Package: deaviz (via r-universe)

July 2, 2026

Type Package

Title Visualization of Data Envelopment Analysis Problems

Version 0.1.0

Description High-dimensional visualization methods for data envelopment analysis (DEA), gathering in one place techniques that have appeared in the literature but remained scattered and largely unimplemented: cross-efficiency matrix unfolding, the Porembski network with lambda edges, principal component analysis biplots, multidimensional-scaling colour-plots, self-organizing maps, the Costa bi-dimensional efficient frontier, parallel coordinates, radar charts, panel-data trajectory biplots, peer and reference networks, and a set of descriptive plots. The package is built around a single validated dea_data() object and uses the 'Benchmarking' package as its DEA engine. The implemented methods draw on a body of literature; representative references include Doyle and Green (1994) <doi:10.1057/jors.1994.84>, Porembski, Breitenstein and Alpar (2005) <doi:10.1007/s11123-005-1328-5> and Bana e Costa, Soares de Mello and Angulo Meza (2016) <doi:10.1016/j.ejor.2016.05.012>.

License AGPL (>= 3)

Encoding UTF-8

Language en-GB

URL https://github.com/Pomelo64/deaviz

BugReports https://github.com/Pomelo64/deaviz/issues

LazyData true

Depends R (>= 3.6.0)

Imports rlang, stats, utils, grDevices, graphics, ggplot2 (>= 3.4.0)

Suggests Benchmarking, lpSolve, lpSolveAPI, kohonen, smacof, plotly, ggrepel, igraph, graphlayouts, MASS, testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Shahin Ashkiani [aut, cre]

Maintainer Shahin Ashkiani <Contact@Shahin-Ashkiani.com>

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

Date/Publication 2026-07-02 18:30:07 UTC

RemoteUrl https://github.com/cran/deaviz

RemoteRef HEAD

RemoteSha f33301d76b3a57cf97a3a7613cde81298fca354b

Contents

as_dea_data . . . 3
chinese_cities . . . 3
compute_efficiency . . . 4
compute_multiplier_weights . . . 6
compute_som . . . 7
cross-efficiency . . . 8
dea_data . . . 10
plot_cem_heatmap . . . 11
plot_cem_unfolding . . . 13
plot_cem_weights_heatmap . . . 14
plot_efficiency_distributions . . . 16
plot_io_3dscatter . . . 17
plot_io_costa_frontier . . . 19
plot_io_distributions . . . 21
plot_io_efficients . . . 22
plot_io_heatmap . . . 23
plot_io_lambda_network . . . 25
plot_io_mds . . . 26
plot_io_parcoo . . . 28
plot_io_pca_biplot . . . 30
plot_io_peer_network . . . 31
plot_io_radar . . . 33
plot_io_scatter . . . 34
plot_io_som . . . 36
plot_io_som_components . . . 37
plot_panel_io_biplot . . . 39
print.dea_data . . . 41
print.dea_som . . . 41
taiwanese_banks . . . 42

Index **44**

as_dea_data *Coerce to a dea_data object*

Description

Coerce to a dea_data object

Usage

```
as_dea_data(x, ...)
```

Arguments

x	A dea_data object, or a data frame passed on to dea_data.
...	Passed to dea_data when x is not already a dea_data object.

Value

A dea_data object.

Examples

```
df <- data.frame(city = c("A", "B"), i_x = c(4, 7), o_y = c(5, 8))
d <- as_dea_data(df)   # coerces a data frame to dea_data
as_dea_data(d)         # already a dea_data: returned unchanged
```

chinese_cities *Inputs and outputs of 35 major Chinese cities*

Description

A benchmark data envelopment analysis (DEA) dataset describing 35 major Chinese cities by three inputs and three outputs. Originally introduced by Sueyoshi (1992), it is the example dataset used to illustrate the DEA-Viz methods in Ashkiani (2019) and has appeared in several DEA studies.

Usage

```
chinese_cities
```

Format

A data frame with 35 rows and 7 variables:

DMU City name (the decision-making unit label).

industrial_labour_force Input 1: industrial labour force.

working_funds Input 2: working funds.

investments Input 3: investment.

gross_industrial_output Output 1: gross industrial output.

profit_and_tax Output 2: profit and tax.

retail_sales Output 3: retail sales.

Details

Columns 2–4 are inputs and columns 5–7 are outputs; column 1 holds the DMU labels. Because the columns are not `i_`/`o_` prefixed, pass them explicitly to `dea_data` (see Examples).

Source

Sueyoshi, T. (1992). Measuring the industrial performance of Chinese cities by data envelopment analysis. *Socio-Economic Planning Sciences*, 26(2), 75-88. doi:10.1016/00380121(92)90015W

Reproduced as the worked example in Ashkiani, S. (2019), Four Essays on Data Visualization and Anomaly Detection of Data Envelopment Analysis Problems (PhD thesis), Universitat Autonoma de Barcelona.

Examples

```
d <- dea_data(
  chinese_cities,
  inputs  = c("industrial_labour_force", "working_funds", "investments"),
  outputs = c("gross_industrial_output", "profit_and_tax", "retail_sales"),
  id      = "DMU"
)
d
```

`compute_efficiency` *Compute DEA efficiency scores*

Description

Computes data envelopment analysis (DEA) efficiency scores for a set of decision-making units, under any of the returns-to-scale assumptions supported by `dea`. The numerical work is done by the **Benchmarking** package; this function wraps it so the input/output data follow the `dea_data` contract and the call sits inside one consistent entry point.

Usage

```
compute_efficiency(
  x,
  rts = c("crs", "vrs", "drs", "irs", "irs2", "fdh", "add"),
  orientation = "in",
  dual = TRUE,
  slack = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data using the default i_/o_ prefix detection. For non-default column selection (explicit inputs/outputs/id), build the object first with dea_data and pass it in.
rts	Returns to scale, one of "crs", "vrs", "drs", "irs", "irs2", "fdh" or "add".
orientation	Measurement orientation, forwarded to dea (e.g. "in", "out" or "graph").
dual	Logical; if TRUE the multiplier (dual) solution is also returned. Set FALSE to skip it when only the scores are needed.
slack	Logical; if TRUE slacks are computed as well.
...	Further arguments forwarded to dea.

Details

The returns-to-scale assumption is selected with rts: in particular "crs" (constant) and "vrs" (variable) are the two most common models, but "drs", "irs", "irs2", "fdh" and "add" are available as well.

Value

The Farrell object returned by dea, whose $eff component holds the efficiency scores.

See Also

dea, which performs the computation; dea_data for the data contract.

Examples

```
df <- data.frame(i_x = c(2, 3, 4), o_y = c(1, 2, 2))
compute_efficiency(df, rts = "crs")$eff
compute_efficiency(df, rts = "vrs")$eff
```

`compute_multiplier_weights` *Compute DEA multiplier (weight) solutions*

Description

Solves the input-oriented, constant-returns-to-scale (CCR) *multiplier* DEA model for every decision-making unit and returns the optimal efficiency scores together with the input and output weights. A non-Archimedean lower bound `epsilon` keeps all weights strictly positive.

Usage

```
compute_multiplier_weights(x, epsilon = 1e-06)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`epsilon`	Non-Archimedean lower bound on every multiplier weight; must be a single non-negative number. Defaults to `1e-6`.

Details

For unit o the model maximises the weighted output $u^\top y_o$ subject to the normalisation $v^\top x_o = 1$ and $u^\top y_j - v^\top x_j \le 0$ for every unit j, with $u, v \ge \epsilon$. The optimal value is the unit's efficiency score.

DEA multiplier solutions are generally **not unique**: efficient units and degenerate vertices admit alternative optimal weight vectors. This function returns *one* optimal solution (the vertex the solver settles on), not a canonical one. Treat the returned weights accordingly when they feed downstream calculations such as cross-efficiency.

Value

A list with components

`eff` named numeric vector of efficiency scores.

`input_weights` matrix of input multipliers (one row per DMU, one column per input).

`output_weights` matrix of output multipliers (one row per DMU, one column per output).

All rows are labelled with the DMU labels carried by `x`.

See Also

`compute_efficiency` for scores via `dea`; `dea_data` for the data contract.

Examples

```
df <- data.frame(i_x = c(2, 3, 4), o_y = c(1, 2, 2))
w <- compute_multiplier_weights(df)
w$eff
w$input_weights
```

`compute_som` *Fit a self-organizing map to a DEA problem*

Description

Trains a hexagonal self-organizing map (SOM) on the standardised input/output data and records, for every node, the mean DEA efficiency of the DMUs mapped to it. The result feeds `plot_io_som` (and other SOM views), so the map is trained once and reused.

Usage

```
compute_som(
  x,
  rts = c("crs", "vrs"),
  xdim = 8,
  ydim = 8,
  rlen = 1000,
  seed = NULL,
  ...
)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`rts`	Returns to scale, passed to `compute_efficiency`: `"crs"` or `"vrs"`.
`xdim`, `ydim`	Grid dimensions (positive integers; default `8` each).
`rlen`	Number of training iterations (default `1000`).
`seed`	Optional single number; if supplied, SOM training is made reproducible without permanently changing the session's random state.
`...`	Further arguments passed to `som`.

Value

An object of class `dea_som`: a list with the fitted `som` (a `kohonen` object), the DMU `labels`, per-DMU `efficiency`, per-node mean `node_efficiency` (`NA` for empty nodes), and the grid dimensions.

See Also

`plot_io_som`, `som`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:12),
  i_x1 = runif(12, 2, 9), i_x2 = runif(12, 1, 5),
  o_y1 = runif(12, 3, 9), o_y2 = runif(12, 1, 4)
)
som <- compute_som(df, xdim = 4, ydim = 4, seed = 1)
som
```

`cross-efficiency` *Cross-efficiency and benevolent weights*

Description

Functions for the cross-efficiency analysis of a DEA problem: the cross-efficiency matrix itself (`compute_cross_efficiency`), the benevolent multiplier weights used to build it (`compute_cross_efficiency_weights`), and a helper to put those weights on a common scale (`standardize_weights`).

Usage

```
compute_cross_efficiency(
  x,
  approach = c("benevolent", "aggressive"),
  epsilon = 0
)

compute_cross_efficiency_weights(
  x,
  approach = c("benevolent", "aggressive"),
  epsilon = 0,
  normalize = TRUE
)

standardize_weights(weights)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`approach`	Secondary goal: `"benevolent"` (maximise others' weighted output) or `"aggressive"` (minimise it).
`epsilon`	Lower bound applied to every multiplier weight. The default `0` reproduces the plain model (weights may be zero); set a small positive value (e.g. `1e-6`) to enforce strictly positive, non-Archimedean weights.
`normalize`	Logical; if `TRUE` (default) inputs and outputs are max-normalised column-wise before solving, so the returned weights are on a comparable scale (useful for plotting).

`weights` A list with `input_weights` and `output_weights` matrices, such as the result of `compute_cross_efficiency_weights`.

Value

`compute_cross_efficiency` returns an $n \times n$ numeric matrix whose entry `[k, j]` is the efficiency of DMU `j` evaluated with the weights of DMU `k` (rows are rating units, columns are rated units). Row and column names are the DMU labels.

`compute_cross_efficiency_weights` returns a list with matrices `input_weights` and `output_weights` (one row per DMU, labelled).

`standardize_weights` returns a list of the same shape as `weights`, row-standardised and rounded to 5 digits.

Functions

- `compute_cross_efficiency()`: Compute the cross-efficiency matrix.
- `compute_cross_efficiency_weights()`: Secondary-goal multiplier weights for every DMU.
- `standardize_weights()`: Row-standardise a set of weights so that, within each DMU, the input weights sum to one and the output weights sum to one.

Secondary-goal model

Multiplier (weight) solutions in DEA are not unique, so cross-efficiency requires a secondary goal to choose among a unit's optimal weights. For unit o the model fixes the unit's own efficiency at its CRS score θ_o and then, among all weight vectors that keep it there, optimises the total weighted output of the *other* units – maximising it for the `"benevolent"` approach, minimising it for the `"aggressive"` one. The unit's own efficiencies θ_o and the fall-back weights come from a single `dea` solve.

References

Doyle, J., & Green, R. (1994). Efficiency and cross-efficiency in DEA: Derivations, meanings and uses. *Journal of the Operational Research Society*, 45(5), 567–578. doi:10.1057/jors.1994.84

Doyle, J., & Green, R. (1994). Efficiency and cross-efficiency in DEA: Derivations, meanings and uses. *Journal of the Operational Research Society*, 45(5), 567–578. doi:10.1057/jors.1994.84

Doyle, J., & Green, R. (1994). Efficiency and cross-efficiency in DEA: Derivations, meanings and uses. *Journal of the Operational Research Society*, 45(5), 567–578. doi:10.1057/jors.1994.84

See Also

`compute_efficiency`, `dea_data`

Examples

```
df <- data.frame(
  dmu   = c("A", "B", "C", "D"),
  i_x1  = c(4, 7, 8, 4),
  i_x2  = c(3, 3, 1, 2),
```

```
  o_y   = c(1, 1, 1, 1)
)
ce <- compute_cross_efficiency(df, approach = "benevolent")
round(ce, 3)
colMeans(ce)   # each unit's average cross-efficiency
```

`dea_data` *Create a DEA data object*

Description

Builds a validated `dea_data` object from a data frame, classifying columns into inputs and outputs. By default columns are recognised by an `i_` prefix (inputs) and an `o_` prefix (outputs); the prefixes are stripped from the stored names for display. Columns can also be selected explicitly via `inputs`/`outputs`, which overrides the prefixes. Any remaining column is treated as metadata; a non-numeric one (or an explicit `id`) supplies the DMU labels.

Usage

```
dea_data(data, inputs = NULL, outputs = NULL, id = NULL)
```

Arguments

`data`	A data frame (or object coercible to one) with named columns: at least one input column and one output column.
`inputs`, `outputs`	Optional column selectors (character names or integer positions). When supplied they take precedence over the `i_`/`o_` prefixes. A column may not be assigned to both.
`id`	Optional single column (name or position) holding DMU labels. If `NULL`, the first unclassified *non-numeric* column is used, falling back to the row names. Numeric id columns (e.g. integer DMU codes) are *not* auto-detected, because they cannot be told apart from an input or output that lacks an `i_`/`o_` prefix; name such a column here explicitly.

Value

An object of class `dea_data`: a list with components

`X` numeric matrix of inputs (one row per DMU).

`Y` numeric matrix of outputs (one row per DMU).

`labels` character vector of DMU labels.

See Also

`compute_efficiency`, `as_dea_data`

Examples

```
df <- data.frame(
  city  = c("A", "B", "C"),
  i_lab = c(10, 12, 9),
  i_cap = c(5, 6, 4),
  o_gdp = c(100, 130, 90)
)
dea_data(df)

# explicit selection, no prefixes needed:
df2 <- setNames(df, c("city", "lab", "cap", "gdp"))
dea_data(df2, inputs = c("lab", "cap"), outputs = "gdp", id = "city")

# numeric DMU id codes must be named explicitly:
df3 <- data.frame(dmu = 1001:1003, i_x = c(2, 3, 4), o_y = c(1, 2, 2))
dea_data(df3, id = "dmu")
```

---

plot_cem_heatmap *Heatmap of a cross-efficiency matrix*

Description

Draws the cross-efficiency matrix (CEM) as a heatmap: each cell is the efficiency of the rated DMU (x-axis) evaluated with the weights of the rating DMU (y-axis). The diagonal holds the simple (self) efficiencies.

Usage

```
plot_cem_heatmap(
  x,
  labels = "all",
  max.overlaps.value = 10,
  transparency = 1,
  fade = TRUE,
  x_angle = NULL,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A cross-efficiency matrix from compute_cross_efficiency, or a dea_data object / data frame from which one is computed.
labels	DMU tick labels: "all" (default) shows them, "none" hides them, and the name/id of a single DMU highlights that DMU's label(s) in colour.

`max.overlaps.value`
: Accepted for API consistency; unused here (default `10`).

`transparency`
: Opacity of the markers/areas, a single number in `[0, 1]` (default `1`).

`fade`
: Controls the single-DMU focus view. When one DMU is given to `labels`, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. `TRUE` (default) uses a sensible fade level; `FALSE` disables it; a single number in `[0, 1]` sets the alpha of the faded marks directly, where larger values fade them less (e.g. `0.4` leaves them more visible).

`x_angle`
: Angle in degrees for the x-axis tick labels, useful when the input/output (or DMU) names on the x-axis are long and overlap. `NULL` (default) keeps the plot's standard orientation; for example `x_angle = 45` tilts the labels to make them readable.

`subtitle`
: Optional subtitle shown beneath the title.

`title`
: Optional plot title.

`interactive`
: Logical; static **ggplot2** (default) or interactive **plotly**.

`...`
: When `x` is data rather than a matrix, further arguments (e.g. `approach`, `epsilon`) passed to `compute_cross_efficiency`.

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

References

Doyle, J., & Green, R. (1994). Efficiency and cross-efficiency in DEA: Derivations, meanings and uses. *Journal of the Operational Research Society*, 45(5), 567–578. doi:10.1057/jors.1994.84

See Also

`compute_cross_efficiency`, `plot_cem_unfolding`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_cem_heatmap(compute_cross_efficiency(df))
```

plot_cem_unfolding *Unfolding map of a cross-efficiency matrix*

Description

Applies multidimensional unfolding (Ashkiani and Mar-Molinero, 2017) to a cross-efficiency matrix, placing each DMU twice – once as a "rating" unit (the weights it applies) and once as a "rated" unit – so that a rating unit sits close to the units it scores highly. Dissimilarities are taken as `1 - CEM`.

Usage

```
plot_cem_unfolding(
  x,
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  fade = TRUE,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

`x`	A cross-efficiency matrix from `compute_cross_efficiency`, or a `dea_data` object / data frame from which one is computed.
`labels`	Which DMUs to label: `"none"` (default), `"all"` (label every DMU; **ggrepel** with `max.overlaps = Inf`), `"max.overlaps"` (**ggrepel** using `max.overlaps.value`), or the name/id of a single DMU to highlight just that one. Use `"id"` to print each DMU's row number inside its own marker.
`max.overlaps.value`	Passed to **ggrepel** when `labels = "max.overlaps"` (default `10`); larger values keep more crowded labels.
`transparency`	Opacity of the markers/areas, a single number in `[0, 1]` (default `0.7`).
`fade`	Controls the single-DMU focus view. When one DMU is given to `labels`, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. `TRUE` (default) uses a sensible fade level; `FALSE` disables it; a single number in `[0, 1]` sets the alpha of the faded marks directly, where larger values fade them less (e.g. `0.4` leaves them more visible).
`subtitle`	Optional subtitle shown beneath the title.
`title`	Optional plot title.
`interactive`	Logical; static **ggplot2** (default) or interactive **plotly**.
`...`	When `x` is data rather than a matrix, further arguments (e.g. `approach`, `epsilon`) passed to `compute_cross_efficiency`.

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

References

Ashkiani, S., & Mar-Molinero, C. (2017). Visualization of cross-efficiency matrices using multidimensional unfolding. In *Recent Applications of Data Envelopment Analysis*.

See Also

`compute_cross_efficiency`, `unfolding`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
ce <- compute_cross_efficiency(df)
plot_cem_unfolding(ce)
```

---

plot_cem_weights_heatmap

Heatmap of secondary-goal multiplier weights

Description

Draws a heatmap of each DMU's secondary-goal (benevolent or aggressive) multiplier weights, one row per DMU and one column per input/output. By default the weights are row-standardised (input weights sum to one, output weights sum to one within each DMU) so the relative emphasis is comparable across units.

Usage

```
plot_cem_weights_heatmap(
  x,
  approach = c("benevolent", "aggressive"),
  standardize = TRUE,
  labels = "all",
  max.overlaps.value = 10,
  transparency = 1,
  fade = TRUE,
  x_angle = NULL,
  subtitle = NULL,
  title = NULL,
```

```
  interactive = FALSE,
  ...
)
```

Arguments

x	A weights list from compute_cross_efficiency_weights, or a dea_data object / data frame from which the weights are computed.
approach	Secondary goal used when the weights are computed from data: "benevolent" (default) or "aggressive". Ignored when x is an already-computed weights list.
standardize	Logical; if TRUE (default) the weights are row-standardised with standardize_weights before plotting.
labels	DMU tick labels: "all" (default) shows them, "none" hides them, and the name/id of a single DMU highlights that DMU's label(s) in colour.
max.overlaps.value	Accepted for API consistency; unused here (default 10).
transparency	Opacity of the markers/areas, a single number in [0, 1] (default 1).
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
x_angle	Angle in degrees for the x-axis tick labels, useful when the input/output (or DMU) names on the x-axis are long and overlap. NULL (default) keeps the plot's standard orientation; for example x_angle = 45 tilts the labels to make them readable.
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	When x is data, further arguments passed to compute_cross_efficiency_weights.

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

References

Doyle, J., & Green, R. (1994). Efficiency and cross-efficiency in DEA: Derivations, meanings and uses. *Journal of the Operational Research Society*, 45(5), 567–578. doi:10.1057/jors.1994.84

See Also

compute_cross_efficiency_weights, standardize_weights

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_cem_weights_heatmap(df, approach = "benevolent")
```

`plot_efficiency_distributions` *Efficiency scores of the DMUs*

Description

Summarises the DEA efficiency scores of the decision-making units as a histogram (default), a boxplot, or a per-DMU bar chart. More than one returns-to-scale model can be requested at once: histograms are then stacked vertically (one panel per model) and boxplots are drawn side by side, each model in its own colour.

Usage

```
plot_efficiency_distributions(
  x,
  rts = "crs",
  orientation = "in",
  type = c("histogram", "box", "bar"),
  bins = 30,
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`rts`	Returns-to-scale model(s) passed to `compute_efficiency`; a character vector with any of `"crs"` (default), `"vrs"`, `"drs"`, `"irs"`, `"fdh"`, `"add"`. Supplying more than one compares models.
`orientation`	Measurement orientation passed to `compute_efficiency`.
`type`	`"histogram"` (default), `"box"`, or `"bar"` (one bar per DMU).

`bins`	Number of histogram bins (used when `type = "histogram"`).
`labels`	Which DMUs to mark: `"none"` (default), `"all"` (rug on histograms, jittered points on boxplots), or the name/id of a single DMU to mark with a dashed line (histogram), a point (boxplot), or a highlighted axis label (bar).
`max.overlaps.value`	Accepted for API consistency; unused here (default `10`).
`transparency`	Opacity of the markers/areas, a single number in `[0, 1]` (default `0.7`).
`subtitle`	Optional subtitle shown beneath the title.
`title`	Optional plot title.
`interactive`	Logical; static **ggplot2** (default) or interactive **plotly**.
`...`	Additional arguments passed to the underlying geom.

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

See Also

`compute_efficiency`, `plot_io_efficients`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_efficiency_distributions(df, rts = "crs")
plot_efficiency_distributions(df, rts = c("crs", "vrs"), type = "box")
```

---

plot_io_3dscatter *Interactive 3-D scatter plot of a DEA problem*

Description

Plots the DMUs in three dimensions, with each axis and the point colour set to any input/output variable or any DEA efficiency model (so a returns-to- scale efficiency such as `"crs"` or `"vrs"` can be used as a dimension). Efficiencies are computed only when referenced, each at most once. This plot is always interactive (**plotly**); it has no static form.

Usage

```
plot_io_3dscatter(
  x,
  dim_x,
  dim_y,
  dim_z,
  color = "crs",
  orientation = "in",
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  subtitle = NULL,
  title = NULL,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
dim_x, dim_y, dim_z	Names of the quantities to map to the three axes. Each is either an input/output variable name or an efficiency model ("crs", "vrs", "drs", "irs", "fdh", "add"; a trailing "_efficiency" is also accepted).
color	Quantity mapping to point colour (same options as the axes; default "crs").
orientation	Measurement orientation for efficiency scores, passed to compute_efficiency (default "in").
labels	Which DMUs to label: "none" (default), "all" / "max.overlaps" (label every point), or the name/id of a single DMU to label only that one.
max.overlaps.value	Accepted for API consistency; unused here (default 10).
transparency	Opacity of the markers/areas, a single number in [0, 1] (default 0.7).
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
...	Additional arguments passed to plotly::plot_ly.

Value

A **plotly** object.

See Also

compute_efficiency

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_3dscatter(df, dim_x = "x1", dim_y = "x2", dim_z = "y", color = "crs")
```

`plot_io_costa_frontier`

Costa bi-dimensional efficient frontier

Description

Draws the Costa et al. (2016) two-dimensional representation of a DEA problem. Each DMU's multiplier weights are standardised and used to collapse its inputs and outputs into a single weighted input `I` and weighted output `O`; efficient units fall on the `O = I` diagonal (the frontier) and inefficient units below it. Uses the constant-returns-to-scale model; weights come from `compute_efficiency`.

Usage

```
plot_io_costa_frontier(
  x,
  orientation = c("in", "out"),
  point_size = 2,
  transparency = 0.7,
  fade = TRUE,
  labels = "none",
  max.overlaps.value = 10,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`orientation`	Measurement orientation, `"in"` or `"out"`.
`point_size`	Point size (default `2`).
`transparency`	Point alpha in `[0, 1]` (default `0.7`).

`fade`	Controls the single-DMU focus view. When one DMU is given to `labels`, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. `TRUE` (default) uses a sensible fade level; `FALSE` disables it; a single number in `[0, 1]` sets the alpha of the faded marks directly, where larger values fade them less (e.g. `0.4` leaves them more visible).
`labels`	Which DMUs to label: `"none"` (default), `"all"` (label every DMU; **ggrepel** with `max.overlaps = Inf`), `"max.overlaps"` (**ggrepel** using `max.overlaps.value`), or the name/id of a single DMU to highlight just that one. Use `"id"` to print each DMU's row number inside its own marker.
`max.overlaps.value`	Passed to **ggrepel** when `labels = "max.overlaps"` (default `10`); larger values keep more crowded labels.
`subtitle`	Optional subtitle shown beneath the title.
`title`	Optional plot title.
`interactive`	Logical; static **ggplot2** (default) or interactive **plotly**.
`...`	Additional arguments passed to `geom_point`.

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

References

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

See Also

`compute_efficiency`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_costa_frontier(df, orientation = "in")
```

plot_io_distributions *Distributions of inputs and outputs*

Description

Shows the distribution of every input and output, either as faceted histograms (default) or as boxplots. By default the variables are standardised so they share a common scale.

Usage

```
plot_io_distributions(
  x,
  type = c("histogram", "box"),
  bins = 30,
  scale = TRUE,
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  x_angle = NULL,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
type	"histogram" (default) or "box".
bins	Number of histogram bins (used when type = "histogram").
scale	Logical; if TRUE (default) variables are standardised (z-scored) before plotting.
labels	Which DMUs to mark: "none" (default), "all" (a rug of every DMU on histograms, jittered points on boxplots), or the name/id of a single DMU to mark it with a dashed vertical line (histogram) or a point (boxplot).
max.overlaps.value	Accepted for API consistency with the other plots; unused here (default 10).
transparency	Opacity of the markers/areas, a single number in [0, 1] (default 0.7).
x_angle	Angle in degrees for the x-axis tick labels, useful when the input/output (or DMU) names on the x-axis are long and overlap. NULL (default) keeps the plot's standard orientation; for example x_angle = 45 tilts the labels to make them readable.
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; if FALSE (default) returns a static **ggplot2** object, if TRUE an interactive **plotly** object.
...	Additional arguments passed to the underlying geom (geom_histogram or geom_boxplot).

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

See Also

`dea_data`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_distributions(df)
plot_io_distributions(df, type = "box")
```

---

plot_io_efficients *Counts of efficient and inefficient DMUs*

Description

Classifies the DMUs as efficient or inefficient (efficiency score equal to 1 within a tolerance) and draws a bar chart of the counts, annotated with the percentage of DMUs in each group.

Usage

```
plot_io_efficients(
  x,
  rts = c("crs", "vrs", "drs", "irs", "fdh", "add"),
  orientation = "in",
  tol = 1e-06,
  labels = TRUE,
  transparency = 0.7,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`rts`	Returns to scale passed to `compute_efficiency`.
`orientation`	Measurement orientation passed to `compute_efficiency`.

`tol`	Tolerance for treating a score as efficient (default `1e-6`).
`labels`	Logical; if `TRUE` (default) the count and percentage are printed on each bar.
`transparency`	Opacity of the markers/areas, a single number in `[0, 1]` (default `0.7`).
`subtitle`	Optional subtitle shown beneath the title.
`title`	Optional plot title.
`interactive`	Logical; static **ggplot2** (default) or interactive **plotly**.
`...`	Additional arguments passed to `geom_col`.

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

See Also

`plot_efficiency_distributions`, `compute_efficiency`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_efficients(df, rts = "crs")
```

---

plot_io_heatmap *Heatmap of standardised inputs and outputs*

Description

Draws a DMU-by-variable heatmap of the input and output values, with DMUs on the y-axis and variables on the x-axis (inputs labelled `I_`, outputs `O_`). By default the values are standardised column-wise so they are comparable across variables.

Usage

```
plot_io_heatmap(
  x,
  scale = TRUE,
  labels = "all",
  max.overlaps.value = 10,
  transparency = 1,
  fade = TRUE,
  x_angle = NULL,
```

```
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
scale	Logical; if TRUE (default) each variable is standardised (z-scored) before plotting.
labels	DMU tick labels: "all" (default) shows them, "none" hides them, and the name/id of a single DMU highlights that DMU's label(s) in colour.
max.overlaps.value	Accepted for API consistency; unused here (default 10).
transparency	Opacity of the markers/areas, a single number in [0, 1] (default 1).
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
x_angle	Angle in degrees for the x-axis tick labels, useful when the input/output (or DMU) names on the x-axis are long and overlap. NULL (default) keeps the plot's standard orientation; for example x_angle = 45 tilts the labels to make them readable.
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	Additional arguments passed to geom_tile.

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

See Also

dea_data

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
```

```
plot_io_heatmap(df)
```

`plot_io_lambda_network` *DEA reference (lambda) network*

Description

Projects the DMUs into two dimensions with Sammon mapping and draws the DEA reference network on top: each DMU is a node coloured by efficiency status, and an edge joins unit r to a reference peer c whenever the envelopment weight λ_{rc} exceeds `edge_threshold`. Edge width is proportional to λ, shown in the legend.

Usage

```
plot_io_lambda_network(
  x,
  rts = c("crs", "vrs"),
  edge_threshold = 0.1,
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  fade = TRUE,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`rts`	Returns to scale, passed to `compute_efficiency`: `"crs"` or `"vrs"`.
`edge_threshold`	Minimum envelopment weight for an edge to be drawn (single non-negative number, default `0.1`).
`labels`	Which DMUs to label: `"none"` (default), `"all"` (label every DMU; **ggrepel** with `max.overlaps = Inf`), `"max.overlaps"` (**ggrepel** using `max.overlaps.value`), or the name/id of a single DMU to highlight just that one. Use `"id"` to print each DMU's row number inside its own marker.
`max.overlaps.value`	Passed to **ggrepel** when `labels = "max.overlaps"` (default `10`); larger values keep more crowded labels.
`transparency`	Node alpha in `[0, 1]` (default `0.7`); lower values reveal overlapping nodes.

fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	Further arguments passed to sammon (e.g. niter).

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

References

Porembski, M., Breitenstein, K., & Alpar, P. (2005). Visualizing efficiency and reference relations in data envelopment analysis with an application to the branches of a German bank. *Journal of Productivity Analysis*, 23(2), 203–221. doi:10.1007/s1112300513285

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5), 401–409. doi:10.1109/TC.1969.222678

See Also

compute_efficiency, plot_io_peer_network, sammon

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_lambda_network(df, rts = "crs")
```

---

plot_io_mds *MDS co-plot of a DEA problem*

Description

Places the DMUs in two dimensions by multidimensional scaling (Adler and Raveh, 2008) and colours them by a chosen quantity: a DEA efficiency score, or one of the input/output variables (or their output-to-input ratios).

Usage

```
plot_io_mds(
  x,
  transform = c("none", "ratio"),
  dist_method = c("euclidean", "manhattan", "maximum", "canberra", "minkowski"),
  mds_type = c("ratio", "interval", "ordinal", "mspline"),
  encode = "crs",
  point_size = 2,
  transparency = 0.7,
  fade = TRUE,
  labels = "none",
  max.overlaps.value = 10,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
transform	What to compute distances on: "none" (the original inputs and outputs) or "ratio" (all output/input ratios).
dist_method	Distance measure, passed to dist.
mds_type	Scaling level, passed to smacofSym.
encode	What to colour by: an efficiency model ("crs", "vrs", "drs", "irs", "fdh", "add") or the name of one of the distance-space columns.
point_size	Point size (default 2).
transparency	Point alpha in [0, 1] (default 0.7).
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
labels	Which DMUs to label: "none" (default), "all" (label every DMU; **ggrepel** with max.overlaps = Inf), "max.overlaps" (**ggrepel** using max.overlaps.value), or the name/id of a single DMU to highlight just that one. Use "id" to print each DMU's row number inside its own marker.
max.overlaps.value	Passed to **ggrepel** when labels = "max.overlaps" (default 10); larger values keep more crowded labels.
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title. If NULL (default) one is generated.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	Additional arguments passed to geom_point.

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

References

Adler, N., & Raveh, A. (2008). Presenting DEA graphically. *Omega*, 36(5), 715–729.

de Leeuw, J., & Mair, P. (2009). Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software*, 31(3), 1–30. doi:10.18637/jss.v031.i03

See Also

`compute_efficiency`, `smacofSym`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:8),
  i_x1 = c(4, 7, 8, 4, 2, 5, 6, 3),
  i_x2 = c(3, 3, 1, 2, 4, 2, 5, 1),
  o_y  = c(5, 8, 6, 7, 3, 9, 4, 6)
)
plot_io_mds(df, encode = "crs")
```

---

plot_io_parcoo *Parallel coordinates plot of a DEA problem*

Description

Draws a parallel-coordinates plot with one axis per input and output, each axis min-max scaled to `[0, 1]` so they are comparable. Each DMU is one line; lines are coloured by efficient/inefficient status.

Usage

```
plot_io_parcoo(
  x,
  efficiency = c("crs", "vrs", "none"),
  orientation = "in",
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  fade = TRUE,
  x_angle = NULL,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
efficiency	Efficiency model used to colour the lines by efficient/inefficient status: "crs" (default), "vrs" or "none".
orientation	Measurement orientation for the efficiency scores, passed to compute_efficiency (default "in").
labels	Which DMUs to label at the right-hand axis: "none" (default), "all", "max.overlaps", or the name/id of a single DMU (also highlights that DMU's line).
max.overlaps.value	Passed to **ggrepel** when labels = "max.overlaps" (default 10).
transparency	Opacity of the markers/areas, a single number in [0, 1] (default 0.7).
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
x_angle	Angle in degrees for the x-axis tick labels, useful when the input/output (or DMU) names on the x-axis are long and overlap. NULL (default) keeps the plot's standard orientation; for example x_angle = 45 tilts the labels to make them readable.
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	Additional arguments passed to geom_line.

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

See Also

compute_efficiency, plot_io_radar

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_parcoo(df, efficiency = "crs")
```

`plot_io_pca_biplot` *PCA biplot of a DEA problem*

Description

Runs a principal component analysis on the standardised input/output data and draws a biplot: DMUs as points coloured by efficiency status, and the variables as labelled vectors. Efficiency comes from `compute_efficiency`.

Usage

```
plot_io_pca_biplot(
  x,
  rts = c("crs", "vrs"),
  vector_size = 1,
  text_size = 3,
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  fade = TRUE,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  seed = NA,
  ...
)
```

Arguments

`x`	A `dea_data` object, or a data frame coerced by `as_dea_data`.
`rts`	Returns to scale, passed to `compute_efficiency`: `"crs"` or `"vrs"`.
`vector_size`	Positive multiplier scaling the length of the variable vectors relative to the point cloud (default `1`).
`text_size`	Size of the variable labels (default `3`).
`labels`	Which DMUs to label: `"none"` (default), `"all"` (label every DMU; **ggrepel** with `max.overlaps = Inf`), `"max.overlaps"` (**ggrepel** using `max.overlaps.value`), or the name/id of a single DMU to highlight just that one. Use `"id"` to print each DMU's row number inside its own marker.
`max.overlaps.value`	Passed to **ggrepel** when `labels = "max.overlaps"` (default `10`); larger values keep more crowded labels.
`transparency`	Point alpha in `[0, 1]` (default `0.7`).
`fade`	Controls the single-DMU focus view. When one DMU is given to `labels`, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen

	DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
seed	Optional seed for geom_text_repel label placement (static mode only). Default NA.
...	Additional arguments passed to geom_point.

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

See Also

compute_efficiency

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y1 = c(5, 8, 6, 7, 3, 9),
  o_y2 = c(2, 3, 1, 2, 1, 4)
)
plot_io_pca_biplot(df, rts = "crs")
```

---

plot_io_peer_network *DEA peer (target) network*

Description

Draws a directed network of DMUs: each inefficient unit has arrows pointing to its efficient peers (the targets in its DEA reference set, i.e. the efficient units with a positive envelopment weight). Arrowheads stop short of the target node so the direction stays readable. Nodes are coloured by efficiency status and, optionally, efficient nodes are sized by how often they are a target.

Usage

```
plot_io_peer_network(
  x,
  rts = c("crs", "vrs"),
  layout = c("pca", "mds", "circle", "fr", "stress"),
  size_by_peers = FALSE,
```

```
  tol = 1e-06,
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  fade = TRUE,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
rts	Returns to scale, passed to compute_efficiency: "crs" or "vrs".
layout	Node layout: "pca" (default, first two principal components of the standardised inputs/outputs), "mds" (classical multidimensional scaling), "circle", "fr" (Fruchterman-Reingold force-directed) or "stress" (stress majorization). The last two require the **igraph** package (and **graphlayouts** for true stress majorization; otherwise a Kamada-Kawai layout is used).
size_by_peers	Logical; if TRUE, efficient nodes are sized by the number of inefficient units that target them. If FALSE (default) all nodes have the same size.
tol	Tolerance for a positive envelopment weight (default 1e-6).
labels	Which DMUs to label: "none" (default), "all" (label every DMU; **ggrepel** with max.overlaps = Inf), "max.overlaps" (**ggrepel** using max.overlaps.value), or the name/id of a single DMU to highlight just that one. Use "id" to print each DMU's row number inside its own marker.
max.overlaps.value	Passed to **ggrepel** when labels = "max.overlaps" (default 10); larger values keep more crowded labels.
transparency	Node alpha in [0, 1] (default 0.7); lower values reveal overlapping nodes.
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's subnetwork; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	Additional arguments passed to geom_point.

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

See Also

compute_efficiency, plot_io_lambda_network

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_peer_network(df, rts = "crs", size_by_peers = TRUE)
```

---

plot_io_radar *Radar plot of inputs and outputs*

Description

Draws a radar (spider) plot with one axis per input and output, each min-max scaled to [0, 1], and one closed polygon per DMU with straight edges. Polygons are coloured by efficient/inefficient status. A radial counterpart to plot_io_parcoo.

Usage

```
plot_io_radar(
  x,
  efficiency = c("crs", "vrs", "none"),
  orientation = "in",
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  fade = TRUE,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
efficiency	Efficiency model used to colour the DMUs by efficient/inefficient status: "crs" (default), "vrs" or "none".
orientation	Measurement orientation for the efficiency scores.

labels	DMU emphasis: "none" (default) or "all" draw every DMU normally; the name/id of a single DMU outlines that DMU's polygon in bold so it stands out. (Variable axis labels are always shown.)
max.overlaps.value	Accepted for API consistency; unused here (default 10).
transparency	Opacity of the markers/areas, a single number in [0, 1] (default 0.7).
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; if FALSE (default) a static **ggplot2** radar; if TRUE an interactive **plotly** scatterpolar.
...	Additional arguments passed to geom_polygon (static mode).

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

See Also

plot_io_parcoo, compute_efficiency

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_radar(df, efficiency = "crs")
```

---

plot_io_scatter *Pairwise scatter plots of inputs, outputs and efficiency scores*

Description

Builds the pairwise scatter plots of a chosen set of columns: any subset of the inputs and outputs, plus optionally one or more efficiency scores. For k selected columns it draws all choose(k, 2) pairwise panels. For example, selecting one input, one output and the CRS score gives three panels.

Usage

```
plot_io_scatter(
  x,
  vars = NULL,
  efficiency = NULL,
  color = c("crs", "vrs", "drs", "irs", "fdh", "add", "none"),
  correlation = TRUE,
  orientation = "in",
  labels = "none",
  max.overlaps.value = 10,
  transparency = 0.7,
  fade = TRUE,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

x	A dea_data object, or a data frame coerced by as_dea_data.
vars	Character vector of input/output column names to include. Defaults to all inputs and outputs.
efficiency	Optional character vector of efficiency models to compute and include as columns, any of "crs", "vrs", "drs", "irs", "fdh", "add".
color	Efficiency model used to colour points by efficient/inefficient status: one of "crs" (default), "vrs", "drs", "irs", "fdh", "add", or "none".
correlation	Logical; if TRUE (default) the Pearson correlation is printed in the top-left corner of each panel.
orientation	Measurement orientation used when computing efficiency scores.
labels	Which DMUs to label: "none" (default), "all", "max.overlaps", or the name/id of a single DMU to highlight it. (Uses geom_text with overlap-thinning rather than **ggrepel**, since repel is unsafe inside facets.)
max.overlaps.value	Accepted for API consistency; unused here (default 10).
transparency	Point alpha in [0, 1] (default 0.7).
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	Additional arguments passed to geom_point.

Value

A **ggplot2** object, or a **plotly** object when `interactive = TRUE`.

See Also

`dea_data`, `compute_efficiency`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:6),
  i_x1 = c(4, 7, 8, 4, 2, 5),
  i_x2 = c(3, 3, 1, 2, 4, 2),
  o_y  = c(5, 8, 6, 7, 3, 9)
)
plot_io_scatter(df, vars = c("x1", "y"), efficiency = "crs")
```

---

plot_io_som *Plot a self-organizing map of DEA efficiency*

Description

Draws the hexagonal SOM as a property map, colouring each node by the mean efficiency of the DMUs assigned to it, and (optionally) labels the nodes with the DMU names. Uses base graphics, so the plot is produced as a side effect on the active device.

Usage

```
plot_io_som(
  som,
  labels = "all",
  max.overlaps.value = 10,
  jitter_sd = 0.1,
  seed = NULL,
  transparency = 0.7,
  subtitle = NULL,
  title = "SOM: mean efficiency per node",
  ...
)
```

Arguments

`som`	A `dea_som` object from `compute_som`, or a `dea_data` object / data frame, in which case the map is fitted first.
`labels`	Which DMUs to write on the map: `"all"` (default), `"none"`, or the name/id of a single DMU to label only that one.

`max.overlaps.value`
: Accepted for API consistency; unused here (default `10`).

`jitter_sd`
: Standard deviation of the random jitter applied to label positions so co-located labels do not overlap (default `0.1`).

`seed`
: Optional single number making the label jitter reproducible without changing the session's random state.

`transparency`
: Opacity of the markers/areas, a single number in `[0, 1]` (default `0.7`).

`subtitle`
: Optional subtitle shown beneath the title.

`title`
: Plot title.

`...`
: When `som` is data rather than a `dea_som`, further arguments passed to `compute_som`.

Value

The `dea_som` object, invisibly.

See Also

`compute_som`

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:12),
  i_x1 = runif(12, 2, 9), i_x2 = runif(12, 1, 5),
  o_y1 = runif(12, 3, 9), o_y2 = runif(12, 1, 4)
)
som <- compute_som(df, xdim = 4, ydim = 4, seed = 1)
plot_io_som(som)
```

---

plot_io_som_components

Plot the component planes of a DEA self-organizing map

Description

Draws one SOM property map per input/output variable – the "component planes" – each node coloured by that variable's codebook weight. This shows how each input and output varies across the trained map. Uses base graphics, so the panels are produced as a side effect on the active device.

Usage

```
plot_io_som_components(
  som,
  variables = NULL,
  ncol = 3,
  labels = TRUE,
  transparency = 0.7,
  subtitle = NULL,
  title = NULL,
  ...
)
```

Arguments

som	A dea_som object from compute_som, or a dea_data object / data frame, in which case the map is fitted first.
variables	Optional character vector selecting which variables to draw (default: all). Names are matched against the input/output column names.
ncol	Number of panels per row (positive integer; default 3).
labels	Logical; if TRUE (default) each panel is titled with its variable name.
transparency	Opacity of the markers/areas, a single number in [0, 1] (default 0.7).
subtitle	Optional subtitle shown beneath the title.
title	Optional overall title drawn above the panel grid.
...	When som is data rather than a dea_som, further arguments passed to compute_som.

Value

The dea_som object, invisibly.

See Also

compute_som, plot_io_som

Examples

```
df <- data.frame(
  dmu  = paste0("D", 1:12),
  i_x1 = runif(12, 2, 9), i_x2 = runif(12, 1, 5),
  o_y1 = runif(12, 3, 9), o_y2 = runif(12, 1, 4)
)
som <- compute_som(df, xdim = 4, ydim = 4, seed = 1)
plot_io_som_components(som)
```

plot_panel_io_biplot *Panel PCA biplot with DMU trajectories*

Description

For panel (long-format) DEA data, computes a single principal-component embedding of the pooled, standardised inputs and outputs (so the axes and loading vectors are fixed), then draws each DMU's path through time as a connected trajectory. Points are coloured by a DEA efficiency score and the arrow on each trajectory points from the earliest to the latest period.

Usage

```
plot_panel_io_biplot(
  panel_data,
  inputs = NULL,
  outputs = NULL,
  id = "Label",
  period = "Period",
  color = c("crs", "vrs"),
  orientation = "in",
  vector_size = 1,
  transparency = 0.7,
  fade = TRUE,
  labels = "none",
  max.overlaps.value = 10,
  period_labels = FALSE,
  size_by_efficiency = FALSE,
  subtitle = NULL,
  title = NULL,
  interactive = FALSE,
  ...
)
```

Arguments

panel_data	A long-format data frame with one row per DMU-period, containing an identifier column and a period column (see id and period); the remaining columns are inputs/outputs.
inputs, outputs	Optional input/output column selectors, given either as character names or as integer column positions (e.g. inputs = 3:5). If NULL (default), columns are recognised by the i_ and o_ prefixes, as in dea_data.
id, period	The identifier and period columns, given as a name or an integer position (default "Label" and "Period").
color	Efficiency model for point colour, computed per period: "crs" (default) or "vrs".
orientation	Measurement orientation for the efficiency scores.

vector_size	Positive multiplier scaling the loading vectors.
transparency	Point opacity. Either a single number in [0, 1] (constant alpha, default 0.7) or the string "efficiency", in which case point alpha is mapped to the efficiency score (more efficient DMUs drawn more opaque).
fade	Controls the single-DMU focus view. When one DMU is given to labels, all other marks (and, for the network plots, everything outside that DMU's sub-network; for the panel biplot, the other trajectories) are faded so the chosen DMU stands out. TRUE (default) uses a sensible fade level; FALSE disables it; a single number in [0, 1] sets the alpha of the faded marks directly, where larger values fade them less (e.g. 0.4 leaves them more visible).
labels	Which DMU trajectories to label at their latest period. One of "none" (default, no labels), "all" (label every DMU, with **ggrepel** max.overlaps = Inf), "max.overlaps" (label with **ggrepel** using max.overlaps.value), or the name/id of a single DMU to highlight just that one. Use "id" to print each DMU's row number inside its own marker.
max.overlaps.value	Passed to **ggrepel** when labels = "max.overlaps" (default 10); larger values keep more crowded labels.
period_labels	Logical; if TRUE each point is labelled with its period. Default FALSE.
size_by_efficiency	Logical; if TRUE point size grows with the efficiency score. Default FALSE.
subtitle	Optional subtitle shown beneath the title.
title	Optional plot title.
interactive	Logical; static **ggplot2** (default) or interactive **plotly**.
...	Additional arguments passed to geom_point.

Value

A **ggplot2** object, or a **plotly** object when interactive = TRUE.

See Also

plot_io_pca_biplot, compute_efficiency

Examples

```
# Real multi-period example: 22 Taiwanese banks over 2009-2011.
# Columns can be given by position (inputs are columns 3-5, outputs 6-8) ...
plot_panel_io_biplot(
  taiwanese_banks, id = "DMU", period = "Year",
  inputs = 3:5, outputs = 6:8, labels = "Cathay"
)

# ... or by name, with the loading vectors and a single bank's trajectory.
plot_panel_io_biplot(
  taiwanese_banks, id = "DMU", period = "Year",
  inputs  = c("labour", "physical_capital", "purchased_funds"),
```

```
  outputs = c("demand_deposits", "short_term_loans", "long_term_loans"),
  transparency = "efficiency", labels = "all"
)

# A toy long-format frame using the i_/o_ prefix convention also works.
panel <- data.frame(
  Label  = rep(paste0("D", 1:4), each = 3),
  Period = rep(2019:2021, times = 4),
  i_x1   = c(4,5,5, 7,6,6, 8,8,7, 4,3,4),
  i_x2   = c(3,3,2, 3,4,3, 1,2,2, 2,2,1),
  o_y    = c(5,6,7, 8,8,9, 6,6,7, 7,8,8)
)
plot_panel_io_biplot(panel, labels = "D2")   # highlight a single DMU
```

`print.dea_data` *Print a DEA data object*

Description

Print a DEA data object

Usage

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

Arguments

`x`	A `dea_data` object, as returned by `dea_data`.
`...`	Ignored.

Value

`x`, invisibly.

`print.dea_som` *Print a fitted DEA self-organizing map*

Description

Print a fitted DEA self-organizing map

Usage

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

Arguments

x	A dea_som object, as returned by compute_som.
...	Ignored.

Value

x, invisibly.

---

taiwanese_banks *Inputs and outputs of 22 Taiwanese commercial banks, 2009-2011*

Description

A balanced panel (long-format) data envelopment analysis (DEA) dataset describing 22 Taiwanese commercial banks over three years (2009-2011) by three inputs and three outputs. It is the worked example of Kao and Liu (2014) on multi-period efficiency measurement, and is the example dataset for plot_panel_io_biplot.

Usage

```
taiwanese_banks
```

Format

A data frame with 66 rows (22 banks $\times$ 3 years) and 8 variables:

DMU Bank name (the decision-making unit label).

Year Period: 2009, 2010 or 2011.

labour Input 1 (I1): labour.

physical_capital Input 2 (I2): physical capital.

purchased_funds Input 3 (I3): purchased funds.

demand_deposits Output 1 (O1): demand deposits.

short_term_loans Output 2 (O2): short-term loans.

long_term_loans Output 3 (O3): medium- and long-term loans.

Details

Columns 3-5 are inputs (I1-I3) and columns 6-8 are outputs (O1-O3); DMU and Year identify each observation. The input/output columns are not i_/o_ prefixed, so pass them by position (e.g. inputs = 3:5, outputs = 6:8) or by name to plot_panel_io_biplot (see Examples).

Source

Kao, C. and Liu, S.-T. (2014). Multi-period efficiency measurement in data envelopment analysis: The case of Taiwanese commercial banks. *Omega*, 47, 90-98. doi:10.1016/j.omega.2013.09.001

Examples

```
plot_panel_io_biplot(
  taiwanese_banks, id = "DMU", period = "Year",
  inputs = 3:5, outputs = 6:8, labels = "Cathay"
)
```

Index

∗ **datasets**
chinese_cities, 3
taiwanese_banks, 42

as_dea_data, 3, *5–8*, *10*, *16*, *18, 19*, *21, 22*, *24, 25*, *27*, *29, 30*, *32, 33*, *35*

chinese_cities, 3
compute_cross_efficiency, *11–14*
compute_cross_efficiency *(*cross-efficiency*)*, 8
compute_cross_efficiency_weights, *15*
compute_cross_efficiency_weights *(*cross-efficiency*)*, 8
compute_efficiency, 4, *6, 7*, *9, 10*, *16–20*, *22, 23*, *25, 26*, *28–34*, *36*, *40*
compute_multiplier_weights, 6
compute_som, 7, *36–38*, *42*
cross-efficiency, 8

dea, *4–6*, *9*
dea_data, *3–6*, *9*, 10, *22*, *24*, *36*, *39*, *41*
dist, *27*

geom_text_repel, *31*

plot_cem_heatmap, 11
plot_cem_unfolding, *12*, 13
plot_cem_weights_heatmap, 14
plot_efficiency_distributions, 16, *23*
plot_io_3dscatter, 17
plot_io_costa_frontier, 19
plot_io_distributions, 21
plot_io_efficients, *17*, 22
plot_io_heatmap, 23
plot_io_lambda_network, 25, *33*
plot_io_mds, 26
plot_io_parcoo, 28, *33, 34*
plot_io_pca_biplot, 30, *40*
plot_io_peer_network, *26*, 31
plot_io_radar, *29*, 33
plot_io_scatter, 34
plot_io_som, *7*, 36, *38*
plot_io_som_components, 37
plot_panel_io_biplot, 39, *42*
print.dea_data, 41
print.dea_som, 41

sammon, *26*
smacofSym, *27, 28*
som, *7*
standardize_weights, *15*
standardize_weights *(*cross-efficiency*)*, 8

taiwanese_banks, 42

unfolding, *14*