

DEA Analysis – Hope Valley Health Care Association

Jaipreet Kaur

Note: This file implements the *Module 8 – DEA (Hope Valley)* dataset and includes CRS, VRS, IRS, DRS, FDH, and FRH models with peers and lambdas. Orientation and packages are globally defined so the file knits cleanly.

1. Setup

```
# Define orientation globally
orientation <- "in" # or "out" if professor requires output-oriented analysis

# Install and load required packages
need <- c("Benchmarking", "dplyr", "tidyverse", "stringr", "purrr")
to_install <- setdiff(need, rownames(installed.packages()))
if (length(to_install)) install.packages(to_install, repos = "https://cloud.r-project.org")

library(Benchmarking)
library(dplyr)
library(tidyverse)
library(tibble)
library(stringr)
library(purrr)

# Helper functions
get_lambda <- function(obj) {
  if (!is.null(obj$lambda)) return(obj$lambda)
  if (!is.null(obj$Lambda)) return(obj$Lambda)
  if ("lambdas" %in% getNamespaceExports("Benchmarking")) {
    tryCatch(Benchmarking::lambdas(obj), error = function(e) NULL)
  } else NULL
}

get_peers <- function(lambda_mat, dmu_names) {
  if (is.null(lambda_mat)) return(rep(NA_character_, length(dmu_names)))
  lambda_mat <- as.matrix(lambda_mat)
  out <- vector("character", ncol(lambda_mat))
  for (j in seq_len(ncol(lambda_mat))) {
    idx <- which(lambda_mat[, j, drop = TRUE] > 1e-9)
    if (length(idx)) {
      out[j] <- paste(dmu_names[idx], collapse = ", ")
    } else {
      out[j] <- NA_character_
    }
  }
}
```

```

    out
}

```

2. Data (Hope Valley Health Care Association)

```

# Inputs: Staff Hours and Supplies per Day
# Outputs: Reimbursed and Privately Paid Patient-Days

hope_valley <- tribble(
  ~Facility,      ~StaffHours, ~Supplies, ~Reimbursed, ~Private,
  "Facility 1",   150,        0.2,       14000,     3500,
  "Facility 2",   400,        0.7,       14000,     21000,
  "Facility 3",   320,        1.2,       42000,     10500,
  "Facility 4",   520,        2.0,       28000,     42000,
  "Facility 5",   350,        1.2,       19000,     25000,
  "Facility 6",   320,        0.7,       14000,     15000
)

hope_valley

X <- as.matrix(hope_valley %>% select(StaffHours, Supplies))
Y <- as.matrix(hope_valley %>% select(Reimbursed, Private))
dmu <- hope_valley$Facility

```

3. DEA Models (Input-Oriented)

```

# CRS / VRS / IRS / DRS
dea_crs <- dea(X, Y, RTS = "crs", ORIENTATION = orientation)
dea_vrs <- dea(X, Y, RTS = "vrs", ORIENTATION = orientation)
dea_irs <- dea(X, Y, RTS = "irs", ORIENTATION = orientation)
dea_drs <- dea(X, Y, RTS = "drs", ORIENTATION = orientation)

# FDH / FRH (if available)
fdh_fit <- tryCatch(fdh(X, Y, ORIENTATION = orientation), error = function(e) e)
frh_fit <- tryCatch(frh(X, Y, ORIENTATION = orientation), error = function(e) e)

# Efficiency summary
eff_tbl <- tibble(
  Facility = dmu,
  Eff_CRS  = eff(dea_crs),
  Eff_VRS  = eff(dea_vrs),
  Eff_IRS  = eff(dea_irs),
  Eff_DRS  = eff(dea_drs),
  Eff_FDH  = if (inherits(fdh_fit, "error")) NA_real_ else eff(fdh_fit),
  Eff_FRH  = if (inherits(frh_fit, "error")) NA_real_ else eff(frh_fit)
)

eff_tbl

```

4. Peers and Lambdas

```
lam_crs <- get_lambda(dea_crs)
lam_vrs <- get_lambda(dea_vrs)
lam_irs <- get_lambda(dea_irs)
lam_drs <- get_lambda(dea_drs)
lam_fdh <- if (!inherits(fdh_fit, "error")) get_lambda(fdh_fit) else NULL
lam_frh <- if (!inherits(frh_fit, "error")) get_lambda(frh_fit) else NULL

peers_tbl <- tibble(
  Facility = dmu,
  Peers_CRS = get_peers(lam_crs, dmu),
  Peers_VRS = get_peers(lam_vrs, dmu),
  Peers_IRS = get_peers(lam_irs, dmu),
  Peers_DRS = get_peers(lam_drs, dmu),
  Peers_FDH = if (!is.null(lam_fdh)) get_peers(lam_fdh, dmu) else NA_character_,
  Peers_FRH = if (!is.null(lam_frh)) get_peers(lam_frh, dmu) else NA_character_
)

peers_tbl
```

5. Combined Summary Table

```
summary_tbl <- eff_tbl %>% left_join(peers_tbl, by = "Facility")
summary_tbl
```

6. Comparison and Interpretation

```
efficient_by_model <- eff_tbl %>%
  pivot_longer(cols = starts_with("Eff_"),
               names_to = "Model", values_to = "Efficiency") %>%
  group_by(Model) %>%
  summarize(
    Efficient_DMUs = paste(Facility[which(abs(Efficiency - 1) < 1e-9)], collapse = ", "),
    .groups = "drop"
  )
efficient_by_model
```

Narrative Guide (Edit for your submission): - **CRS vs VRS:** VRS scores are usually CRS. Differences imply scale inefficiency. - **IRS vs DRS:** If IRS efficiency > DRS, larger scale may be beneficial; the opposite indicates possible diseconomies of scale. - **FDH / FRH:** FDH is nonconvex and usually produces equal or higher efficiencies. FRH sits between VRS and FDH. - **Peers:** The peer set identifies the efficient units serving as benchmarks. Multiple peers suggest composite targets.

7. Reproducibility

```
# Orientation and environment details  
orientation  
sessionInfo()
```