Package ‘adea’

February 10, 2022

Type Package

Depends Benchmarking, combinat, lpSolveAPI, methods, R (>= 3.5.0), rmarkdown

Imports knitr

VignetteBuilder knitr

Version 1.3.1

Date 2022-02-09

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/>.

License GPL (>= 3)

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/testthat/parallel true

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

Repository CRAN

Repository/R-Forge/Project adea

Repository/R-Forge/Revision 13

Repository/R-Forge/DateTimeStamp 2022-02-09 20:07:47

Date/Publication 2022-02-10 09:50:02 UTC

NeedsCompilation no

R topics documented:

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

Description

```
Package: adea

Version: 1.3.1

Date: 2022-02-09
```

`License: GPL (>= 3)`

DEA, that means Data Envelopment Analysis, consider a set of DMUs (Decision Making Units) and computes an efficiency score for each DMU. Such score is defined as a weighted ratio between several inputs and outputs values for such DMU.

This package provides an implementation of the ADEA method for variable selection in DEA. ADEA methodology includes a new phase in the classical DEA analysis that measures the relative importance of each input and output variable. This above-mentioned measure is called load ratio or contribution. A load for the whole model is also defined. Using such measure, a procedure to select an optimised or relevant set of variables has been developed.

A variable load is a number between 0 and 1. Where 0 means that the contribution of that variable to the efficiency values is negligible. In an ideal case, each input or output variable will have a load of 1.

As it is usually done in DEA, these loads are computed as its maximum allowable value. Using alternative sets of weights, this procedure don't change efficiency scores. But because the sum of all of them is 1, 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 important properties that easy its interpretation:

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

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

The package is named after its main function. adea makes a DEA analysis using alternative optimal solution of the same programs that DEA uses to compute efficiency scores. For a detailed description of the maths behind the model, see the references.

The main functions that this package provides are:

- adea: It makes ADEA analysis giving, efficiency scores for each DMU, a set of weight, and a load for each input and output variable, and also model load.
- cadea: Constrained ADEA analysis to force that variable loads fall in a given range. The efficiencies scores will change in order to allow this.
- adea_load_average: Search for DMU's with higher impact on ADEA model.

Note

The package is ready for translations, so contributions with translated versions of po files will be very welcomed.

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>.

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

ADEA analysis, computes a score for each DMU and load ratio for each variable.

Usage

```
adea(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  name = "",
  eff.tolerance = 0.001
)
```

Arguments

`input` A matrix or a data frame with the inputs of units to be evaluated, one row for each DMU and one column for each input.

`output` A matrix or a data frame with the outputs of units to be evaluated, one row for each DMU and one column for each output.

`orientation` Use "input" for input orientation or use "output" for output orientation in DEA model.

`load.orientation`
It allows the selection of variables to be included in load analysis. Its default value is "inoutput" which means that all input and all 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. It will be shown in print and summary results.

`eff.tolerance` A value between 0 and 1 to tolerance when considering a DMU as efficient in reports.

Details

This function computes an efficiency score for each DMU, the same as in standard DEA model.

Then a load ratio for each variable is computed searching two new set of weights while keeping DMU's scores.

The load ratio of a variable is a number between 0 and 1. Where 0 means that the contribution of that variable to the efficiency computations is negligible. In an ideal case, each input or output variable will have a load of 1 divide by the number of them.

As it is usually done in DEA this load ratio has been computed as its maximum allowable value. But because the sum of all of them is 1, when one increases its load ratio any other decreases its value. So only the lowest value of all load ratios, this is load model, has a real meaning. This lowest value can be taken as a significance measure of the entire model.

Value

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

- eff is a vector with DMU's scores
- neff is the number of efficient DMU with eff.tolerance. It means DMUs with efficiencies between 1 - eff.tolerance and 1 + eff.tolerance.
- load.orientation is the load orientation, one of 'input', 'output' or 'inoutput'. The last is the default value.
- name: A label of the model
- orientation: DEA model orientation 'input' or 'output'
- ux: A set of weights for inputs
- vy: A set of weights for output
- load: A list with all information about loads:
  - ratios$input: A vector with load ratios of input variables
  - ratios$output: A vector with load ratios of output variables
  - load: The lowest load ratio, which is the load of the ADEA model
  - lp: A pointer to the linear programming program of the model. Mainly for research purpose
  - iinput: A vector of index of inputs that almost reach the load level
  - ioutput: A vector of index of outputs that almost reach the load level
  - 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`.

See Also

`adea-package`.

Examples

```
data('cardealers4')
input <- cardealers4[, 1:2]
output <- cardealers4[, 3:4]

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

# Get model's load ratios
model$load$ratios
# $input
# Employees Depreciation
# 0.6666667    1.3333333
# $output
# CarsSold WorkOrders
# 1.2663476  0.7336524
```

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

Description

The function returns a list of DEA models dropping one variable in each step 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"),
  verbose = 0
)
```

Arguments

`input` A matrix or a data frame with the inputs of units to be evaluated, one row for each DMU and one column for each input.

`output` A matrix or a data frame with the outputs of units to be evaluated, one row for each DMU and one column for each output.

`orientation` Use "input" for input orientation or use "output" for output orientation in DEA model.

`load.orientation`
It allows the selection of variables to be included in load analysis. Its default value is "inoutput" which means that all input and all 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. It will be shown in print and summary results.

`direction` The direction in which the variables go in and out of the model. Until now, only backward option is implemented.

`verbose` Use 0 for minimal output, only final model. 1 or more to get detailed information of each step. This option affects only to printed output but 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 lower value of load is dropped. In this case, the load of a new model can be lower than a previous one.

See examples for more details.

Value

A list with all computed models. So, you can retrieve each model individually.

See Also

adea_parametric

Examples

```
# Read data
data('cardealers4')
input <- cardealers4[, 1:2]
output <- cardealers4[, 3:4]

# Compute all dea models in hierarchical way
adea_hierarchical(input, output)
#       Load #Efficient Factors #Inputs #Outputs               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

# Compute again and store result in sol.ah
sol.ah <- adea_hierarchical(input, output)

# Summary the model with 3 variables
```

```
summary(sol.ah$models[[3]])
# Model name:
# Orientation is input
# Inputs: Depreciation
# Outputs: CarsSold WorkOrders
# Load: 0.9575672
# Input loads: 1
# Output loads: 1.042433 0.9575672
# Summary of efficiencies:
# Mean        sd      Min.   1st Qu.    Median   3rd Qu.      Max.
# 0.9002232 0.1351949 0.6515044 0.8722527 0.9422250 0.9978982 1.0000000
# Efficiencies:
# 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 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_loads *Compute the variables load ratios in DEA*

Description

For the given input, output and two sets of weights, the function computes the loads of each variable for such values.

Usage

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

Arguments

input	A matrix or a data frame with the inputs of units to be evaluated, one row for each DMU and one column for each input.
output	A matrix or a data frame with the outputs of units to be evaluated, 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`
: It allows the selection of variables to be included in load analysis. Its default value is "inoutput" which means that all input and all 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 are a great level of freedom to select the sets of weights for input and output variables.

Not all those sets of weights attaches the same importance to the variables. This function allows to compute the load of each variable for the given weights in order to compare different sets of weights for the same efficiencies scores. Also compute load.levels which are the minimum value of such loads.

Take into account that different sets of weights means different ways to model the efficiency.

This function does not solve any model, only if ux and vy are the optimal values for the adea model, this function provides the load-levels as described in the theoretical adea model.

Value

Loads ratios and load for input and output variables

Examples

```
# Load data
data('cardealers4')
# Define input and output
input <- cardealers4[, 1:2]
output <- cardealers4[, 3:4]
# Make dea analysis
model <- dea(input, output, RTS = 'crs', DUAL = TRUE)
# Show results
model
# Compute loads for such weights
adea_loads(input, output, model$ux, model$vy)
```

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

Description

Search for leverage units (DMU's) with higher 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
)
```

Arguments

`input` A matrix or a data frame with the inputs of units to be evaluated, one row for each DMU and one column for each input.

`output` A matrix or a data frame with the outputs of units to be evaluated, one row for each DMU and one column for each output.

`orientation` Use "input" for input orientation or use "output" for output orientation in DEA model.

`load.orientation` It allows the selection of variables to be included in load analysis. Its default value is "inoutput" which means that all input and all 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.

Value

The indexes of units with higher impact on load levels.

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[, 1:2]
output <- cardealers4[, 3:4]
adea_load_leverage(input, output, ndel = 2)
```

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

Description

The function returns a list of DEA models removing at least one variable in each step, to get a sequence of models with increasing values of 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"),
  verbose = 0
)
```

Arguments

`input` A matrix or a data frame with the inputs of units to be evaluated, one row for each DMU and one column for each input.

`output` A matrix or a data frame with the outputs of units to be evaluated, one row for each DMU and one column for each output.

`orientation` Use "input" for input orientation or use "output" for output orientation in DEA model.

`load.orientation` It allows the selection of variables to be included in load analysis. Its default value is "inoutput" which means that all input and all 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. It will be shown in print and summary results.

`direction` The direction in which the variables go in and out of the model. Until now, only backward option is implemented.

`verbose` Use 0 for minimal output, only final model. 1 or more to get detailed information of each step. This option affects only to printed output but not the result.

Details

This procedure provides a list of all DEA models for all different threshold for load values.

The models are sorted by increasing values of loads from initially given model 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 in each step.

See example for more details.

Value

A list with all computed models. So, you can retrieve each model individually.

See Also

adea_hierarchical

Examples

```
# Read data
data('cardealers4')
input <- cardealers4[, 1:2]
output <- cardealers4[, 3:4]

# Compute all dea models in parametric way
adea_parametric(input, output)
#       Load #Efficient Factors #Inputs #Outputs               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

# Compute again and store result in sol.ap
sol.ap <- adea_parametric(input, output)

# Summary the model with 3 variables
summary(sol.ap$models[[3]])
# Model name:
# Orientation is input
# Inputs: Depreciation
# Outputs: CarsSold WorkOrders
# Load: 0.9575672
# Input loads: 1
# Output loads: 1.042433 0.9575672
# Summary of efficiencies:
# Mean        sd      Min.   1st Qu.    Median   3rd Qu.      Max.
# 0.9002232 0.1351949 0.6515044 0.8722527 0.9422250 0.9978982 1.0000000
# Efficiencies:
#  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 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. This is a back end function for adea_hierarchical and adea_parametric functions. So, it is not for end user use.

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,
  verbose = 0
)
```

Arguments

`input`	A matrix or a data frame with the inputs of units to be evaluated, one row for each DMU and one column for each input.
`output`	A matrix or a data frame with the outputs of units to be evaluated, one row for each DMU and one column for each output.
`orientation`	Use "input" for input orientation or use "output" for output orientation in DEA model.
`load.orientation`	It allows the selection of variables to be included in load analysis. Its default value is "inoutput" which means that all input and all 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. It will be shown in print and summary results.
`direction`	The direction in which the variables go in and out of the model. Until now, only backward option is implemented.
`load.critical`	Minimum values for load.ratios 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.
`verbose`	Use 0 for minimal output, only final model. 1 or more to get detailed information of each step. This option affects only to printed output but not the result.

Value

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

cadea *Make a constrained ADEA analysis*

Description

This function computes an efficiency score for each DMU and a load for each variable in the current model. But loads or contributions of input or output variables are constrained between the given values. So DMU's efficiencies could change from regular DEA or ADEA model.

Usage

```
cadea(
  input,
  output,
  orientation = c("input", "output"),
  load.orientation = c("inoutput", "input", "output"),
  name = "",
  load.min,
  load.max,
  eff.tolerance = 1e-04
)
```

Arguments

`input` A matrix or a data frame with the inputs of units to be evaluated, one row for each DMU and one column for each input.

`output` A matrix or a data frame with the outputs of units to be evaluated, one row for each DMU and one column for each output.

`orientation` Use "input" for input orientation or use "output" for output orientation in DEA model.

`load.orientation` It allows the selection of variables to be included in load analysis. Its default value is "inoutput" which means that all input and all 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. It will be shown in print and summary 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.

`eff.tolerance` A value between 0 and 1 to tolerance when considering a DMU as efficient in reports.

Details

A variable load is a number between 0 and 1. Where 0 means that the contribution of that variable to the efficiency computations is negligible. In an ideal case, each input or output variable will have a load of 1 divide by the number of them. This model force input and output weights in such a way that final variable loads fall between the given values.

For more information about loads or ADEA model see adea

Value

The function returns an object with efficiency scores, one set of weights for inputs and other for outputs, number of efficient units, load ratios and load levels.

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[, 1:2]
output <- cardealers4[, 3:4]

# 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 model load ratios
model$load$ratios
# $input
# Employees Depreciation
# 0.6666667    1.3333333
# $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 0.6515044

# Get model load ratios
cmodel$load$ratios
# $input
# Employees Depreciation
#       0.8          1.2
# $output
# CarsSold WorkOrders
#      1.2        0.8

# 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  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  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.

`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 adea_stepwise function prints a summary, the input variables, and outputs variables.

Usage

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

Arguments

x	Object of class adeastepwise 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".

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

summary method for adea class

Usage

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

Arguments

`object`	is the object of class adea to summarise
`...`	optional arguments to 'print'

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.
`...`	Optional arguments to "print".

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
`...`	optional arguments to 'print'

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.
...	Optional arguments to "print".

summary.cadea *summary method for cadea class*

Description

summary method for cadea class

Usage

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

Arguments

object	is the object of class cadea to summarise
...	optional arguments to ’print’

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