply chain management, or service delivery practices to improve its performance.  
  
##Uniform Performance Under FDH and FRH: All facilities achieve perfect efficiency (1.0000) under both FDH and FRH models. This implies that regardless of the assumption regarding returns to scale, all facilities can produce at least the same output with their given inputs, reinforcing the idea that inefficiencies may be scale-related rather than absolute in nature.  
  
dea.plot(x, y, RTS = "crs")  
  
# DEA Plot under Constant Returns to Scale (CRS)  
dea.plot(x, y, RTS='crs')



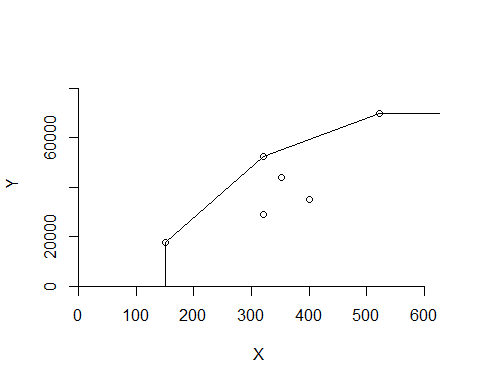
##This plot illustrates the efficiency frontier assuming that increasing inputs will result in a proportional increase in outputs. It helps identify DMUs that maximize their efficiency under this assumption.  
  
# DEA Plot under Free Replicability Hull (FRH)  
dea.plot(x, y, RTS='add') # 'add' specifies FRH in the Benchmarking package



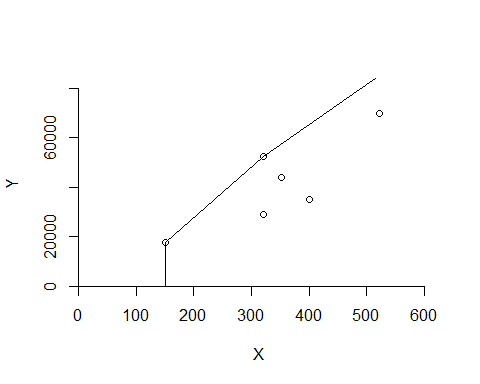
##The FRH plot shows how DMUs can achieve efficiency when inputs can be freely replicated. It highlights the optimal combinations of inputs that lead to efficient output levels, useful for understanding scalability.  
  
# DEA Plot under Free Disposability Hull (FDH)  
dea.plot(x, y, RTS='fdh')



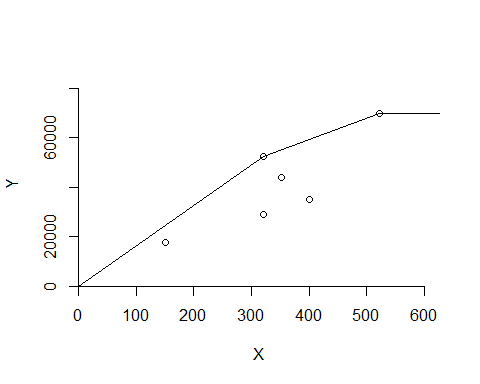
##This plot reflects the concept that outputs can be freely disposed of without affecting input levels. It identifies efficient DMUs and visualizes their performance based on the available resources.  
  
# DEA Plot under Variable Returns to Scale (VRS)  
dea.plot(x, y, RTS='vrs')



##The VRS plot shows the efficiency frontier when the relationship between inputs and outputs is not constant. It provides insights into how DMUs perform under varying scales of operation.  
  
# DEA Plot under Increasing Returns to Scale (IRS)  
dea.plot(x, y, RTS='irs')



##This plot indicates that as inputs increase, outputs increase at an increasing rate. It helps identify DMUs that could benefit from expanding operations to improve efficiency.  
  
# DEA Plot under Decreasing Returns to Scale (DRS)  
dea.plot(x, y, RTS='drs')



##The DRS plot illustrates a situation where increasing inputs leads to a less-than-proportional increase in outputs. It highlights DMUs that may be overextending their resources.  
  
##The DEA analysis provides a robust framework for assessing operational efficiency within healthcare settings. The findings suggest that while many facilities operate efficiently, targeted interventions are necessary for those that do not. Continuous monitoring and improvement strategies should be implemented to maintain high standards of care and resource utilization in nursing homes. This analysis not only serves as a tool for accountability but also as a guide for future investments in operational improvements and resource allocation.