Package ‘deaR’

February 24, 2026

Type Package

Title Conventional and Fuzzy Data Envelopment Analysis

Version 1.5.4

Maintainer Vicente Bolos <vicente.bolos@uv.es>

Description Set of functions for Data Envelopment Analysis, including classical, fuzzy, cross-efficiency, bootstrapping, and Malmquist models. See: Banker, R.; Charnes, A.; Cooper, W.W. (1984). <doi:10.1287/mnsc.30.9.1078>, Charnes, A.; Cooper, W.W.; Rhode
2217(78)90138-8> and Charnes, A.; Cooper, W.W.; Rhodes, E. (1981). <doi:10.1287/mnsc.27.6.668>.

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.5)

Imports lpSolve, ggplot2, tidyr, plotly, igraph, writexl, dplyr, methods, gridExtra, grDevices, optiSolve

NeedsCompilation no

Author Vicente Bolos [aut, cre],
Vicente Coll-Serrano [aut],
Rafael Benitez Suarez [aut]

Repository CRAN

Date/Publication 2026-02-24 12:00:02 UTC

Contents

`Airlines` *Data: Coelli, Grifell-Tatje, and Perelman (2002).*

Description

Data of 28 airlines with 2 outputs and 4 inputs.

Usage

```
data("Airlines")
```

Format

Data frame with 28 rows and 7 columns. Definition of outputs (Y) and inputs (X):

y1 = Pass Passenger-kilometers flown

y2 = Cargo Freight tonne-kilometers flown

x1 = Lab Labor (number of employees)

x2 = Fuel Fuel (millions of gallons)

x3 = Matl Other inputs (millions of U.S. dollar equivalent) consisting of operating and maintenance expenses excluding labor and fuel expenses

x4 = Cap Capital (sum of the maximum takeoff weights of all aircraft flown multiplied by the number of days flown)

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Coelli, T.; Griffel-Tatje, E.; Perelman, S. (2002). "Capacity Utilization and Profitability: A Decomposition of Short-Run Profit Efficiency", International Journal of Production Economics 79, 261–278.

See Also

make_deadata, model_sbmeff

Examples

```
# Example. Replication of results in Aparicio et al. (2007).
data("Airlines")
data_example <- make_deadata(Airlines,
                             inputs = 4:7,
                             outputs = 2:3)
result <- model_sbmeff(data_example)
efficiencies(result)
result2 <- model_sbmeff(data_example,
                        kaizen = TRUE)
efficiencies(result2)
```

bootstrap_basic *Bootstrapping DEA*

Description

To bootstrap efficiency scores, deaR uses the algorithm proposed by Simar and Wilson (1998). For now, the function bootstrap_basic can only be used with basic DEA models.

Usage

```
bootstrap_basic(datadea,
                orientation = c("io", "oo"),
                rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
                L = 1,
                U = 1,
                B = 2000,
                h = NULL,
                alpha = 0.05)
```

Arguments

datadea	A deadata object with n DMUs, m inputs and s outputs.
orientation	A string, equal to "io" (input oriented) or "oo" (output oriented).
rts	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).
L	Lower bound for the generalized returns to scale (grs).
U	Upper bound for the generalized returns to scale (grs).
B	Number of bootstrap iterations.
h	Bandwidth of smoothing window. By default h = 0.014 (you can set h equal to any other value). The optimal bandwidth factor can also be calculated following the proposals of Silverman (1986) and Daraio y Simar (2007). So, h = "h1" is the optimal h referred as "robust normal-reference rule" (Daraio and Simar, 2007 p.60), h = "h2" is the value of h1 but instead of the factor 1.06 with the factor 0.9, h = "h3" is the value of h1 adjusted for scale and sample size (Daraio and Simar, 2007 p.61), and h = "h4" is the bandwidth provided by a Gaussian kernel density estimate.
alpha	Between 0 and 1 (for confidence intervals).

Value

A list with the results (score and score_bc components) along with any other necessary information to replicate the results, such as the name of the model and parameters orientation, rts, bandwidth, bias, CI (confidence intervals) and descriptives.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Behr, A. (2015). Production and Efficiency Analysis with R. Springer.

Bogetoft, P.; Otto, L. (2010). Benchmarking with DEA, SFA, and R. Springer.

Daraio, C.; Simar, L. (2007). Advanced Robust and Nonparametric Methods in Efficiency Analysis: Methodology and Applications. New York: Springer.

Färe, R.; Grosskopf, S.; Kokkenlenberg, E. (1989). "Measuring Plant Capacity, Utilization and Technical Change: A Nonparametric Approach". International Economic Review, 30(3), 655-666.

Löthgren, M.; Tambour, M. (1999). "Bootstrapping the Data Envelopment Analysis Malmquist Productivity Index". Applied Economics, 31, 417-425.

Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis. London: Chapman and Hall.

Simar, L.; Wilson, P.W. (1998). "Sensitivity Analysis of Efficiency Scores: How to Bootstrap in Nonparametric Frontier Models". Management Science, 44(1), 49-61.

Simar, L.; Wilson, P.W. (1999). "Estimating and Bootstrapping Malmquist Indices". European Journal of Operational Research, 115, 459-471.

Simar, L.; Wilson, P.W. (2008). Statistical Inference in Nonparametric Frontier Models: Recent Developments and Perspective. In H.O. Fried; C.A. Knox Lovell and S.S. Schmidt (eds.) The Measurement of Productive Efficiency and Productivity Growth. New York: Oxford University Press. doi:10.1093/acprof:oso/9780195183528.001.0001

Examples

```
# To replicate the results in Simar y Wilson (1998, p. 58) you have to
# set B=2000 (in the example B = 100 to save time)
data("Electric_plants")
data_example <- make_deadata(Electric_plants,
                             ni = 3,
                             no = 1)
result <- bootstrap_basic(datadea = data_example,
                             orientation = "io",
                             rts = "vrs",
                             B = 100)
result$score_bc
result$CI
```

Coelli_1998 *Data: Coelli, Rao and Battese (1998).*

Description

Data of five DMUs with two inputs and one output. Prices for inputs are available. Price for output is not from Coelli et al. (1998).

Usage

```
data("Coelli_1998")
```

Format

Data frame with 6 rows and 5 columns. Definition of inputs (X) and outputs (Y):

Input1 Input 1

Input2 Input 2

Output Output

Price_input1 Price input 1

Price_input2 Price input 2

Price_output Price output

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Coelli, T.; Prasada Rao, D.S.; Battese, G.E. An introduction to efficiency and productivity analysis. Boston: Kluwer Academic Publishers.

See Also

make_deadata

Examples

```
# Example. Replication of results in Coelli et al. (1998, p.166).
# Cost efficiency model.
data("Coelli_1994")
# Selection of prices: data_prices is the trasnpose where the prices for inputs are.
data_prices <- t(Coelli_1998[, 5:6])

data_example <- make_deadata(Coelli_1998,
                             dmus = 1,
                             ni = 2,
                             no = 1)
result <- model_profit(data_example,
                       price_input = data_prices,
                       rts = "crs",
                       restricted_optimal = FALSE)
# notice that the option by default is restricted_optimal=TRUE
```

```
efficiencies(result)
```

`Coll_Blasco_2006` *Data: Coll and Blasco (2006).*

Description

Data of six authorized dealers with two inputs and two outputs.

Usage

```
data("Coll_Blasco_2006")
```

Format

Data frame with 6 rows and 5 columns. Definition of inputs (X) and outputs (Y):

x1 = Employees Number of employees

x2 = Capital Impairment of assets

y1 = Vehicles Number of vehicles sold

y2 = Orders Number of orders received at the garage

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Coll-Serrano, V.; Blasco-Blasco, O. (2006). Evaluacion de la Eficiencia mediante el Análisis Envolvente de Datos. Introduccion a los Modelos Básicos.

See Also

make_deadata

Examples

```
# Example. How to read data with deaR
data("Coll_Blasco_2006")
data_example <- make_deadata(Coll_Blasco_2006,
                             dmus = 1,
                             ni = 2,
                             no = 2)
```

`cross_efficiency` *Cross efficiency analysis*

Description

Computes arbitrary, benevolent and aggressive formulations of cross-efficiency under any returns-to-scale. Doyle and Green (1994) present three alternatives ways of formulating the secondary goal (wich will minimize or maximize the other DMUs' cross-efficiencies in some way). Methods II and III are implemented in deaR with any returns-to-scale. The maverick index is also calculated.

Usage

```
cross_efficiency(datadea,
                 dmu_eval = NULL,
                 dmu_ref = NULL,
                 epsilon = 0,
                 orientation = c("io", "oo"),
                 rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
                 L = 1,
                 U = 1,
                 selfapp = TRUE,
                 correction = FALSE,
                 M2 = TRUE,
                 M3 = TRUE)
```

Arguments

`datadea` An object of class `dea` or `deadata`. If it is of class `dea` it must have been obtained with some of the multiplier DEA models.

`dmu_eval` A numeric vector. Only the multipliers of DMUs in `dmu_eval` are computed. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`epsilon` Numeric, multipliers must be >= `epsilon`.

`orientation` A string, equal to "io" (input-oriented) or "oo" (output-oriented).

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`selfapp` Logical. If it is `TRUE`, self-appraisal is included in the average scores of `A` and `e`.

`correction` Logical. If it is `TRUE`, a correction is applied in the "vrs" input-oriented model in order to avoid negative cross-efficiencies, according to Lim & Zhu (2015).

M2	Logical. If it is TRUE, it computes Method II for aggresive/benevolent estimations.
M3	Logical. If it is TRUE, it computes Method III for aggresive/benevolent estimations.

Value

A list with the results for each method (Arbitrary, M2_agg, M2_ben, M3_agg and M3_ben components), along with any other necessary information to replicate the results, such as the name of the model and parameters orientation, rts, selfapp, dmu_eval and dmu_ref.

Note

(1) We can obtain negative cross-efficiency in the input-oriented DEA model under no constant returns-to-scale. However, the same does not happen in the case of the output-oriented VRS DEA model. For this reason, the proposal of Lim and Zhu (2015) is implemented in deaR to calculate the input-oriented cross-efficiency model under no constant returns-to-scale.

(2) The multiplier model can have alternate optimal solutions (see note 1 in model_multiplier). So, depending on the optimal weights selected we can obtain different cross-efficiency scores.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Sexton, T.R., Silkman, R.H.; Hogan, A.J. (1986). Data envelopment analysis: critique and extensions. In: Silkman RH (ed) Measuring efficiency: an assessment of data envelopment analysis, vol 32. Jossey-Bass, San Francisco, pp 73–104. doi:10.1002/ev.1441

Doyle, J.; Green, R. (1994). “Efficiency and cross efficiency in DEA: derivations, meanings and the uses”, Journal of Operational Research Society, 45(5), 567–578. doi:10.2307/2584392

Cook, W.D.; Zhu, J. (2015). DEA Cross Efficiency. In: Zhu, J. (ed) Data Envelopment Analysis. A Handbook of Models and Methods. International Series in Operations Research & Management Science, vol 221. Springer, Boston, MA, 23-43. doi:10.1007/9781489975539_2

Lim, S.; Zhu, J. (2015). "DEA Cross-Efficiency Under Variable Returns to Scale". Journal of Operational Research Society, 66(3), p. 476-487. doi:10.1057/jors.2014.13

See Also

model_multiplier, cross_efficiency_fuzzy

Examples

```
# Example 1.
# Arbitrary formulation. Input-oriented model under constant returns-to-scale.
data("Golany_Roll_1989")
data_example <- make_deadata(datadea = Golany_Roll_1989,
                             inputs = 2:4,
                             outputs = 5:6)
result <- cross_efficiency(data_example,
                           orientation = "io",
                           rts = "crs",
                           selfapp = TRUE)
result$Arbitrary$cross_eff
result$Arbitrary$e

# Example 2.
# Benevolent formulation (method II). Input-oriented.
data("Golany_Roll_1989")
data_example <- make_deadata(datadea = Golany_Roll_1989,
                             inputs = 2:4,
                             outputs = 5:6)
result <- cross_efficiency(data_example,
                           orientation = "io",
                           selfapp = TRUE)
result$M2_ben$cross_eff
result$M2_ben$e

# Example 3.
# Benevolent formulation (method III). Input-oriented.
data("Golany_Roll_1989")
data_example <- make_deadata(datadea = Golany_Roll_1989,
                             inputs = 2:4,
                             outputs = 5:6)
result <- cross_efficiency(data_example,
                           orientation = "io",
                           selfapp = TRUE)
result$M3_ben$cross_eff
result$M3_ben$e

# Example 4.
# Arbitrary formulation. Output-oriented.
data("Golany_Roll_1989")
data_example <- make_deadata(datadea = Golany_Roll_1989,
                             inputs = 2:4,
                             outputs = 5:6)
result <- cross_efficiency(data_example,
                           orientation = "oo",
                           selfapp = TRUE)
result$Arbitrary$cross_eff
result$Arbitrary$e

# Example 5.
# Arbitrary formulation. Input-oriented model under vrs returns-to-scale.
```

```
data("Lim_Zhu_2015")
data_example <- make_deadata(Lim_Zhu_2015,
                             ni = 1,
                             no = 5)
cross <- cross_efficiency(data_example,
                          epsilon = 0,
                          orientation = "io",
                          rts = "vrs",
                          selfapp = TRUE,
                          M2 = FALSE,
                          M3 = FALSE)
cross$Arbitrary$e
```

cross_efficiency_fuzzy *Cross efficiency fuzzy analysis*

Description

Computes the cross-efficiency fuzzy tables from DEA fuzzy data or a Guo-Tanaka DEA model solution. The (crisp) relative efficiencies for the case h = 1 are obtained from the CCR model (model_multiplier).

Usage

```
cross_efficiency_fuzzy(datadea,
                       orientation = c("io", "oo"),
                       h = 1,
                       selfapp = TRUE)
```

Arguments

datadea	An object of class dea_fuzzy or deadata_fuzzy. If it is of class dea_fuzzy it must have been obtained with modelfuzzy_guotanaka.
orientation	A string, equal to "io" (input-oriented) or "oo" (output-oriented).
h	A numeric vector with the h-levels (in [0,1]).
selfapp	Logical. If it is TRUE, self-appraisal is included in the average scores of A and e.

Value

A list with the results along with any other necessary information to replicate the results, such as the name of the model and parameters orientation, rts, selfapp and h.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Doyle, J.; Green, R. (1994). “Efficiency and Cross Efficiency in DEA: Derivations, Meanings and the Uses”, Journal of Operational Research Society, 45(5), 567–578. doi:10.2307/2584392

Guo, P.; Tanaka, H. (2001). "Fuzzy DEA: A Perceptual Evaluation Method", Fuzzy Sets and Systems, 119, 149–160. doi:10.1016/S01650114(99)001062

León, T.; Liern, V.; Ruiz, J.L.; Sirvent, I. (2003). "A Fuzzy Mathematical Programming Approach to the assessment of efficiency with DEA Models", Fuzzy Sets Systems, 139(2), 407–419. doi:10.1016/S01650114(02)006085

Sexton, T.R., Silkman, R.H.; Hogan, A.J. (1986). Data envelopment analysis: critique and extensions. In: Silkman RH (ed) Measuring efficiency: an assessment of data envelopment analysis, vol 32. Jossey-Bass, San Francisco, pp 73–104. doi:10.1002/ev.1441

Examples

```
data("Guo_Tanaka_2001")
datadea <- make_deadata_fuzzy(datadea = Guo_Tanaka_2001,
                              inputs.mL = 2:3,
                              inputs.dL = 4:5,
                              outputs.mL = 6:7,
                              outputs.dL = 8:9)
result <- cross_efficiency_fuzzy(datadea = datadea,
                                 h = seq(0, 1, 0.2))
```

`Departments` *Data: Tomkins and Green (1988).*

Description

Data from 20 University accounting departments in the UK.

Usage

```
data("Departments")
```

Format

Data frame with 20 rows and 11 columns. Definition of inputs (X) and outputs (Y):

x1 = Staff Average Full Time Academic Staff 82/3-84/5)

x2 = Salaries 1984-5 Salaries Academics and Related (in pounds))

x3 = Other_Exp 1984-5 Other Expenses (in pounds)

y1 = Undergrad Average Number Undergraduates 82/3-84/5

y2 = Research_post Research Postgraduates

y3 = Taught_post Taught Postgraduates

y4 = Res_co_income Research council income (in pounds)

y5 = Other_res_income Other research income (in pounds)

y6 = Other_income Other income (in pounds)

y7 = Publications Number of publications

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Tomkins, C.; Green, R. (1988). "An Experiment in the Use of Data Envelopment Analysis for Evaluating the Efficiency of UK University Departments of Accounting", Financial Accountability and Management, 4(2), 147-164. doi:10.1111/j.14680408.1988.tb00296.x

See Also

make_deadata, model_basic

Examples

```
# Example.
# Replication of results DEA1 in Tomkins and Green (1988)
data("Departments")
# Calculate Total income
Departments$Total_income <- Departments[, 5] + Departments[, 6] + Departments[, 7]
data_example <- make_deadata(Departments,
                             inputs = 9,
                             outputs = c(2, 3, 4, 12))
result <- model_basic(data_example,
                      orientation = "io",
                      rts = "crs")
efficiencies(result) # Table 3 (p.156)
references(result) # Table 3 (p.157)
```

`Doyle_Green_1994` *Data: Doyle and Green (1994).*

Description

Data adapted from Tomkins and Green (1988). 13 DMUs using 3 inputs to produce 2 outputs.

Usage

```
data("Doyle_Green_1994")
```

Format

Data frame with 13 rows and 6 columns. Definition of inputs (X) and outputs (Y):

y1 = Undergraduate Number of undergraduates

y2 = Postgraduates Number of postgraduates (taught and research)

y3 = Research_income Research and other income

y4 = Publications Number of publications

x1 = Salaries Salaries of academic and related staff

x2 = Other_expenses Other expenses

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Doyle, J.; Green, R. (1994). “Efficiency and cross efficiency in DEA: derivations, meanings and the uses”, Journal of Operational Research Society, 45(5), 567–578. doi:10.2307/2584392

See Also

`make_deadata`, `model_multiplier`, `cross_efficiency`

Examples

```
# Example.
data("Doyle_Green_1994")
data_example <- make_deadata(datadea = Doyle_Green_1994,
                             dmus = 1,
                             inputs = 6:7,
                             outputs = 2:5)
result <- cross_efficiency(data_example,
                           orientation = "io",
                           selfapp = TRUE)
result$Arbitrary$cross_eff
result$Arbitrary$e
# Aggressive using method II
result$M2_agg$cross_eff
# Aggressive using method III
result$M3_agg$cross_eff
```

Economy *Data: Wang and Lan (2011).*

Description

Data of the industrial economy of China in 2005-2009 (data in wide format).

Usage

```
data("Economy")
```

Format

Data frame with 31 rows and 16 columns. Definition of inputs (X) and outputs (Y):

x1 = Capital Total assets (in 100 million RMB)

x2 = Labor Annual average employed persons (in 10000 persons)

y1 = GIOV Gross industrial output value (in 100 million RMB)

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Wang, Y.; Lan, Y. (2011). "Measuring Malmquist Productiviy Index: A New Approach Based on Double Frontiers Data Envelopment Analysis". Mathematical and Computer Modelling, 54, 2760-2771. doi:10.1016/j.mcm.2011.06.064

See Also

`make_malmquist`, `malmquist_index`

Examples

```
# Example . Data in wide format.
# Replication of results in Wang and Lan (2011, p. 2768)
data("Economy")
data_example <- make_malmquist(Economy,
                               nper = 5,
                               arrangement = "horizontal",
                               ni = 2,
                               no = 1)
result <- malmquist_index(data_example)
```

`EconomyLong` *Data: Wang and Lan (2011).*

Description

Data of the industrial economy of China in 2005-2009 (data in long format).

Usage

```
data("EconomyLong")
```

Format

Data frame with 155 rows and 5 columns. Definition of inputs (X) and outputs (Y):

x1 = Capital Total assets (in 100 million RMB)

x2 = Labor Annual average employed persons (in 10000 persons)

y1 = GIOV Gross industrial output value (in 100 million RMB)

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Wang, Y.; Lan, Y. (2011). "Measuring Malmquist Productiviy Index: A New Approach Based on Double Frontiers Data Envelopment Analysis". Mathematical and Computer Modelling, 54, 2760-2771. doi:10.1016/j.mcm.2011.06.064

See Also

`make_malmquist`, `malmquist_index`

Examples

```
# Example. Data in long format.
# Replication of results in Wang and Lan (2011, p. 2768)
data("EconomyLong")
data_example <- make_malmquist(EconomyLong,
                               percol = 2,
                               arrangement = "vertical",
                               ni = 2,
                               no = 1)
result <- malmquist_index(data_example)
```

`efficiencies` *Objective scores*

Description

Extract the scores (optimal objective values) of the evaluated DMUs from a conventional, fuzzy or stochastic DEA solution. Note that these scores may not always be interpreted as efficiencies.

Usage

```
efficiencies(x, ...)
```

Arguments

`x`	An object of class `dea`, `dea_fuzzy` or `dea_stoch`.
`...`	ignored.

Value

A matrix or numeric vector with the optimal objective scores for each evaluated DMU.

efficiencies.dea *Objective scores*

Description

Extract the scores (optimal objective values) of the evaluated DMUs from a conventional DEA solution. Note that these scores may not always be interpreted as efficiencies.

Usage

```
## S3 method for class 'dea'
efficiencies(x, ...)
```

Arguments

x	Object of class dea obtained with some of the conventional DEA model_* functions.
...	Other options (for compatibility reasons).

Value

A numeric vector with the optimal objective scores for each evaluated DMU.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Tomkins, C.; Green, R. (1988). “An Experiment in the Use of Data Envelopment Analysis for Evaluating the Efficiency of UK University Departments of Accounting”. Financial Accountability and Management 4(2): 147.

Examples

```
# Replication results model DEA1 in Tomkins and Green (1988)
data("Departments")
# Calculate Total income
Departments$Total_income <- Departments[, 5] + Departments[, 6] + Departments[, 7]
data_DEA1 <- make_deadata(Departments,
                          inputs = 9,
                          outputs = c(2, 3, 4, 12))
result <- model_basic(data_DEA1,
                      orientation = "io",
                      rts = "crs")
efficiencies(result) # Table 3 (p.156)
```

efficiencies.dea_fuzzy *Efficiencies*

Description

Extract the scores (optimal objective values) of the evaluated DMUs from a fuzzy DEA solution. Note that these scores may not always be interpreted as efficiencies.

Usage

```
## S3 method for class 'dea_fuzzy'
efficiencies(x, ...)
```

Arguments

x	Object of class dea_fuzzy obtained with some of the fuzzy DEA modelfuzzy_* functions.
...	Other options (for compatibility).

Value

A matrix of size (number of evaluated DMUs) x (number of h-levels) with the optimal objective scores.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Boscá, J.E.; Liern, V.; Sala, R.; Martínez, A. (2011). "Ranking Decision Making Units by Means of Soft Computing DEA Models". International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 19(1), p.115-134.

Examples

```
# Replication of results in Boscá, Liern, Sala and Martínez (2011, p.125)
data("Leon2003")
data_example <- make_deadata_fuzzy(datadea = Leon2003,
                                   inputs.mL = 2,
                                   inputs.dL = 3,
                                   outputs.mL = 4,
                                   outputs.dL = 5)
result <- modelfuzzy_kaoliu(data_example,
                            kaoliu_modelname = "basic",
                            alpha = seq(0, 1, by = 0.1),
                            orientation = "io",
                            rts = "vrs")
efficiencies(result)
```

eff_dmus *Efficient DMUs.*

Description

Returns the efficient DMUs evaluated in a dea class object.

Usage

```
eff_dmus(deasol, tol = 1e-04)
```

Arguments

deasol	An object of class "dea" obtained by a DEA model function.
tol	Numeric. Absolute tolerance for numeric comparisons in efficiency scores. By default, it is 1e-4.

Value

A numeric vector containing which DMUs has been evaluated as efficient. This vector is empty if there is not any efficient DMU.

Note

If maxslack is FALSE, the slacks computed in the first stage are supposed to be the max slacks.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
dataFortune <- make_deadata(Fortune500,
                            ni = 3,
                            no = 2)
ccrFortune <- model_basic(dataFortune)
eff_dmus(ccrFortune)
```

`Electric_plants` *Data: Färe, Grosskopf and Kokkelenberg (1989).*

Description

Data of 19 coal-fired steam-electric generating plants operating in Illinois in 1978. Each plant uses 3 inputs to produce 1 output.

Usage

```
data("Electric_plants")
```

Format

Data frame with 18 rows and 5 columns. Definition of inputs (X) and outputs (Y):

x1 = Labor Labor average annual employment

x2 = Fuel Fuel 10^{10} Btu

x3 = Capital Capital MW (fixed input)

y1 = Output Output 10^{6} Kwh

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Färe, R.; Grosskopf, S.; Kokkenlenberg, E. (1989). "Measuring Plant Capacity, Utilization and Technical Change: A Nonparametric Approach". International Economic Review, 30(3), 655-666.

Simar, L.; Wilson, P.W. (1998). "Sensitivity Analysis of Efficiency Scores: How to Bootstrap in Nonparametric Frontier Models". Management Science, 44(1), 49-61.

See Also

make_deadata, model_basic

Examples

```
# Example. Replication of results in Simar and Wilson (1998, p.59)
data("Electric_plants")
data_example <- make_deadata(Electric_plants,
                             dmus = 1,
                             ni = 3,
                             no = 1)
result <- model_basic(data_example,
                      orientation = "io",
                      rts = "vrs")
efficiencies(result)
```

`extreme_efficient` *Extreme efficient DMUs*

Description

Find a set of extreme efficient DMUs from a `deadata` object.

Usage

```
extreme_efficient(datadea,
              dmu_ref = NULL,
              rts = c("crs", "vrs", "nirs", "ndrs"),
              tol = 1e-6)
```

Arguments

`datadea`	A `deadata` object with `n` DMUs, `m` inputs and `s` outputs.
`dmu_ref`	A numeric vector containing which DMUs are the evaluation reference set, i.e. the cluster of DMUs from which we want to find an extreme efficient DMUs subset. If `NULL` (default), all DMUs are considered.
`rts`	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing) or "ndrs" (non-decreasing).
`tol`	Numeric, a tolerance margin for checking efficiency. It is 1e-6 by default.

Value

A numeric vector representing an extreme efficient subset of DMUs.

Author(s)

Vicente Coll-Serrano (`<vicente.coll@uv.es>`). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (`<vicente.bolos@uv.es>`). *Department of Business Mathematics*

Rafael Benítez (`<rafael.suarez@uv.es>`). *Department of Business Mathematics*

University of Valencia (Spain)

References

Charnes, A.; Cooper, W.W.; Thrall, R.M. (1991) "A structure for classifying and characterizing efficiency and inefficiency in data envelopment analysis", Journal of Productivity Analisys, 2, 197–237.

Examples

```
data("PFT1981")
datadea <- make_deadata(PFT1981,
                        ni = 5,
                        no = 3)
# We find an extreme efficient subset from a cluster formed by the first 20 DMUs
result <- extreme_efficient(datadea = datadea,
                            dmu_ref = 1:20)
```

Fortune500 *Data: Zhu (2014).*

Description

This dataset consists of 15 firms from the Fortune 500 list 1995 (https://fortune.com/ranking/fortune500/) with 3 inputs and 2 outputs.

Usage

```
data("Fortune500")
```

Format

Data frame with 15 rows and 6 columns. Definition of inputs (X) and outputs (Y):

x1 = Assets Assets (millions of dollars)

x2 = Equity Equity (millions of dollars)

x3 = Employees Number of employees

y1 = Revenue Revenue (millions of dollars)

y2 = Profit Profit (millions of dollars)

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Zhu, J. (2014). Quantitative Models for Performance Evaluation and Benchmarking. Data Envelopment Analysis with Spreadsheets. 3rd Edition Springer, New York. doi:10.1007/9783319066479

See Also

`make_deadata`, `model_multiplier`

Examples

```
data("Fortune500")
data_Fortune <- make_deadata(datadea = Fortune500,
                             dmus = 1,
                             inputs = 2:4,
                             outputs = 5:6)
result <- model_multiplier(data_Fortune,
                           epsilon = 1e-6,
                           orientation = "io",
                           rts = "crs")
# results for General Motors and Ford Motor are not shown
# by deaR because the solution is infeasible
efficiencies(result)
multipliers(result)
```

`Fried1993` *Data: Fried, Knox Lovell and Schmidt (1993).*

Description

Data of 11 DMUs with two inputs and one output.

Usage

```
data("Fried1993")
```

Format

Data frame with 11 rows and 4 columns. Definition of inputs (X) and outputs (Y):

x1 Input 1

x2 Input 2

y1 Output 1

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Ali, A.I.; Seiford, L.M. (1993). The Mathematical Programming Approach to Efficiency Analysis. In Fried, H.O.; Knox Lovell, C.A.; Schmidt, S.S.(eds.), The Measurement of Productive Efficiency. Techniques and Applications. New York: Oxford University Press.

See Also

make_deadata, model_basic

Examples

```
# Example. Replication of results in Ali and (1993, p.143).
data("Fried1993")
data_example <- make_deadata(Fried1993,
                             ni = 2,
                             no = 1)
result <- model_basic(data_example,
                      orientation = "oo",
                      rts = "vrs")
efficiencies(result)
targets(result)
```

FuzzyExample *Data: Fuzzy data reading example.*

Description

Synthetic dataset of 5 DMUs with 3 inputs and 3 outputs containing fuzzy and crisp data.

Usage

```
data("FuzzyExample")
```

Format

Data frame with 5 rows and 15 columns.

DMU DMU names

Input1.mL First Input (crisp numbers)

Input2.mL Second Input (left centers)

Input2.mR Second Input (right centers)

Input2.dL Second Input (left radii)

Input2.dR Second Input (right radii)

Input3.mL Third Input (centers)

Input3.dL Third Input (radii)

Output1.mL First Output (crisp numbers)

Output2.mL Second Output (left centers)

Output2.mR Second Output (right centers)

Output2.dL Second Output (radii)

Output3.mL Third Output (centers)

Output3.dL Third Output (left radii)

Output3.dR Third Output (right radii)

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

See Also

make_deadata_fuzzy

Examples

```
# Example. Reading the data.
data("FuzzyExample")
datafuzzy <- make_deadata_fuzzy(FuzzyExample,
                                inputs.mL = c(2, 3, 7),
                                inputs.mR = c(NA, 4, NA),
                                inputs.dL = c(NA, 5, 8),
                                inputs.dR = c(NA, 6, NA),
                                outputs.mL = c(9, 10 , 13),
                                outputs.mR = c(NA, 11, NA),
                                outputs.dL = c(NA, 12, 14),
                                outputs.dR = c(NA, NA, 15))
```

Golany_Roll_1989 *Data: Golany and Roll (1989).*

Description

Data of 13 DMUs using 3 inputs to produce 2 outputs.

Usage

```
data("Golany_Roll_1989")
```

Format

Data frame with 13 rows and 6 columns. Definition of inputs (X) and outputs (Y):

x1 Input 1

x2 Input 2

x3 Input 3

y1 Output 1

y2 Output 2

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Golany, B.; Roll, Y. (1989). "An Application Procedure for DEA". Omega, International Journal of Management Science, 17(3), 237-250. doi:10.1016/03050483(89)900297

See Also

make_deadata, model_multiplier, cross_efficiency

Examples

```
# Example.
data("Golany_Roll_1989")
data_example <- make_deadata(datadea = Golany_Roll_1989,
                             dmus = 1,
                             inputs = 2:4,
                             outputs = 5:6)
```

```
result <- cross_efficiency(data_example,
                           orientation = "io",
                           selfapp = TRUE)
result$Arbitrary$cross_eff
result$Arbitrary$e
```

Grifell_Lovell_1999 *Data: Grifell-Tatjé and Lovell (1999).*

Description

Data of 8 DMUs producing 1 output (Y) by using 1 input (X) for two periods of time.

Usage

```
data("Grifell_Lovell_1999")
```

Format

Data frame with 16 rows and 4 columns. Definition of inputs (X) and outputs (Y):

X Input

Y Output

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Grifell-Tatjé, E.; Lovel, C.A.K. (1999). "A Generalized Malmquist productivity index". Top, 7(1), 81-101.

See Also

make_malmquist, malmquist_index

Examples

```
# Example. Replication of results in Grifell-Tatjé and Lovell (1999, p. 100).
data("Grifell_Lovell_1999")
data_example <- make_malmquist(Grifell_Lovell_1999,
                               percol = 1,
                               dmus = 2,
                               inputs = 3,
                               outputs = 4,
                               arrangement = "vertical")

result_fgnz <- malmquist_index(data_example,
                               orientation = "oo",
                               rts = "vrs",
                               type1 = "cont",
                               type2 = "fgnz")

result_fgnz$mi
```

Guo_Tanaka_2001 *Data: Guo and Tanaka (2001).*

Description

Data of 5 DMUs with two symmetric triangular fuzzy inputs, Xj = (xj, alphaj), and two symmetric triangular fuzzy outputs, Yj = (yj, betaj).

Usage

```
data("Guo_Tanaka_2001")
```

Format

Data frame with 5 rows and 9 columns. Definition of fuzzy inputs (X) and fuzzy outputs (Y):

x1 Input 1

x2 Input 2

alpha1 spread vector Input 1

alpha2 spread vector Input 2

y1 Output 1

y2 Output 2

beta1 spread vector Output 1

beta2 spread vector Output 2

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Guo, P.; Tanaka, H. (2001). "Fuzzy DEA: A Perceptual Evaluation Method", Fuzzy Sets and Systems, 119, 149–160. doi:10.1016/S01650114(99)001062

See Also

`make_deadata_fuzzy`, `modelfuzzy_guotanaka`, `cross_efficiency_fuzzy`

Examples

```
data("Guo_Tanaka_2001")
data_example <- make_deadata_fuzzy(Guo_Tanaka_2001,
                                   dmus = 1,
                                   inputs.mL = 2:3,
                                   inputs.dL = 4:5,
                                   outputs.mL = 6:7,
                                   outputs.dL = 8:9)
result <- modelfuzzy_guotanaka(data_example,
                               h = seq(0, 1, by = 0.1),
                               orientation = "io")
efficiencies(result)
```

`Hotels` *Data: Wu, Tsai and Zhou (2011).*

Description

This dataset consists of 23 four- and five-plum ITHs in Taipei in 2006. Authors consider 4 inputs and 3 outputs.

Usage

```
data("Hotels")
```

Format

Data frame with 23 rows and 8 columns. Definition of inputs (X) and outputs (Y):

x1 = Employees Total number of employees)

x2 = Guest_rooms Total number of guest rooms)

x3 = Area_F&B Total area of F&B departments (in 36 square-feet)

x4 = Operating_cost Total operating cost (in NT$)

y1 = Room_revenue Room revenues (in NT$)

y2 = F&B_revenue F&B revenues (in NT$)

y3 = Other_revenue Other revenues (in NT$)

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Wu, J.; Tsai, H. and Zhou, Z. (2011). "Improving efficiency in International tourist hotels in Taipei using a non-radial DEA mode", Internationl Journal of Contemporary Hospitality Management, 23(1), 66-83. doi:10.1108/09596111111101670

See Also

make_deadata, model_nonradial

Examples

```
# Example. Replication of results in Wu,Tsai and Zhou (2011)
data("Hotels")
data_hotels <- make_deadata(Hotels,
                            dmus = 1,
                            inputs = 2:5,
                            outputs = 6:8)
result <- model_nonradial(data_hotels,
                          orientation = "oo",
                          rts = "vrs")
efficiencies(result)
```

Hua_Bian_2007 *Data: Hua and Bian (2007).*

Description

Data of 30 DMUs with two desirable inputs, two desirable outputs and one udesirable output.

Usage

```
data("Hua_Bian_2007")
```

Format

Data frame with 30 rows and 6 columns. Definition of inputs (X) and outputs (Y):

x1 = D-Input1 Desirable Input 1

x2 = D-Input2 Desirable Input 2

y1 = D-Output1 Desirable Output 1

y2 = D-Output2 Desirable Output 2

y3 = UD-Output1 Undesirable Output 1

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Hua Z.; Bian Y. (2007). DEA with Undesirable Factors. In: Zhu J., Cook W.D. (eds) Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis. Springer, Boston, MA. doi:10.1007/9780387716077_6

See Also

make_deadata, model_basic

Examples

```
# Example. Replication of results in Hua and Bian (2007).
data("Hua_Bian_2007")
# The third output is an undesirable output
data_example <- make_deadata(Hua_Bian_2007,
                             ni = 2,
                             no = 3,
                             ud_outputs = 3)

# Translation parameter (vtrans_o) is set to 1500
result <- model_basic(data_example,
                      orientation = "oo",
                      rts = "vrs",
                      vtrans_o = 1500)
eff <- efficiencies(result)
1 / eff # results M5 in Table 6-5 (p.119)
```

is.dea *dea class check.*

Description

Checks whether an R object is of dea class or not.

Usage

```
is.dea(x)
```

Arguments

x	Any **R** object.

Value

Returns TRUE if its argument is a dea object (that is, has "dea" amongst its classes) and FALSE otherwise.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

`is.deadata` *deadata class check.*

Description

Checks whether an R object is of deadata class or not.

Usage

```
is.deadata(x)
```

Arguments

`x`	Any **R** object.

Value

Returns `TRUE` if its argument is a `deadata` object (that is, has "deadata" amongst its classes) and `FALSE` otherwise.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

`is.deadata_fuzzy` *deadata_fuzzy class check.*

Description

Checks whether an R object is of deadata_fuzzy class or not.

Usage

```
is.deadata_fuzzy(x)
```

Arguments

`x`	Any **R** object.

Value

Returns `TRUE` if its argument is a `deadata_fuzzy` object (that is, has "deadata_fuzzy" amongst its classes) and `FALSE` otherwise.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

`is.dea_fuzzy` *dea_fuzzy class check.*

Description

Checks whether an R object is of dea_fuzzy class or not.

Usage

```
is.dea_fuzzy(x)
```

Arguments

`x`	Any **R** object.

Value

Returns `TRUE` if its argument is a `dea_fuzzy` object (that is, has "dea_fuzzy" amongst its classes) and `FALSE` otherwise.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

is.friends *Friends check.*

Description

Checks whether a subset of DMUs is friends or not, according to Tone (2010).

Usage

```
is.friends(datadea,
             dmu_eval = NULL,
             dmu_ref = NULL,
             rts = c("crs", "vrs", "nirs", "ndrs"),
             tol = 1e-6)
```

Arguments

datadea	The data, including n DMUs, m inputs and s outputs.
dmu_eval	A numeric vector containing the subset of DMUs to be checked. If NULL (default), all DMUs are considered.
dmu_ref	A numeric vector containing which DMUs are the evaluation reference set. If NULL (default), all DMUs are considered.
rts	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing) or "ndrs" (non-decreasing).
tol	Numeric, a tolerance margin for checking efficiency. It is 1e-6 by default.

Value

Returns TRUE if dmu_eval is friends of dmu_ref, and FALSE otherwise.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Tone, K. (2010). "Variations on the theme of slacks-based measure of efficiency in DEA", European Journal of Operational Research, 200, 901-907. doi:10.1016/j.ejor.2009.01.027

See Also

maximal_friends, model_sbmeff

Examples

```
data("PFT1981")
datadea <- make_deadata(PFT1981,
                        ni = 5,
                        no = 3)
subset1 <- c(15, 16, 17, 19) # Subset of DMUs to be checked
result1 <- is.friends(datadea = datadea,
                      dmu_eval = subset1,
                  dmu_ref = 1:20) # We only consider a cluster formed by the first 20 DMUs
subset2 <- c(15, 16, 17, 20) # Another subset of DMUs to be checked
result2 <- is.friends(datadea = datadea,
                      dmu_eval = subset2,
                  dmu_ref = 1:20) # We only consider a cluster formed by the first 20 DMUs
```

Kao_Liu_2003 *Data: Kao and Liu (2003).*

Description

Data of 24 university libraries in Taiwan with one input and five outputs.

Usage

```
data("Kao_Liu_2003")
```

Format

Data frame with 24 rows and 11 columns. Definition of fuzzy inputs (X) and fuzzy outputs (Y):

x1 = Patronage It is a weighted sum of the standardized scores of faculty, graduate students, undergraduate students, and extension students in the range of 0 and 1.

y1 = Collections Books, serials, microforms, audiovisual works, and database.

y2 = Personnel Classified staff, unclassified staff, and student assistants.

y3 = Expenditures Capital expenditure, operating expenditure, and special expenditure.

y4 = Buildings Area and seats

y5 = Services Operating hours, attendance, circulation, communication channels, range of services, amount of services, etc.

beta3_l lower spread vector Expenditures

beta3_u upper spread vector Expenditures

beta5_l lower spread vector Services

beta5_u upper spread vector Services

Note

There are three observations that are missing: expenditures of Library 24 and services of Library 22 and Library 23. Kao and Liu (2000b) represent the expenditures of Library 24 by the triangular fuzzy number Y = (0.11; 0.41; 1.0). The services of Library 22 and Library 23 are expressed by a same triangular fuzzy number Y = (0.41; 0.69; 1.0).

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Kao, C., Liu, S.T. (2003). “A mathematical programming approach to fuzzy efficiency ranking”, International Journal of Production Economics, 85. doi:10.1016/S09255273(03)000264

See Also

make_deadata_fuzzy, model_basic

Examples

```
# Example. Replication of results in Kao and Liu (2003, p.152)
data_example <- make_deadata_fuzzy(Kao_Liu_2003,
                                   dmus = 1,
                                   inputs.mL = 2,
                                   outputs.mL = 3:7,
                                   outputs.dL = c(NA, NA, 8, NA, 10),
                                   outputs.dR = c(NA, NA, 9, NA, 11))
result <- modelfuzzy_kaoliu(data_example,
                            kaoliu_modelname = "basic",
                            orientation = "oo",
                            rts = "vrs",
                            alpha = 0)
eff <- efficiencies(result)
eff
```

---

lambdas *Lambdas*

Description

Extract the lambdas of the DMUs from a dea or dea_fuzzy solution.

Usage

```
lambdas(deasol)
```

Arguments

deasol	Object of class dea or dea_fuzzy obtained with some of the DEA model functions.

Value

A matrix with parameters lambda for each evaluated DMU.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
data("Coll_Blasco_2006")
data_example <- make_deadata(Coll_Blasco_2006,
                             ni = 2,
                             no = 2)
result <- model_multiplier(data_example,
                           orientation = "io",
                           rts = "crs")
lambdas(result)
```

Leon2003 *Data: Leon, Liern, Ruiz and Sirvent (2003).*

Description

Data of 8 DMUs with one symmetric triangular fuzzy inputs: Xj = (xj, alphaj), and one symmetric triangular fuzzy outputs: Yj = (yj, betaj).

Usage

```
data("Leon2003")
```

Format

Data frame with 8 rows and 5 columns. Definition of fuzzy inputs (X) and fuzzy outputs (Y):

x1 Input 1

alpha1 spread vector Input 1

y1 Output 1

beta1 spread vector Output 1

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Leon, T.; Liern, V. Ruiz, J.; Sirvent, I. (2003). "A Possibilistic Programming Approach to the Assessment of Efficiency with DEA Models", Fuzzy Sets and Systems, 139, 407–419. doi:10.1016/S01650114(02)006085

See Also

make_deadata_fuzzy, modelfuzzy_possibilistic, cross_efficiency_fuzzy, modelfuzzy_guotanaka

Examples

```
# Example. Replication of results in Leon et. al (2003, p. 416)
data("Leon2003")
data_example <- make_deadata_fuzzy(Leon2003,
                                   dmus = 1,
                                   inputs.mL = 2,
                                   inputs.dL = 3,
                                   outputs.mL = 4,
                                   outputs.dL = 5)
result <- modelfuzzy_possibilistic(data_example,
                                   h = seq(0, 1, by = 0.1),
                                   orientation = "io",
                                   rts = "vrs")
efficiencies(result)
```

Libraries *Data: Cooper, Seiford and Tone (2007).*

Description

Data for 23 public libraries of the Tokyo Metropolitan Area in 1986.

Usage

```
data("Libraries")
```

Format

Data frame with 23 rows and 7 columns. Definition of inputs (X) and outputs (Y):

x1 = AREA Floor area (unit=1000 m2)

x2 = BOOKS Number of books (unit=1000)

x3 = STAFF Staff

x4 = POPULATION Population (unit=1000)

y1 = REGISTERED Registered residents (unit=1000)

y2 = BORROWED Borrowed books (unit=1000)

Author(s)

Vicente Coll-Serrano (`<vicente.coll@uv.es>`). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (`<vicente.bolos@uv.es>`). *Department of Business Mathematics*

Rafael Benitez (`<rafael.suarez@uv.es>`). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Cooper, W.W.; Seiford, L.M. and Tone, K. (2007). Data Envelopment Analysis. A Comprehensive Text with Models, Applications, References and DEA-Solver Software. Springer.

See Also

`make_deadata`, `model_basic`

Examples

```
# Example 1. Non-controllable input (POPULATION).
# Replication of results in Cooper, Seiford and Tone (2007, p.221)
data(Libraries)
# POPULATION (non-controllable input) is the 4th input.
data_example <- make_deadata(Libraries,
                             dmus = 1,
                             inputs = 2:5,
                             nc_inputs = 4,
                             outputs = 6:7)
result <- model_basic(data_example,
                      orientation = "io",
                      rts = "crs")
efficiencies(result)
targets(result)

# Example 2. Non-discretionary input (POPULATION).
data(Libraries)
# POPULATION (non-controllable input) is the 4th input.
data_example2 <- make_deadata(Libraries,
                              dmus=1,
                              inputs=2:5,
                              nd_inputs=4,
                              outputs=6:7)
result2 <- model_basic(data_example2,
                       orientation="io",
                       rts="crs")
efficiencies(result2)
targets(result2)
```

Lim_Zhu_2015 *Data: Lim and Zhu (2015).*

Description

Data of 37 R&D project proposal relating to the Turkish iron and steel industry. Authors consider one input and five outputs.

Usage

```
data("Lim_Zhu_2015")
```

Format

Data frame with 37 rows and 7 columns. Definition of inputs (X) and outputs (Y):

x1 = Budget Budget

y1 = Indirect_economic Indirect economic contribution

y2 = Direct_economic Direct economic contribution

y3 = Technical Technical contribution

y4 = Social Social contribution

y5 = Scientific Scientific contribution

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Lim, S.; Zhu, J. (2015). "DEA Cross-Efficiency Under Variable Returns to Scale". Journal of Operational Research Society, 66(3), p. 476-487. doi:10.1057/jors.2014.13

See Also

make_deadata, model_multiplier, cross_efficiency

Examples

```
# Example. Arbitrary formulation.
# Input-oriented model under variable returns-to-scale.
data("Lim_Zhu_2015")
data_example <- make_deadata(Lim_Zhu_2015,
                             dmus = 1,
                             ni = 1,
                             no = 5)
cross <- cross_efficiency(data_example,
                          epsilon = 0,
                          orientation = "io",
                          rts = "vrs",
                          selfapp = TRUE,
                          M2 = FALSE,
                          M3 = FALSE)
cross$Arbitrary$e
```

make_deadata *make_deadata*

Description

This function creates, from a data frame, a deadata structure, which is as list with fields input, output, dmunames, nc_inputs, nc_outputs, nd_inputs, nd_outputs.

Usage

```
make_deadata(datadea = NULL,
          ni = NULL,
          no = NULL,
          dmus = 1,
          inputs = NULL,
          outputs = NULL,
          nc_inputs = NULL,
          nc_outputs = NULL,
          nd_inputs = NULL,
          nd_outputs = NULL,
          ud_inputs = NULL,
          ud_outputs = NULL,
          bnd_inputs = NULL,
          bnd_outputs = NULL)
```

Arguments

datadea Data frame with DEA data.

ni Number of inputs, if inputs are in columns 2:(ni + 1) (if DMUs are in the first column) or 1:ni (no DMUs column).

no Number of outputs, if outputs are in columns (ni + 2):(ni + no + 1) (if DMUs are in the first column) or (ni + 1):(ni + no) (no DMUs column). If not specified, DMUs are in the first column.

dmus Column (number or name) of DMUs (optional). By default, it is the first column. If there is not any DMU column, then it must be NULL.

inputs Columns (numbers or names) of inputs (optional). It prevails over ni. Alternatively to datadea, it can be a matrix with the inputs (DMUs in columns). In this case, DMUs names are taken from the columns names.

outputs Columns (numbers or names) of outputs (optional). It prevails over no. Alternatively to datadea, it can be a matrix with the outputs (DMUs in columns).

nc_inputs A numeric vector containing the indices of non-controllable inputs.

nc_outputs A numeric vector containing the indices of non-controllable outputs.

nd_inputs A numeric vector containing the indices of non-discretionary inputs.

nd_outputs A numeric vector containing the indices of non-discretionary outputs.

`ud_inputs` A numeric vector containing the indices of undesirable (good) inputs.

`ud_outputs` A numeric vector containing the indices of undesirable (bad) outputs.

`bnd_inputs` A numeric vector of length `ni` with the lower bounds of inputs and upper bounds of undesirable (good) inputs. NA for unbounded inputs.

`bnd_outputs` A numeric vector of length `no` with the upper bounds of outputs and lower bounds of undesirable (bad) outputs. NA for unbounded outputs.

Value

An object of class `deadata`. Any function `model_*` can be applied.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

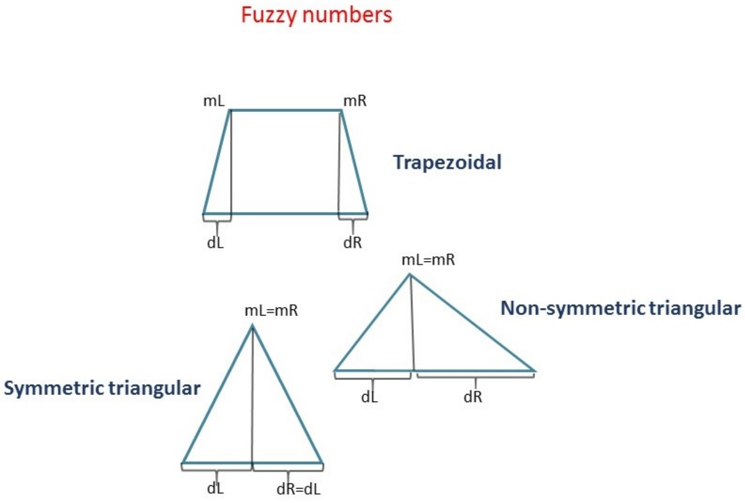
Examples

```
data("Coll_Blasco_2006")
data_example <- make_deadata(datadea = Coll_Blasco_2006,
                             ni = 2,
                             no = 2)
# This is the same as:
data_example <- make_deadata(Coll_Blasco_2006,
                             inputs = 2:3,
                             outputs = 4:5)
# And the same as:
dmunames <- c("A", "B", "C", "D", "E", "F")
nd <- length(dmunames) # Number of DMUs
inputnames <- c("Employees", "Capital")
ni <- length(inputnames) # Number of Inputs
outputnames <- c("Vehicles", "Orders")
no <- length(outputnames) # Number of Outputs
inputs <- matrix(c(8, 8, 11, 15, 14, 12, 12, 13, 11, 18, 18, 20),
                 nrow = ni, ncol = nd, dimnames = list(inputnames, dmunames))
outputs <- matrix(c(14, 20, 25, 42, 8, 30, 25, 8, 40, 22, 24, 30),
                  nrow = no, ncol = nd, dimnames = list(outputnames, dmunames))
data_example <- make_deadata(inputs = inputs,
                             outputs = outputs)
# If the first input is a non-controllable input:
data_example <- make_deadata(Coll_Blasco_2006,
                             inputs = 2:3,
                             outputs = 4:5,
                             nc_inputs = 1)
# If the second output is a non-discretionary output:
data_example <- make_deadata(Coll_Blasco_2006,
                             inputs = 2:3,
                             outputs = 4:5,
                             nd_outputs = 2)
# If the second input is a non-discretionary input and the second output is undesirable:
data_example <- make_deadata(Coll_Blasco_2006,
                             inputs = 2:3,
                             outputs = 4:5,
                             nd_inputs = 2,
                             ud_outputs = 2)
# If the first input is lower bounded by 5 and the second output is upper bounded by 50:
data_example <- make_deadata(Coll_Blasco_2006,
                             inputs = 2:3,
                             outputs = 4:5,
                             bnd_inputs = c(5, NA),
                             bnd_outputs = c(NA, 50))
```

make_deadata_fuzzy   *make_deadata_fuzzy*

Description

This function creates, from a data frame, a deadata_fuzzy structure, which is as list with fields input, output and dmunames. At the same time, input and output are lists with fields mL, mR, dL and dR.



Usage

```
make_deadata_fuzzy(datadea,
                   dmus = 1,
                   inputs.mL = NULL,
                   inputs.mR = NULL,
                   inputs.dL = NULL,
                   inputs.dR = NULL,
                   outputs.mL = NULL,
                   outputs.mR = NULL,
                   outputs.dL = NULL,
                   outputs.dR = NULL,
                   nc_inputs = NULL,
                   nc_outputs = NULL,
                   nd_inputs = NULL,
                   nd_outputs = NULL,
                   ud_inputs = NULL,
                   ud_outputs = NULL)
```

Arguments

`datadea` Data frame with DEA data.

`dmus` Column (number or name) of DMUs (optional). By default, it is the first column. If there is not any DMU column, then it must be `NULL`.

`inputs.mL` Where are (columns) the `mL` (left centers) of trapezoidal fuzzy inputs in `datadea`. If an input is triangular or crisp, we put the column where the centers or the crisp values are, respectively.

Alternatively to `datadea`, `inputs.mL` can be a matrix of size (number of inputs x number of DMUs) with the `mL` of trapezoidal fuzzy inputs, the centers of triangular inputs, and the crisp values of crisp inputs. In this case, DMUs names are taken from the columns names.

`inputs.mR` Where are (columns) the `mR` (right centers) of trapezoidal fuzzy inputs in `datadea`. If an input is triangular or crisp, we put `NA`.

Alternatively to `datadea`, `inputs.mR` can be a matrix of size (number of inputs x number of DMUs) with the `mR` of trapezoidal fuzzy inputs, the centers of triangular inputs, and the crisp values of crisp inputs. If all inputs are triangular or crisp, then `inputs.mR` must be NULL (default) or equal to `inputs.mL`.

`inputs.dL` Where are (columns) the `dL` (left radii) of trapezoidal and triangular fuzzy inputs in `datadea`. If an input is symmetric, we put the column where the radii are. If an input is rectangular or crisp, we put `NA`.

Alternatively to `datadea`, `inputs.dL` can be a matrix of size (number of inputs x number of DMUs) with the `dL` of trapezoidal and triangular fuzzy inputs. If an input is rectangular or crisp, its radius is zero. If all inputs are rectangular or crisp, then `inputs.dL` must be NULL (default) or a zero matrix.

`inputs.dR` Where are (columns) the `dR` (right radii) of trapezoidal and triangular fuzzy inputs in `datadea`. If an input is symmetric, rectangular or crisp, we put `NA`.

Alternatively to `datadea`, `inputs.dR` can be a matrix of size (number of inputs x number of DMUs) with the `dR` of trapezoidal and triangular fuzzy inputs. If an input is rectangular or crisp, its radius is zero. If all inputs are symmetric, rectangular or crisp, then `inputs.dR` must be NULL (default) or equal to `inputs.dL`.

outputs.mL Analogous to inputs.mL, but relating to outputs.

outputs.mR Analogous to inputs.mR, but relating to outputs.

outputs.dL Analogous to inputs.dL, but relating to outputs.

outputs.dR Analogous to inputs.dR, but relating to outputs.

nc_inputs A numeric vector containing the indices of non-controllable inputs.

nc_outputs A numeric vector containing the indices of non-controllable outputs.

nd_inputs A numeric vector containing the indices of non-discretionary inputs.

nd_outputs A numeric vector containing the indices of non-discretionary outputs.

ud_inputs A numeric vector containing the indices of undesirable (good) inputs.

ud_outputs A numeric vector containing the indices of undesirable (bad) outputs.

Value

An object of class deadata_fuzzy. Any function modelfuzzy_* can be applied.

Examples

```
# Example 1. If inputs and/or outputs are symmetric triangular fuzzy numbers
data("Leon2003")
data_example <- make_deadata_fuzzy(datadea = Leon2003,
                                   inputs.mL = 2,
                                   inputs.dL = 3,
                                   outputs.mL = 4,
                                   outputs.dL = 5)
# Example 2. If inputs and/or outputs are non-symmetric triangular fuzzy numbers
data("Kao_Liu_2003")
data_example <- make_deadata_fuzzy(Kao_Liu_2003,
                                   inputs.mL = 2,
                                   outputs.mL = 3:7,
                                   outputs.dL = c(NA, NA, 8, NA, 10),
                                   outputs.dR = c(NA, NA, 9, NA, 11))
```

make_malmquist *make_malmquist*

Description

This function creates, from a data frame, a list of deadata objects.

Usage

```
make_malmquist(datadea,
               nper = NULL,
               percol = NULL,
               arrangement  = c("horizontal", "vertical"),
               ...)
```

Arguments

datadea	Data frame with DEA data.
nper	Number of time periods (with dataset in wide format).
percol	Column of time period (with dataset in long format).
arrangement	Horizontal with data in wide format. Vertical with data in long format.
...	Other options to be passed to the make_deadata function.

Value

An object of class deadata. Function malmquist_index can be applied.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
# Example 1. If you have a dataset in wide format.
data("Economy")
data_example <- make_malmquist(datadea = Economy,
                               nper = 5,
                               arrangement = "horizontal",
                               ni = 2,
                               no = 1)
# This is the same as:
data_example <- make_malmquist(datadea = Economy,
                               nper = 5,
                               arrangement = "horizontal",
                               inputs = 2:3,
                               outputs = 4)
# Example 2. If you have a dataset in long format.
data("EconomyLong")
data_example2 <- make_malmquist(EconomyLong,
                                percol = 2,
                                arrangement = "vertical",
                                inputs = 3:4,
                                outputs = 5)
```

malmquist_index *Malmquist index*

Description

This function calculates the input/output oriented Malmquist productivity index under constant or variable returns-to-scale.

Usage

```
malmquist_index(datadealist,
                dmu_eval = NULL,
                dmu_ref = NULL,
                orientation = c("io", "oo"),
                rts = c("crs", "vrs"),
                type1 = c("cont", "seq", "glob"),
                type2 = c("fgnz", "rd", "gl", "bias"),
                tc_vrs = FALSE,
                vtrans_i = NULL,
                vtrans_o = NULL)
```

Arguments

datadealist	A list with the data (deadata objects) at different times, including DMUs, inputs and outputs.
dmu_eval	A numeric vector containing which DMUs have to be evaluated. If NULL (default), all DMUs are considered.
dmu_ref	A numeric vector containing which DMUs are the evaluation reference set. If NULL (default), all DMUs are considered.
orientation	A string, equal to "io" (input oriented) or "oo" (output oriented).
rts	A string, determining the type of returns to scale, equal to "crs" (constant) or "vrs" (variable).
type1	A string, equal to "cont" (contemporary), "seq" (sequential) or "glob" (global).
type2	A string, equal to "fgnz" (Fare et al. 1994), "rd" (Ray and Desli 1997), "gl" (generalized) or "bias" (biased).
tc_vrs	Logical. If it is FALSE, it computes the vrs bias malmquist index by using the technical change under crs (Fare and Grosskopf 1996). Otherwise, it uses the technical change under vrs.
vtrans_i	Numeric vector of translation for undesirable inputs in non-directional basic models. If vtrans_i[i] is NA, then it applies the "max + 1" translation to the i-th undesirable input. If vtrans_i is a constant, then it applies the same translation to all undesirable inputs. If vtrans_i is NULL, then it applies the "max + 1" translation to all undesirable inputs.
vtrans_o	Numeric vector of translation for undesirable outputs in non-directional basic models, analogous to vtrans_i, but applied to outputs.

Value

A numeric list with Malmquist index and other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Note

In the results: EC = Efficiency Change, PTEC = Pure Technical Efficiency Change, SEC = Scale Efficiency Change, TC = Technological Change, MI = Malmquist Index

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benitez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Caves, D.W.; Christensen, L. R.; Diewert, W.E. (1982). “The Economic Theory of Index Numbers and the Measurement of Input, Output, and Productivity”. Econometrica, 50(6), 1393-1414.

Fare, R.; Grifell-Tatje, E.; Grosskopf, S.; Lovell, C.A.K. (1997). "Biased Technical Change and the Malmquist Productivity Index". Scandinavian Journal of Economics, 99(1), 119-127.

Fare, R.; Grosskopf, S.; Lindgren, B.; Roos, P. (1989). “Productivity Developments in Swedish Hospitals: A Malmquist Output Index Approach”. Discussion paper n. 89-3. Southern Illinois University. Illinois.

Fare, R.; Grosskopf, S.; Lindgren, B.; Roos, P. (1992). “Productivity changes in Swedish Pharmacies 1980-89: A nonparametric Malmquist Approach”. Journal of productivity Analysis, 3(3), 85-101.

Fare, R.; Grosskopf, S.; Norris, M.; Zhang, Z. (1994). “Productivity Growth, Technical Progress, and Efficiency Change in Industrialized Countries”. American Economic Review, 84(1), 66-83.

Fare, R.; Grosskopf, S.; Roos, P. (1998), Malmquist Productivity Indexes: A Survey of Theory and Practice. In: Fare R., Grosskopf S., Russell R.R. (eds) Index Numbers: Essays in Honour of Sten Malmquist. Springer.

Grifell-Tatje, E.; Lovell, C.A.K. (1999). "A Generalized Malmquist productivity index". Top, 7(1), 81-101.

Pastor, J.T.; Lovell, C.A.k. (2005). "A global Malmquist productiviyt index". Economics Letters, 88, 266-271.

Ray, S.C.; Desli, E. (1997). "Productivity Growth, Technical Progress, and Efficiency Change in Industrialized Countries: Comment". The American Economic Review, 87(5), 1033-1039.

Shestalova, V. (2003). "Sequential Malmquist Indices of Productivity Growth: An Application to OECD Industrial Activities". Journal of Productivity Analysis, 19, 211-226.

Examples

```
# Example 1. With dataset in wide format.
# Replication of results in Wang and Lan (2011, p. 2768)
data("Economy")
data_example <- make_malmquist(datadea = Economy,
                               nper = 5,
                               arrangement = "horizontal",
                               ni = 2,
                               no = 1)
result <- malmquist_index(data_example, orientation = "io")
mi <- result$mi
effch <- result$ec
tech <- result$tc

# Example 2. With dataset in long format.
# Replication of results in Wang and Lan (2011, p. 2768)
data("EconomyLong")
data_example2 <- make_malmquist(EconomyLong,
                                percol = 2,
                                arrangement = "vertical",
                                inputs = 3:4,
                                outputs = 5)
result2 <- malmquist_index(data_example2, orientation = "io")
mi2 <- result2$mi
effch2 <- result2$ec
tech2 <- result2$tc

# Example 3. Replication of results in Grifell-Tatje and Lovell (1999, p. 100).
data("Grifell_Lovell_1999")
data_example <- make_malmquist(Grifell_Lovell_1999,
                               percol = 1,
                               dmus = 2,
                               inputs = 3,
                               outputs = 4,
                               arrangement = "vertical")
result_fgnz <- malmquist_index(data_example,
                               orientation = "oo",
                               rts = "vrs",
                               type1 = "cont",
                               type2 = "fgnz")
mi_fgnz <- result_fgnz$mi

result_rd <- malmquist_index(data_example,
                             orientation = "oo",
                             rts = "vrs",
                             type1 = "cont",
                             type2 = "rd")
mi_rd <- result_rd$mi

result_gl <- malmquist_index(data_example,
                             orientation = "oo",
                             rts = "vrs",
                             type1 = "cont",
                             type2 = "gl")
mi_gl <- result_gl$mi
```

maximal_friends *Maximal friends of a set of DMUs.*

Description

Finds the maximal friends subsets of a given set of DMUs, according to Tone (2010). It uses an ascending algorithm in order to find directly maximal subsets.

Usage

```
maximal_friends(datadea,
                dmu_ref = NULL,
                rts = c("crs", "vrs", "nirs", "ndrs"),
                tol = 1e-6,
                silent = FALSE)
```

Arguments

datadea	A deadata object with n DMUs, m inputs and s outputs.
dmu_ref	A numeric vector containing which DMUs are the evaluation reference set, i.e. the cluster of DMUs from which we want to find maximal friends. If NULL (default), all DMUs are considered.
rts	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing) or "ndrs" (non-decreasing).
tol	Numeric, a tolerance margin for checking efficiency. It is 1e-6 by default.
silent	Logical, if FALSE (default) steps are printed.

Value

A list with numeric vectors representing maximal friends subsets of DMUs.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Tone, K. (2010). "Variations on the theme of slacks-based measure of efficiency in DEA", European Journal of Operational Research, 200, 901-907. doi:10.1016/j.ejor.2009.01.027

See Also

is.friends, model_sbmeff

Examples

```
## Not run:
data("PFT1981")
datadea <- make_deadata(PFT1981,
                        ni = 5,
                        no = 3)
# We find maximal friends of a cluster formed by the first 20 DMUs
result <- maximal_friends(datadea = datadea,
                          dmu_ref = 1:20)

## End(Not run)
```

modelfuzzy_guotanaka *Fuzzy DEA model*

Description

Solve the Fuzzy input-oriented and output-oriented DEA model proposed by Guo and Tanaka (2001) under constant returns to scale. In deaR is implemented the LP poblem given by the model (16) in Guo and Tanaka (2001, p.155). The fuzzy efficiencies are calculated according to equations in (17) (Guo and Tanaka, 2001, p.155). The (crisp) relative efficiencies and multipliers for the case h = 1 are obtained from the CCR model (`model_multiplier`).

Usage

```
modelfuzzy_guotanaka(datadea,
                       dmu_eval = NULL,
                       dmu_ref = NULL,
                       orientation = c("io", "oo"),
                       h = 1)
```

Arguments

datadea	A deadata_fuzzy object, including DMUs, inputs and outputs.
dmu_eval	A numeric vector containing which DMUs have to be evaluated. If NULL (default), all DMUs are considered.
dmu_ref	A numeric vector containing which DMUs are the evaluation reference set. If NULL (default), all DMUs are considered.

`orientation` A string, equal to "io" (input oriented) or "oo" (output oriented).

`h` A numeric vector with the h-levels (in [0,1]).

Value

A list of class `dea_fuzzy` with the results for the evaluated DMUs (`DMU` component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `h`, `dmu_eval` and `dmu_ref`.

Note

The optimal solution of model (16) is not unique.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Emrouznejad, A.; Tavana, M.; Hatami-Marbini, A. (2014). “The State of the Art in Fuzzy Data Envelopment Analysis”, in A. Emrouznejad and M. Tavana (eds.), Performance Measurement with Fuzzy Data Envelopment Analysis. Studies in Fuzziness and Soft Computing 309. Springer, Berlin. doi:10.1007/9783642413728_1

Guo, P.; Tanaka, H. (2001). "Fuzzy DEA: A Perceptual Evaluation Method", Fuzzy Sets and Systems, 119, 149–160. doi:10.1016/S01650114(99)001062

Hatami-Marbini, A.; Emrouznejad, A.; Tavana, M. (2011). "A Taxonomy and Review of the Fuzzy Data Envelopment Analysis Literature: Two Decades in the Making", European Journal of Operational Research, 214, 457–472. doi:10.1016/j.ejor.2011.02.001

See Also

`model_basic`, `model_multiplier`, `modelfuzzy_kaoliu`, `modelfuzzy_possibilistic`, `cross_efficiency_fuzzy`

Examples

```
# Example 1.
# Replication results in Guo and Tanaka (2001, p. 159).
# In deaR is implemented the LP poblem given by the model 16 in Guo and Tanaka (2001, p. 155).
# The fuzzy efficiencies are calculated according to equations in (17) (Guo and Tanaka, 2001,p.155).
data("Guo_Tanaka_2001")
data_example <- make_deadata_fuzzy(Guo_Tanaka_2001,
                                   inputs.mL = 2:3,
                                   inputs.dL = 4:5,
                                   outputs.mL = 6:7,
                                   outputs.dL = 8:9)
result <- modelfuzzy_guotanaka(data_example,
                               h = c(0, 0.5, 0.75, 1),
                               orientation = "io")
efficiencies(result)

# Example 2.
data("Guo_Tanaka_2001")
data_example <- make_deadata_fuzzy(Guo_Tanaka_2001,
                                   inputs.mL = 2:3,
                                   inputs.dL = 4:5,
                                   outputs.mL = 6:7,
                                   outputs.dL = 8:9)
result2 <- modelfuzzy_guotanaka(data_example,
                                h = seq(0, 1, by = 0.1),
                                orientation = "io")
efficiencies(result2)
```

modelfuzzy_kaoliu *Fuzzy DEA model.*

Description

Solve the fuzzy DEA model by Kao and Liu (2000)

Usage

```
modelfuzzy_kaoliu(datadea,
                  dmu_eval = NULL,
                  kaoliu_modelname = c("basic", "additive", "addmin",
                                       "addsupereff", "deaps", "dir", "fdh",
                                       "lgo", "multiplier", "nonradial",
                                       "profit", "qgo", "rdm", "sbmeff",
                                       "sbmsupereff", "supereff"),
                  alpha = 1,
                  ...)
```

Arguments

`datadea` A `deadata_fuzzy` object, including DMUs, inputs and outputs.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`kaoliu_modelname` a string containing the name of the model.

`alpha` A numeric vector with the alpha-cuts (in [0,1]). If `alpha`>1, it determines the number of alpha-cuts, equispatially distributed in [0,1].

`...` `dmu_ref`, `orientation`, `rts` and other model parameters.

Value

A list of class dea_fuzzy with the results for the evaluated DMUs (DMU component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters orientation, rts, alpha, dmu_eval and dmu_ref.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Boscá, J.E.; Liern, V.; Sala, R.; Martínez, A. (2011). "Ranking Decision Making Units by Means of Soft Computing DEA Models". International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 19(1), p.115-134.

Emrouznejad, A.; Tavana, M.; Hatami-Marbini, A. (2014). “The State of the Art in Fuzzy Data Envelopment Analysis”, in A. Emrouznejad and M. Tavana (eds.), Performance Measurement with Fuzzy Data Envelopment Analysis. Studies in Fuzziness and Soft Computing 309. Springer, Berlin.

Hatami-Marbini, A.; Emrouznejad, A.; Tavana, M. (2011). "A Taxonomy and Review of the Fuzzy Data Envelopment Analysis Literature: Two Decades in the Making", European Journal of Operational Research, 214, 457–472.

Kao, C.; Liu, S.T. (2000). “Fuzzy efficiency measures in data envelopment analysis, Fuzzy Sets and Systems”, 119, 149–160.

Kao, C., Liu, S.T., (2000). “Data envelopment analysis with missing data: An application to university libraries in Taiwan”, Journal of the Operational Research Society, 51, 897–905.

Kao, C., Liu, S.T. (2003). “A mathematical programming approach to fuzzy efficiency ranking”, International Journal of Production Economics, 85.

See Also

model_basic, model_multiplier, modelfuzzy_possibilistic, modelfuzzy_guotanaka

Examples

```
# Example 1.
# Replication of results in Boscá, Liern, Sala and Martínez (2011, p.125)
data("Leon2003")
data_example <- make_deadata_fuzzy(datadea = Leon2003,
                                   inputs.mL = 2,
                                   inputs.dL = 3,
                                   outputs.mL = 4,
                                   outputs.dL = 5)
result <- modelfuzzy_kaoliu(data_example,
                            kaoliu_modelname = "basic",
                            alpha = seq(0, 1, by = 0.1),
                            orientation = "io",
                            rts = "vrs")
efficiencies(result)

# Example 2.
# Replication of results in Kao and Liu (2003, p.152)
data("Kao_Liu_2003")
data_example <- make_deadata_fuzzy(Kao_Liu_2003,
                                   inputs.mL = 2,
                                   outputs.mL = 3:7,
                                   outputs.dL = c(NA, NA, 8, NA, 10),
                                   outputs.dR = c(NA, NA, 9, NA, 11))
result <- modelfuzzy_kaoliu(data_example,
                            kaoliu_modelname = "basic",
                            orientation = "oo",
                            rts = "vrs",
                            alpha = 0)
sol <- efficiencies(result)
eff <- data.frame(1 / sol$Worst, 1 / sol$Best)
names(eff) <- c("eff_lower", "eff_upper")
eff
```

modelfuzzy_possibilistic

Possibilistic Fuzzy DEA model.

Description

Solve the possibilistic fuzzy DEA model proposed by León et. al (2003).

Usage

```
modelfuzzy_possibilistic(datadea,
                            dmu_eval = NULL,
                            poss_modelname = c("basic"),
                            h = 1,
                            ...)
```

Arguments

`datadea`	A `deadata_fuzzy` object, including DMUs, inputs and outputs.
`dmu_eval`	A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.
`poss_modelname`	a string containing the name of the model.
`h`	A numeric vector with the h-levels (in [0,1]).
`...`	`dmu_ref`, `orientation`, `rts` and other model parameters.

Value

A list of class dea_fuzzy with the results for the evaluated DMUs (DMU component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters orientation, rts, h, dmu_eval and dmu_ref.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Emrouznejad, A.; Tavana, M.; Hatami-Marbini, A. (2014). “The State of the Art in Fuzzy Data Envelopment Analysis”, in A. Emrouznejad and M. Tavana (eds.), Performance Measurement with Fuzzy Data Envelopment Analysis. Studies in Fuzziness and Soft Computing 309. Springer, Berlin. doi:10.1007/9783642413728_1

Hatami-Marbini, A.; Emrouznejad, A.; Tavana, M. (2011). "A Taxonomy and Review of the Fuzzy Data Envelopment Analysis Literature: Two Decades in the Making", European Journal of Operational Research, 214, 457–472. doi:10.1016/j.ejor.2011.02.001

Leon, T.; Liern, V. Ruiz, J.; Sirvent, I. (2003). "A Possibilistic Programming Approach to the Assessment of Efficiency with DEA Models", Fuzzy Sets and Systems, 139, 407–419. doi:10.1016/S01650114(02)006085

See Also

model_basic, modelfuzzy_kaoliu, modelfuzzy_guotanaka

Examples

```
# Replication of results in Leon et. al (2003, p. 416)
data("Leon2003")
data_example <- make_deadata_fuzzy(Leon2003,
                                   inputs.mL = 2,
                                   inputs.dL = 3,
                                   outputs.mL = 4,
                                   outputs.dL = 5)
result <- modelfuzzy_possibilistic(data_example,
                                   h = seq(0, 1, by = 0.1),
                                   orientation = "io",
                                   rts = "vrs")
efficiencies(result)
```

`model_additive` *Additive DEA model.*

Description

Solve the additive model of Charnes et. al (1985). With the current version of deaR, it is possible to solve input-oriented, output-oriented, and non-oriented additive model under constant and non-constant returns to scale.

Besides, the user can set weights for the input slacks and/or output slacks. So, it is also possible to solve weighted additive models. For example: Measure of Inefficiency Proportions (MIP), Range Adjusted Measure (RAM), etc.

Usage

```
model_additive(datadea,
               dmu_eval = NULL,
               dmu_ref = NULL,
               orientation = NULL,
               weight_slack_i = 1,
               weight_slack_o = 1,
               rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
               L = 1,
               U = 1,
               compute_target = TRUE,
               returnlp = FALSE,
               ...)
```

Arguments

`datadea`	A `deadata` object with `n` DMUs, `m` inputs and `s` outputs.
`dmu_eval`	A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.
`dmu_ref`	A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.
`orientation`	This parameter is either `NULL` (default) or a string, equal to "io" (input-oriented) or "oo" (output-oriented). It is used to modify the weight slacks. If input-oriented, `weight_slack_o` are taken 0. If output-oriented, `weight_slack_i` are taken 0.
`weight_slack_i`	A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input slacks. If 0, output-oriented.
`weight_slack_o`	A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output slacks. If 0, input-oriented.
`rts`	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L`	Lower bound for the generalized returns to scale (grs).
`U`	Upper bound for the generalized returns to scale (grs).
`compute_target`	Logical. If it is `TRUE`, it computes targets. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.
`returnlp`	Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints).
`...`	Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Note

In this model, the efficiency score is the sum of the slacks. Therefore, a DMU is efficient when the objective value (`objval`) is zero.

Author(s)

Vicente Coll-Serrano (`<vicente.coll@uv.es>`). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (`<vicente.bolos@uv.es>`). *Department of Business Mathematics*

Rafael Benítez (`<rafael.suarez@uv.es>`). *Department of Business Mathematics*

University of Valencia (Spain)

References

Charnes, A.; Cooper, W.W.; Golany, B.; Seiford, L.; Stuz, J. (1985) "Foundations of Data Envelopment Analysis for Pareto-Koopmans Efficient Empirical Production Functions", Journal of Econometrics, 30(1-2), 91-107. doi:10.1016/03044076(85)901332

Charnes, A.; Cooper, W.W.; Lewin, A.Y.; Seiford, L.M. (1994). Data Envelopment Analysis: Theory, Methology, and Application. Boston: Kluwer Academic Publishers. doi:10.1007/97894011-06375

Cooper, W.W.; Park, K.S.; Pastor, J.T. (1999). "RAM: A Range Adjusted Measure of Inefficiencies for Use with Additive Models, and Relations to Other Models and Measures in DEA". Journal of Productivity Analysis, 11, p. 5-42. doi:10.1023/A:1007701304281

See Also

`model_addsupereff`

Examples

```
# Example 1.
# Replication of results in Charnes et. al (1994, p. 27)
x <- c(2, 3, 6, 9, 5, 4, 10)
y <- c(2, 5, 7, 8, 3, 1, 7)
data_example <- data.frame(dmus = letters[1:7], x, y)
data_example <- make_deadata(data_example,
                             ni = 1,
                             no = 1)
result <- model_additive(data_example,
                         rts = "vrs")
efficiencies(result)
slacks(result)
lambdas(result)

# Example 2.
# Measure of Inefficiency Proportions (MIP).
x <- c(2, 3, 6, 9, 5, 4, 10)
y <- c(2, 5, 7, 8, 3, 1, 7)
data_example <- data.frame(dmus = letters[1:7], x, y)
data_example <- make_deadata(data_example,
                             ni = 1,
                             no = 1)
result2 <- model_additive(data_example,
                          rts = "vrs",
                          weight_slack_i = 1 / data_example[["input"]],
                          weight_slack_o = 1 / data_example[["output"]])
slacks(result2)

# Example 3.
# Range Adjusted Measure of Inefficiencies (RAM).
x <- c(2, 3, 6, 9, 5, 4, 10)
y <- c(2, 5, 7, 8, 3, 1, 7)
data_example <- data.frame(dmus = letters[1:7], x, y)
data_example <- make_deadata(data_example,
                             ni = 1,
                             no = 1)
range_i <- apply(data_example[["input"]], 1, max) -
           apply(data_example[["input"]], 1, min)
range_o <- apply(data_example[["output"]], 1, max) -
           apply(data_example[["output"]], 1, min)
w_range_i <- 1 / (range_i * (dim(data_example[["input"]])[1] +
                             dim(data_example[["output"]])[1]))
w_range_o <- 1 / (range_o * (dim(data_example[["input"]])[1] +
                             dim(data_example[["output"]])[1]))
result3 <- model_additive(data_example,
                          rts = "vrs",
                          weight_slack_i = w_range_i,
                          weight_slack_o = w_range_o)
slacks(result3)
```

`model_addmin` *Additive-min DEA model.*

Description

Solve the weighted version of the additive-min (mADD) model of Aparicio et. al (2007) with different returns to scale. For non constant returns to scale, a modification given by Zhu et al. (2018) is done.

Usage

```
model_addmin(datadea,
               dmu_eval = NULL,
               dmu_ref = NULL,
               orientation = NULL,
               weight_slack_i = 1,
               weight_slack_o = 1,
               rts = c("crs", "vrs", "nirs", "ndrs"),
               method = c("mf", "milp"),
               extreff = NULL,
               M_d = NULL,
               M_lambda = 1e3,
               maxfr = NULL,
               tol = 1e-6,
               silent = TRUE,
               compute_target = TRUE,
               check_target = FALSE,
               returnlp = FALSE,
               ...)
```

Arguments

`datadea`	A `deadata` object with `n` DMUs, `m` inputs and `s` outputs.
`dmu_eval`	A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.
`dmu_ref`	A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.
`orientation`	This parameter is either `NULL` (default) or a string, equal to "io" (input-oriented) or "oo" (output-oriented). It is used to modify the weight slacks. If input-oriented, `weight_slack_o` are taken 0. If output-oriented, `weight_slack_i` are taken 0.
`weight_slack_i`	A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input slacks. If 0, output-oriented.
`weight_slack_o`	A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output slacks. If 0, input-oriented.

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing) or "ndrs" (non-decreasing). Under non-increasing or non-decreasing returns to scale, you may set `check_target = TRUE` because methods are not reliable. Generalized returns to scale are not available.

`method` A string with the method: "mf" (default) for maximal friends, or "milp" for the mixed integer linear program of Aparicio et al. (2007). MILP method is faster but very problematic numerically.

`extreff` A vector with the extreme efficient DMUs for "milp" method, as it is returned by function `extreme_efficient`. If `NULL` (default) this vector is computed internally.

`M_d` Numeric, a big positive quantity for "milp" method. It is an upper bound for auxiliary variables named "d" in Aparicio (2007). If `NULL` (default), it is estimated automatically. A very big value can produce catastrophic cancellations. If the results are not correct or the solver hangs, try to change its value.

`M_lambda` Numeric, a big positive quantity for "milp" method. It is an upper bound for lambda variables. A very big value can produce catastrophic cancellations. If the results are not correct or the solver hangs, try to change its value (1e3 by default).

`maxfr` A list with the maximal friends sets for "mf" method, as it is returned by function `maximal_friends`. If `NULL` (default) this list is computed internally.

`tol` Numeric, a tolerance margin for checking efficiency in `extreme_efficient` or `maximal_friends` functions, and for checking targets.

`silent` Logical. If `FALSE`, it prints all the messages from function `maximal_friends`.

`compute_target` Logical. If it is `TRUE` (default), it computes targets. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

`check_target` Logical. If it is `TRUE`, it checks the efficiency of targets. If a target is not efficient, the method has failed.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints).

`...` For compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Note

In this model, the efficiency score is the sum of the slacks. Therefore, a DMU is efficient when the objective value (`objval`) is zero.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Aparicio, J.; Ruiz, J.L.; Sirvent, I. (2007) "Closest targets and minimum distance to the Pareto-efficient frontier in DEA", Journal of Productivity Analysis, 28, 209-218. doi:10.1007/s11123007-00395

Zhu, Q.; Wu, J.; Ji, X.; Li, F. (2018) "A simple MILP to determine closest targets in non-oriented DEA model satisfying strong monotonicity", Omega, 79, 1-8. doi:10.1016/j.omega.2017.07.003

See Also

model_additive, extreme_efficient, maximal_friends

Examples

```
# Example 1.
data("Airlines")
datadea <- make_deadata(Airlines,
                        inputs = 4:7,
                        outputs = 2:3)
result <- model_addmin(datadea = datadea,
                       method = "milp")
targets(result)

## Not run:
# Example 2. Directional model with Additive-min model in second stage
data("Airlines")
datadea <- make_deadata(Airlines,
                        inputs = 4:7,
                        outputs = 2:3)
resdir <- model_basic(datadea = datadea,
                      orientation = "dir",
                      maxslack = FALSE)
proj_input <- targets(resdir)[[1]] + slacks(resdir)[[1]]
proj_output <- targets(resdir)[[2]] - slacks(resdir)[[2]]
nd <- ncol(datadea$dmunames) # Number of DMUs
maxfr <- maximal_friends(datadea = datadea)
for (i in 1:nd) {
  datadea2 <- datadea
  datadea2$input[, i] <- proj_input[i, ]
  datadea2$output[, i] <- proj_output[i, ]
  DMUaux <- model_addmin(datadea = datadea2,
                         method = "mf",
                         maxfr = maxfr,
                         dmu_eval = i)$DMU[[1]]
  resdir$DMU[[i]]$slack_input <- DMUaux$slack_input
  resdir$DMU[[i]]$slack_output <- DMUaux$slack_output
  resdir$DMU[[i]]$target_input <- DMUaux$target_input
  resdir$DMU[[i]]$target_output <- DMUaux$target_output
}
targets(resdir)

## End(Not run)
```

model_addsupereff *Additive super-efficiency DEA model.*

Description

Solve the additive super-efficiency model proposed by Du, Liang and Zhu (2010). It is an extension of the SBM super-efficiency to the additive DEA model.

Usage

```
model_addsupereff(datadea,
                  dmu_eval = NULL,
                  dmu_ref = NULL,
                  orientation = NULL,
                  weight_slack_i = NULL,
                  weight_slack_o = NULL,
                  rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
                  L = 1,
                  U = 1,
                  compute_target = TRUE,
                  returnlp = FALSE,
                  ...)
```

Arguments

`datadea` A `deadata` object with `n` DMUs, `m` inputs and `s` outputs.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`orientation` This parameter is either `NULL` (default) or a string, equal to "io" (input-oriented) or "oo" (output-oriented). It is used to modify the weight slacks. If input-oriented, `weight_slack_o` are taken 0. If output-oriented, `weight_slack_i` are taken 0.

`weight_slack_i` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input super-slacks (`t_input`). If 0, output-oriented. If `weight_slack_i` is the matrix of the inverses of inputs of DMUS in `dmu_eval` (default), the model is unit invariant.

`weight_slack_o` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output super-slacks (`t_output`). If 0, input-oriented. If `weight_slack_o` is the matrix of the inverses of outputs of DMUS in `dmu_eval` (default), the model is unit invariant.

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`compute_target` Logical. If it is `TRUE`, it computes targets, projections and slacks. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints).

`...` Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Du, J.; Liang, L.; Zhu, J. (2010). "A Slacks-based Measure of Super-efficiency in Data Envelopment Analysis. A Comment", European Journal of Operational Research, 204, 694-697. doi:10.1016/j.ejor.2009.12.007

Zhu, J. (2014). Quantitative Models for Performance Evaluation and Benchmarking. Data Envelopment Analysis with Spreadsheets. 3rd Edition Springer, New York. doi:10.1007/9783319066479

See Also

`model_additive`, `model_supereff`, `model_sbmsupereff`

Examples

```
# Replication of results in Du, Liang and Zhu (2010, Table 6, p.696)
data("Power_plants")
Power_plants <- make_deadata(Power_plants,
                             ni = 4,
                             no = 2)
result <- model_addsupereff(Power_plants,
                            rts = "crs")
efficiencies(result)
```

---

model_basic *Basic (radial and directional) DEA model.*

Description

It solves input and output oriented, along with directional, basic DEA models (envelopment form) under constant (CCR model), variable (BCC model), non-increasing, non-decreasing or generalized returns to scale. By default, models are solved in a two-stage process (slacks are maximized).

You can use the `model_basic` function to solve directional DEA models by choosing `orientation` = "dir".

The model_basic function allows to treat with non-discretional, non-controllable and undesirable inputs/outputs.

Usage

```
model_basic(datadea,
            dmu_eval = NULL,
            dmu_ref = NULL,
            orientation = c("io", "oo", "dir"),
            dir_input = NULL,
            dir_output = NULL,
            rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
            L = 1,
            U = 1,
            maxslack = TRUE,
            weight_slack_i = 1,
            weight_slack_o = 1,
            vtrans_i = NULL,
            vtrans_o = NULL,
            compute_target = TRUE,
            compute_multiplier = FALSE,
            returnlp = FALSE,
            silent_ud = FALSE,
            ...)
```

Arguments

`datadea` A `deadata` object with `n` DMUs, `m` inputs and `s` outputs.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`orientation` A string, equal to "io" (input oriented), "oo" (output oriented), or "dir" (directional).

`dir_input` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the input directions. If `dir_input` == input matrix (of DMUS in `dmu_eval`) and `dir_output` == 0, it is equivalent to input oriented (`beta` = 1 - `efficiency`). If `dir_input` is omitted, input matrix (of DMUS in `dmu_eval`) is assigned.

`dir_output` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the output directions. If `dir_input` == 0 and `dir_output` == output matrix (of DMUS in `dmu_eval`), it is equivalent to output oriented (`beta` = `efficiency` - 1). If `dir_output` is omitted, output matrix (of DMUS in `dmu_eval`) is assigned.

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`maxslack` Logical. If it is `TRUE`, it computes the max slack solution.

`weight_slack_i` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input slacks for the max slack solution.

`weight_slack_o` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output slacks for the max slack solution.

`vtrans_i` Numeric vector of translation for undesirable inputs with non-directional orientation. If `vtrans_i[i]` is `NA`, then it applies the "max + 1" translation to the i-th undesirable input. If `vtrans_i` is a constant, then it applies the same translation to all undesirable inputs. If `vtrans_i` is `NULL`, then it applies the "max + 1" translation to all undesirable inputs.

`vtrans_o` Numeric vector of translation for undesirable outputs with non-directional orientation, analogous to `vtrans_i`, but applied to outputs.

`compute_target` Logical. If it is `TRUE`, it computes targets of the max slack solution. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

`compute_multiplier` Logical. If it is `TRUE`, it computes multipliers (dual solution) when `orientation` is "io" or "oo". If there are undesirable variables, multipliers are only computed for variable returns to scale.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints) of stage 1.

`silent_ud` Logical. For internal use, to avoid multiple warnings in the execution of `malmquist_index` function with undesirable variables.

`...` Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Note

(1) Model proposed by Seiford and Zhu (2002) is applied for undesirable inputs/outputs and non-directional orientation (i.e., input or output oriented). You should select "vrs" returns to scale (BCC model) in order to maintain translation invariance. If deaR detects that you are not specifying `rts` = "vrs", it makes the change to "vrs" automatically.

(2) With undesirable inputs and non-directional orientation use input-oriented BCC model, and with undesirable outputs and non-directional orientation use output-oriented BCC model. Alternatively, you can also treat the undesirable outputs as inputs and then apply the input-oriented BCC model (similarly with undesirable inputs).

(3) Model proposed by Fare and Grosskopf (2004) is applied for undesirable inputs/outputs and directional orientation.

(4) With `orientation` = "dir" (directional distance function model), efficient DMUs are those for which `beta` = 0.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Charnes, A.; Cooper, W.W.; Rhodes, E. (1978). “Measuring the efficiency of decision making units”, European Journal of Operational Research 2, 429–444.

Charnes, A.; Cooper, W.W.; Rhodes, E. (1979). “Short communication: Measuring the efficiency of decision making units”, European Journal of Operational Research 3, 339.

Charnes, A.; Cooper, W.W.; Rhodes, E. (1981). "Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through", Management Science, 27(6), 668-697.

Banker, R.; Charnes, A.; Cooper, W.W. (1984). “Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis”, Management Science; 30; 1078-1092.

Undesirable inputs/outputs:

Pastor, J.T. (1996). "Translation Invariance in Data Envelopment Analysis: a Generalization", Annals of Operations Research, 66(2), 91-102.

Seiford, L.M.; Zhu, J. (2002). “Modeling undesirable factors in efficiency evaluation”, European Journal of Operational Research 142, 16-20.

Färe, R. ; Grosskopf, S. (2004). “Modeling undesirable factors in efficiency evaluation: Comment”, European Journal of Operational Research 157, 242-245.

Hua Z.; Bian Y. (2007). DEA with Undesirable Factors. In: Zhu J., Cook W.D. (eds) Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis. Springer, Boston, MA.

Non-discretionary/Non-controllable inputs/outputs:

Banker, R.; Morey, R. (1986). “Efficiency Analysis for Exogenously Fixed Inputs and Outputs”, Operations Research; 34; 513-521.

Ruggiero J. (2007). Non-Discretionary Inputs. In: Zhu J., Cook W.D. (eds) Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis. Springer, Boston, MA.

Directional DEA model:

Chambers, R.G.; Chung, Y.; Färe, R. (1996). "Benefit and Distance Functions", Journal of Economic Theory, 70(2), 407-419.

Chambers, R.G.; Chung, Y.; Färe, R. (1998). "Profit Directional Distance Functions and Nerlovian Efficiency", Journal of Optimization Theory and Applications, 95, 351-354.

See Also

model_multiplier, model_supereff

Examples

```
# Example 1. Basic DEA model with desirable inputs/outputs.
# Replication of results in Charnes, Cooper and Rhodes (1981).
data("PFT1981")
# Selecting DMUs in Program Follow Through (PFT)
PFT <- PFT1981[1:49, ]
PFT <- make_deadata(PFT,
                    inputs = 2:6,
                    outputs = 7:9)
eval_pft <- model_basic(PFT,
                        orientation = "io",
                        rts = "crs")
eff <- efficiencies(eval_pft)
s <- slacks(eval_pft)
lamb <- lambdas(eval_pft)
tar <- targets(eval_pft)
ref <- references(eval_pft)
returns <- rts(eval_pft)

# Example 2. Basic DEA model with undesirable outputs.
# Replication of results in Hua and Bian (2007).
data("Hua_Bian_2007")
# The third output is an undesirable output.
data_example <- make_deadata(Hua_Bian_2007,
                             ni = 2,
                             no = 3,
                             ud_outputs = 3)
# Translation parameter (vtrans_o) is set to 1500
```

```
result <- model_basic(data_example,
                      orientation = "oo",
                      rts = "vrs",
                      vtrans_o = 1500)
eff <- efficiencies(result)
1 / eff # results M5 in Table 6-5 (p.119)

# Example 3. Basic DEA model with non-discretionary (fixed) inputs.
# Replication of results in Ruggiero (2007), Table 5.1, BM column.
data("Ruggiero2007")
# The second input is a non-discretionary input.
datadea <- make_deadata(Ruggiero2007,
                        ni = 2,
                        no = 1,
                        nd_inputs = 2)
result <- model_basic(datadea,
                      orientation = "io",
                      rts = "crs")
efficiencies(result)
```

model_deaps *Preference Structure DEA model.*

Description

With this non-radial DEA model (Zhu, 1996), the user can specify the preference input (or output) weigths that reflect the relative degree of desirability of the adjustments of the current input (or output) levels.

Usage

```
model_deaps(datadea,
            dmu_eval = NULL,
            dmu_ref = NULL,
            weight_eff = 1,
            orientation = c("io", "oo"),
            rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
            L = 1,
            U = 1,
            restricted_eff = TRUE,
            maxslack = TRUE,
            weight_slack = 1,
            compute_target = TRUE,
            returnlp = FALSE,
            ...)
```

Arguments

`datadea` A `deadata` object, including `n` DMUs, `m` inputs and `s` outputs.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`weight_eff` Preference weights. If input-oriented, it is a value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the lenght of `dmu_eval`) with the weights applied to the input efficiencies. If output-oriented, it is a value, vector of length `s`, or matrix `s x ne` with the weights applied to the output efficiencies.

`orientation` A string, equal to "io" (input-oriented) or "oo" (output-oriented).

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`restricted_eff` Logical. If it is `TRUE`, the efficiencies are restricted to be <=1 (input-oriented) or >=1 (output-oriented).

`maxslack` Logical. If it is `TRUE`, it computes the max slack solution.

`weight_slack` If input-oriented, it is a value, vector of length `s`, or matrix `s x ne` with the weights of the output slacks for the max slack solution. If output-oriented, it is a value, vector of length `m`, or matrix `m x ne` with the weights of the input slacks for the max slack solution.

`compute_target` Logical. If it is `TRUE`, it computes targets of the max slack solution. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints) of stage 1.

`...` Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Author(s)

Vicente Coll-Serrano (`<vicente.coll@uv.es>`). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (`<vicente.bolos@uv.es>`). *Department of Business Mathematics*

Rafael Benítez (`<rafael.suarez@uv.es>`). *Department of Business Mathematics*

University of Valencia (Spain)

References

Zhu, J. (1996). “Data Envelopment Analysis with Preference Structure”, The Journal of the Operational Research Society, 47(1), 136. doi:10.2307/2584258

Zhu, J. (2014). Quantitative Models for Performance Evaluation and Benchmarking. Data Envelopment Analysis with Spreadsheets. 3rd Edition Springer, New York. doi:10.1007/9783319066479

See Also

model_nonradial, model_profit, model_sbmeff

Examples

```
data("Fortune500")
data_deaps <- make_deadata(datadea = Fortune500,
                           ni = 3,
                           no = 2)
result <- model_deaps(data_deaps,
                      weight_eff = c(1, 2, 3),
                      orientation = "io",
                      rts = "vrs")
efficiencies(result)
```

---

model_dir *Directional DEA model.*

Description

It solves directional, basic DEA models under constant, variable, non-increasing, non-decreasing or generalized returns to scale. By default, models are solved in a two-stage process (slacks are maximized).

Usage

```
model_dir(datadea,
            dmu_eval = NULL,
            dmu_ref = NULL,
            dir_input = NULL,
            dir_output = NULL,
            d_input = 1,
            d_output = 1,
            rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
            L = 1,
            U = 1,
            maxslack = TRUE,
            weight_slack_i = 1,
            weight_slack_o = 1,
            returnlp = FALSE,
            ...)
```

Arguments

`datadea` A `deadata` object with `n` DMUs, `m` inputs and `s` outputs.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`dir_input` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the input directions. If `dir_input` == input matrix (of DMUS in `dmu_eval`) and `dir_output` == 0, it is equivalent to input oriented (`beta` = 1 - `efficiency`). If `dir_input` is omitted, input matrix (of DMUS in `dmu_eval`) is assigned.

`dir_output` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the output directions. If `dir_input` == 0 and `dir_output` == output matrix (of DMUS in `dmu_eval`), it is equivalent to output oriented (`beta` = `efficiency` - 1). If `dir_output` is omitted, output matrix (of DMUS in `dmu_eval`) is assigned.

`d_input` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the input orientation parameters. If `d_input` == 1 (default) and `d_output` == 0, it is equivalent to input oriented. It is an alternative for `dir_input`.

`d_output` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the output orientation parameters. If `d_input` == 0 and `d_output` == 1 (default), it is equivalent to output oriented. It is an alternative to `dir_output`.

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`maxslack` Logical. If it is `TRUE`, it computes the max slack solution.

`weight_slack_i` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input slacks for the max slack solution.

`weight_slack_o` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output slacks for the max slack solution.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints) of stage 1.

`...` Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation_param`, `rts`, `dmu_eval` and `dmu_ref`.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Chambers, R.G.; Chung, Y.; Färe, R. (1996). "Benefit and Distance Functions", Journal of Economic Theory, 70(2), 407-419.

Chambers, R.G.; Chung, Y.; Färe, R. (1998). "Profit Directional Distance Functions and Nerlovian Efficiency", Journal of Optimization Theory and Applications, 95, 351-354.

See Also

`model_basic`, `model_lgo`, `model_qgo`

Examples

```
data("PFT1981")
# Selecting DMUs in Program Follow Through (PFT)
PFT <- PFT1981[1:49, ]
PFT <- make_deadata(PFT,
                    inputs = 2:6,
                    outputs = 7:9 )
eval_pft <- model_dir(PFT)
efficiencies(eval_pft)
```

`model_fdh` *Free disposal hull (FDH) model.*

Description

FDH model allows the free disposability to construct the production possibility set. The central feature of the FDH model is the lack of convexity for its production possibility set (Thrall, 1999).

Usage

```
model_fdh(datadea,
          fdh_modelname = c("basic"),
          ...)
```

Arguments

datadea	A deadata object, including DMUs, inputs and outputs.
fdh_modelname	A string containing the name of the model to apply FDH. For now, only "basic" is available.
...	dmu_eval, dmu_ref, orientation and other model parameters. Parameters like rts, max_slack and returnlp are ignored.

Value

A list of class dea with the results for the evaluated DMUs (DMU component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Cherchye, L.; Kuosmanen, T.; Post, T. (2000). "What Is the Economic Meaning of FDH? A Reply to Thrall". Journal of Productivity Analysis, 13(3), 263–267.

Deprins, D.; Simar, L. and Tulkens, H. (1984). Measuring Labor-Efficiency in Post Offices. In M. Marchand, P. Pestieau and H. Tulkens (eds.), The Performance of Public Entreprises: Concepts and Measurement. Amsterdam: North-Holland.

Sanei, M.; Mamizadeh Chatghayeb, S. (2013). "Free Disposal Hull Models in Supply Chain Management", International Journal of Mathematical Modelling and Computations, 3(3), 125-129.

Thrall, R. M. (1999). "What Is the Economic Meaning of FDH?", Journal of Productivity Analysis, 11(3), 243–50.

Examples

```
# Example 1. FDH input-oriented.
# Replication of results in Sanei and Mamizadeh Chatghayeb (2013)
data("Supply_Chain")
data_fdh1 <- make_deadata(Supply_Chain,
                          inputs = 2:4,
                          outputs = 5:6)
result <- model_fdh(data_fdh1) # by default orientation = "io"
efficiencies(result)

# Example 2. FDH output-oriented.
# Replication of results in Sanei and Mamizadeh Chatghayeb (2013)
data("Supply_Chain")
```

```
data_fdh2 <- make_deadata(Supply_Chain,
                          inputs = 5:6,
                          outputs = 7:8)
result2 <- model_fdh(data_fdh2,
                     orientation = "oo")
efficiencies(result2)
```

model_lgo *Linear Generalized Oriented DEA model.*

Description

It solves linear generalized oriented DEA models (see Bolós et al. 2026). By default, models are solved in a two-stage process (slacks are maximized).

Usage

```
model_lgo(datadea,
          dmu_eval = NULL,
          dmu_ref = NULL,
          d_input = 1,
          d_output = 1,
          rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
          L = 1,
          U = 1,
          maxslack = TRUE,
          weight_slack_i = 1,
          weight_slack_o = 1,
          returnlp = FALSE,
          ...)
```

Arguments

datadea	A deadata object with n DMUs, m inputs and s outputs.
dmu_eval	A numeric vector containing which DMUs have to be evaluated. If NULL (default), all DMUs are considered.
dmu_ref	A numeric vector containing which DMUs are the evaluation reference set. If NULL (default), all DMUs are considered.
d_input	A value, vector of length m, or matrix m x ne (where ne is the length of dmu_eval) with the input orientation parameters. If d_input == 1 (default) and d_output == 0, it is equivalent to input oriented.
d_output	A value, vector of length s, or matrix s x ne (where ne is the length of dmu_eval) with the output orientation parameters. If d_input == 0 and d_output == 1 (default), it is equivalent to output oriented.

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`maxslack` Logical. If it is `TRUE`, it computes the max slack solution.

`weight_slack_i` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input slacks for the max slack solution.

`weight_slack_o` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output slacks for the max slack solution.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints) of stage 1.

`...` Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation_param`, `rts`, `dmu_eval` and `dmu_ref`.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Bolós, V.J.; Benítez, R.; Coll-Serrano, V (2026). "A new family of models with generalized orientation in data envelopment analysis". International Transactions in Operational Research. doi:10.1111/itor.70063

Chambers, R.G.; Chung, Y.; Färe, R. (1996). "Benefit and Distance Functions", Journal of Economic Theory, 70(2), 407-419.

Chambers, R.G.; Chung, Y.; Färe, R. (1998). "Profit Directional Distance Functions and Nerlovian Efficiency", Journal of Optimization Theory and Applications, 95, 351-354.

See Also

`model_basic`, `model_dir`, `model_qgo`

Examples

```
data("PFT1981")
# Selecting DMUs in Program Follow Through (PFT)
PFT <- PFT1981[1:49, ]
PFT <- make_deadata(PFT,
                    inputs = 2:6,
                    outputs = 7:9 )
eval_pft <- model_lgo(PFT)
efficiencies(eval_pft)
```

model_multiplier *Multiplier DEA model*

Description

Solve input-oriented and output-oriented basic DEA models (multiplicative form) under constant (CCR DEA model), variable (BCC DEA model), non-increasing, non-decreasing or generalized returns to scale. It does not take into account non-controllable, non-discretionary or undesirable inputs/outputs.

Usage

```
model_multiplier(datadea,
                 dmu_eval = NULL,
                 dmu_ref = NULL,
                 epsilon = 0,
                 orientation = c("io", "oo"),
                 rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
                 L = 1,
                 U = 1,
                 returnlp = FALSE,
                 compute_lambda = TRUE,
                 ...)
```

Arguments

`datadea`	A `deadata` object, including DMUs, inputs and outputs.
`dmu_eval`	A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.
`dmu_ref`	A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.
`epsilon`	Numeric, multipliers must be >= `epsilon`.
`orientation`	A string, equal to "io" (input-oriented) or "oo" (output-oriented).
`rts`	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints).

`compute_lambda` Logical. If it is `TRUE`, it computes the dual problem and lambdas.

`...` Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Note

(1) Very important with the multiplier model: "The optimal weights for an efficient DMU need not be unique" (Cooper, Seiford and Tone, 2007:31). "Usually, the optimal weights for inefficient DMUs are unique, the exception being when the line of the DMU is parallel to one of the boundaries of the feasible region" (Cooper, Seiford and Tone, 2007:32).

(2) The measure of technical input (or output) efficiency obtained by using multiplier DEA models is better the smaller the value of epsilon.

(3) Epsilon is usually set equal to 10^-6. However, if epsilon is not set correctly, the multiplier model can be infeasible (Zhu,2014:49).

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Charnes, A.; Cooper, W.W. (1962). “Programming with Linear Fractional Functionals”, Naval Research Logistics Quarterly 9, 181-185. doi:10.1002/nav.3800090303

Charnes, A.; Cooper, W.W.; Rhodes, E. (1978). “Measuring the Efficiency of Decision Making Units”, European Journal of Operational Research 2, 429–444. doi:10.1016/03772217(78)901388

Charnes, A.; Cooper, W.W.; Rhodes, E. (1979). “Short Communication: Measuring the Efficiency of Decision Making Units”, European Journal of Operational Research 3, 339. doi:10.1016/0377-2217(79)902297

Golany, B.; Roll, Y. (1989). "An Application Procedure for DEA", OMEGA International Journal of Management Science, 17(3), 237-250. doi:10.1016/03050483(89)900297

Seiford, L.M.; Thrall, R.M. (1990). “Recent Developments in DEA. The Mathematical Programming Approach to Frontier Analysis”, Journal of Econometrics 46, 7-38. doi:10.1016/03044076(90)90045-U

Zhu, J. (2014). Quantitative Models for Performance Evaluation and Benchmarking. Data Envelopment Analysis with Spreadsheets. 3rd Edition Springer, New York. doi:10.1007/9783319066479

See Also

`model_basic`, `cross_efficiency`

Examples

```
# Example 1.
# Replication of results in Golany and Roll (1989).
data("Golany_Roll_1989")
data_example <- make_deadata(datadea = Golany_Roll_1989[1:10, ],
                             inputs = 2:4,
                             outputs = 5:6)
result <- model_multiplier(data_example,
                           epsilon = 0,
                           orientation = "io",
                           rts = "crs")
efficiencies(result)
multipliers(result)

# Example 2.
# Multiplier model with infeasible solutions (See note).
data("Fortune500")
data_Fortune <- make_deadata(datadea = Fortune500,
                             inputs = 2:4,
                             outputs = 5:6)
result2 <- model_multiplier(data_Fortune,
                           epsilon = 1e-6,
                           orientation = "io",
                           rts = "crs")
# Results for General Motors and Ford Motor are not shown by deaR
# because the solution is infeasible.
efficiencies(result2)
multipliers(result2)
```

`model_nonradial` *Non-radial DEA model.*

Description

Non-radial DEA model allows for non-proportional reductions in each input or augmentations in each output.

Usage

```
model_nonradial(datadea,
                dmu_eval = NULL,
                dmu_ref = NULL,
                orientation = c("io", "oo"),
                rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
                L = 1,
                U = 1,
                maxslack = TRUE,
                weight_slack = 1,
                compute_target = TRUE,
                returnlp = FALSE,
                ...)
```

Arguments

datadea	A deadata object, including n DMUs, m inputs and s outputs.
dmu_eval	A numeric vector containing which DMUs have to be evaluated. If NULL (default), all DMUs are considered.
dmu_ref	A numeric vector containing which DMUs are the evaluation reference set. If NULL (default), all DMUs are considered.
orientation	A string, equal to "io" (input-oriented) or "oo" (output-oriented).
rts	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).
L	Lower bound for the generalized returns to scale (grs).
U	Upper bound for the generalized returns to scale (grs).
maxslack	Logical. If it is TRUE, it computes the max slack solution.
weight_slack	If input-oriented, it is a value, vector of length s, or matrix s x ne (where ne is the length of dmu_eval) with the weights of the output slacks for the max slack solution. If output-oriented, it is a value, vector of length m, or matrix m x ne with the weights of the input slacks for the max slack solution.
compute_target	Logical. If it is TRUE, it computes targets of the max slack solution. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.
returnlp	Logical. If it is TRUE, it returns the linear problems (objective function and constraints) of stage 1.
...	Ignored, for compatibility issues.

Value

A list of class dea with the results for the evaluated DMUs (DMU component), along with any other necessary information to replicate the results, such as the name of the model and parameters orientation, rts, dmu_eval and dmu_ref.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Banker, R.D.; Morey, R.C. (1986). "Efficiency Analysis for Exogenously Fixed Inputs and Outputs", Operations Research, 34, 80-97. doi:10.1287/opre.34.4.513

Färe, R.; Lovell, C.K. (1978). "Measuring the Technical Efficiency of Production", Journal of Economic Theory, 19(1), 150-162. doi:10.1016/00220531(78)900601

Wu, J.; Tsai, H.; Zhou, Z. (2011). "Improving Efficiency in International Tourist Hotels in Taipei Using a Non-Radial DEA Model", International Journal of Contemporary Hospitatlity Management, 23(1), 66-83. doi:10.1108/09596111111101670

Zhu, J. (1996). “Data Envelopment Analysis with Preference Structure”, The Journal of the Operational Research Society, 47(1), 136. doi:10.2307/2584258

See Also

`model_deaps`, `model_profit`, `model_sbmeff`

Examples

```
# Replication of results in Wu, Tsai and Zhou (2011)
data("Hotels")
data_hotels <- make_deadata(Hotels,
                            inputs = 2:5,
                            outputs = 6:8)
result <- model_nonradial(data_hotels,
                          orientation = "oo",
                          rts = "vrs")
efficiencies(result)
```

`model_profit` *Profit efficiency DEA model.*

Description

Cost, revenue and profit efficiency DEA models.

Usage

```
model_profit(datadea,
             dmu_eval = NULL,
             dmu_ref = NULL,
             price_input = NULL,
             price_output = NULL,
             rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
             L = 1,
             U = 1,
             restricted_optimal = TRUE,
             returnlp = FALSE,
             ...)
```

Arguments

`datadea`	A `deadata` object, including `n` DMUs, `m` inputs and `s` outputs.
`dmu_eval`	A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.
`dmu_ref`	A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.
`price_input`	Unit prices of inputs for cost or profit efficiency models. It is a value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`).
`price_output`	Unit prices of outputs for revenue or profit efficiency models. It is a value, vector of length `s`, or matrix `s` x `ne`.
`rts`	A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).
`L`	Lower bound for the generalized returns to scale (grs).
`U`	Upper bound for the generalized returns to scale (grs).
`restricted_optimal`	Logical. If it is `TRUE`, the optimal inputs are restricted to be <= inputs (for cost efficiency models) or the optimal outputs are restricted to be >= outputs (for revenue efficiency models).
`returnlp`	Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints) of stage 1.
`...`	Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Coelli, T.; Prasada Rao, D.S.; Battese, G.E. (1998). An introduction to efficiency and productivity analysis. Jossey-Bass, San Francisco, pp 73–104. doi:10.1002/ev.1441

See Also

model_deaps, model_nonradial, model_sbmeff

Examples

```
# Example 1. Replication of results in Coelli et al. (1998, p.166).
# Cost efficiency model.
data("Coelli_1998")
# Selection of prices: input_prices is the transpose where the prices for inputs are.
input_prices <- t(Coelli_1998[, 5:6])

data_example1 <- make_deadata(Coelli_1998,
                              ni = 2,
                              no = 1)
result1 <- model_profit(data_example1,
                       price_input = input_prices,
                       rts = "crs",
                       restricted_optimal = FALSE)
# notice that the option by default is restricted_optimal = TRUE
efficiencies(result1)

# Example 2. Revenue efficiency model.
data("Coelli_1998")
# Selection of prices for output: output_prices is the transpose where the prices for outputs are.
output_prices <- t(Coelli_1998[, 7])
data_example2 <- make_deadata(Coelli_1998,
                             ni = 2,
                             no = 1)
result2 <- model_profit(data_example2,
                       price_output = output_prices,
                       rts = "crs",
                       restricted_optimal = FALSE)
# notice that the option by default is restricted_optimal = TRUE
efficiencies(result2)

# Example 3. Profit efficiency model.
data("Coelli_1998")
```

```
# Selection of prices for inputs and outputs: input_prices and output_prices are
# the transpose where the prices (for inputs and outputs) are.
input_prices <- t(Coelli_1998[, 5:6])
output_prices <- t(Coelli_1998[, 7])
data_example3 <- make_deadata(Coelli_1998,
                              ni = 2,
                              no = 1)
result3 <- model_profit(data_example3,
                        price_input = input_prices,
                        price_output = output_prices,
                        rts = "crs",
                        restricted_optimal = FALSE)
# notice that the option by default is restricted_optimal = TRUE
efficiencies(result3)
```

model_qgo *Quadratically Constrained CRS Generalized Oriented DEA model.*

Description

It solves quadratically constrained CRS generalized oriented DEA models (see Bolós et al. 2026), using alabama solver. By default, models are solved in a two-stage process (slacks are maximized).

Usage

```
model_qgo(datadea,
             dmu_eval = NULL,
             dmu_ref = NULL,
             d_input = 1,
             d_output = 1,
             rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
             L = 1,
             U = 1,
             give_X = TRUE,
             n_attempts_max = 5,
             force_quad = FALSE,
             maxslack = TRUE,
             weight_slack_i = 1,
             weight_slack_o = 1,
             returnqp = FALSE,
             ...)
```

Arguments

datadea	A deadata object with n DMUs, m inputs and s outputs.
dmu_eval	A numeric vector containing which DMUs have to be evaluated. If NULL (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`d_input` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the input orientation parameters. If `d_input` == 1 (default) and `d_output` == 0, it is equivalent to input oriented.

`d_output` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the output orientation parameters. If `d_input` == 0 and `d_output` == 1 (default), it is equivalent to output oriented.

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`give_X` Logical. If it is `TRUE` (default), it uses an initial vector (given by the evaluated DMU) for the solver. If it is `FALSE`, the initial vector is given internally by the solver and it is usually randomly generated.

`n_attempts_max` A value with the maximum number of attempts if the solver does not converge. Each attempt uses a different initial vector.

`force_quad` Logical. If it is `FALSE` (default) it uses the linear model `model_lgo` to get the results, in some particular cases in which the results of `model_qgo` can be deduced from the results of `model_lgo`, see V. J. Bolós et al. (2026).

`maxslack` Logical. If it is `TRUE`, it computes the max slack solution.

`weight_slack_i` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input slacks for the max slack solution.

`weight_slack_o` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output slacks for the max slack solution.

`returnqp` Logical. If it is `TRUE`, it returns the quadratic problems (objective function and constraints) of stage 1.

`...` Other parameters to be passed to the solver `solvecop` from package `optiSolve`.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters `d_input`, `d_output`, `rts`, `dmu_eval` and `dmu_ref`.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Bolós, V.J.; Benítez, R.; Coll-Serrano, V (2026). "A new family of models with generalized orientation in data envelopment analysis". International Transactions in Operational Research. doi:10.1111/itor.70063

See Also

model_basic, model_dir, model_lgo

Examples

```
data("PFT1981")
# Selecting DMUs in Program Follow Through (PFT)
PFT <- PFT1981[1:49, ]
PFT <- make_deadata(PFT,
                    inputs = 2:6,
                    outputs = 7:9 )
eval_pft <- model_qgo(PFT, dmu_eval = 1:5)
efficiencies(eval_pft)
```

---

model_rdm *Range directional model.*

Description

Range directional model from Portela et al. (2004).

Usage

```
model_rdm(datadea,
            dmu_eval = NULL,
            dmu_ref = NULL,
            orientation = c("no", "io", "oo"),
            irdm = FALSE,
            maxslack = TRUE,
            weight_slack_i = 1,
            weight_slack_o = 1,
            compute_target = TRUE,
            returnlp = FALSE,
            ...)
```

Arguments

datadea	A deadata object, including n DMUs, m inputs and s outputs.
dmu_eval	A numeric vector containing which DMUs have to be evaluated. If NULL (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`orientation` A string, equal to "no" (non-oriented), "io" (input oriented), or "oo" (output oriented).

`irdm` Logical. If it is `TRUE`, it applies the IRDM (inverse range directional model).

`maxslack` Logical. If it is `TRUE`, it computes the max slack solution.

`weight_slack_i` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the input slacks for the max slack solution.

`weight_slack_o` A value, vector of length `s`, or matrix `s` x `ne` (where `ne` is the length of `dmu_eval`) with the weights of the output slacks for the max slack solution.

`compute_target` Logical. If it is `TRUE`, it computes targets of the max slack solution. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints) of stage 1.

`...` Ignored, for compatibility issues.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Note

Undesirable inputs/outputs are treated as negative inputs/outputs in this model.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Portela, M.; Thanassoulis, E.; Simpson, G. (2004). "Negative data in DEA: a directional distance approach applied to bank branches", Journal of the Operational Research Society, 55 1111-1121.

`model_sbmeff` *Slack based measure (SBM) of efficiency model.*

Description

Calculate the SBM model proposed by Tone (2001).

Usage

```
model_sbmeff(datadea,
             dmu_eval = NULL,
             dmu_ref = NULL,
             weight_input = 1,
             weight_output = 1,
             orientation = c("no", "io", "oo"),
             rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
             L = 1,
             U = 1,
             kaizen = FALSE,
             maxfr = NULL,
             tol = 1e-6,
             silent = FALSE,
             compute_target = TRUE,
             returnlp = FALSE,
             ...)
```

Arguments

`datadea` A `deadata` object with `n` DMUs, `m` inputs and `s` outputs.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`weight_input` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with weights to inputs corresponding to the relative importance of items.

`weight_output` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with weights to outputs corresponding to the relative importance of items.

`orientation` A string, equal to "no" (non-oriented), "io" (input-oriented) or "oo" (output-oriented).

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

kaizen Logical. If TRUE, the kaizen version of SBM (Tone 2010), also known as SBM-Max, is computed.

maxfr A list with the maximal friends sets, as it is returned by function maximal_friends. If NULL (default) this list is computed internally.

tol Numeric, a tolerance margin for checking efficiency (only for the kaizen version).

silent Logical. If FALSE (default) it prints all the messages from function maximal_friends.

compute_target Logical. If it is TRUE, it computes targets. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

returnlp Logical. If it is TRUE, it returns the linear problems (objective function and constraints). If kaizen is TRUE it is ignored.

... Other options (currently not implemented)

Value

A list of class dea with the results for the evaluated DMUs (DMU component), along with any other necessary information to replicate the results, such as the name of the model and parameters orientation, rts, dmu_eval and dmu_ref.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Tone, K. (2001). "A Slacks-Based Measure of Efficiency in Data Envelopment Analysis", European Journal of Operational Research, 130, 498-509. doi:10.1016/S03772217(99)004075

Tone, K. (2010). "Variations on the theme of slacks-based measure of efficiency in DEA", European Journal of Operational Research, 200, 901-907. doi:10.1016/j.ejor.2009.01.027

Cooper, W.W.; Seiford, L.M.; Tone, K. (2007). Data Envelopment Analysis. A Comprehensive Text with Models, Applications, References and DEA-Solver Software. 2nd Edition. Springer, New York. doi:10.1007/9780387452838

Aparicio, J.; Ruiz, J.L.; Sirvent, I. (2007) "Closest targets and minimum distance to the Pareto-efficient frontier in DEA", Journal of Productivity Analysis, 28, 209-218. doi:10.1007/s11123007-00395

See Also

model_nonradial, model_deaps, model_profit, model_sbmsupereff

Examples

```
# Example 1. Replication of results in Tone (2001, p.505)
data("Tone2001")
data_example <- make_deadata(Tone2001,
                             ni = 2,
                             no = 2)
result_SBM <- model_sbmeff(data_example,
                           orientation = "no",
                           rts = "crs")
result_CCR <- model_basic(data_example,
                          orientation = "io",
                          rts = "crs")
efficiencies(result_SBM)
efficiencies(result_CCR)
slacks(result_SBM)
slacks(result_CCR)

# Example 2. Replication of results in Tone (2003), pp 10-11 case 1:1.
data("Tone2003")
data_example <- make_deadata(Tone2003,
                             ni = 1,
                             no = 2,
                             ud_outputs = 2)
result <- model_sbmeff(data_example,
                       rts = "vrs")
efficiencies(result)
targets(result)

# Example 3. Replication of results in Aparicio (2007).
data("Airlines")
datadea <- make_deadata(Airlines,
                        inputs = 4:7,
                        outputs = 2:3)
result <- model_sbmeff(datadea = datadea, kaizen = TRUE)
efficiencies(result)
targets(result)
```

---

model_sbmsupereff *Slack based measure of superefficiency model*

Description

Slack based measure of superefficiency model (Tone 2002) with n DMUs, m inputs and s outputs.

Usage

```
model_sbmsupereff(datadea,
                  dmu_eval = NULL,
                  dmu_ref = NULL,
                  weight_input = 1,
                  weight_output = 1,
                  orientation = c("no", "io", "oo"),
                  rts = c("crs", "vrs", "nirs", "ndrs", "grs"),
                  L = 1,
                  U = 1,
                  compute_target = TRUE,
                  compute_rho = FALSE,
                  kaizen = FALSE,
                  silent = FALSE,
                  returnlp = FALSE)
```

Arguments

`datadea` A `deadata` object, including DMUs, inputs and outputs.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`weight_input` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with weights to inputs corresponding to the relative importance of items.

`weight_output` A value, vector of length `m`, or matrix `m` x `ne` (where `ne` is the length of `dmu_eval`) with weights to outputs corresponding to the relative importance of items.

`orientation` A string, equal to "no" (non-oriented), "io" (input-oriented) or "oo" (output-oriented).

`rts` A string, determining the type of returns to scale, equal to "crs" (constant), "vrs" (variable), "nirs" (non-increasing), "ndrs" (non-decreasing) or "grs" (generalized).

`L` Lower bound for the generalized returns to scale (grs).

`U` Upper bound for the generalized returns to scale (grs).

`compute_target` Logical. If it is `TRUE`, it computes targets, superslacks (`t_input` and `t_output`) and slacks. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

`compute_rho` Logical. If it is `TRUE`, it computes the SBM efficiency score (applying `model_sbmeff`) of the DMU (`project_input`, `project_output`).

`kaizen` Logical. If `TRUE`, the kaizen version of SBM (Tone 2010), also known as SBM-Max, is computed for the efficiency score of the DMU (`project_input`, `project_output`).

`silent` Logical. If `FALSE` (default) it prints all the messages from function `maximal_friends`.

`returnlp` Logical. If it is `TRUE`, it returns the linear problems (objective function and constraints).

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Tone, K. (2002). "A slacks-based measure of super-efficiency in data envelopment analysis", European Journal of Operational Research, 143, 32-41. doi:10.1016/S03772217(01)003241

Tone, K. (2010). "Variations on the theme of slacks-based measure of efficiency in DEA", European Journal of Operational Research, 200, 901-907. doi:10.1016/j.ejor.2009.01.027

Cooper, W.W.; Seiford, L.M.; Tone, K. (2007). Data Envelopment Analysis. A Comprehensive Text with Models, Applications, References and DEA-Solver Software. 2nd Edition. Springer, New York. doi:10.1007/9780387452838

See Also

model_sbmeff, model_supereff, model_addsupereff

Examples

```
# Replication of results in Tone(2002, p.39)
data("Power_plants")
data_example <- make_deadata(Power_plants,
                             ni = 4,
                             no = 2)
result <- model_sbmsupereff(data_example,
                            orientation = "io",
                            rts = "crs")
efficiencies(result)
slacks(result)$slack_input
references(result)
```

model_supereff *Radial super-efficiency basic DEA model*

Description

Solve Andersen and Petersen radial Super-efficiency DEA model.

Usage

```
model_supereff(datadea,
               dmu_eval = NULL,
               dmu_ref = NULL,
               supereff_modelname = c("basic"),
               ...)
```

Arguments

`datadea` An object of class `deadata`.

`dmu_eval` A numeric vector containing which DMUs have to be evaluated. If `NULL` (default), all DMUs are considered.

`dmu_ref` A numeric vector containing which DMUs are the evaluation reference set. If `NULL` (default), all DMUs are considered.

`supereff_modelname` A string containing the name of the radial model to apply super-efficiency.

`...` `orientation`, `rts` and other model parameters.

Value

A list of class `dea` with the results for the evaluated DMUs (`DMU` component, we note that we call "targets" to the "efficient projections" in the strongly efficient frontier), along with any other necessary information to replicate the results, such as the name of the model and parameters `orientation`, `rts`, `dmu_eval` and `dmu_ref`.

Note

(1) Radial super-efficiency model under variable (vrs, nirs, ndrs, grs) returns to scale can be infeasible for certain DMUs. See example 2.

(2) DMUs with infeasible solution are not shown in the results.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Andersen, P.; Petersen, N.C. (1993). "A procedure for ranking efficient units in data envelopment analysis", Management Science, 39, 1261-1264.

Tone, K. (2002). "A slacks-based measure of super-efficiency in data envelopment analysis", European Journal of Operational Research, 143, 32-41.

See Also

model_basic, model_sbmsupereff, model_addsupereff

Examples

```
# Example 1.
# Replication of results in Tone (2002, p.38)
data("Power_plants")
data_example <- make_deadata(Power_plants,
                             ni = 4,
                             no = 2)
result <- model_supereff(data_example,
                         orientation = "io",
                         rts = "crs")
eff <- efficiencies(result)

# Example 2.
# Results of Super-efficiency with vrs returns to scale show infeasibility solutions
# for DMUs D4 and D6 (these DMUs are not shown in deaR results).
data("Power_plants")
data_example2 <- make_deadata(Power_plants,
                              ni = 4,
                              no = 2)
result2 <- model_supereff(data_example2,
                          orientation = "io",
                          rts = "vrs")
eff2 <- efficiencies(result2)
```

multipliers *Multipliers*

Description

Extract the multipliers of the DMUs from a dea or dea_fuzzy solution.

Usage

```
multipliers(deasol)
```

Arguments

deasol	Object of class dea or dea_fuzzy obtained with some of the DEA model functions.

Value

A list with the multipliers of the evaluated DMUs.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
data("Coll_Blasco_2006")
data_example <- make_deadata(Coll_Blasco_2006,
                             ni = 2,
                             no = 2)
result <- model_multiplier(data_example,
                           orientation = "io",
                           rts = "crs")
multipliers(result)
```

PFT1981 *Data: Charnes, Cooper and Rhodes (1981).*

Description

Data from Project Follow Through (PTF) in public school education. There are 49 DMUs (school sites) in PFT and 21 DMUs in Non-Follow Through (NFT). Authors consider 3 outputs (Y) and 5 inputs (X).

Usage

```
data("PFT1981")
```

Format

Data frame with 70 rows and 10 columns. Definition of inputs (X) and outputs (Y):

Y1 = Reading Total Reading Scores (as measured by the Metropolitan Achievement Test).

Y2 = Math Total Math Scores (total mathematics score by the Metropolitan Achievement Test.

Y3 = Coopersmith Total Coopersmith Scores (Coopersmith self-esteem inventory, intended as a measure of self-esteem).

X1 = Education Education level of mother (as measured in terms of percentage of high school graduates among female parents).

X2 = Occupation Occupation Index (highest occupation of a family member according to a prearranged rating scale).

X3 = Parental Parental Visit Index (representing the number of visits to the school site).

X4 = Counseling Counseling Index (parent counselling index calculated from data on time spent with child on school-related topics such as reading together, etc.).

X5 = Teachers Number of Teachers (number of teachers at a given site).

Program PFT or NFT.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Charnes, A.; Cooper, W.W.; Rhodes, E. (1981). "Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through", Management Science, 27(6), 668-697. doi:10.1287/mnsc.27.6.668

See Also

make_deadata, model_basic

Examples

```
# Example 1. Replication of results in Charnes, Cooper and Rhodes (1981)
data("PFT1981")
# selecting DMUs in Project Follow Through (PFT)
PFT <- PFT1981[1:49, ]
PFT <- make_deadata(PFT,
                    dmus = 1,
                    inputs = 2:6,
                    outputs = 7:9 )
eval_pft <- model_basic(PFT,
                        orientation = "io",
                        rts = "crs")
eff_pft <- efficiencies(eval_pft)

# Example 2. Replication of results in Charnes, Cooper and Rhodes (1981)
data("PFT1981")
# selecting DMUs in Non-Follow Through (NFT)
NFT <- PFT1981[50:70,]
NFT <- make_deadata(NFT,
                    dmus = 1,
                    inputs = 2:6,
                    outputs = 7:9 )
eval_nft <- model_basic(NFT,
                        orientation = "io",
                        rts = "crs")
eff_nft <- efficiencies(eval_nft)
```

plot.dea *Plot for DEA models.*

Description

Plot some attribute of a DEA model.

Usage

```
## S3 method for class 'dea'
plot(x, tol = 1e-04, showPlots = TRUE, ...)
```

Arguments

`x`	An object of class `dea` obtained by a DEA model function.
`tol`	Numeric. Absolute tolerance for numeric comparisons. By default, it is 1e-4.
`showPlots`	Logical. When TRUE (default) the plots are shown one by one. When it is FALSE the plots are not shown and are returned by the function (invisibly) as a list.
`...`	Ignored, for compatibility issues.

Value

Depending on the model, it returns some plots.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Zhu, J. (2014). Quantitative Models for Performance Evaluation and Benchmarking. Data Envelopment Analysis with Spreadsheets. 3rd Edition Springer, New York. doi:10.1007/9783319066479

Examples

```
data_example <- make_deadata(datadea = Fortune500,
                             inputs = 2:4,
                             outputs = 5:6)
result <- model_basic(data_example)
plot(result)
```

`plot.dea_fuzzy` *Plot for fuzzy DEA models.*

Description

Plot some attributes of a fuzzy DEA model (Guo-Tanaka, Kao-Liu and possibilistic models).

Usage

```
## S3 method for class 'dea_fuzzy'
plot(x, showPlots = TRUE, ...)
```

Arguments

`x`	An object of class `dea_fuzzy` obtained by a fuzzy DEA model function.
`showPlots`	Logical. When TRUE (default) the plots are shown one by one. When it is FALSE the plots are not shown and are returned by the function (invisibly) as a list.
`...`	Ignored, for compatibility issues.

Value

Depending on the model, it returns some plots.

Author(s)

Vicente Coll-Serrano (`<vicente.coll@uv.es>`). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (`<vicente.bolos@uv.es>`). *Department of Business Mathematics*

Rafael Benítez (`<rafael.suarez@uv.es>`). *Department of Business Mathematics*

University of Valencia (Spain)

References

Zhu, J. (2014). Quantitative Models for Performance Evaluation and Benchmarking. Data Envelopment Analysis with Spreadsheets. 3rd Edition Springer, New York. doi:10.1007/9783319066479

Power_plants *Data: Tone (2002).*

Description

This dataset consists of six power plants with 4 inputs (X) and 2 outputs (Y).

Usage

```
data("Power_plants")
```

Format

Data frame with 15 rows and 7 columns. Definition of inputs (X) and outputs (Y):

x1 Manpower requiered

x2 Construction costs in millions of dollars

x3 Annual maintenance costs in millions of dollars

x4 Number of villages to be evacuated

y1 Power generated in megawatts

y2 Safety level

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Andersen, P.; Petersen, N.C. (1993). "A procedure for ranking efficient units in data envelopment analysis", Management Science, 39, 1261-1264.

Doyle, J. and Green R. (1993). "Data envelopment analysis and multiple criteria decision making", Omega, 21 (6), 713-715. doi:10.1016/03050483(93)90013B

Tone, K. (2002). "A slacks-based measure of super-efficiency in data envelopment analysis", European Journal of Operational Research, 143, 32-41. doi:10.1016/S03772217(01)003241

See Also

make_deadata, model_supereff, model_sbmsupereff

Examples

```
# Example 1. Radial super-efficiency model.
# Replication of results in Tone (2002)
data("Power_plants")
data_example <- make_deadata(Power_plants,
                             ni = 4,
                             no = 2)
result <- model_supereff(data_example,
                         orientation = "io",
                         rts = "crs")
eff <- efficiencies(result)
eff

# Example 2. SBM super-efficiency model.
data("Power_plants")
data_example <- make_deadata(Power_plants,
                             ni = 4,
                             no = 2)
result2 <- model_sbmsupereff(data_example,
                             orientation = "io",
                             rts = "crs")
efficiencies(result2)
slacks(result2)$input
references(result2)
```

print.deadata *deadata class print method*

Description

Print method for deadata class.

Usage

```
## S3 method for class 'deadata'
print(x, ...)
```

Arguments

x	A deadata object (as returned by make_deadata function).
...	For compatibility issues.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

print.deadata_fuzzy *deadata_fuzzy class print method*

Description

Print method for deadata_fuzzy class.

Usage

```
## S3 method for class 'deadata_fuzzy'
print(x, ...)
```

Arguments

x	A deadata_fuzzy object (as returned by make_deadata_fuzzy function).
...	For compatibility issues.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

read_data *read_data*

Description

This function is deprecated. Use make_deadata instead.

Usage

```
read_data(datadea = NULL,
          ni = NULL,
          no = NULL,
          dmus = 1,
          inputs = NULL,
          outputs = NULL,
          nc_inputs = NULL,
          nc_outputs = NULL,
          nd_inputs = NULL,
          nd_outputs = NULL,
          ud_inputs = NULL,
          ud_outputs = NULL)
```

Arguments

`datadea` Data frame with DEA data.

`ni` Number of inputs, if inputs are in columns 2:(`ni` + 1) (if DMUs are in the first column) or 1:`ni` (no DMUs column).

`no` Number of outputs, if outputs are in columns (`ni` + 2):(`ni` + `no` + 1) (if DMUs are in the first column) or (`ni` + 1):(`ni` + `no`) (no DMUs column). If not specified, DMUs are in the first column.

`dmus` Column (number or name) of DMUs (optional). By default, it is the first column. If there is not any DMU column, then it must be `NULL`.

`inputs` Columns (numbers or names) of inputs (optional). It prevails over `ni`. Alternatively to `datadea`, it can be a matrix with the inputs (DMUs in columns). In this case, DMUs names are taken from the columns names.

`outputs` Columns (numbers or names) of outputs (optional). It prevails over `no`. Alternatively to `datadea`, it can be a matrix with the outputs (DMUs in columns).

`nc_inputs` A numeric vector containing the indices of non-controllable inputs.

`nc_outputs` A numeric vector containing the indices of non-controllable outputs.

`nd_inputs` A numeric vector containing the indices of non-discretionary inputs.

`nd_outputs` A numeric vector containing the indices of non-discretionary outputs.

`ud_inputs` A numeric vector containing the indices of undesirable (good) inputs.

`ud_outputs` A numeric vector containing the indices of undesirable (bad) outputs.

Value

An object of class `deadata`. Any function `model_*` can be applied.

read_data_fuzzy *read_data_fuzzy*

Description

This function is deprecated. Use make_deadata_fuzzy instead.

Usage

```
read_data_fuzzy(datadea,
                dmus = 1,
                inputs.mL = NULL,
                inputs.mR = NULL,
                inputs.dL = NULL,
                inputs.dR = NULL,
                outputs.mL = NULL,
                outputs.mR = NULL,
                outputs.dL = NULL,
                outputs.dR = NULL,
                nc_inputs = NULL,
                nc_outputs = NULL,
                nd_inputs = NULL,
                nd_outputs = NULL,
                ud_inputs = NULL,
                ud_outputs = NULL)
```

Arguments

datadea Data frame with DEA data.

dmus Column (number or name) of DMUs (optional). By default, it is the first column. If there is not any DMU column, then it must be NULL.

inputs.mL Where are (columns) the mL (left centers) of trapezoidal fuzzy inputs in datadea. If an input is triangular or crisp, we put the column where the centers or the crisp values are, respectively.

Alternatively to datadea, inputs.mL can be a matrix of size (number of inputs x number of DMUs) with the mL of trapezoidal fuzzy inputs, the centers of triangular inputs, and the crisp values of crisp inputs. In this case, DMUs names are taken from the columns names.

inputs.mR Where are (columns) the mR (right centers) of trapezoidal fuzzy inputs in datadea. If an input is triangular or crisp, we put NA.

Alternatively to datadea, inputs.mR can be a matrix of size (number of inputs x number of DMUs) with the mR of trapezoidal fuzzy inputs, the centers of triangular inputs, and the crisp values of crisp inputs. If all inputs are triangular or crisp, then inputs.mR must be NULL (default) or equal to inputs.mL.

`inputs.dL` Where are (columns) the `dL` (left radii) of trapezoidal and triangular fuzzy inputs in `datadea`. If an input is symmetric, we put the column where the radii are. If an input is rectangular or crisp, we put `NA`.

Alternatively to `datadea`, `inputs.dL` can be a matrix of size (number of inputs x number of DMUs) with the `dL` of trapezoidal and triangular fuzzy inputs. If an input is rectangular or crisp, its radius is zero. If all inputs are rectangular or crisp, then `inputs.dL` must be NULL (default) or a zero matrix.

`inputs.dR` Where are (columns) the `dR` (right radii) of trapezoidal and triangular fuzzy inputs in `datadea`. If an input is symmetric, rectangular or crisp, we put `NA`.

Alternatively to `datadea`, `inputs.dR` can be a matrix of size (number of inputs x number of DMUs) with the `dR` of trapezoidal and triangular fuzzy inputs. If an input is rectangular or crisp, its radius is zero. If all inputs are symmetric, rectangular or crisp, then `inputs.dR` must be NULL (default) or equal to `inputs.dL`.

`outputs.mL` Analogous to `inputs.mL`, but relating to outputs.

`outputs.mR` Analogous to `inputs.mR`, but relating to outputs.

`outputs.dL` Analogous to `inputs.dL`, but relating to outputs.

`outputs.dR` Analogous to `inputs.dR`, but relating to outputs.

`nc_inputs` A numeric vector containing the indices of non-controllable inputs.

`nc_outputs` A numeric vector containing the indices of non-controllable outputs.

`nd_inputs` A numeric vector containing the indices of non-discretionary inputs.

`nd_outputs` A numeric vector containing the indices of non-discretionary outputs.

`ud_inputs` A numeric vector containing the indices of undesirable (good) inputs.

`ud_outputs` A numeric vector containing the indices of undesirable (bad) outputs.

Value

An object of class `deadata_fuzzy`. Any function `modelfuzzy_*` can be applied.

`read_malmquist` *read_malmquist*

Description

This function is deprecated. Use `make_malmquist` instead.

Usage

```
read_malmquist(datadea,
               nper = NULL,
               percol = NULL,
               arrangement  = c("horizontal", "vertical"),
               ...)
```

Arguments

`datadea`	Data frame with DEA data.
`nper`	Number of time periods (with dataset in wide format).
`percol`	Column of time period (with dataset in long format).
`arrangement`	Horizontal with data in wide format. Vertical with data in long format.
`...`	Other options to be passed to the `make_deadata` function.

Value

An object of class `deadata`. Function `malmquist_index` can be applied.

`references` *References*

Description

Extract the reference set for each DMU (inefficient DMUs and efficicent DMUs that are combination of other efficient DMUs) from a DEA model solution.

Usage

```
references(deasol,
           thr = 1e-4)
```

Arguments

`deasol`	Object of class `dea` obtained with some of the DEA model functions.
`thr`	Tolerance threshold (for avoiding miss detection of efficient DMUs due to round off errors).

Value

A list with the references for each evaluated DMU.

Author(s)

Vicente Coll-Serrano (`<vicente.coll@uv.es>`). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (`<vicente.bolos@uv.es>`). *Department of Business Mathematics*

Rafael Benítez (`<rafael.suarez@uv.es>`). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
# Replication results model DEA1 in Tomkins and Green (1988).
data("Departments")
# Calculate Total income
Departments$Total_income <- Departments[, 5] + Departments[, 6] + Departments[, 7]
data_DEA1 <- make_deadata(Departments,
                          inputs = 9,
                          outputs = c(2, 3, 4, 12))
result <- model_basic(data_DEA1,
                      orientation = "io",
                      rts = "crs")
references(result) # Table 3 (p.157)
```

rts *Returns to scale*

Description

Extract the returns to scale.

Usage

```
rts(deamodel,
            thr = 1e-4)
```

Arguments

deamodel	Object of class dea obtained with some of the DEA model functions.
thr	Threshold for the tolerance for considering something equal to 1. Defaults to 1e-4.

Value

A dataframe with some information about returns to scale (rts), depending on the model.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
data("Coll_Blasco_2006")
data_example <- make_deadata(Coll_Blasco_2006,
                             ni = 2,
                             no = 2)
result <- model_basic(data_example,
                      orientation = "io",
                      rts ="crs")
rts(result)
```

Ruggiero2007 *Data: Ruggiero (2007).*

Description

Simulated data of 35 DMUs with two inputs and one output.

Usage

```
data("Ruggiero2007")
```

Format

Data frame with 35 rows and 4 columns. Definition of inputs (X) and outputs (Y):

x1 Input 1

x2 Input 2

y1 Output 1

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Ruggiero J. (2007). Non-Discretionary Inputs. In: Zhu J., Cook W.D. (eds) Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis. Springer, Boston, MA. doi:10.1007/9780387716077_5

See Also

make_deadata, model_basic

Examples

```
# Example. Replication of results in Ruggiero (2007).
data("Ruggiero2007")
# the second input is a non-discretionary input
datadea <- make_deadata(Ruggiero2007,
                        ni = 2,
                        no = 1,
                        nd_inputs = 2)
result <- model_basic(datadea,
                      orientation = "io",
                      rts = "crs")
efficiencies(result)
slacks(result)
```

`slacks` *Slacks*

Description

Extract the slacks of the DMUs from a `dea` or `dea_fuzzy` solution.

Usage

```
slacks(deasol)
```

Arguments

`deasol`	Object of class `dea` or `dea_fuzzy` obtained with some of the DEA model functions.

Value

A list with the optimal slacks for each evaluated DMU.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
data("Coll_Blasco_2006")
data_example <- make_deadata(Coll_Blasco_2006,
                             ni = 2,
                             no = 2)
result <- model_multiplier(data_example,
                           orientation = "io",
                           rts = "crs")
slacks(result)
```

`summary.dea` *Summary conventional DEA models.*

Description

Summary of the results obtained by a conventional DEA model.

Usage

```
## S3 method for class 'dea'
summary(object, exportExcel = FALSE, filename = NULL, returnList = FALSE, ...)
```

Arguments

`object` An object of class `"dea"` obtained by a DEA model function.

`exportExcel` Logical value. If TRUE (FALSE by default) the results are also exported to an Excel file.

`filename` Character string. Absolute file name (including path) of the exported Excel file. If NULL, then the file name will be "ResultsDEA" + timestamp.xlsx.

`returnList` Logical value. If TRUE then the results are given as a list of data frames. If FALSE (default) all the data frames are merged into a single data frame.

`...` Ignored. Used for compatibility issues.

Value

Depending on the model it returns a single data frame containing: efficiencies, slacks, lambdas, targets, references or a list of data frames with the cross-efficiencies computed with different methods (Arbitrary, Method II or Method III (see CITA)) or, in case the model is a Malmquist index, a single data frame with the coefficients for the different periods.

Author(s)

Vicente Coll-Serrano (<`vicente.coll@uv.es`>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<`vicente.bolos@uv.es`>). *Department of Business Mathematics*

Rafael Benítez (<`rafael.suarez@uv.es`>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Charnes, A.; Cooper, W.W.; Rhodes, E. (1981). "Evaluating Program and Managerial Efficiency: An Application of Data Envelopment Analysis to Program Follow Through", Management Science, 27(6), 668-697. doi:10.1287/mnsc.27.6.668

Examples

```
data("PFT1981")
# Selecting DMUs in Program Follow Through (PFT)
PFT <- PFT1981[1:49, ]
PFT <- make_deadata(PFT,
                    inputs = 2:6,
                    outputs = 7:9 )
eval_pft <- model_basic(PFT,
                        orientation = "io",
                        rts = "crs")
summary(eval_pft)
```

`summary.dea_fuzzy` *Summary Fuzzy DEA models.*

Description

Summary of the results obtained by a fuzzy DEA model.

Usage

```
## S3 method for class 'dea_fuzzy'
summary(object, ..., exportExcel = FALSE, filename = NULL, returnList = FALSE)
```

Arguments

`object`	An object of class `"dea_fuzzy"` obtained with a fuzzy DEA model function (`modelfuzzy_guotanaka`, `modelfuzzy_kaoliu`, `modelfuzzy_possibilistic`).
`...`	Extra options.
`exportExcel`	Logical value. If TRUE (FALSE by default) the results are also exported to an Excel file.
`filename`	Character string. Absolute file name (including path) of the exported Excel file. If NULL, then the file name will be "ResultsDEA" + timestamp.xlsx.
`returnList`	Logical value. If TRUE then the results are given as a list of data frames. If FALSE (default) all the data frames are merged into a single data frame.

Value

If the model is that from Guo and Tanaka (`modelfuzzy_guotanaka`), it returns a data frame with columns: DMU, alpha cuts and efficiencies. For the possibilistic model (`modelfuzzy_possibilistic`) it returns a data frame with columns: DMU, alpha-cuts, efficiencies and the corresponding lambda values For the Kao-Liu model (`modelfuzzy_kaoliu`), the result may depend on the crisp sub-model used. It will contain a data frame with the efficiencies (if any), the slacks and superslacks (if any), the lambda values and the targets.

If `exportExcel` is TRUE, then an Excel file will be created containing as many sheets as necessary depending on the variables returned.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Leon, T.; Liern, V. Ruiz, J.; Sirvent, I. (2003). "A Possibilistic Programming Approach to the Assessment of Efficiency with DEA Models", Fuzzy Sets and Systems, 139, 407–419. doi:10.1016/S01650114(02)006085

Examples

```
data("Leon2003")
data_example <- make_deadata_fuzzy(Leon2003,
                                   inputs.mL = 2,
                                   inputs.dL = 3,
                                   outputs.mL = 4,
                                   outputs.dL = 5)
result <- modelfuzzy_possibilistic(data_example,
                                   h = seq(0, 1, by = 0.1),
                                   orientation = "io",
                                   rts = "vrs")
summary(result)
```

`Supply_Chain` *Data: Sanei and Mamizadeh Chatghayeb (2013).*

Description

Data of 17 supply chain (buyer-supplier relationship in manufacturing).

Usage

```
data("Supply_Chain")
```

Format

Data frame with 17 rows and 8 columns. Definition of inputs (X) and outputs (Y):

X1 to X3 Inputs of buyers

I1 to I2 Outputs of buyers, Inputs of suppliers

Y1 to Y2 Outputs of suppliers

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Sanei, M.; Mamizadeh Chatghayeb, S. (2013). "Free Disposal Hull Models in Supply Chain Management", International Journal of Mathematical Modelling and Computations, 3(3), 125-129.

See Also

make_deadata, model_fdh

Examples

```
# Example. FDH input-oriented.
# Replication of results in Sanei and Mamizadeh Chatghayeb (2013)
data("Supply_Chain")
data_fdh1 <- make_deadata(Supply_Chain,
                          dmus = 1,
                          inputs = 2:4,
                          outputs = 5:6)
# by default orientation = "io"
result <- model_fdh(data_fdh1)
efficiencies(result)
```

targets *Targets*

Description

Extract the targets of the DMUs from a dea or dea_fuzzy solution. We note that we call "targets" to the "efficient projections" in the strongly efficient frontier.

Usage

```
targets(deasol)
```

Arguments

deasol	Object of class dea or dea_fuzzy obtained with some of the DEA model functions.

Value

A list with the targets for each evaluated DMU.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Examples

```
data("Coll_Blasco_2006")
data_example <- make_deadata(Coll_Blasco_2006,
                             ni = 2,
                             no = 2)
result <- model_multiplier(data_example,
                           orientation = "io",
                           rts = "crs")
targets(result)
```

Tone2001 *Data: Tone (2001).*

Description

Data of 5 DMUs producing 2 outputs by using 2 inputs

Usage

```
data("Tone2001")
```

Format

Data frame with 5 rows and 5 columns. Definition of inputs (X) and outputs (Y):

x1 Input1

x2 Input2

y1 Output1

y2 Output2

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Tone, K. (2001). "A Slacks-Based Measure of Efficiency in Data Envelopment Analysis", European Journal of Operational Research, 130, 498-509. doi:10.1016/S03772217(99)004075

See Also

make_deadata, model_sbmeff

Examples

```
# Example. Replication of results in Tone (2001, p. 505)
data("Tone2001")
data_example <- make_deadata(Tone2001,
                             ni = 2,
                             no = 2)
result <- model_sbmeff(data_example,
                       orientation = "no",
                       rts = "crs")
```

```
efficiencies(result)
slacks(result)
```

Tone2003 *Data: Tone (2003).*

Description

Data of 9 DMUs producing 2 outputs, being second output undesirable, by using 1 input.

Usage

```
data("Tone2003")
```

Format

Data frame with 9 rows and 4 columns. Definition of inputs (X) and outputs (Y):

x Input

yg Output1 ("good" output)

yb Output2 (undesirable "bad" output)

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolos (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benitez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

Source

Tone, K. (2003). "Dealing with undesirable outputs in DEA: A Slacks-Based Measure (SBM) approach", GRIPS Research Report Series I-2003-0005.

See Also

make_deadata, model_sbmeff

Examples

```
# Example. Replication of results in Tone (2003), pp 10-11.
data("Tone2003")
data_example <- make_deadata(Tone2003,
                             ni = 1,
                             no = 2,
                             ud_outputs = 2)
result <- model_sbmeff(data_example,
                       rts = "vrs")
efficiencies(result)
targets(result)
```

---

undesirable_basic *Undesirable inputs and outputs for basic DEA model.*

Description

This function transforms a deadata or deadata_fuzzy class with undesirable inputs/outputs according to Seiford and Zhu (2002). Onwards, it is recommended to use a DEA model with variable returns to scale (vrs).

Usage

```
undesirable_basic(datadea,
                  vtrans_i = NULL,
                  vtrans_o = NULL)
```

Arguments

datadea	A deadata object, including DMUs, inputs and outputs.
vtrans_i	Numeric vector of translation for undesirable inputs. If vtrans_i[i] is NA, then it applies the "max + 1" translation to the i-th undesirable input. If vtrans_i is a constant, then it applies the same translation to all undesirable inputs. If vtrans_i is NULL, then it applies the "max + 1" translation to all undesirable inputs.
vtrans_o	Numeric vector of translation for undesirable outputs, analogous to vtrans_i, but applied to outputs.

Value

An list with the transformed object of class deadata or deadata_fuzzy and the corresponding translation vectors vtrans_i and vtrans_o.

Author(s)

Vicente Coll-Serrano (<vicente.coll@uv.es>). *Quantitative Methods for Measuring Culture (MC2). Applied Economics.*

Vicente Bolós (<vicente.bolos@uv.es>). *Department of Business Mathematics*

Rafael Benítez (<rafael.suarez@uv.es>). *Department of Business Mathematics*

University of Valencia (Spain)

References

Seiford, L.M.; Zhu, J. (2002). “Modeling undesirable factors in efficiency evaluation”, European Journal of Operational Research 142, 16-20.

Hua Z.; Bian Y. (2007). DEA with Undesirable Factors. In: Zhu J., Cook W.D. (eds) Modeling Data Irregularities and Structural Complexities in Data Envelopment Analysis. Springer, Boston, MA.

Examples

```
data("Hua_Bian_2007")
# The third output is an undesirable output.
data_example <- make_deadata(Hua_Bian_2007,
                             ni = 2,
                             no = 3,
                             ud_outputs = 3)
# rts must be "vrs" for undesirable inputs/outputs:
# Translation parameter is set to (max + 1)
result <- model_basic(data_example,
                      orientation = "oo",
                      rts = "vrs")
```

Index