Package 'ecpromethee'

June 29, 2026

Type Package

Title EC-PROMETHEE Multi-Criteria Decision Method

Version 0.1.0

Description Implements the EC-PROMETHEE multi-criteria decision method described by Basilio, Pereira and Yigit (2023) <doi:10.3390/math11214432>. The method combines objective criteria weights from ENTROPY and CRITIC with optional subjective weights, generates random normalized weights inside criterion-specific ranges, and aggregates repeated PROMETHEE II rankings into a final score.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Marcio Basilio [aut, cre],
Valdecy Pereira [aut],
Fatih Yigit [aut]

Maintainer Marcio Basilio <marciopbasilio@gmail.com>

Repository CRAN

Date/Publication 2026-06-29 13:00:35 UTC

Contents

`ecpromethee-package` *EC-PROMETHEE Multi-Criteria Decision Method*

Description

Implements ENTROPY, CRITIC, PROMETHEE II and EC-PROMETHEE aggregation.

References

Basilio, M. P.; Pereira, V.; Yigit, F. (2023). New Hybrid EC-Promethee Method with Multiple Iterations of Random Weight Ranges: Applied to the Choice of Policing Strategies. *Mathematics*, 11(21), 4432. doi:10.3390/math11214432

`critic_weights` *Calculate CRITIC Criteria Weights*

Description

Computes objective criteria weights using Criteria Importance Through Intercriteria Correlation (CRITIC).

Usage

```
critic_weights(x, directions = "max")
```

Arguments

`x`	Numeric decision matrix. Rows are alternatives and columns are criteria.
`directions`	Character vector indicating whether each criterion should be maximized or minimized. Use `"max"` or `"min"`.

Value

A numeric vector of normalized criteria weights.

Examples

```
x <- matrix(c(1, 2, 3, 2, 3, 4, 4, 4, 5), nrow = 3)
critic_weights(x)
```

ec_promethee *Apply the EC-PROMETHEE Method*

Description

EC-PROMETHEE combines ENTROPY and CRITIC objective weights with optional subjective weights. It creates criterion-specific weight ranges, generates random normalized weights inside those ranges, runs PROMETHEE II for each random weight set, and aggregates ordinal positions into a final ranking.

Usage

```
ec_promethee(
  x,
  subjective_weights = NULL,
  iterations = 1000,
  directions = "max",
  preference = "usual",
  q = 0,
  p = 0,
  s = 1,
  seed = NULL
)
```

Arguments

x	Numeric decision matrix. Rows are alternatives and columns are criteria.
subjective_weights	Optional numeric vector of decision-maker weights.
iterations	Number of PROMETHEE II rankings to generate.
directions	Character vector indicating whether each criterion should be maximized or minimized. Use "max" or "min".
preference	Preference function type for each criterion.
q	Indifference threshold for each criterion.
p	Preference threshold for each criterion.
s	Gaussian threshold for each criterion.
seed	Optional random seed.

Value

A list with final ranking, component weights, weight ranges, random weights and the iteration rank matrix.

Examples

```
x <- matrix(c(7, 9, 6, 8, 7, 7, 6, 8, 9), nrow = 3, byrow = TRUE)
ec_promethee(x, iterations = 20, seed = 123)$final_ranking
```

entropy_weights *Calculate ENTROPY Criteria Weights*

Description

Computes objective criteria weights using the ENTROPY method described in Basilio, Pereira and Yigit (2023).

Usage

```
entropy_weights(x)
```

Arguments

x	Numeric decision matrix. Rows are alternatives and columns are criteria.

Value

A numeric vector of normalized criteria weights.

Examples

```
x <- matrix(c(1, 2, 3, 2, 3, 4, 4, 4, 5), nrow = 3)
entropy_weights(x)
```

policing_mode *Policing Strategy Decision Matrix*

Description

A 14 by 20 decision matrix using the "Mode" scenario from Basilio, Pereira and Yigit (2023). Rows are policing strategy alternatives and columns are crime-demand criteria.

Usage

```
policing_mode
```

Format

A numeric matrix with 14 alternatives and 20 criteria.

Value

A numeric matrix with class `matrix`. Rows represent policing strategy alternatives, columns represent crime-demand criteria, and the cell values are the criterion scores used by EC-PROMETHEE.

Source

Basilio, M. P.; Pereira, V.; Yigit, F. (2023). doi:10.3390/math11214432

Examples

```
dim(policing_mode)
ec_promethee(policing_mode, iterations = 5, seed = 123)$final_ranking
```

`preference_function` *Apply PROMETHEE Preference Functions*

Description

Applies one of the six standard PROMETHEE preference functions.

Usage

```
preference_function(d, type = "usual", q = 0, p = 0, s = 1)
```

Arguments

`d`	Numeric vector of deviations between two alternatives.
`type`	Preference function type. Supported values are `"usual"`, `"u_shape"`, `"v_shape"`, `"level"`, `"linear"` and `"gaussian"`.
`q`	Indifference threshold.
`p`	Preference threshold.
`s`	Gaussian threshold.

Value

A numeric vector of preference degrees in `[0, 1]`.

Examples

```
preference_function(c(-1, 0, 1), type = "usual")
preference_function(c(0, 1, 2), type = "v_shape", p = 2)
```

promethee_ii *Rank Alternatives With PROMETHEE II*

Description

Runs PROMETHEE II and returns the complete ranking of alternatives.

Usage

```
promethee_ii(
  x,
  weights,
  directions = "max",
  preference = "usual",
  q = 0,
  p = 0,
  s = 1
)
```

Arguments

x	Numeric decision matrix. Rows are alternatives and columns are criteria.
weights	Numeric vector of criteria weights.
directions	Character vector indicating whether each criterion should be maximized or minimized. Use "max" or "min".
preference	Preference function type for each criterion.
q	Indifference threshold for each criterion.
p	Preference threshold for each criterion.
s	Gaussian threshold for each criterion.

Value

A data frame with alternatives, positive flow, negative flow, net flow and rank.

Examples

```
x <- matrix(c(7, 9, 6, 8, 7, 7, 6, 8, 9), nrow = 3, byrow = TRUE)
promethee_ii(x, weights = c(1 / 3, 1 / 3, 1 / 3))
```

random_weights *Generate Random Normalized Weights*

Description

Generates random normalized weights from EC-PROMETHEE criterion ranges.

Usage

```
random_weights(lower, upper, iterations, seed = NULL)
```

Arguments

lower	Numeric vector with lower limits.
upper	Numeric vector with upper limits.
iterations	Number of random weight sets to generate.
seed	Optional random seed.

Value

A numeric matrix with `iterations` rows and one column per criterion.

Examples

```
random_weights(c(0.1, 0.2), c(0.4, 0.5), iterations = 3, seed = 1)
```

weight_ranges *Build EC-PROMETHEE Weight Ranges*

Description

Creates criterion-specific lower and upper limits from ENTROPY, CRITIC and optional subjective decision-maker weights.

Usage

```
weight_ranges(entropy, critic, subjective_weights = NULL)
```

Arguments

entropy	Numeric vector of ENTROPY weights.
critic	Numeric vector of CRITIC weights.
subjective_weights	Optional numeric vector of subjective weights.

Value

A data frame with columns `criterion`, `lower` and `upper`.

Examples

```
weight_ranges(c(0.4, 0.6), c(0.5, 0.5))
```

Index