

# Package ‘PrometheeTools’

May 7, 2026

**Title** PROMETHEE and GLNF for Ranking and Sorting Problems

**Version** 0.1.0

**Description** PROMETHEE (Preference Ranking Organisation METHod for Enrichment of Evaluations) based method assesses alternatives to obtain partial and complete rankings. The package also provides the GLNF (Global Local Net Flow) sorting algorithm to classify alternatives into ordered categories, as well as an index function to measure the classification quality. Barrera, F., Segura, M., & Maroto, C. (2023) <[doi:10.1111/itor.13288](https://doi.org/10.1111/itor.13288)>. Brans, J.P.; De Smet, Y., (2016) <[doi:10.1007/978-1-4939-3094-4\\_6](https://doi.org/10.1007/978-1-4939-3094-4_6)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** ggplot2

**URL** <https://github.com/ifelipebj/PrometheeTools>

**BugReports** <https://github.com/ifelipebj/PrometheeTools/issues>

**NeedsCompilation** no

**Author** Felipe Barrera [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0003-3236-6966>>),  
Marina Segura [aut] (ORCID: <<https://orcid.org/0000-0002-6068-960X>>),  
Concepcion Maroto [aut] (ORCID: <<https://orcid.org/0000-0001-8512-3197>>)

**Maintainer** Felipe Barrera <[ifelipebj@hotmail.com](mailto:ifelipebj@hotmail.com)>

**Repository** CRAN

**Date/Publication** 2023-09-01 11:30:11 UTC

## Contents

|                       |          |
|-----------------------|----------|
| GLNF . . . . .        | 2        |
| PROMETHEEII . . . . . | 4        |
| SILS . . . . .        | 5        |
| <b>Index</b>          | <b>8</b> |

**Description**

This function applies the GLNF Sorting (Global Local Net Flow Sorting) algorithm to classify the alternatives into ordered groups according to the decision-maker's preferences in multiple criteria context. GLNF sorting is based on PROMETHEE net flows and a set of limiting profiles. This algorithm starts from a global classification (global search) that is enhanced by two local searches, intra-categorical and inter-categorical.

**Usage**

```
GLNF(matrix_evaluation, data_criteria)
```

**Arguments**

`matrix_evaluation`

The matrix includes the values for all alternatives. The alternatives and limiting profiles are row and columns correspond to the evaluation criteria.

`data_criteria`

Matrix with the parameter information (rows) for each criterion (columns). The rows of parameters are in the following order: Function Type, Indifference Threshold, Preference Threshold, Objective and Weight.

**Details**

- The Limiting Profiles should be presented as rows in the `matrix_evaluation`. The name must start with the letter "r" followed by the profile number (e.g., "r1", "r2").
- For  $k$  categories, there should be  $k + 1$  limiting profiles. To create  $k$  groups the set of limit profiles are defined, where  $r_1$  is preferred to  $r_2, \dots, r_{k+1}$ .
- The types of preference function are as follows: "linear", "v-shape", "usual", "u-shape", "level" and "gaussian".
- The preference and indifference thresholds depend on the type of function selected. The preference threshold requires definition (is non-zero) for all functions except for "usual" and "u-shaped". The indifference threshold is non-zero for "linear", "level" and "u-shaped" functions.
- In the objective write "max" to maximize or "min" to minimize.
- The sum of the weights of all criteria must be equal to 1.

**Value**

-Global Matrix with the results of the global search where positive, negative and net flow, and its preclassification are defined for each alternative.

-Local Matrices with the results of the first local search. PROMETHEE is applied to each group obtained in the global search. The alternatives are divided according to their positive or negative sign from the net flows obtained from PROMETHEE.

-Local2 Matrices with the results of the second local search, where the alternatives are divided according to their sign from net flows are obtained after applying PROMETHEE between each pair of neighbour categories.

-Class Final classification of the alternatives results.

## References

Barrera, F., Segura, M., & Maroto, C. (2023) Online. Multicriteria sorting method based on global and local search for supplier segmentation. *International Transactions in Operational Research*. DOI:10.1111/itor.13288

## See Also

[PROMETHEEII](#)

## Examples

```
matrix_evaluation <- data.frame (
  Alternative = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
                 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
                 "r1", "r2", "r3", "r4", "r5"),
  Monetary = c(21.52, 68.09, 184.94, 237.62, 14.29, 12.78, 91.53, 11.39, 264.79, 12.74,
               274.41, 3.75, 47.92, 34.5, 45.89, 39.92, 31.18, 273.23, 16.39, 3.91,
               20.09, 6.52, 26.62, 28.47, 7.57, 69.2, 420.95, 12.01, 85.88, 8.78,
               6816.80, 120, 40, 20, 0),
  Recency = c(0, 0, 0, 0, 3, 5, 0, 6, 0, 3,
              1, 0, 1, 0, 0, 0, 0, 0, 2, 1,
              0, 0, 0, 0, 5, 1, 0, 0, 1, 4,
              0, 1, 7, 8, 12),
  Frequency = c(7, 5, 12, 12, 1, 3, 9, 2, 12, 4,
                11, 3, 10, 10, 11, 11, 12, 12, 7, 1,
                5, 2, 9, 11, 4, 10, 12, 3, 10, 2,
                12, 10, 8, 4, 1),
  Financial_score = c(66, 58, 83, 68, 68, 69, 77, 55, 77, 53,
                    78, 35, 84, 75, 71, 64, 56, 55, 52, 30,
                    66, 50, 65, 53, 54, 82, 68, 53, 62, 43,
                    100, 80, 75, 65, 0),
  Length = c(4, 3, 3, 2, 2, 2, 2, 3, 2, 4,
             3, 3, 1, 1, 2, 5, 4, 2, 2, 5,
             4, 5, 1, 4, 2, 1, 5, 1, 1, 2,
             5, 4, 3, 2, 1))
data_criteria <- data.frame(
  Parameter = c("Function Type", "Indifference Threshold",
               "Preference Threshold", "Objective", "Weight"),
  Frequency = c("linear", 0, 3, "max", 0.2),
  Monetary = c("linear", 30.00, 120, "max", 0.4),
  Recency = c("usual", 0.00, 0.00, "min", 0.1),
  Financial_score = c("linear", 0.00, 10, "max", 0.2),
  Length = c("usual", 0.00, 0.00, "max", 0.1))
RS <- GLNF(matrix_evaluation, data_criteria)
```

```
RS$Class
RS$Global
RS$Local1
RS$Local2
```

---

 PROMETHEEII

*The PROMETHEE Outranking Method*


---

### Description

PROMETHEE is a multicriteria method that quantifies preference relationships and obtains the positive, negative and net flows of the alternatives, generating rankings that reflect the decision-maker's preferences. This function applies PROMETHEE I (partial ranking) and PROMETHEE II (full ranking). This function can handle a large number of alternatives.

### Usage

```
PROMETHEEII(matrix_evaluation, data_criteria)
```

### Arguments

`matrix_evaluation`

The matrix includes the values for all alternatives. The alternatives profiles are rows and columns correspond to the evaluation criteria.

`data_criteria`

Matrix with the parameter information (rows) for each criterion (columns). The rows of parameters are in the following order: Function Type, Indifference Threshold, Preference Threshold, Objective and Weight.

### Details

- The types of preference function are as follows: "linear", "v-shape", "usual", "u-shape", "level" and "gaussian".
- The preference and indifference thresholds depend on the type of function selected. The preference threshold requires definition (is non-zero) for all functions except for "usual" and "u-shaped". The indifference threshold is non-zero for "linear", "level" and "u-shaped" functions.
- In the objective write "max" to maximize or "min" to minimize.
- The sum of the weights of all criteria must be equal to 1.
- This implementation of PROMETHEEII is designed to handle a large number of alternatives (it has been tested with 10,000 alternatives) much higher than the previous implementations in R (promethee123 and PROMETHEE).

### Value

-NF Matrix with positive and negative flows (PROMETHEE I) and net flows for complete ranking (PROMETHEE II).

-NFC Net flows matrix by criterion.

## References

Brans, J.P.; De Smet, Y., (2016). PROMETHEE Methods. In: Multiple Criteria Decision Analysis. State of the Art Surveys, Figuera, J., Greco, S., Ehrgott, M.; Springer: New York, USA, pp. 187-219. DOI: 10.1007/978-1-4939-3094-4\_6.

## Examples

```
matrix_evaluation <- data.frame (

Alternative = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
               11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
               21, 22, 23, 24, 25, 26, 27, 28, 29, 30),
Monetary = c(21.52, 68.09, 184.94, 237.62, 14.29, 12.78, 91.53, 11.39, 264.79, 12.74,
             274.41, 3.75, 47.92, 34.5, 45.89, 39.92, 31.18, 273.23, 16.39, 3.91,
             20.09, 6.52, 26.62, 28.47, 7.57, 69.2, 420.95, 12.01, 85.88, 8.78),
Recency = c(0, 0, 0, 0, 3, 5, 0, 6, 0, 3,
            1, 0, 1, 0, 0, 0, 0, 0, 2, 1,
            0, 0, 0, 0, 5, 1, 0, 0, 1, 4),
Frequency = c(7, 5, 12, 12, 1, 3, 9, 2, 12, 4,
             11, 3, 10, 10, 11, 11, 12, 12, 7, 1,
             5, 2, 9, 11, 4, 10, 12, 3, 10, 2),
Financial_score = c(66, 58, 83, 68, 68, 69, 77, 55, 77, 53,
                  78, 35, 84, 75, 71, 64, 56, 55, 52, 30,
                  66, 50, 65, 53, 54, 82, 68, 53, 62, 43),
Length = c(4, 3, 3, 2, 2, 2, 2, 3, 2, 4,
          3, 3, 1, 1, 2, 5, 4, 2, 2, 5,
          4, 5, 1, 4, 2, 1, 5, 1, 1, 2))
data_criteria <- data.frame(
Parameter = c("Function Type", "Indifference Threshold",
             "Preference Threshold", "Objective", "Weight"),
Frequency = c("linear", 0, 3, "max", 0.2),
Monetary = c("linear", 30.00, 120, "max", 0.4),
Recency = c("usual", 0.00, 0.00, "min", 0.1),
Financial_score = c("linear", 0.00, 10, "max", 0.2),
Length = c("usual", 0.00, 0.00, "max", 0.1))
RS <- PROMETHEEII(matrix_evaluation, data_criteria)
RS$NF
RS$NFC
```

---

SILS

*Quality Index of Silhouette for Sorting*

---

## Description

This function computes the quality index for SILS (Silhouette for Sorting), which relies on PROMETHEE II net flows to assess the classifications generated by PROMETHEE-based ordered sorting methods.

## Usage

```
SILS(matrix_evaluation, data_criteria, k, SILS_plot = FALSE)
```

### Arguments

|                                |  |
|--------------------------------|--|
| <code>matrix_evaluation</code> | The matrix includes the values for all alternatives and limiting profiles are rows and columns correspond to the evaluation criteria. The last column indicates the alternative classification.                  |
| <code>data_criteria</code>     | Matrix with the parameter information (rows) for each criterion (columns). The rows of parameters are in the following order: Function Type, Indifference Threshold, Preference Threshold, Objective and Weight. |
| <code>k</code>                 | The number of categories to be evaluated.  |
| <code>SILS_plot</code>         | Boolean value indicating whether to generate a stacked bar chart representing the SILS values.   |

### Details

- The categories corresponding to the classifications to be assessed should be indicated in the last column of `matrix_evaluation` in ordinal numbers, where 1 is the most preferred group.
- Enter the same criteria parameters and limiting profiles that you used to obtain the classifications with PROMETHEE II.
- The Limiting Profiles used must be presented as rows in the `matrix_evaluation`. The name must start with the letter "r" followed by the profile number (e.g., "r1", "r2"). For k categories, there should define k + 1 limiting profiles.
- The types of preference function are as follows: "linear", "v-shape", "usual", "u-shape", "level" and "gaussian".
- The preference and indifference thresholds depend on the type of function selected. The preference threshold requires definition (is non-zero) for all functions except for "usual" and "u-shaped". The indifference threshold is non-zero for "linear", "level" and "u-shaped" functions.
- In the objective write "max" to maximize or "min" to minimize.
- The sum of the weights of all criteria must be equal to 1.

### Value

- A data frame with the SILS values for each alternative according to the classification entered.
- Stacked bar chart representing the SILS values for each alternative with horizontal control limits.

### References

Barrera, F., Segura, M., & Maroto, C. (2023) Online. Multicriteria sorting method based on global and local search for supplier segmentation. *International Transactions in Operational Research*. DOI:10.1111/itor.13288

### See Also

[PROMETHEEII](#)

**Examples**

```

k <- 4
matrix_evaluation <- data.frame (

Alternative = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
               11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
               21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
               "r1", "r2", "r3", "r4", "r5"),
Monetary = c(21.52, 68.09, 184.94, 237.62, 14.29, 12.78, 91.53, 11.39, 264.79, 12.74,
             274.41, 3.75, 47.92, 34.5, 45.89, 39.92, 31.18, 273.23, 16.39, 3.91,
             20.09, 6.52, 26.62, 28.47, 7.57, 69.2, 420.95, 12.01, 85.88, 8.78,
             6816.80, 120, 40, 20, 0),
Recency = c(0, 0, 0, 0, 3, 5, 0, 6, 0, 3,
            1, 0, 1, 0, 0, 0, 0, 0, 2, 1,
            0, 0, 0, 0, 5, 1, 0, 0, 1, 4,
            0, 1, 7, 8, 12),
Frequency = c(7, 5, 12, 12, 1, 3, 9, 2, 12, 4,
            11, 3, 10, 10, 11, 11, 12, 12, 7, 1,
            5, 2, 9, 11, 4, 10, 12, 3, 10, 2,
            12, 10, 8, 4, 1),
Financial_score = c(66, 58, 83, 68, 68, 69, 77, 55, 77, 53,
                  78, 35, 84, 75, 71, 64, 56, 55, 52, 30,
                  66, 50, 65, 53, 54, 82, 68, 53, 62, 43,
                  100, 80, 75, 65, 0),
Length = c(4, 3, 3, 2, 2, 2, 2, 3, 2, 4,
           3, 3, 1, 1, 2, 5, 4, 2, 2, 5,
           4, 5, 1, 4, 2, 1, 5, 1, 1, 2,
           5, 4, 3, 2, 1),
Category = c(3, 3, 1, 1, 4, 3, 2, 4, 1, 3,
            1, 4, 2, 2, 2, 2, 2, 1, 3, 4,
            3, 3, 3, 2, 4, 2, 1, 4, 3, 4,
            NA, NA, NA, NA, NA))

data_criteria <- data.frame(
  Parameter = c("Function Type", "Indifference Threshold",
               "Preference Threshold", "Objective", "Weight"),
  Frequency = c("linear", 0, 3, "max", 0.2),
  Monetary = c("linear", 30.00, 120, "max", 0.4),
  Recency = c("usual", 0.00, 0.00, "min", 0.1),
  Financial_score = c("linear", 0.00, 10, "max", 0.2),
  Length = c("usual", 0.00, 0.00, "max", 0.1))
RS <- SILS(matrix_evaluation, data_criteria, k, SILS_plot = TRUE)
print(RS)

```

# Index

GLNF, [2](#)

PROMETHEEII, [3](#), [4](#), [6](#)

SILS, [5](#)