Package: adea (via r-universe)

November 6, 2026

Type Package

Depends methods, R (>= 3.5.0), rmarkdown, ROI.plugin.glpk, ROI.plugin.lpsolve, ROI.plugin.symphony

Imports combinat, graphics, knitr, lpSolveAPI, parallel, parallelly, ROI, slam

VignetteBuilder knitr

Version 1.5.2

Date 2024-11-06

Encoding UTF-8

ByteCompile true

Title Alternate DEA Package

Maintainer Manuel Munoz-Marquez <manuel.munoz@uca.es>

Description The meaning of adea is ``alternate DEA''. This package is devoted to provide the alternative method of DEA described in the paper entitled ``Stepwise Selection of Variables in DEA Using Contribution Load'', by F. Fernandez-Palacin, M. A. Lopez-Sanchez and M. Munoz-Marquez. Pesquisa Operacional 38 (1), pg. 1-24, 2018. <doi:10.1590/0101-7438.2018.038.01.0031>. A full functional on-line and interactive version is available at <https://knuth.uca.es/shiny/DEA/>.

URL http://knuth.uca.es/dea/

License GPL (>= 3)

RoxygenNote 7.3.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/testthat/parallel true

NeedsCompilation no

Author Fernando Fernandez-Palacin [aut] (<https://orcid.org/0000-0002-3525-2924>), Manuel Munoz-Marquez [aut, cre] (<https://orcid.org/0000-0003-4157-8784>)

Config/pak/sysreqs cmake coinor-symphony coinor-libsymphony-dev libglpk-dev make libuv1-dev

Repository https://cranhaven.r-universe.dev

Date/Publication 2026-11-06 03:02:00 UTC

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/adea

RemoteSha 1a17f409eed2a4dc1771737692a474c3c7b9b59a

RemoteSubdir adea

Contents

adea-package . . . 3
adea . . . 4
adea_hierarchical . . . 8
adea_load_leverage . . . 10
adea_loads . . . 12
adea_parametric . . . 14
adea_stepwise . . . 16
cadea . . . 18
cardealers1 . . . 21
cardealers2 . . . 22
cardealers3 . . . 22
cardealers4 . . . 23
dea . . . 24
fsdea . . . 26
plot.adea . . . 29
print.adea . . . 30
print.adeahierarchical . . . 30
print.adealoadleverage . . . 31
print.adeaparametric . . . 31
print.adeastepwise . . . 32
print.cadea . . . 32
print.dea . . . 33
print.fsdea . . . 33
spanishuniversities2018 . . . 34
summary.adea . . . 35
summary.adeahierarchical . . . 35
summary.adeaparametric . . . 36
summary.adeastepwise . . . 36
summary.cadea . . . 37
summary.dea . . . 37
summary.fsdea . . . 38
tokyo_libraries . . . 39

Index **41**

adea-package *Data Envelopment Analysis: Variable Selection, Constrained ADEA and Leverage Units*

Description

Package: adea

Version: 1.5.2

Date: 2024-11-06

License: GPL (>= 3)

Data Envelopment Analysis (DEA) involves evaluating the efficiency of a set of Decision Making Units (DMUs) and calculating a relative efficiency score for each DMU. These scores are determined as a weighted ratio between all inputs and outputs amounts for such DMU.

DEA methodology assumes that all DMUs use the same set of inputs to produce the same set of outputs.

Variable selection

The selection of input and output variables for inclusion in a DEA model is a crucial aspect, as numerous studies have shown. This package offers two variable selection procedures. The first is ADEA, which is based on a measure of the relative importance of variables in the entire set of scores. For more information on this methodology, see adea.

An alternative way to select the variables for the model is by solving a mathematical optimization problem that determines the optimal selection based on some performance criteria. fsdea function provides an implementation of this procedure.

Function Index

The main functions provided by this package are:

- adea: Conducts ADEA analysis, providing efficiency scores for each DMU, a set of weights, and loads for each input and output variable, along with a model load.
- adea_parametric: Does a stepwise analysis of removing variables step by steps. adea_hierarchical works in similar way.
- fsea: Selects an optimal subset of input and output variables based on some performance criteria.
- cadea: Performs Constrained ADEA analysis to enforce variable load constraints within a specified range, resulting in changes to efficiency scores.
- adea_load_average: Identifies DMUs with a higher impact on the ADEA model.

Note

This package is translation-ready, and contributions of translated versions of po files are highly welcome.

Author(s)

Fernando Fernandez-Palacin <fernando.fernandez@uca.es> and Manuel Munoz-Marquez <manuel.munoz@uca.es>

Mantainer: Manuel Munoz-Marquez <manuel.munoz@uca.es>

References

A new approach to the bi-dimensional representation of the DEA efficient frontier with multiple inputs and outputs. *Carlos A. Bana e Costa* and *Joao Carlos C. B. Soares de Mello*, and *Lidia Angulo Meza*. European Journal of Operational Research, 255 (1), pg. 175-186, 2016, <DOI:10.1016/j.ejor.2016.05.012>.

Stepwise Selection of Variables in DEA Using Contribution Load. *F. Fernandez-Palacin*, *M. A. Lopez-Sanchez*, and *M. Munoz-Marquez*. Pesquisa Operacional 38 (1), pg. 1-24, 2018. <DOI:10.1590/0101-7438.2018.038.01.0000>.

Feature Selection in Data Envelopment Analysis: A Mathematical Optimization approach. *Benitez-Pena, S.*, *Bogetoft, P.*, and *Romero Morales, D.*. Omega, Elsevier BV, 96, pp. 102068, 2020. <DOI:10.1016/j.omega.2019.05.004>

Methodology for calculating critical values of relevance measures in variable selection methods in data envelopment analysis. *Jeyms Villanueva-Cantillo* and *Manuel Munoz-Marquez*. European Journal of Operational Research, 290 (2), pg. 657-670, 2021. <DOI:10.1016/j.ejor.2020.08.021>.

`adea` *ADEA analysis to variable selection in DEA*

Description

It does an ADEA analysis. In particular it computes a score for each DMU and a load for each variable.

Usage

```
adea(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  name = "",
  solver = "auto"
)
```

Arguments

`input`	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
`output`	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
`orientation`	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
`load.orientation`	This parameter allows the selection of variables to be included in load analysis. The default is "inoutput" which means that all input and output variables will be included. Use "input" or "output" to include only input or output variables in load analysis.
`name`	An optional descriptive name for the model. The default is an empty string. This name will be displayed in printed and summarized results.
`solver`	The solver used by ROI to solve the DEA optimization problem. The default is "auto." The solver must be installed and capable of solving linear programming problems. Use `ROI_installed_solvers()` to list them.

Details

The selection of input and output variables for inclusion in a DEA model is a critical aspect, as many studies have revealed. `adea` function provides an implementation of the ADEA method for variable selection in DEA.

ADEA methodology introduces a new phase in classical DEA analysis, measuring the relative importance of each input and output variable. This measure is referred to as load or contribution. It also defines a load for the entire model. Using this measure, a procedure has been developed to select an optimised or relevant set of variables.

A variable's load is a standardized ratio of the efficiency scores of all DMUs due to that variable. These loads quantify the contribution of each variable to the overall model. Where 0 means that the contribution of that variable to the efficiency values is negligible. The ideal load is 1, and values range from 0 to the number of input or output variables. The lowest load value has a real significance as it represents the variable with the least contribution to efficiency.

As it is usually done in DEA, these loads are computed as its maximum allowable value using alternative sets of weights but without changing the efficiency scores. But because the sum of all of them is fixed, when one variable increases its load, any other decrease in value. So only the lowest value of all loads has a real meaning. This lowest value can be taken as a significance measure of the entire model.

This measure, load, has two key properties that easy its interpretation:

- It has a bounded range from 0 to the number of input or output variables, with 1 as the ideal value.
- It is invariant by changes in scale.

ADEA analysis can be done considering only input variables, in this case ADEA analysis has input as `load.orientation` value. output when only output variables are considered. And inoutput `load.orientation` when all variables in the model are taken into account.

Adea models, as the DEA models, can be input or output orientated. Input orientated DEA models propose to reach the efficiency of DMUs through a reduction in the amount of input required by non efficient DMUs. On the contrary, output orientated DEA models propose to increase the amount of output of non efficient DMUs.

For a detailed description of the maths behind the model, see the references.

Value

The function returns an adea class object with the following named members:

- name: A label of the model
- orientation: DEA model orientation 'input' or 'output'
- load.orientation: load DEA model orientation 'input', 'output', or 'inoutput'
- inputnames: Variable input names
- outputnames: Variable output names
- eff: is a vector with DMU's scores
- loads: A list with all information about loads:
  - load: The lowest load, which is the load of the ADEA model
  - input: A vector with loads of input variables
  - iinput: Index of input variable that reach the load of the model
  - output: A vector with loads of output variables
  - ioutput: Index of output variable that reach the load of the model
- ux: A set of weights for inputs
- vy: A set of weights for output
- vinput: Standardized virtual input dividing by the sum of the weights, see [Costa2006] in adea-package.
- voutput: Standardized virtual output dividing by the sum of the weights, see [Costa2006] in adea-package
- solver: The solver used for the resolution of the optimization problem

References

Stepwise Selection of Variables in DEA Using Contribution Load. *F. Fernandez-Palacin*, *M. A. Lopez-Sanchez* and *M. Munoz-Marquez*. Pesquisa Operacional 38 (1), pg. 1-24, 2018. <DOI:10.1590/0101-7438.2018.038.01.0000>.

Methodology for calculating critical values of relevance measures in variable selection methods in data envelopment analysis. *Jeyms Villanueva-Cantillo* and *Manuel Munoz-Marquez*. European Journal of Operational Research, 290 (2), pg. 657-670, 2021. <DOI:10.1016/j.ejor.2020.08.021>.
#

See Also

adea-package.

Examples

```
# Load data
data('cardealers4')

# Define input and output
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]

# Compute adea model
model <- adea(input, output, name = 'ADEA for cardealers4 dataset')
model
# Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044

# Get model's load
model$loads$load
# [1] 0.6666667

# Get variable loads
model$loads
# $load
# [1] 0.6666667
# $input
# Employees Depreciation
# 0.6666667    1.3333333
# $iinput
# Employees
#         1
# $output
# CarsSold WorkOrders
# 1.2663476  0.7336524
# $ioutput
# WorkOrders
#          2

# Summarize the model and print additional information
summary(model)
# Model name              ADEA for cardealers4 dataset
# Orientation                                    input
# Load orientation                             inoutput
# Model load                        0.666666666666659
# Input load.Employees              0.666666666666659
# Input load.Depreciation            1.33333333333334
# Output load.CarsSold                1.1025075271907
# Output load.WorkOrders              0.8974924728093
# Inputs                         Employees Depreciation
# Outputs                           CarsSold WorkOrders
# nInputs                                              2
# nOutputs                                             2
# nVariables                                           4
# nEfficients                                          2
# Eff. Mean                          0.90022318389575
```

```
# Eff. sd                          0.135194867030839
# Eff. Min.                        0.651504424778761
# Eff. 1st Qu.                     0.872252747252747
# Eff. Median                      0.942225031605562
# Eff. 3rd Qu.                     0.997898230088495
# Eff. Max.                                        1
```

`adea_hierarchical` *Selection of an optimal subset of variables for DEA analysis*

Description

This function returns a list of DEA models by systematically eliminating one variable at a time, following ADEA methodology.

Usage

```
adea_hierarchical(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  name = "",
  direction = c("backward", "backward/input", "backward/output"),
  solver = "auto",
  verbose = 0
)
```

Arguments

`input`	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
`output`	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
`orientation`	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
`load.orientation`	This parameter allows the selection of variables to be included in load analysis. The default is "inoutput" which means that all input and output variables will be included. Use "input" or "output" to include only input or output variables in load analysis.
`name`	An optional descriptive name for the model. The default is an empty string. This name will be displayed in printed and summarized results.
`direction`	The direction in which the variables enter or exit the model. Currently, only the "backward" option is implemented.

solver	The solver used by ROI to solve the DEA optimization problem. The default is "auto." The solver must be installed and capable of solving linear programming problems. Use `ROI_installed_solvers()` to list them.
verbose	Use 0 for minimal output, displaying only the final model. Use 1 or higher values to get detailed information for each step. The default is 0. This option affects only the printed output and not the result.

Details

This procedure provides a list of all DEA models for all nested sets of variables. In each model, the variable with lowest load is dropped. It's important to note that the load of the new model can be lower than that of the previous one. For more details, please refer to the examples section.

Value

The function returns an object of the adeaparametric class with the following named members:

- name: A label of the model.
- orientation: DEA model orientation ('input' or 'output').
- load.orientation: load DEA model orientation ('input', 'output', or 'inoutput').
- loads: Load of each model.
- models: List of all adea models.
- ninputs: Number of input variables in each model.
- noutputs: Number of output variables in each model.
- nvariables: Number of variables in each model.
- inputnames: Names of input variables in each model.
- outputnames: Names of output variables in each model.
- out: Variable dropped at each step.
- solver: The solver used for resolving the optimization problem.

See Also

adea_parametric

Examples

```
# Load data
data('cardealers4')

# Define input and output
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]

# Compute all dea models in hierarchical manner
sol.ah <- adea_hierarchical(input, output)

# Print the result
```

```
sol.ah
#     Loads nEfficients nVariables nInputs nOutputs             Inputs           Outputs
# 4 0.6666667        2        4      2       2 Employees, Depreciation CarsSold, WorkOrders
# 3 0.9575672         2        3      1       2          Depreciation CarsSold, WorkOrders
# 2 1.0000000         1        2      1       1           Depreciation           CarsSold

# Summary the model with 3 variables
summary(sol.ah$models[[3]])
# Model name
# Orientation                           input
# Load orientation                    inoutput
# Model load                0.957567163474156
# Input load.Depreciation                   1
# Output load.CarsSold       1.04243283652584
# Output load.WorkOrders    0.957567163474156
# Inputs                         Depreciation
# Outputs                 CarsSold WorkOrders
# nInputs                                   1
# nOutputs                                  2
# nVariables                                3
# nEfficients                               2
# Eff. Mean                  0.90022318389575
# Eff. sd                   0.135194867030839
# Eff. Min.                 0.651504424778761
# Eff. 1st Qu.              0.872252747252747
# Eff. Median               0.942225031605563
# Eff. 3rd Qu.              0.997898230088496
# Eff. Max.                                 1

# Get efficiencies for the model with 3 variables
sol.ah$models[[3]]$eff
# Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044
```

`adea_load_leverage` *Search for leverage units (DMU's) with a greater impact on load levels in DEA analysis*

Description

Search for leverage units (DMU's) with a greater impact on load levels in DEA analysis.

Usage

```
adea_load_leverage(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  load.diff = 0.05,
```

```
  ndel = 1,
  nmax = 0,
  solver = "auto"
)
```

Arguments

input	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
output	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
orientation	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
load.orientation	This parameter allows the selection of variables to be included in load analysis. The default is "inoutput" which means that all input and output variables will be included. Use "input" or "output" to include only input or output variables in load analysis.
load.diff	Minimum difference in load to consider a subset of DMUs as a leverage one
ndel	Maximum number of units to drop out in each try.
nmax	Maximum number of DMU sets to include in results. 0 for no limit.
solver	The solver used by ROI to solve the DEA optimization problem. The default is "auto." The solver must be installed and capable of solving linear programming problems. Use `ROI_installed_solvers()` to list them.

Details

A leverage unit is a DMU that significantly alters the results of the current procedure, in this case, a DMU that produce a large change in variable loads.

Value

The function returns a list with the following named members:

- loads: Load of each model after removing the corresponding DMUs
- loads.diff: For each model the difference between its load and the original one
- dmu.indexs: Index of DMUs removed in each model

Note

This function has to solve a large number of large linear programs that grows with DMUs. So computation time required may be very large, be patient.

Examples

```
data('cardealers4')
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]
adea_load_leverage(input, output, ndel = 2)
#         load  load.diff DMUs
# 1  1.0000000 0.33333333 1, 6
# 2  1.0000000 0.33333333 3, 4
# 3  1.0000000 0.33333333 2, 3
# 4  1.0000000 0.33333333 2, 5
# 5  1.0000000 0.33333333 4, 6
# 6  1.0000000 0.33333333    2
# 7  1.0000000 0.33333333 1, 4
# 8  1.0000000 0.33333333 2, 6
# 9  1.0000000 0.33333333 1, 2
# 10 0.9635628 0.29689609 2, 4
# 11 0.8743243 0.20765766 5, 6
# 12 0.8479940 0.18132736 1, 3
# 13 0.8420551 0.17538843 3, 6
# 14 0.8243243 0.15765766 1, 5
# 15 0.8000000 0.13333333    6
# 16 0.8000000 0.13333333    4
# 17 0.8000000 0.13333333    1
# 18 0.8000000 0.13333333    3
# 19 0.7461771 0.07951041 3, 5
# 20 0.7358231 0.06915643    5
```

---

adea_loads *Compute the variables loads for DEA analysis with specified weights*

Description

The adea_loads function calculates variable loads for Data Envelopment Analysis (DEA) with user-specified weights for input and output variables.

Usage

```
adea_loads(
  input,
  output,
  ux,
  vy,
  load.orientation = c("inoutput", "input", "output")
)
```

Arguments

input	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
output	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
ux	A matrix of weights for DMUs and input variables.
vy	A matrix of weights for DMUs and output variables.
load.orientation	This parameter allows the selection of variables to be included in load analysis. The default is "inoutput" which means that all input and output variables will be included. Use "input" or "output" to include only input or output variables in load analysis.

Details

In DEA analysis, even when the efficiency scores remain constants, there is a significant degree of freedom in selecting the sets of weights for input and output variables.

Not all sets of weights assign the same importance to the variables. This function allows you to compute the load of each variable based on the provided weights. It also computes load.levels, which represents the minimum values of such loads.

It's important to note that different sets of weights result in different ways to model efficiency. This function does not solve any model. It provides the loads for the specified weights, as described in the theoretical ADEA model.

This function is primarily intended for research and internal use.

Value

Loads for model, input and output variables

Examples

```
# Load data
data('cardealers4')
# Define input and output
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]
# Make dea analysis
model <- dea(input, output)
# Show results
model
#   Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044
# Compute loads for such weights
adea_loads(input, output, model$ux, model$vy)
# $load
# [1] 0
# $input
#    Employees Depreciation
```

```
#           0           2
# $iinput
# Employees
#         1
# $output
#   CarsSold WorkOrders
#  1.1025075  0.8974925
# $ioutput
# WorkOrders
```

---

adea_parametric *Selection of an optimal subset of variables for DEA analysis*

Description

The function returns a list of DEA models, progressively removing at least one variable at each step. This process results in a sequence of models with increasing values ADEA loads.

Usage

```
adea_parametric(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  name = "",
  direction = c("backward", "backward/input", "backward/output"),
  solver = "auto",
  verbose = 0
)
```

Arguments

input	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
output	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
orientation	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
load.orientation	This parameter allows the selection of variables to be included in load analysis. The default is "inoutput" which means that all input and output variables will be included. Use "input" or "output" to include only input or output variables in load analysis.
name	An optional descriptive name for the model. The default is an empty string. This name will be displayed in printed and summarized results.

direction	The direction in which the variables enter or exit the model. Currently, only the "backward" option is implemented.
solver	The solver used by ROI to solve the DEA optimization problem. The default is "auto." The solver must be installed and capable of solving linear programming problems. Use ROI_installed_solvers() to list them.
verbose	Use 0 for minimal output, displaying only the final model. Use 1 or higher values to get detailed information for each step. The default is 0. This option affects only the printed output and not the result.

Details

The models are sorted by increasing values of loads, starting from initially given model and progressing to the one with one input and one output variable. Note that the load value for the model with one input and one output is 1. In each step, at least one variable is dropped, but more than one can be dropped if necessary.

See example for more details.

Value

The function returns an adeaparametric class object with the following named members:

- name: A label of the model.
- orientation: DEA model orientation ('input' or 'output').
- load.orientation: load DEA model orientation ('input', 'output', or 'inoutput').
- loads: Load of each model.
- models: List of all ADEA models.
- ninputs: Number of input variables in each model.
- noutputs: Number of output variables in each model.
- nvariables: Number of variables in each model.
- inputnames: Names of input variables in each model.
- outputnames: Names of output variables in each model.
- out: Variable, or variables, dropped in each step.
- solver: The solver used for the resolution of the optimization problem.

See Also

adea_hierarchical

Examples

```
# Read data
data('cardealers4')
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]

# Compute all dea models in parametric way and store in sol.ap
```

```
sol.ap <- adea_parametric(input, output)
# Show result
sol.ap
#     Loads nEfficients nVariables nInputs nOutputs              Inputs           Outputs
# 4 0.6666667         2          4       2        2 Employees, Depreciation CarsSold, WorkOrders
# 3 0.9575672         2          3       1        2           Depreciation CarsSold, WorkOrders
# 2 1.0000000         1          2       1        1             Depreciation           CarsSold

# Summary the model with 3 variables
summary(sol.ap$models[[3]])
# Model name
# Orientation                            input
# Load orientation                     inoutput
# Model load                 0.957567163474156
# Input load.Depreciation                    1
# Output load.CarsSold        1.04243283652584
# Output load.WorkOrders     0.957567163474156
# Inputs                           Depreciation
# Outputs                   CarsSold WorkOrders
# nInputs                                    1
# nOutputs                                   2
# nVariables                                 3
# nEfficients                                2
# Eff. Mean                   0.90022318389575
# Eff. sd                    0.135194867030839
# Eff. Min.                  0.651504424778761
# Eff. 1st Qu.               0.872252747252747
# Eff. Median                0.942225031605563
# Eff. 3rd Qu.               0.997898230088496
# Eff. Max.                                  1
# Get efficiencies for the model with 3 variables
sol.ap$models[[3]]$eff
#  Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044
```

---

adea_stepwise *Select an optimal subset of variables for DEA analysis*

Description

Stepwise procedure for variable selection in DEA models.

Usage

```
adea_stepwise(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  name = "",
```

```
  direction = c("backward", "backward/input", "backward/output"),
  load.critical = 0.5,
  max.steps = ncol(input) + ncol(output) - 2,
  solver = "auto",
  verbose = 0
)
```

Arguments

input	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
output	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
orientation	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
load.orientation	This parameter allows the selection of variables to be included in load analysis. The default is "inoutput" which means that all input and output variables will be included. Use "input" or "output" to include only input or output variables in load analysis.
name	An optional descriptive name for the model. The default is an empty string. This name will be displayed in printed and summarized results.
direction	The direction in which the variables enter or exit the model. Currently, only the "backward" option is implemented.
load.critical	Minimum values for loads to consider that a variable should be considered in the model. It can be also a vector with two values, the first value input loads and the second for output loads.
max.steps	The maximum number of steps allowed.
solver	The solver used by ROI to solve the DEA optimization problem. The default is "auto." The solver must be installed and capable of solving linear programming problems. Use `ROI_installed_solvers()` to list them.
verbose	Use 0 for minimal output, displaying only the final model. Use 1 or higher values to get detailed information for each step. The default is 0. This option affects only the printed output and not the result.

Details

This function serves as a backend utility for variable selection in DEA models and is not intended for end-user use. It is used as part of the adea_hierarchical and adea_parametric functions.

This function performs a stepwise procedure for variable selection within DEA models.

Value

The function returns a DEA model with optimised set of variables.

cadea *Make a constrained ADEA analysis*

Description

This function calculates an efficiency score for each DMU and computes a load for each variable within the current model. However, it's essential to note that the loads or contributions of input or output variables are subject to constraints within specified values. As a result, the efficiencies of DMUs may deviate from those obtained in regular DEA or ADEA models.

Usage

```
cadea(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  name = "",
  load.min,
  load.max,
  solver = "lpsolve"
)
```

Arguments

`input`	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
`output`	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
`orientation`	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
`load.orientation`	This parameter allows the selection of variables to be included in load analysis. The default is "inoutput" which means that all input and output variables will be included. Use "input" or "output" to include only input or output variables in load analysis.
`name`	An optional descriptive name for the model. The default is an empty string. This name will be displayed in printed and summarized results.
`load.min`	A numeric value or vector giving minimum values for loads. Values for `load.min` must belongs to [0, 1).
`load.max`	A numeric value or vector giving maximum values for loads. Values for `load.max` must be greater than 1. If `load.min` or `load.max` are vectors then its length must be the same as the number of loads to compute. This means, number of inputs when `load.orientation` is input, number of outputs when `load.orientation` is output, and the sum of both when `load.orientation` is inoutput.

`solver` The solver used by ROI to solve the DEA optimization problem. The default is "auto." The solver must be installed and capable of solving linear programming problems. Use `ROI_installed_solvers()` to list them.

Details

A variable load is a numerical value between 0 and 1, with 0 signifying that the variable's contribution to the efficiency calculations is negligible. In an ideal scenario, each input or output variable would have a load of 1. This model enforces input and output weights in a manner that ensures the final variable loads fall within specified values.

In a raw variable selection procedure, it's possible to inadvertently remove a variable from a DEA model, leading to a non-natural model. In other cases, there may be political or tactical reasons for certain variables to be considered. In a standard DEA model, the weights associated with these variables might be reduced to very small values, effectively rendering their contributions nearly negligible.

The constraints for variable loads in these models prevent such scenarios by ensuring that the contributions of variables reach the desired levels. This maintains the integrity and significance of variables in the DEA model.

For more information about loads or ADEA model see `adea`

Value

The function returns a cadea object with the follosing named members:

- name: A label of the model
- orientation: DEA model orientation 'input' or 'output'
- load.orientation: load DEA model orientation 'input', 'output', or 'inoutput'
- inputnames: Variable input names
- outputnames: Variable output names
- eff: is a vector with DMU's scores
- loads: A list with all information about loads:
  - load: The lowest load, which is the load of the ADEA model
  - input: A vector with loads of input variables
  - iinput: Index of input variable that reach the load of the model
  - output: A vector with loads of output variables
  - ioutput: Index of output variable that reach the load of the model
  - load.min: Miniminum allowed value for variable loads when load.orientation is ioutput, for input variable loads when load.orientation is input, and for output variable loads when load.orientation is output.
  - load.max: Mamixum allowed value for all, input or output variable loads as for load.min.
- ux: A set of weights for inputs
- vy: A set of weights for output
- iterations: Number of iterations to reach the stop rule

- vinput: Standardized virtual input dividing by the sum of the weights, see [Costa2006] in adea-package.
- voutput: Standardized virtual output dividing by the sum of the weights, see [Costa2006] in adea-package efficiency scores, one set of weights for inputs and other for outputs, number of efficient units, variable loads and model load.
- solver: The solver used for the resolution of the optimization problem

Note

If the given limits are too narrow, then the model is infeasible, which will result in an error.

See Also

adea-package.

Examples

```
data('cardealers4')
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]

# Compute adea model
model <- adea(input, output)
model
# Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044

# Get input variable loads
model$loads$input
# Employees Depreciation
# 0.6666667    1.3333333
# Get output variable loads
model$loads$output
#  CarsSold WorkOrders
# 1.2663476  0.7336524

# Compute a constrained adea model to force load between .8 and 1.5
cmodel <- cadea(input, output, load.min = .8, load.max = 1.5)
cmodel
#  Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.5920826

# Get loads
cmodel$loads
# $load
# [1] 0.8
# $input
# Employees Depreciation
#       0.8          1.2
# $iinput
# Employees
```

```
# 1
# $output
# CarsSold WorkOrders
#      1.2        0.8
# $ioutput
# WorkOrders
# 2
# $load.min
# [1] 0.8 0.8 0.8 0.8
# $load.max
# [1] 1.5 1.5 1.5 1.5
# See differences of efficiencies in both models
model$eff - cmodel$eff
#      Dealer A      Dealer B      Dealer C      Dealer D      Dealer E      Dealer F
# -2.220446e-16 -1.332268e-15 -1.110223e-16  2.220446e-16 -1.110223e-16  5.942183e-02
```

`cardealers1` *A data set about car dealers, 1 of 4, to be used in DEA*

Description

An example dataset with six car dealers data to be use in DEA (Data Envelopment Analysis).

Format

`Employees`: The number of employees in the car dealer to be considered as input variable in DEA.

`Sold`: The sold amount by the car dealer to be considered as output variable in DEA.

Details

The purpose of this dataframe is to be used in DEA (Data Envelopment Analysis). DEA rate each unit, in this case car dealers, in terms of efficiency. Less input and more output lead to higher efficiency scores.

Usage

data('cardealers1')

Source

Vicente Coll Serrano y Olga M. Blasco Blasco (2006) Evaluacion de la eficiencia mediante el analisis envolvente de datos. Electronic edition. https://www.eumed.net/libros-gratis/2006c/197/

See Also

See also adea-package.

cardealers2 *A data set about cardealers, 2 of 4, to be used in DEA*

Description

An example dataset with six car dealers data to be use in DEA (Data Envelopment Analysis).

Format

`Employess`: The number of employees in the car dealer to be considered as input variable in DEA.

`CarsSold`: The number of car sold by the car dealer to be considered as output variable in DEA.

`WorkOrders`: The number of works orders for the mechanical garage to be considered as output variable in DEA.

Details

The purpose of this dataframe is to be used in DEA (Data Envelopment Analysis). DEA rate each unit, in this case car dealers, in terms of efficiency. Less input and more output lead to higher efficiency scores.

Usage

data('cardealers2')

Source

Vicente Coll Serrano y Olga M. Blasco Blasco (2006) Evaluacion de la eficiencia mediante el analisis envolvente de datos. Electronic edition. https://www.eumed.net/libros-gratis/2006c/197/

See Also

See also adea-package.

cardealers3 *A data set about car dealers, 3 of 4, to be used in DEA*

Description

An example dataset with six car dealers data to be use in DEA (Data Envelopment Analysis).

Format

`Employess`: The number of employees in the car dealer to be considered as input variable in DEA.

`Amortization`: As an indirect measure of the capital invested to be considered as input variable in DEA.

`NetSales`: Net amount of sales to be considered as output variable in DEA.

Details

The purpose of this dataframe is to be used in DEA (Data Envelopment Analysis). DEA rate each unit, in this case car dealers, in terms of efficiency. Less input and more output lead to higher efficiency scores.

Usage

data('cardealers3')

Source

Vicente Coll Serrano y Olga M. Blasco Blasco (2006) Evaluacion de la eficiencia mediante el analisis envolvente de datos. Electronic edition. https://www.eumed.net/libros-gratis/2006c/197/

See Also

See also adea-package.

`cardealers4` *A data set about car dealers, 4 of 4, to be used in DEA*

Description

An example dataset with six car dealers to be use in DEA (Data envelopment analysis).

Format

`Employees`: Number of employees to be considered as input variable in DEA.

`Depreciation`: Depreciation of stored to be considered as input variable in DEA.

`Sold`: Number of cars sold as output variable in DEA.

`Orders`: Number of work orders in garage as output variable in DEA.

Details

The purpose of this dataframe is to be used in DEA (Data Envelopment Analysis). DEA rate each unit, in this case car dealers, in terms of efficiency. Less input and more output lead to higher efficiency scores.

Usage

```
data('cardealers4')
```

Source

Vicente Coll Serrano y Olga M. Blasco Blasco (2006) Evaluacion de la eficiencia mediante el analisis envolvente de datos. Electronic edition. https://www.eumed.net/libros-gratis/2006c/197/

See Also

See also adea-package.

---

dea *DEA - Data Envelopment Analysis*

Description

Data Envelopment Analysis, DEA, computes, for a set of Decision Making Units, DMU, a relative efficiency score, comparing one unit with the others.

Usage

```
dea(
  input,
  output,
  orientation = c("input", "output"),
  name = "",
  solver = "auto"
)
```

Arguments

input	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
output	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
orientation	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
name	An optional descriptive name for the model. The default is an empty string. This name will be displayed in printed and summarized results.
solver	The solver used by ROI to solve the DEA optimization problem. The default is "auto." The solver must be installed and capable of solving linear programming problems. Use ROI_installed_solvers() to list them.

Details

Each DMU transforms inputs into outputs. The set of inputs and outputs is the same for all the DMUs, but not their quantities.

This function computes a relative efficiency score and weights for each input and output variable in the model. All these for each DMU.

Value

This function return a dea class object with the following named members:

- name: A label of the model
- orientation: DEA model orientation 'input' or 'output'
- inputnames: Variable input names
- outputnames: Variable output names
- eff: is a vector with DMU's scores
- ux: A set of weights for inputs
- vy: A set of weights for output
- vinput: Standardized virtual input dividing by the sum of the weights, see [Costa2006] in adea-package.
- voutput: Standardized virtual output dividing by the sum of the weights, see [Costa2006] in adea-package
- solver: The solver used for the resolution of the optimization problem

See Also

adea-package.

Examples

```
# Load data
data('cardealers4')

# Define input and output
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]

# Compute dea model
model <- dea(input, output, name = 'DEA for cardealers4 dataset')

# Print DMU efficiencies
model
# Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044

# Summarize the model and print aditional information
summary(model)
# Model name   DEA for cardealers4 dataset
```

```
# Orientation                      input
# Inputs           Employees Depreciation
# Outputs             CarsSold WorkOrders
# nInputs                              2
# nOutputs                             2
# nVariables                           4
# nEfficients                          2
# Eff. Mean              0.90022318389575
# Eff. sd                0.135194867030839
# Eff. Min.              0.651504424778761
# Eff. 1st Qu.           0.872252747252747
# Eff. Median            0.942225031605562
# Eff. 3rd Qu.           0.997898230088496
# Eff. Max.                            1
```

`fsdea` *Feature Selection in Data Envelopment Analysis with Mathematical Programming*

Description

Data Envelopment Analysis (DEA) calculates a relative efficiency score for a set of Decision Making Units (DMUs) by comparing one unit with others.

Usage

```
fsdea(
  input,
  output,
  orientation = c("input", "output"),
  name = "",
  ninputs = ncol(input),
  noutputs = ncol(output),
  nvariables = ncol(input) + ncol(output),
  solver = "auto"
)
```

Arguments

`input`	A matrix or a data frame containing the inputs of the units to be evaluated, with one row for each DMU and one column for each input.
`output`	A matrix or a data frame containing the outputs of the units to be evaluated, with one row for each DMU and one column for each output.
`orientation`	Use "input" for input orientation or "output" for output orientation in DEA model. The default is "input".
`name`	An optional descriptive name for the model. The default is an empty string. This name will be displayed in printed and summarized results.

ninputs	Number of input features (variables) to be selected. Default is the number of input variables.
noutputs	Number of output features (variables) to be selected. Default is the number of output variables.
nvariables	Number of total features (variables) to be selected, only applicable when both ninputs and noutputs are omitted. Default is the number of input plus output variables.
solver	The solver to be used by ROI to solve the DEA optimization problem. The solver must be installed and capable of solving mixed integer linear programming problems. Default is "auto." Use 'ROI_installed_solvers()' to list available solvers.

Details

Each DMU transforms inputs into outputs. The set of inputs and outputs is the same for all the DMUs, but not their quantities.

One of the fundamental steps in the application of data envelopment analysis is the choice of variables to include in the model. One of the methods proposed for this is what is known as the feature selection method. This method constructs a linear programming problem to maximize some objective function related to the dmu efficiencies. This function implements the feature selection method in the article [Benitez-Pena, S., Bogetoft, P. and Romero Morales, D. *Feature Selection in Data Envelopment Analysis: A Mathematical Optimization approach* Omega, Elsevier BV, **2020**, Vol. 96, pp. 102068](http://www.sciencedirect.com/science/article/pii/S0305048318312131)

This function, in the case of input orientation, maximize the sum of all efficiencies, while in the output orientation case, the goal is to minimize this sum. Once the relevant variables are selected, the function calculates the relative efficiency scores for each Decision Making Unit (DMU) and determines the weights for all input and output variables within the model.

Value

This function return a fsdea class object with the following named members:

- orientation: DEA model orientation.
- name: A label of the model.
- ninputs: Number of inputs to be selected.
- noutputs: Number of outputs to be selected.
- nvariables: Number of total variables to be selected.
- inputnames: Names of input variables.
- outputnames: Names of output variables.
- eff: A vector with DMU scores.
- ux: A set of weights for input variables.
- vy: A set of weights for output variables.
- obj: Optimal value of the objective function in the optimization problem.
- iselected: A vector of zeros and ones indicating the selected input variables.

- oselected: A vector of zeros and ones indicating the selected output variables.
- niselected: Number of input selected variables.
- noselected: Number of output selected variables.
- nvselected: Number of selected variables.
- vinput: Standardized virtual input divided by the sum of the weights, see [Costa2006] in adea-package.
- voutput: Standardized virtual output divided by the sum of the weights, see [Costa2006] in adea-package.
- solver: The solver used for the resolution of the optimization problem.

See Also

adea-package.

Examples

```
data('cardealers4')
input <- cardealers4[, c('Employees', 'Depreciation')]
output <- cardealers4[, c('CarsSold', 'WorkOrders')]

# Compute DEA model selecting at most 1 output
model1o <- fsdea(input, output, noutputs = 1)
model1o
#   Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
#  0.7875000 0.7500000 0.3000000 0.8653846 1.0000000 0.5400000
#  Selected inputs : Depreciation
#  Selected outputs: CarsSold

# Compute DEA model selecting at most 1 input
model1i <- fsdea(input, output, ninputs = 1)
model1i
# Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044
# Selected inputs : Depreciation
# Selected outputs: CarsSold, WorkOrders

# Compute DEA model selecting at most 3 variables
model3v <- fsdea(input, output, nvariables = 3)
model3v
# Dealer A  Dealer B  Dealer C  Dealer D  Dealer E  Dealer F
# 0.9915929 1.0000000 0.8928571 0.8653846 1.0000000 0.6515044
# Selected inputs : Depreciation
# Selected outputs: CarsSold, WorkOrders
```

plot.adea *ADEA plot of standarized virtual input and virtual output*

Description

ADEA plot of standarized virtual input and virtual output

Usage

```
## S3 method for class 'adea'
plot(
  x,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  labels = NULL,
  labels.pos = 4,
  lcol = "black",
  ...
)
```

Arguments

x	ADEA object to be plotted
main	if not null then it is used as main in plot. Its default value is the translation of "ADEA efficient frontier". If the adea x has name it is added to the previous value.
xlab	if not null then it is used as xlab in plot. Its default value is the translation of "Virtual input".
ylab	if not null then it is used as ylab in plot. Its default value is the translation of "Virtual output".
labels	if not null then a vector of labels for the DMUs points
labels.pos	position for the labels in the plot. Its default value is 4.
lcol	the color to use to draw the line. Its default value is black.
...	Adittional parameters to plot

Details

This function plots virtual input and virtual outpus in an ADEA model. The virtual input and output vectors are computed as a weighted sum of the inputs and outputs. In addition, it is imposed that the sum of the weights be the unit.

For more information on this calculation process see the references in *adea-package*.

For the calculations of virtual input and virtual output, the weights generated by ADEA have been used, but they are the same as those that would be obtained using standard DEA.

Value

A list with vinput and voutput values. These values are provided mainly for use with the function identify.

See Also

adea

Examples

```
data("cardealers4")
input = cardealers4[, c('Employees', 'Depreciation')]
output = cardealers4[, c('CarsSold', 'WorkOrders')]
adea <- adea(input = input, output = output)
plot(adea)
```

---

print.adea *Print method for adea class*

Description

Print method for adea class

Usage

```
## S3 method for class 'adea'
print(x, ...)
```

Arguments

x	Object of class adea to print.
...	Optional arguments to "print".

---

print.adeahierarchical *Print method for adeahierarchical class*

Description

For the final model of adea_hierarchical function prints a summary, the input variables, and outputs variables.

Usage

```
## S3 method for class 'adeahierarchical'
print(x, ...)
```

Arguments

x	Object of class adeahierarchical to print.
...	Optional arguments to "print".

---

print.adealoadleverage *Print method for adealoadleverage class*

Description

Print adealoadleverage results in human readable way

Usage

```
## S3 method for class 'adealoadleverage'
print(x, ...)
```

Arguments

x	Object of class adealoadleverage to print.
...	Optional arguments to "print".

---

print.adeaparametric *Print method for adeaparametric class*

Description

For the final model of adea_parametric function prints a summary, the input variables, and outputs variables.

Usage

```
## S3 method for class 'adeaparametric'
print(x, ...)
```

Arguments

x	Object of class adeaparametric to print.
...	Optional arguments to "print".

print.adeastepwise *Print method for adeastepwise class*

Description

For the final model of adeastepwise function prints a summary, the input variables, and outputs variables.

Usage

```
## S3 method for class 'adeastepwise'
print(x, ...)
```

Arguments

x	Object of class stepwise to print.
...	Optional arguments to "print".

print.cadea *Print method for cadea class*

Description

Print method for cadea class

Usage

```
## S3 method for class 'cadea'
print(x, ...)
```

Arguments

x	Object of class cadea to print.
...	Optional arguments to "print".

print.dea *Print method for dea class*

Description

Print method for dea class

Usage

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

Arguments

x	Object of class dea to print.
...	Optional arguments to "print".

print.fsdea *Print method for fsdea class*

Description

Print method for fsdea class

Usage

```
## S3 method for class 'fsdea'
print(x, ...)
```

Arguments

x	Object of class fsdea to print.
...	Optional arguments to "print".

spanishuniversities2018 *A data set of Spanish public universities*

Description

Data set spanishuniversities2018 belongs to 47 public Spanish universities, all but Pais Vasco because its staff data is missing.

Format

TeachingStaff: number of people teaching and researching with a doctor degree.

FPU2018: number of predoctoral contracts with state funds done in 2018.

FPI2018: number of postdoctoral contracts with state funds done in 2018.

Patents: number of patents registered from 2013 and 2017.

PhDThesis: number of PhD Thesis in 2017/2018 academic year.

JCR: number of papers published in journals included in JCR index

Six: number of positive reports obtained in the state research evaluation program. All permanent teaching staff from Spanish universities can apply every six year for a program to get a positive evaluation of their research activities. Positive evaluations are considered merits for promotion and represent an increase in salary.

Projects: number of research projects financed with state funds obtained in competitive calls.

Details

The purpose of this dataframe is to be used in DEA (Data Envelopment Analysis).

The only one input variable is TeachingStaff.

The output variables are: TeachingStaff, FPU2018, FPI2018, Patens, PhDThesis, JCR, Six and Projects.

Usage

data('spanishuniversities2018')

Source

Data are taken from public available information systems.

See Also

See also adea-package.

`summary.adea` *summary method for adea class*

Description

Extract information from a fsdea class object and compute some more. The default tolerance to consider a DMU as efficient one is .001 in reports. Use 'eff.tolerance' parameter to consider another tolerance between 0 and 1.

Usage

```
## S3 method for class 'adea'
summary(object, ...)
```

Arguments

`object`	is the object of class adea to summarise
`...`	For compatibility reason, see note about 'eff.tolerance' parameter.

`summary.adeahierarchical` *Summary method for adeahierarchical class*

Description

For the final model of adea_hierarchical function print the model name, orientation, load orientation, a summary, the input variables, and outputs variables.

Usage

```
## S3 method for class 'adeahierarchical'
summary(object, ...)
```

Arguments

`object`	Is the object of class adeahierarchical to summarise.
`...`	For compatibility reason, see note about 'eff.tolerance' parameter.

Details

The default tolerance to consider a DMU as efficient one is .001 in reports. Use 'eff.tolerance' parameter to consider another tolerance between 0 and 1.

summary.adeaparametric *Summary method for adeaparametric class*

Description

For the final model of adea_parametric function print the model name, orientation, load orientation, a summary, the input variables, and outputs variables.

Usage

```
## S3 method for class 'adeaparametric'
summary(object, ...)
```

Arguments

object	is the object of class adeaparametric to summarise
...	For compatibility reason, see note about 'eff.tolerance' parameter.

Details

The default tolerance to consider a DMU as efficient one is .001 in reports. Use 'eff.tolerance' parameter to consider another tolerance between 0 and 1.

summary.adeastepwise *Summary method for adeastepwise class*

Description

For the final model of adea_stepwise function print the model name, orientation, load orientation, a summary, the input variables, and outputs variables.

Usage

```
## S3 method for class 'adeastepwise'
summary(object, ...)
```

Arguments

object	Is the object of class adeastepwise to summarise.
...	For compatibility reason, see note about 'eff.tolerance' parameter.

Details

The default tolerance to consider a DMU as efficient one is .001 in reports. Use 'eff.tolerance' parameter to consider another tolerance between 0 and 1.

summary.cadea *summary method for cadea class*

Description

Extract information from a fsdea class object and compute some more. The default tolerance to consider a DMU as efficient one is .001 in reports. Use 'eff.tolerance' parameter to consider another tolerance between 0 and 1.

Usage

```
## S3 method for class 'cadea'
summary(object, ...)
```

Arguments

object	is the object of class cadea to summarise
...	For compatibility reason, see note about 'eff.tolerance' parameter.

summary.dea *summary method for dea class*

Description

Extract information from a dea class object and compute some more.

Usage

```
## S3 method for class 'dea'
summary(object, ...)
```

Arguments

object	is the object of class dea to summarise
...	For compatibility reason, see note about 'eff.tolerance' parameter.

Details

The default tolerance to consider a DMU as efficient one is .001 in reports. Use 'eff.tolerance' parameter to consider another tolerance between 0 and 1.

Value

This function returns the following information, as a named list:

- Model name: if it is previously set
- Orientation: Model orientation
- Inputs: Names of the input variables in the model
- Outputs: Names of the output variables in the model
- nInputs: Number of the input variables in the model
- nOutputs: Number of the output variables in the model
- nVariables: Total number of input and output variables in the model
- n: Number of DMUs
- nEfficients: Number of efficient DMUs taking into account whithin eff.tolerance tolerance.
- Mean: Mean of efficiency scores.
- sd: Standard deviation of efficiency scores.
- Min.: Minimun value of efficiency scores.
- 1st Qu.: First quater of efficiency scores.
- Median: Median of efficiency scores.
- 3rd Qu.: Third quater of efficiency scores.
- Max.: Maximum value of efficiency scores.

`summary.fsdea` *summary method for fsdea class*

Description

Extract information from a fsdea class object and compute some more. The default tolerance to consider a DMU as efficient one is .001 in reports. Use ‘eff.tolerance‘ parameter to consider another tolerance between 0 and 1.

Usage

```
## S3 method for class 'fsdea'
summary(object, ...)
```

Arguments

`object`	is the object of class dea to summarise
`...`	For compatibility reason, see note about ‘eff.tolerance‘ parameter.

Value

This function returns the following information, as a named vector:

- Model name: if it is previously set
- Orientation: Model orientation
- Inputs: Names of the input variables in the model
- Outputs: Names of the output variables in the model
- n: Number of DMUs
- nEfficients: Number of efficient DMUs taking into account whithin eff.tolerance tolerance.
- Mean: This and following statistical measure of the efficiency scores. Its mean.
- sd: Its standard deviation.
- Min.: Its minimun value.
- 1st Qu.: Its first quater.
- Median: Its median.
- 3rd Qu.: Its third quater.
- Max.: Its maximum value.
- nInputs: Number of input variables to select
- nOutputs: Number of output variables to select
- nTotal: Total number of variables to select
- iSelected: Names of the selected input variables
- oSelected: Names of the selected output variables

---

tokyo_libraries *A data set of Tokyo libraries*

Description

`tokyo_libraries` involves a set of 23 libraries in Tokyo and it has been used frequently in DEA literature.

Format

`Area.I1`: Area of library.

`Books.I2`: Number of books.

`Staff.I3`: Staff members.

`Populations.I4`: Population of influence area.

`Regist.O1`: Registered people.

`Borrow.O2`: Borrowed books.

Details

The purpose of this dataframe is to be used in DEA (Data Envelopment Analysis).

The input variables are `Area.I1`, `Books.I2`, `Staff.I3` and `Populations.I4`.

The output variables are `Regist.O1` and `Borrow.O2`.

Usage

```
data('tokyo_libraries')
```

See Also

See also adea-package.

Index

∗ **DEA**
adea-package, 3
dea, 24
∗ **Data Envelopment Analysis**
dea, 24
fsdea, 26
∗ **Feature selection**
fsdea, 26
∗ **Variable selection**
fsdea, 26
∗ **data**
cardealers1, 21
cardealers2, 22
cardealers3, 22
cardealers4, 23
spanishuniversities2018, 34
tokyo_libraries, 39
∗ **package**
adea-package, 3
∗ **plot**
plot.adea, 29

adea, *3*, 4, *19*, *30*
adea-package, 3
adea_hierarchical, 8, *15*
adea_load_leverage, 10
adea_loads, 12
adea_parametric, *9*, 14
adea_stepwise, 16

cadea, 18
cardealers1, 21
cardealers2, 22
cardealers3, 22
cardealers4, 23

dea, 24

fsdea, *3*, 26

plot *(*plot.adea*)*, 29
plot,adea-method *(*plot.adea*)*, 29
plot.adea, 29
print.adea, 30
print.adeahierarchical, 30
print.adealoadleverage, 31
print.adeaparametric, 31
print.adeastepwise, 32
print.cadea, 32
print.dea, 33
print.fsdea, 33

spanishuniversities2018, 34
summary.adea, 35
summary.adeahierarchical, 35
summary.adeaparametric, 36
summary.adeastepwise, 36
summary.cadea, 37
summary.dea, 37
summary.fsdea, 38

tokyo_libraries, 39