Assignment 5(Quantitative Management Modelling)

Sumit Dutt Mishra

08/11/2021

```
setwd("~/Desktop/SEMESTER 2/QUANT MANAGEMENT/assignment 5")
```

Installing packages

```
library(lpSolveAPI)
library(ucminf)
library(quadprog)
library(Benchmarking)
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.1 ──

## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.5     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.1

## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Computing the Formulation

```
# creating the vectors with our values
input  <-  matrix(c(150,400,320,520,350,  320,  200,  700,  1200,  2000,
1200,  700),ncol  =  2)
output <-
matrix(c(14000,14000,42000,28000,19000,14000,3500,21000,10500,42000,25000,
15000),ncol = 2)
# Assigning column names
colnames(input) <- c("daily_reimbursed_patient",
"daily_privately_paid_patient")
colnames(output) <- c("staff_daily_hours","supplies_daily")

# values of in & out
input

##      daily_reimbursed_patient daily_privately_paid_patient
## [1,]                      150                          200
## [2,]                      400                          700
## [3,]                      320                         1200
## [4,]                      520                         2000
```

```
## [5,]                      350                          1200
## [6,]                      320                           700
```

```
output
```

```
##      staff_daily_hours supplies_daily
## [1,]             14000           3500
## [2,]             14000          21000
## [3,]             42000          10500
## [4,]             28000          42000
## [5,]             19000          25000
## [6,]             14000          15000
```

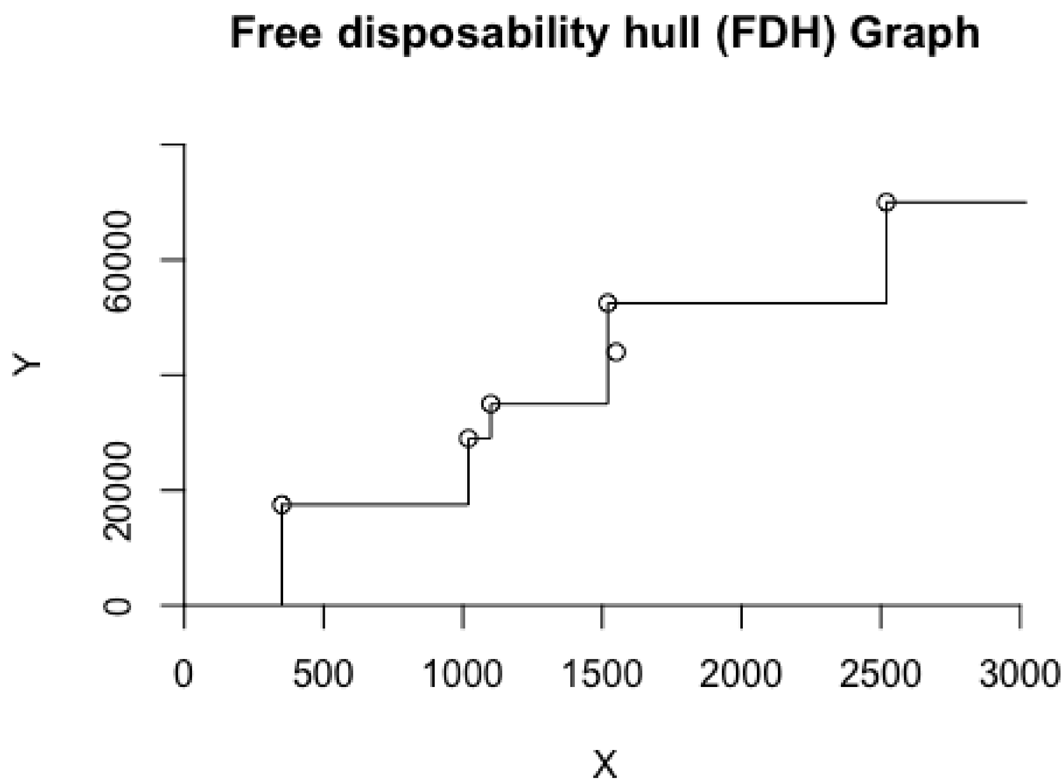
###The results are same as the performance data table from Hope Valley Health Care Association's six nursing facilities.

###Using "DEA" tool that can help organizations to identify and allocate their resources to enhance their efficiency and have better practices and plotting the results

```
analyse_fdh<-  dea(input,output,RTS  =  "fdh")
efficiency_fdh <- as.data.frame(analyse_fdh$eff)
colnames(efficiency_fdh)  <-  c("efficiency_fdh")
peer_fdh  <-  peers(analyse_fdh)
colnames(peer_fdh) <- c("peer1_fdh")
lambda_fdh <- lambda(analyse_fdh)
colnames(lambda_fdh) <- c("L1_fdh", "L2_fdh", "L3_fdh", "L4_fdh", "L5_fdh",
"L6_fdh")
peer_lamb_efficiency_fdh  <-  cbind(peer_fdh,  lambda_fdh,  efficiency_fdh)
peer_lamb_efficiency_fdh
```

```
##   peer1_fdh L1_fdh L2_fdh L3_fdh L4_fdh L5_fdh L6_fdh efficiency_fdh
## 1         1      1      0      0      0      0      0              1
## 2         2      0      1      0      0      0      0              1
## 3         3      0      0      1      0      0      0              1
## 4         4      0      0      0      1      0      0              1
## 5         5      0      0      0      0      1      0              1
## 6         6      0      0      0      0      0      1              1
```

```
dea.plot(input,output,RTS="fdh", main="Free disposability hull (FDH) Graph")
```



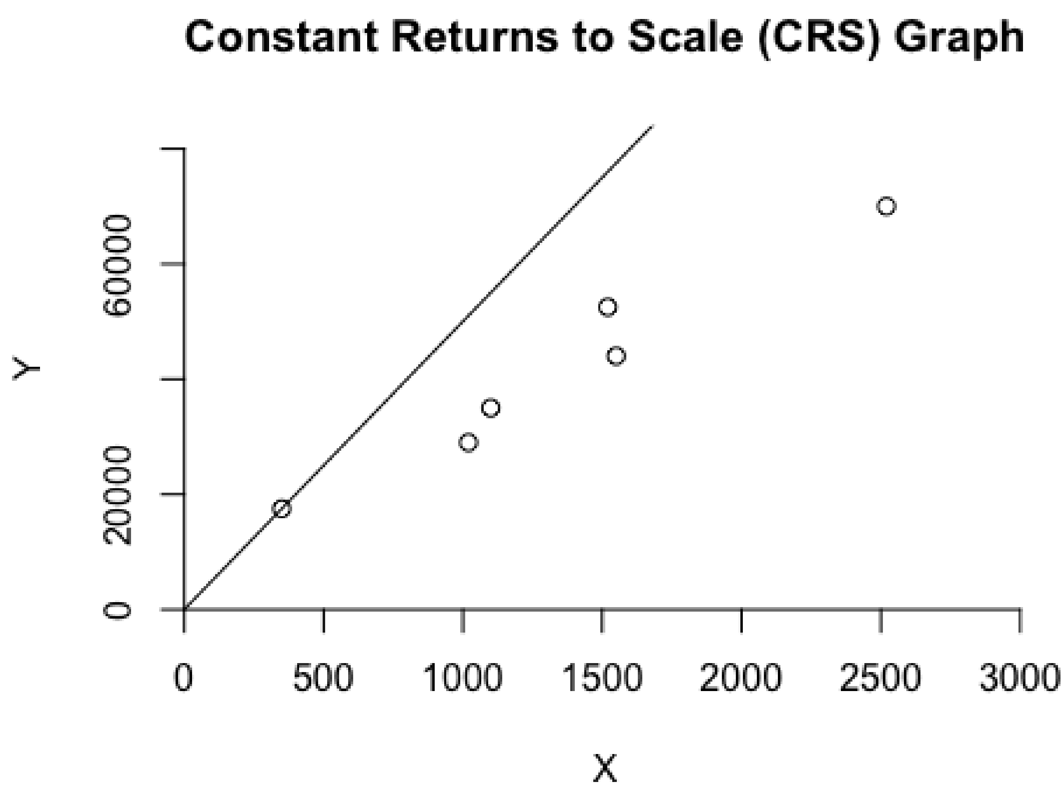
DEA analysis using CRS and plotting the results

```
analyse_crs <- dea(input,output,RTS = "crs")
eff_crs <- as.data.frame(analyse_crs$eff)
colnames(eff_crs) <- c("efficiency_crs")
peer_crs <- peers(analyse_crs)
colnames(peer_crs) <- c("peer1_crs", "peer2_crs", "peer3_crs")
lambda_crs <- lambda(analyse_crs)
colnames(lambda_crs) <- c("L1_crs", "L2_crs", "L3_crs", "L4_crs")
peer_lamb_eff_crs <- cbind(peer_crs, lambda_crs, eff_crs)
peer_lamb_eff_crs

##   peer1_crs peer2_crs peer3_crs    L1_crs     L2_crs L3_crs    L4_crs
## 1         1        NA        NA 1.0000000 0.00000000      0 0.0000000
## 2         2        NA        NA 0.0000000 1.00000000      0 0.0000000
## 3         3        NA        NA 0.0000000 0.00000000      1 0.0000000
## 4         4        NA        NA 0.0000000 0.00000000      0 1.0000000
## 5         1         2         4 0.2000000 0.08048142      0 0.5383307
## 6         1         2         4 0.3428571 0.39499264      0 0.1310751
##   efficiency_crs
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
```

```
## 5      0.9774987
## 6      0.8674521

dea.plot(input,output,RTS="crs", main="Constant Returns to Scale (CRS)
Graph")
```



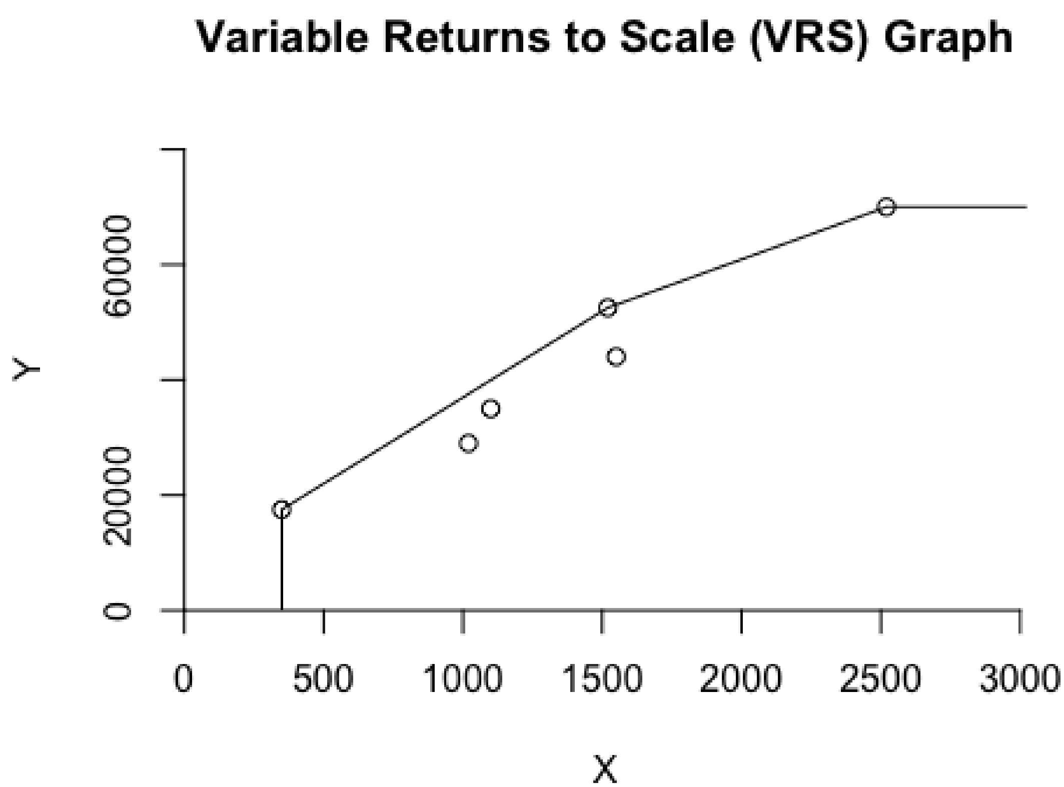
DEA Analysis using VRS and plotting the results

```
analyse_vrs  <-  dea(input,output,RTS  =  "vrs")
eff_vrs <- as.data.frame(analyse_vrs$eff)
colnames(eff_vrs)  <-  c("efficiency_vrs")
peer_vrs <- peers(analyse_vrs)
colnames(peer_vrs) <- c("peer1_vrs", "peer2_vrs", "peer3_vrs")
lambda_vrs <- lambda(analyse_vrs)
colnames(lambda_vrs) <- c("L1_vrs", "L2_vrs", "L3_vrs", "L4_vrs", "L5_vrs")
peer_lamb_eff_vrs <- cbind(peer_vrs, lambda_vrs, eff_vrs)
peer_lamb_eff_vrs

##   peer1_vrs peer2_vrs peer3_vrs    L1_vrs    L2_vrs L3_vrs L4_vrs
L5_vrs
## 1         1        NA        NA 1.0000000 0.0000000      0      0
0.0000000
## 2         2        NA        NA 0.0000000 1.0000000      0      0
0.0000000
```

```
## 3          3        NA        NA 0.0000000 0.0000000      1      0
0.0000000
## 4          4        NA        NA 0.0000000 0.0000000      0      1
0.0000000
## 5          5        NA        NA 0.0000000 0.0000000      0      0
1.0000000
## 6          1         2         5 0.4014399 0.3422606      0      0
0.2562995
##   efficiency_vrs
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
## 5      1.0000000
## 6      0.8963283

dea.plot(input,output,RTS="vrs", main="Variable Returns to Scale (VRS)
Graph")
```



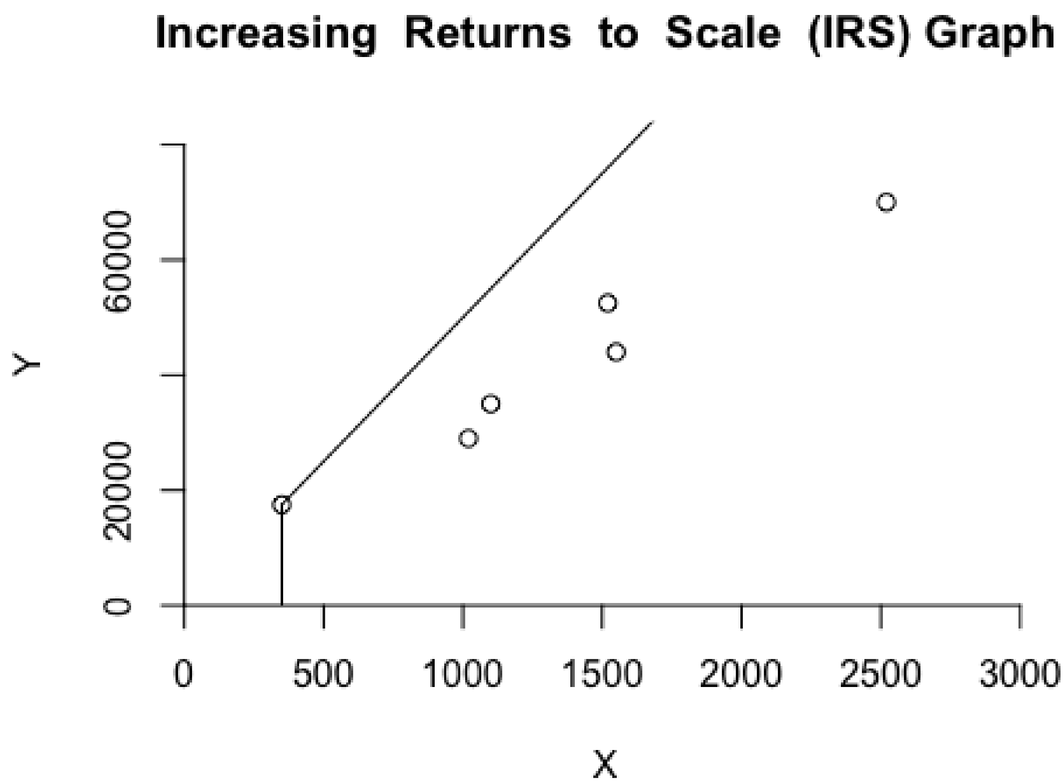
DEA Analysis using IRS and plotting the results

```
analyse_irs  <-  dea(input,output,RTS  =  "irs")
eff_irs <- as.data.frame(analyse_irs$eff)
colnames(eff_irs)  <-  c("efficiency_irs")
```

```
peer_irs <- peers(analyse_irs)
colnames(peer_irs)  <-  c("peer1_irs",  "peer2_irs",  "peer3_irs")
lambda_irs  <-  lambda(analyse_irs)
colnames(lambda_irs) <- c("L1_irs", "L2_irs", "L3_irs", "L4_irs", "L5_irs")
peer_lamb_eff_irs <- cbind(peer_irs, lambda_irs, eff_irs)
peer_lamb_eff_irs

##   peer1_irs peer2_irs peer3_irs    L1_irs    L2_irs L3_irs L4_irs
L5_irs
## 1         1        NA        NA 1.0000000 0.0000000      0      0
0.0000000
## 2         2        NA        NA 0.0000000 1.0000000      0      0
0.0000000
## 3         3        NA        NA 0.0000000 0.0000000      1      0
0.0000000
## 4         4        NA        NA 0.0000000 0.0000000      0      1
0.0000000
## 5         5        NA        NA 0.0000000 0.0000000      0      0
1.0000000
## 6         1         2         5 0.4014399 0.3422606      0      0
0.2562995
##   efficiency_irs
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
## 5      1.0000000
## 6      0.8963283

dea.plot(input,output,RTS="irs",  main="Increasing  Returns  to  Scale  (IRS)
Graph")
```



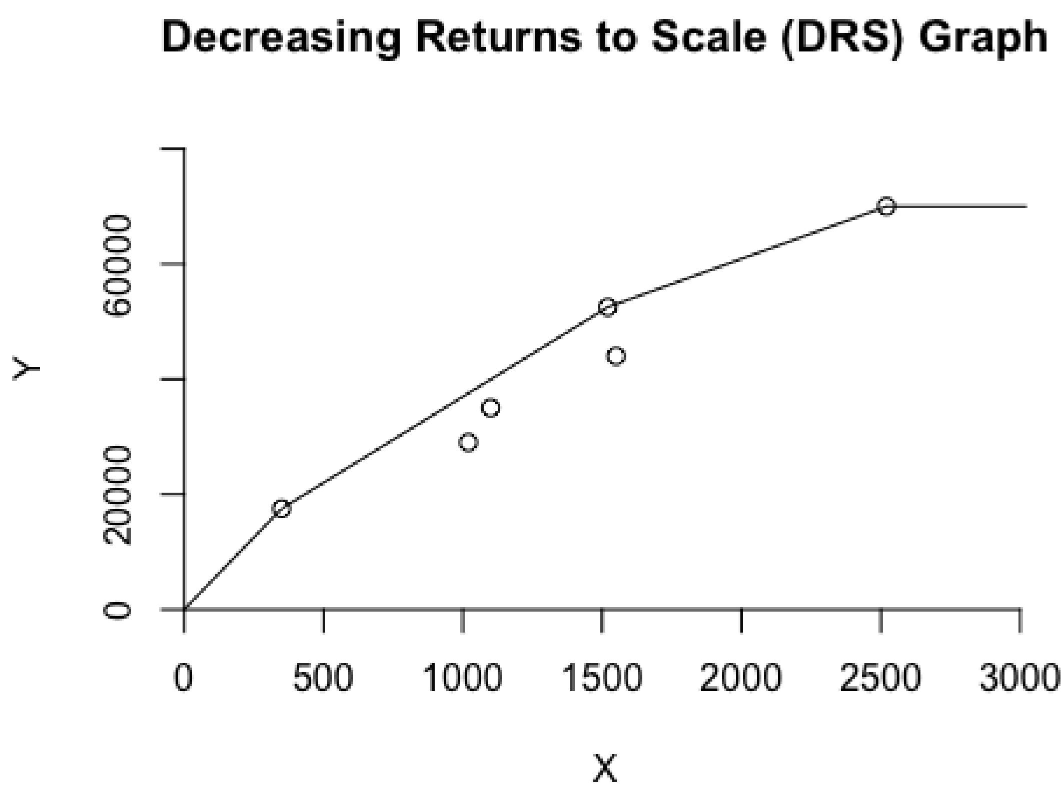
DEA Analysis using DRS and Plotting the result

```
analyse_drs  <-  dea(input,output,RTS  =  "drs")
eff_drs <- as.data.frame(analyse_drs$eff)
colnames(eff_drs)  <-  c("efficiency_drs")
peer_drs <- peers(analyse_drs)
colnames(peer_drs) <- c("peer1_drs", "peer2_drs", "peer3_drs")
lambda_drs <- lambda(analyse_drs)
colnames(lambda_drs) <- c("L1_drs", "L2_drs", "L3_drs", "L4_drs")
peer_lamb_eff_drs <- cbind(peer_drs, lambda_drs, eff_drs)
peer_lamb_eff_drs

##   peer1_drs peer2_drs peer3_drs    L1_drs     L2_drs L3_drs    L4_drs
## 1         1        NA        NA 1.0000000 0.00000000      0 0.0000000
## 2         2        NA        NA 0.0000000 1.00000000      0 0.0000000
## 3         3        NA        NA 0.0000000 0.00000000      1 0.0000000
## 4         4        NA        NA 0.0000000 0.00000000      0 1.0000000
## 5         1         2         4 0.2000000 0.08048142      0 0.5383307
## 6         1         2         4 0.3428571 0.39499264      0 0.1310751
##   efficiency_drs
## 1      1.0000000
## 2      1.0000000
## 3      1.0000000
## 4      1.0000000
```

```
## 5       0.9774987
## 6       0.8674521
```

```
dea.plot(input,output,RTS="drs", main="Decreasing Returns to Scale (DRS)
Graph")
```



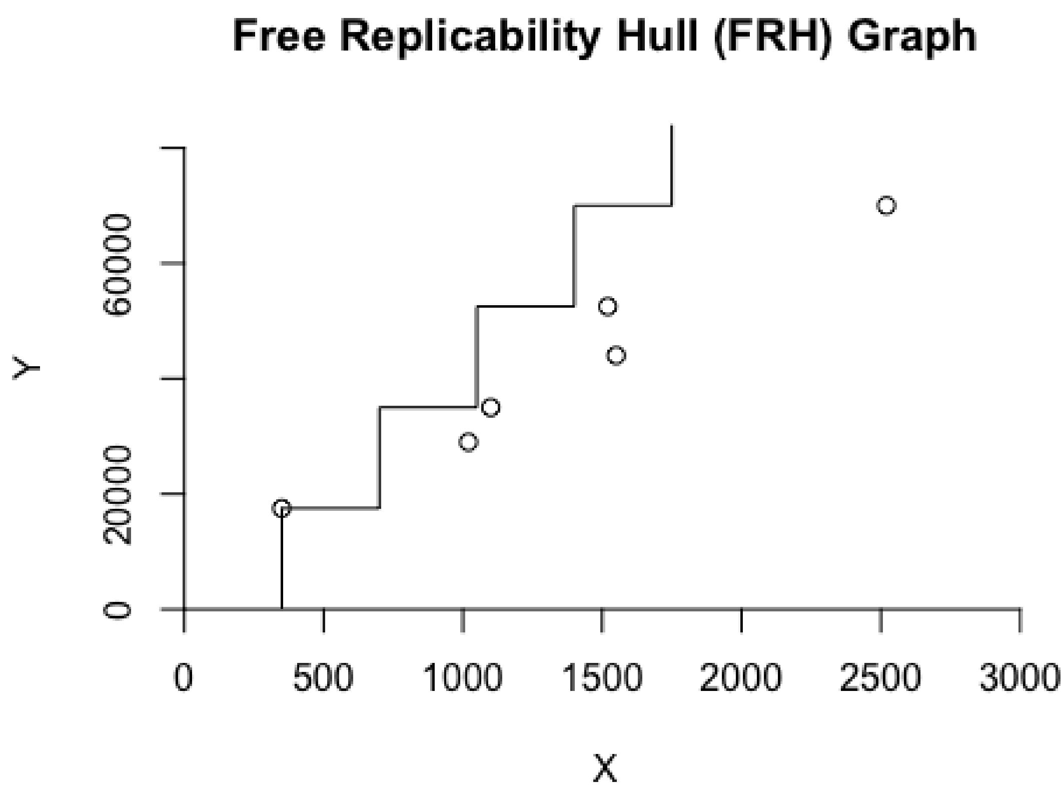
DEA Analysis using FRH and plotting the results

```
analyse_frh  <-  dea(input,output,RTS  =  "add")
eff_frh <- as.data.frame(analyse_frh$eff)
colnames(eff_frh) <- c("efficiency_frh")
peer_frh <- peers(analyse_frh)
colnames(peer_frh) <- c("peer1_frh")
lambda_frh <- lambda(analyse_frh)
colnames(lambda_frh) <- c("L1_frh", "L2_frh", "L3_frh", "L4_frh", "L5_frh",
"L6_frh")
peer_lamb_eff_frh <- cbind(peer_frh, lambda_frh, eff_frh)
peer_lamb_eff_frh
```

```
##   peer1_frh L1_frh L2_frh L3_frh L4_frh L5_frh L6_frh efficiency_frh
## 1         1      1      0      0      0      0      0              1
## 2         2      0      1      0      0      0      0              1
## 3         3      0      0      1      0      0      0              1
## 4         4      0      0      0      1      0      0              1
```

```
## 5          5      0      0      0      0      1      0              1
## 6          6      0      0      0      0      0      1              1
```

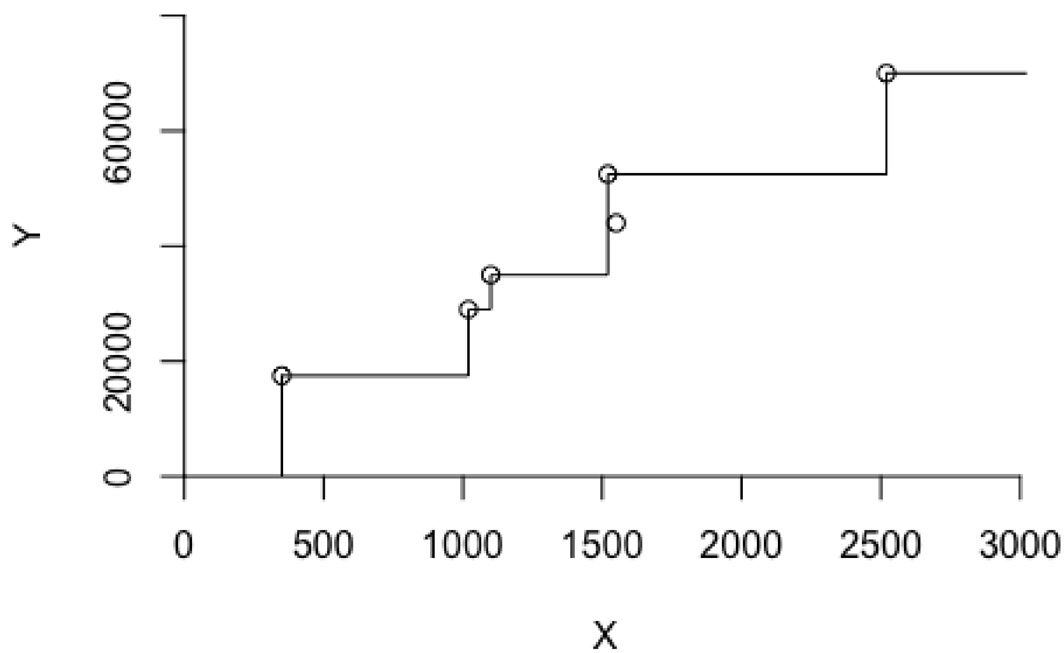
```
dea.plot(input,output,RTS="add", main="Free Replicability Hull (FRH) Graph")
```



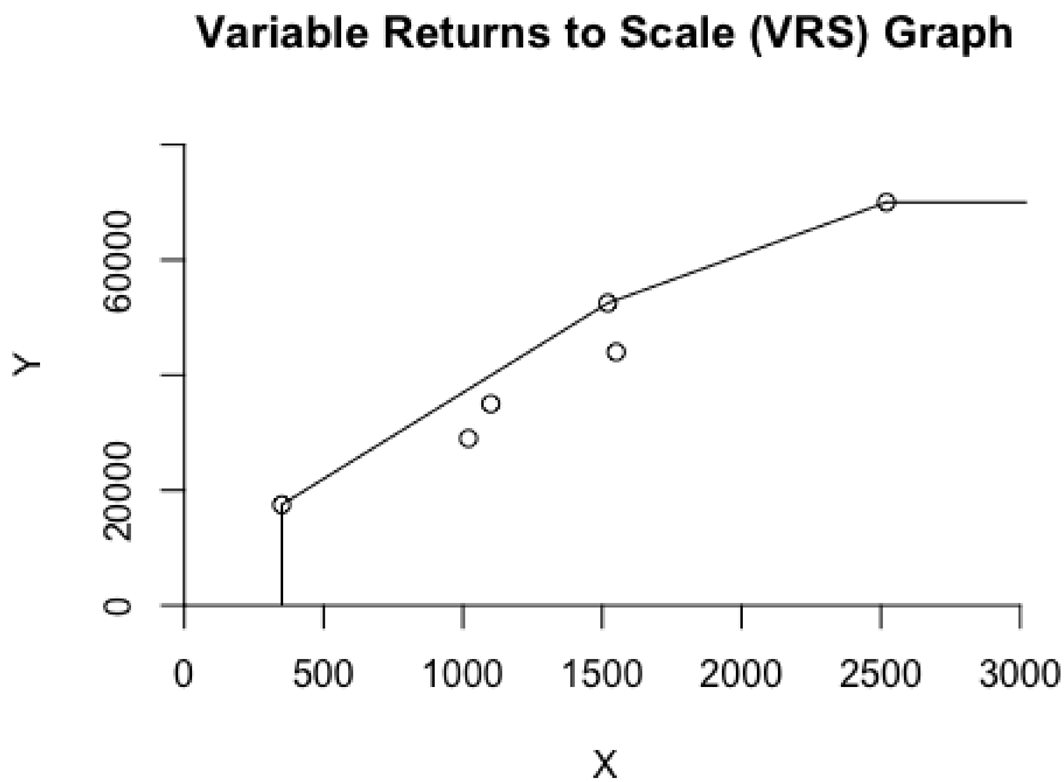
Compare between different assumptions

```
dea.plot(input,output,RTS="fdh", main="Free disposability hull (FDH) Graph")
```

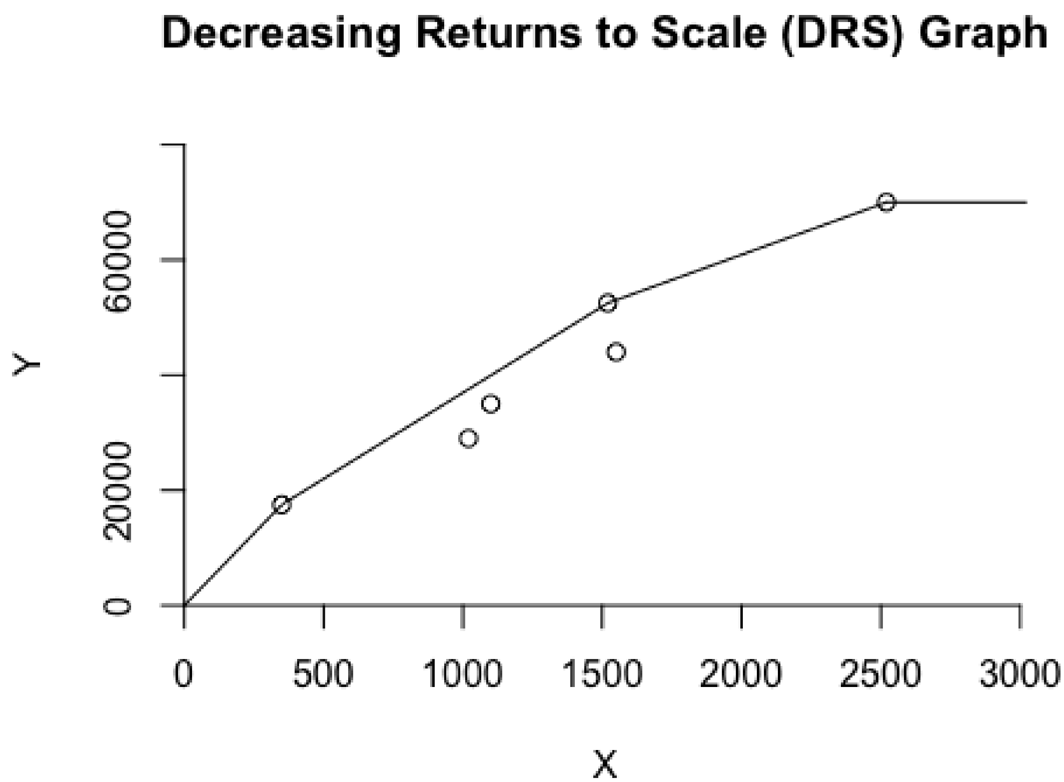
Free disposability hull (FDH) Graph



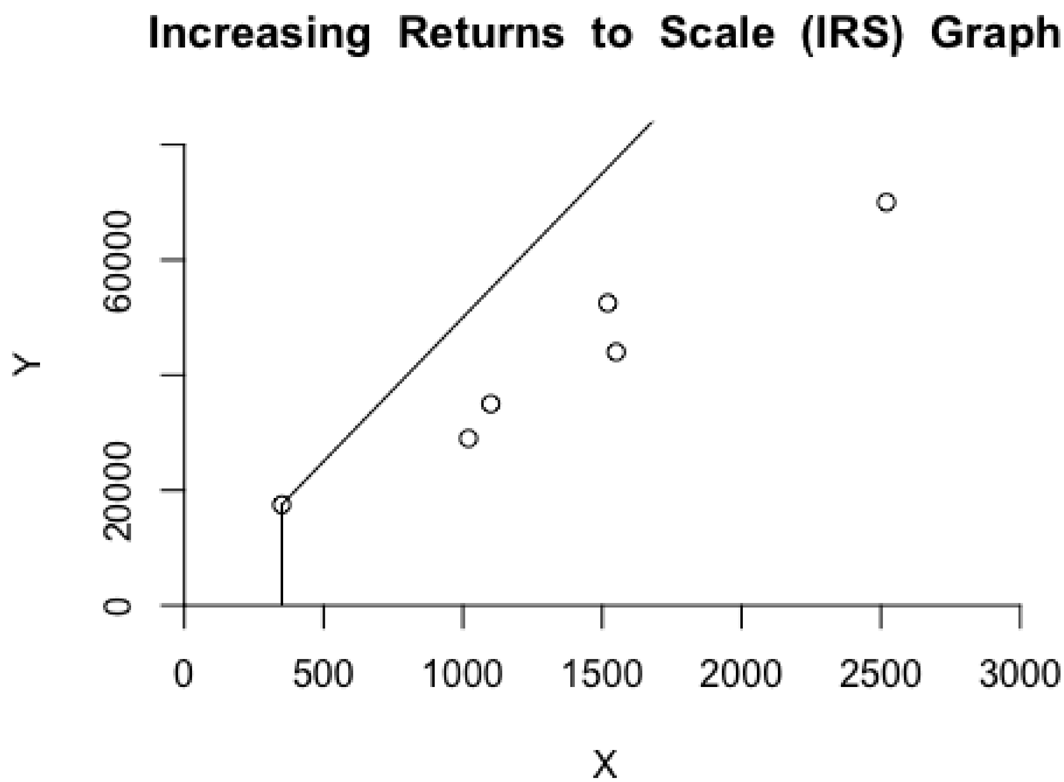

```
dea.plot(input,output,RTS="vrs", main="Variable Returns to Scale (VRS)
Graph")
```



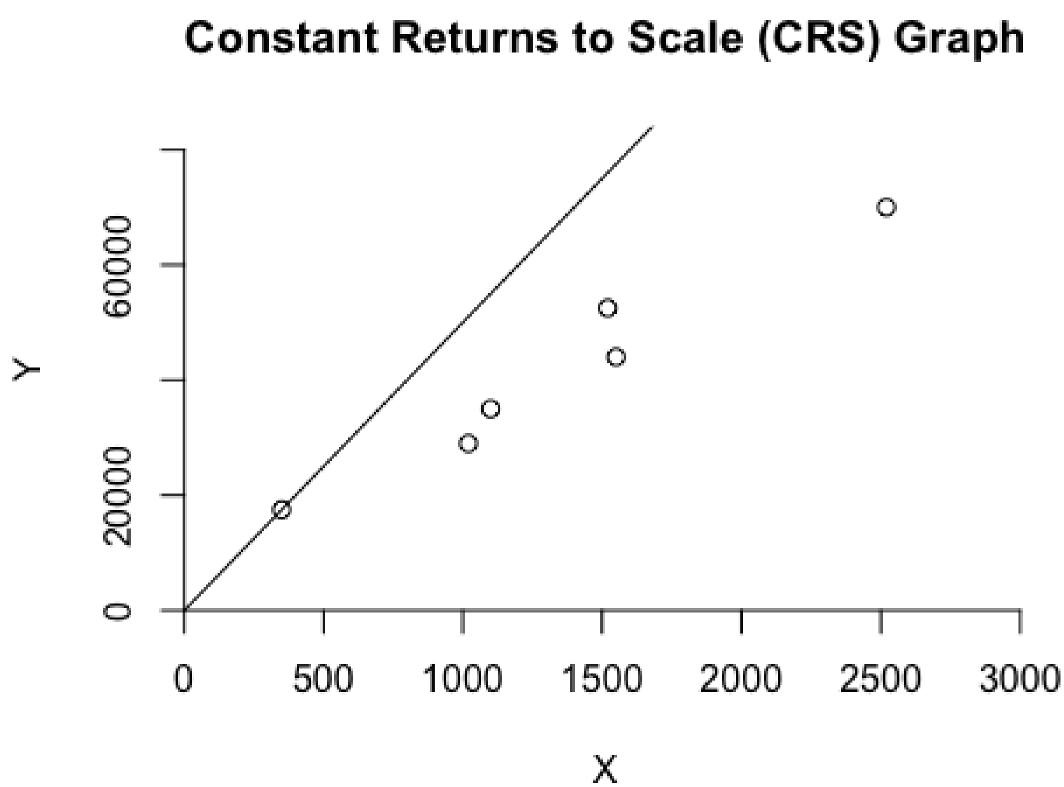

```
dea.plot(input,output,RTS="drs", main="Decreasing Returns to Scale (DRS)
Graph")
```




```
dea.plot(input,output,RTS="irs",  main="Increasing  Returns  to  Scale  (IRS)
Graph")
```

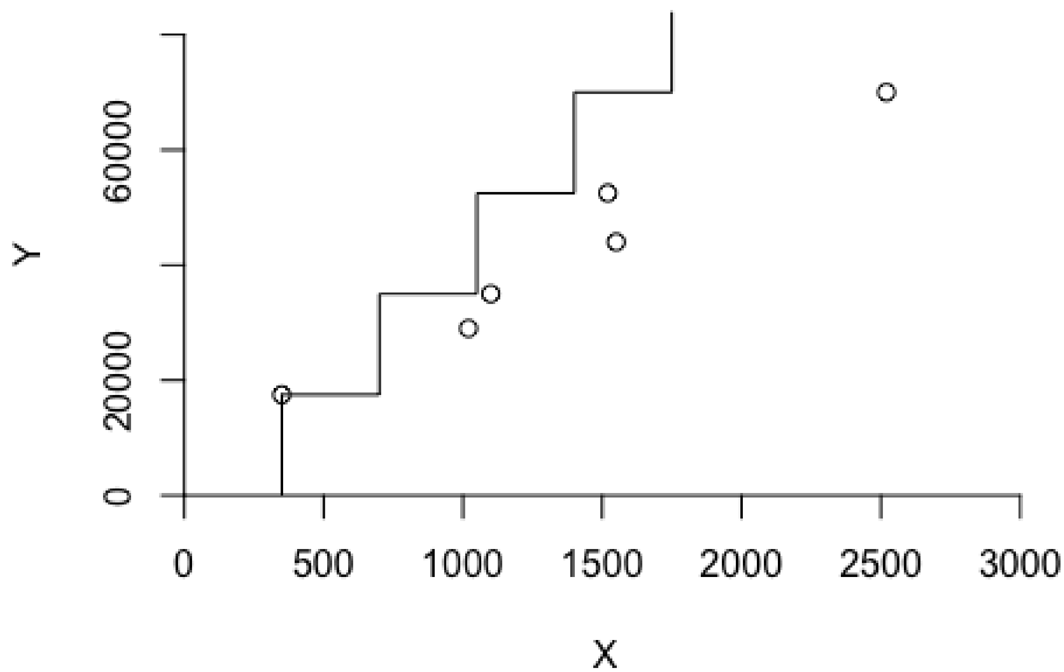



```
dea.plot(input,output,RTS="crs", main="Constant Returns to Scale (CRS)
Graph")
```




```
dea.plot(input,output,RTS="add", main="Free Replicability Hull (FRH) Graph")
```

Free Replicability Hull (FRH) Graph



##Summary ####1)The premise of estimating the technology using a minimum extrapolation technique is shared by all DEA model.

####2)FDH is the smallest technology set, It creates fewer outputs (number of patient days reimbursed by third-party sources and number of patient days reimbursed privately) with more inputs (number of patient days reimbursed by third-party sources and number of patient days reimbursed privately) . FDH is the most popular model among businesses, yet it has several flaws owing to its assumptions. The efficiencies in this model is 1, when we compare it to other models it is not more efficient.

####3)VRS is larger than FDH because it “fills-out” the spaces that FDH reduced.It is clearly visible that unit 6 can improve its efficiency.

####4)According to the graphs DRS and IRS are larger than VRS. For smaller input values, #####DRS: enlarge the set #####IRS: try to raise the technology. #####Units 5 and 6 might increase their efficiency, according to DRS, and facility 6 could improve as well, according to IRS.

####5)CRS is the largest technology set. It allows to assess if there are any scaling up or down combinations.Improvement is required based on the efficiency numbers for unit 5 and 6.

####6)FRH is larger than FDH but smaller than CRS. The main job for FRH is to replace deterministic data with random variables.

##2) Research and Development Division of Emax Corporation

Objective function:

Max: 20 X1 + 15 X2 + 25 X3 - 6 Y1P - 6 Y1M - 3 Y2M Subject to Employment Level 6x1 + 4x2 + 5x3 - (Y1P - Y1M) = 50

Earnings Next Year

8x1 + 7x2 + 5x3 - (Y2P - Y2M) = 75

Non-negativity contraint

X1, X2, X3 >= 0 Y1P, Y1M, Y2P, Y2M >= 0

```
library(lpSolveAPI)
setwd("~/Desktop/SEMESTER 2/QUANT MANAGEMENT/assignment 5")
```

###Loading the data

```
first_e_max <- read.lp("firstemax.lp")
```

###Solving lp

```
solve(first_e_max)

## [1] 0
```

####we can see, the solver is returning 0, which clearly indicates that it is finding a solution.

```
get.objective(first_e_max)

## [1] 225
```

####To maximizing profit while reducing other company goals such as manpower and profits. The penalty for failing to meet the goals is 225.

```
get.variables(first_e_max)

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

get.constraints(first_e_max)

## [1] 50 75
```

####Initially when we solved this we got a deficit of 25000 i,e 25000 people were hired. In order to neutralize it , I have added some higher variable i.e. 2500

```
setwd("~/Desktop/SEMESTER 2/QUANT MANAGEMENT/assignment 5")
```

####Loading the data

```
E_Max <-  read.lp("emax.lp")
E_Max

## Model name:
##               X1     X2     X3    Y1P    Y1M    Y2M    Y2P
## Maximize      20     15     25  -2500     -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
```

###Solving Lp

```
solve(E_Max)

## [1] 0
```

####Getting Objective

```
get.objective(E_Max)

## [1] 208.3333

get.variables(E_Max)

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

get.constraints(E_Max)

## [1] 50 75
```

####Order of the variables in the objective function. X1 = 0,X2 = 8.33,X3 = 3.33,Y1P = 0,Y1M = 0,Y2M = 0,Y2P = 0 clearly depicting that all the expectations are fully satisfied. Hence we can move forward with this.