

Package: psdea (via r-universe)

October 28, 2026

Type Package

Title Data Envelopment Analysis with User-Defined and Learned Reference Sets

Version 1.0.0

Author Jiachao Peng [aut, cre]

Maintainer Jiachao Peng <jiachao.peng@wit.edu.cn>

Description Slacks-based data envelopment analysis for panel data in which the reference set is an explicit argument. A user-supplied weight matrix, or a matrix learned from the data by penalised non-negative reconstruction, selects the units that may enter each decision-making unit's frontier. The package reports the unrestricted and the restricted score, the ratio of the two and an exact decomposition of the shortfall into the part the restricted technology can remove and the part it cannot. Moran's I and local indicators of spatial association are provided as diagnostics. All linear programs are solved by a two-phase simplex with Bland's rule implemented in R, so the package needs no compiler.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 4.0)

Imports utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Repository <https://cran.r-universe.dev>

Date/Publication 2026-10-28 11:10:02 UTC

RemoteUrl <https://github.com/cran/psdea>

RemoteRef HEAD

RemoteSha bba2c330e41bcd7a1a376ad10936f616fe6051f

Contents

psdea-package	2
psdea_china_energy	3
psdea_learn	3
psdea_lp	4
psdea_moran	5
psdea_panel	6
psdea_sbm	7
psdea_spw	8
psdea_w_contiguity	9
Index	10

psdea-package	<i>Data Envelopment Analysis with User-Defined and Learned Reference Sets</i>
---------------	---

Description

Slacks-based data envelopment analysis for panel data in which the reference set is an explicit argument. A weight matrix, supplied by the user or learned from the data, selects the units that may enter each decision-making unit’s frontier. The package reports the unrestricted and the restricted score, the ratio of the two, and an exact decomposition of the shortfall into the part the restricted technology can remove and the part it cannot.

Details

The main function is [psdea_panel](#). Weight matrices are built with [psdea_spw](#) or learned with [psdea_learn](#); scores can be inspected with [psdea_moran](#). Every linear program is solved by a two-phase revised simplex implemented in R ([psdea_lp](#)), so the package has no compiled code and no dependencies beyond base R.

References

Tone, K. (2001) A slacks-based measure of efficiency in data envelopment analysis. *European Journal of Operational Research* 130, 498–509.

psdea_china_energy	<i>Provincial panel of China, 2003–2024</i>
--------------------	---

Description

A panel of 30 Chinese provinces with the inputs, output and undesirable output used in the article. Energy, carbon dioxide and gross domestic product for 2023 and 2024 are provisional; the columns whose names begin with `flag_` record whether a value is observed, official or nowcast.

Format

A data frame with 660 rows and 22 variables. The variables used in the examples are `code` (1–30, matching the weight matrices), `prov`, `year`, `energy` (10\,000 tonnes of coal equivalent), `capital` (100 million yuan at constant 2003 prices), `gdp_real` (same units) and `co2` (million tonnes).

Source

Provincial statistical yearbooks and the CEADs emission inventory.

Examples

```
data(psdea_china_energy)
str(psdea_china_energy[1:5, c("prov", "year", "energy", "gdp_real")])
```

psdea_learn	<i>Learn the reference set from the data</i>
-------------	--

Description

Reconstructs each unit's bundle as a sparse non-negative combination of the other units' bundles by penalised least squares, solved with cyclic coordinate descent. The support of the solution is the peer set.

Usage

```
psdea_learn(Z, lambda = 0.05, maxit = 500L, tol = 1e-4)

psdea_learn_data(data, vars, lambda = 0.05, by = NULL)
```

Arguments

<code>Z</code>	matrix with units in the columns and standardised inputs and outputs in the rows.
<code>lambda</code>	penalty on the sum of the weights. Larger values give sparser peer sets.
<code>maxit</code>	maximum number of coordinate-descent sweeps.
<code>tol</code>	a weight is treated as zero below this value when counting peers.
<code>data</code>	a data frame.
<code>vars</code>	names of the variables forming the bundle.
<code>by</code>	optional name of the period variable; the matrix is then built for each period and averaged.

Details

The problem is a penalised non-negative reconstruction, not a convex-hull representation: the weights are not constrained to sum to one and normalising them afterwards does not change the optimisation problem. The peer set is chosen on the same observations that produce the scores, so the matrix is a selection rule, not an estimate of a technology club.

Value

A row-normalised matrix of weights, with attributes `n_peers` and `iters`.

Examples

```
set.seed(1)
Z <- matrix(runif(40, 1, 2), 4, 10)
W <- psdea_learn(Z, lambda = 0.05)
attr(W, "n_peers")
```

psdea_lp	<i>Two-phase revised simplex</i>
----------	----------------------------------

Description

Solves $\min c'x$ subject to $Ax = b$ and $x \geq 0$ with a two-phase revised simplex and Bland's rule for pricing.

Usage

```
psdea_lp(A, b, c, maxit = 20000L, tol = 1e-10)
```

Arguments

<code>A</code>	constraint matrix, $m \times n$.
<code>b</code>	right-hand side, length m .
<code>c</code>	objective coefficients, length n .
<code>maxit</code>	maximum number of pivots.
<code>tol</code>	tolerance used in the pricing and ratio tests.

Value

A list with `x` (the solution), `obj`, `it` (pivots) and `status`: 0 optimal, 1 infeasible, 2 iteration limit reached.

Examples

```
A <- matrix(c(1, 2, -1, 0,
              3, 1,  0, -1), 2, 4, byrow = TRUE)
psdea_lp(A, c(4, 6), c(1, 1, 0, 0))$obj # 2.8
```

psdea_moran

*Spatial autocorrelation diagnostics***Description**

Moran's I and local indicators of spatial association for a vector of scores, with permutation inference. These are diagnostics: I is the slope of the spatial lag on the score, not a share of variance.

Usage

```
psdea_moran(x, W, nperm = 499L, seed = 1L)
```

```
psdea_lisa(x, W, nperm = 499L, seed = 1L)
```

Arguments

<code>x</code>	numeric vector of scores, one per unit.
<code>W</code>	square weight matrix.
<code>nperm</code>	number of permutations.
<code>seed</code>	seed for the permutations.

Value

`psdea_moran` returns a named vector with the observed statistic and its p-value; `psdea_lisa` returns a data frame with the local statistic and its conditional p-value for each unit.

Examples

```
x <- c(1, 2, 3, 4)
W <- psdea_spw(c(1, 2, 3), c(2, 3, 4), 1:4)
psdea_moran(x, W, nperm = 99)
```

psdea_panel

*Efficiency against an unrestricted and a restricted frontier***Description**

Estimates a slacks-based measure with undesirable outputs for each period of a panel, once against all units and, when a weight matrix is supplied, once against the units selected by that matrix. Returns the decomposed shortfall.

Usage

```
psdea_panel(data, y, x, year, id = NULL, bad = NULL, W = NULL,
            rts = c("crs", "vrs"))

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

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

Arguments

<code>data</code>	a data frame in long format, one row per unit and period.
<code>y</code>	name of the single desirable output.
<code>x</code>	for <code>psdea_panel</code> , a character vector with the names of the inputs; for the <code>print</code> method, the fitted object.
<code>year</code>	name of the period variable; one frontier is built per period.
<code>id</code>	name of the unit identifier. When <code>NULL</code> the row order is used and a variable <code>.id</code> is created.
<code>bad</code>	optional character vector with the names of undesirable outputs.
<code>W</code>	optional square matrix of order equal to the number of units in a period. A non-zero off-diagonal entry means that the column unit may enter the frontier of the row unit. The unit itself is always included.
<code>rts</code>	either <code>"crs"</code> or <code>"vrs"</code> .
<code>object</code>	an object returned by <code>psdea_panel</code> .
<code>...</code>	further arguments, currently ignored.

Details

The shares are $(1 - \rho^N)/(1 - \rho^G)$ and $(\rho^N - \rho^G)/(1 - \rho^G)$; they sum to one and are undefined on the frontier, where the denominator is zero. They are not equal to $1 - \text{TGR}$.

Value

A data frame of class `psdea` with the columns `id`, `year`, `eff` (unrestricted score), `eff_loc` (restricted score), `tgr` (their ratio), `s_local` and `s_gap` (the two shares of the shortfall, missing when the unrestricted score is one) and `n_peers` (the size of the reference set).

See Also

[psdea_spw](#), [psdea_learn](#)

Examples

```
data(psdea_china_energy)
d <- subset(psdea_china_energy, year == 2015 & code <= 6)
W <- psdea_w_contiguity[1:6, 1:6]
fit <- psdea_panel(d, "gdp_real", c("capital", "energy"), "year",
                  id = "code", bad = "co2", W = W)

fit
fit$s_local + fit$s_gap
```

psdea_sbm

Slacks-based measure for one cross-section

Description

Computes the slacks-based measure of Tone (2001) with undesirable outputs for every column of a cross-section, optionally against the restricted reference technology defined by a weight matrix.

Usage

```
psdea_sbm(X, Y, B = NULL, vrs = FALSE, W = NULL)
```

Arguments

<code>X</code>	matrix of inputs, rows are inputs and columns are units.
<code>Y</code>	matrix of desirable outputs, same layout.
<code>B</code>	matrix of undesirable outputs, or <code>NULL</code> .
<code>vrs</code>	logical; variable rather than constant returns to scale.
<code>W</code>	optional square weight matrix; see psdea_panel .

Value

A list with `eff`, and, when `W` is given, `eff_loc`, `n_peers`, `tgr`, `s_local` and `s_gap`.

References

Tone, K. (2004) Dealing with undesirable outputs in DEA: a slacks-based measure approach. *GRIPS Research Report I-2003-0005*.

Examples

```

X <- matrix(c(1, 1, 1), nrow = 1)
Y <- matrix(c(1, 2, 4), nrow = 1)
psdea_sbm(X, Y)$eff      # 0.25 0.50 1.00
W <- matrix(c(0, 1, 0,
              1, 0, 1,
              0, 1, 0), 3, 3, byrow = TRUE)
psdea_sbm(X, Y, W = W)$eff_loc # 0.50 1.00 1.00

```

psdea_spw

Build a weight matrix

Description

Builds a square weight matrix from a table of links, or reads one from a comma-separated file. The matrix is used only through its non-zero pattern.

Usage

```
psdea_spw(from, to, units, row_standardise = TRUE, directed = FALSE)
```

```
psdea_spw_read(file, units = NULL, row_standardise = TRUE)
```

Arguments

from, to	vectors of unit names or indices giving the links.
units	character vector with the unit names, in the order used by the data; the row and column names of the result follow this order.
row_standardise	logical; divide each row by its sum.
directed	logical; if FALSE the links are made symmetric.
file	path to a comma-separated file without header.

Value

A numeric matrix with row and column names units.

Examples

```

units <- c("A", "B", "C")
psdea_spw(c("A", "B"), c("B", "C"), units)

```

psdea_w_contiguity	<i>Spatial weight matrices for the Chinese provinces</i>
--------------------	--

Description

Matrices of order 30 used in the article: land-border contiguity (65 links plus one for Hainan), four nearest neighbours, and inverse distance with a median cut-off. Rows and columns follow the province code.

Format

A 30 by 30 numeric matrix, row-standardised.

Source

Own construction from provincial boundaries and capital coordinates.

Examples

```
data(psdea_w_contiguity)
table(rowSums(psdea_w_contiguity > 0))
```

Index

* package

- psdea-package, [2](#)
- print.psdea (psdea_panel), [6](#)
- psdea (psdea-package), [2](#)
- psdea-package, [2](#)
- psdea_china_energy, [3](#)
- psdea_learn, [2](#), [3](#), [7](#)
- psdea_learn_data (psdea_learn), [3](#)
- psdea_lisa (psdea_moran), [5](#)
- psdea_lp, [2](#), [4](#)
- psdea_moran, [2](#), [5](#)
- psdea_panel, [2](#), [6](#), [7](#)
- psdea_sbm, [7](#)
- psdea_spw, [2](#), [7](#), [8](#)
- psdea_spw_read (psdea_spw), [8](#)
- psdea_w_contiguity, [9](#)
- psdea_w_distance (psdea_w_contiguity), [9](#)
- psdea_w_knn4 (psdea_w_contiguity), [9](#)
- summary.psdea (psdea_panel), [6](#)