Package: ecolRxC (via r-universe)

September 14, 2024

Type Package

Title Ecological Inference of RxC Tables by Latent Structure Approaches

Version 0.1.1-10

Description Estimates RxC (R by C) vote transfer matrices (ecological contingency tables) from aggregate data building on Thomsen (1987) and Park (2008) approaches. References: Park, W.-H. (2008). "Ecological Inference and Aggregate Analysis of Election". PhD Dissertation. University of Michigan. <https://deepblue.lib.umich.edu/bitstream/handle/2027.42/58525/wpark_1.pdf> Thomsen, S.R. (1987, ISBN:87-7335-037-2). "Danish Elections 1920 79: a Logit Approach to Ecological Analysis and Inference". Politica, Aarhus, Denmark.

License GPL (>= 2)

Encoding UTF-8

Imports stats

Suggests ggplot2, scales

RoxygenNote 7.1.1

NeedsCompilation no

Author Jose M. Pavía [aut, cre] (<https://orcid.org/0000-0002-0129-726X>), Søren Risbjerg Thomsen [aut]

Maintainer Jose M. Pavía <jose.m.pavia@uv.es>

Repository CRAN

Date/Publication 2023-03-31 07:50:02 UTC

Contents

`ecolRxC` *Ecological Inference of RxC Tables by Latent Structure Approaches*

Description

Estimates JxK (RxC) vote transfer matrices (ecological contingency tables) based on Thomsen (1987) and Park (2008) approaches.

Usage

```
ecolRxC(
  votes.election1,
  votes.election2,
  scale = "probit",
  method = "Thomsen",
  local = TRUE,
  census.changes = c("adjust", "raw", "regular", "ordinary", "enriched",
    "simultaneous", "semifull", "full", "gold"),
  reference = NULL,
  confidence = NULL,
  B = 500,
  Yule.aprox = FALSE,
  tol = 1e-06,
  ...
)
```

Arguments

`votes.election1`
: data.frame (or matrix) of order IxJ1 with the votes gained by (or the counts corresponding to) the J1 (social classes) political options competing (available) on election 1 (or origin) in the I units considered.

`votes.election2`
: data.frame (or matrix) of order IxK2 with the votes gained by (or the counts corresponding to) the K2 political options competing (available) on election 2 (or destination) in the I (territorial) units considered.

`scale`
: A character string indicating the type of transformation to be applied to the vote fractions for applying ecological inference. Only `logit` and `probit` are allowed. Default, `probit`.

`method`
: A character string indicating the algorithm to be used for adjusting (making congruent with the observed margins) the initial crude fractions attained in a 2x2 fashion. Only `Thomsen` (see sec. 4.3 in Thomsen, 1987) and `IPF` (iterative proportional fitting, also known as raking) are allowed. This argument has no effect in the 2x2 case. Default, `Thomsen`.

`local` A TRUE/FALSE argument indicating whether local solutions (solutions for each polling unit) must be computed. In that case, the global solution is attained as composition/aggregation of local solutions. When `method = "Thomsen"` local solutions are always computed. Default `TRUE`.

`census.changes` A character string informing about the level of information available in `votes.election1` and `votes.election2` regarding new entries and exits of the election censuses between the two elections or indicating how their sum discrepancies should be handled. This argument allows the eight options discussed in Pavia (2022) as well as an adjusting option. This argument admits nine values: `adjust`, `raw`, `regular`, `ordinary`, `simultaneous`, `enriched`, `semifull`, `full` and `gold`. See **Details**. Default, `adjust`.

`reference` A vector of two components indicating (parties) options in election 1 and 2, respectively, to be used as reference with `method = "Thomsen"`. This has not effect with `method = "IPF"`. The references can be indicated by name or by position. If `reference = NULL`, the final solution is constructed as a weighted average of all the congruent solutions attained after considering as references all combinations of options. Default `NULL`.

`confidence` A number between 0 and 1 to be used as level of confidence for the confidence intervals of the transition rates. By default `NULL`. If `confidence = NULL`, confidence intervals are not computed.

`B` An integer indicating the number of samples to be drawn from each crude estimated confidence interval for estimating final confidence intervals when either R (J) or C (K) is higher than two. This is not relevant for the 2x2 case. It can take a while to compute confidence intervals, mainly when `method = Thomsen`. In general computation burden grows with `B`. Default, `500`.

`Yule.aprox` `TRUE/FALSE` argument indicating if either Thomsen (1987)'s formula (3.44), based on a binormal, or Thomsen (1987)'s formula (3.46), based on Yule's approximation of tetrachoric correlation, should be use to estimate cross-proportions. Default `FALSE`, formula (3.44).

`tol` A number indicating the level of precision to be used to stop the adjustment of initial/crude count estimates reached using a 2x2 approach in a general RxC case. This is not relevant for the 2x2 case. Default, `0.000001`.

`...` Other arguments to be passed to the function. Not currently used.

Details

Description of the `census.changes` argument in more detail.

- `adjust`: The default value. This is the simplest solution for handling discrepancies between the total number of counts for the first and second elections. With this value the J1 column-aggregations of the counts in `votes.election1` of the first election are proportionally adjusted to equal the aggregation of the counts in `votes.election2` of the second election. In this scenario, J is equal to J1 and K equal to K2.
- `raw`: This argument accounts for a scenario with two elections elapsed at least some months where only the raw election data recorded in the I (territorial) units, in which the electoral space under study is divided, are available and net entries and net exits are approached from the available information. In this scenario, net exits and net entries are estimated according

to Pavia (2022). When both net entries and exits are no null, constraint (15) of Pavia (2022) applies: no transfer between entries and exits are allowed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.

- `simultaneous`: This is the value to be used in classical ecological inference problems, such as in ecological studies of social or racial voting, and in scenarios with two simultaneous elections. In this scenario, the sum by rows of `votes.election1` and `votes.election2` must coincide.
- `regular`: This value accounts for a scenario with two elections elapsed at least some months where (i) the column J1 of `votes.election1` corresponds to new (young) electors who have the right to vote for the first time, (ii) net exits and maybe other additional net entries are computed according to Pavia (2022). When both net entries and exits are no null, constraints (13) and (15) of Pavia (2022) apply. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `ordinary`: This value accounts for a scenario with two elections elapsed at least some months where (i) the column K1 of `votes.election2` corresponds to electors who died in the interperiod election, (ii) net entries and maybe other additional net exits are computed according to Pavia (2022). When both net entries and net exits are no null, constraints (14) and (15) of Pavia (2022) apply. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `enriched`: This value accounts for a scenario that somewhat combine `regular` and `ordinary` scenarios. We consider two elections elapsed at least some months where (i) the column J1 of `votes.election1` corresponds to new (young) electors who have the right to vote for the first time, (ii) the column K2 of `votes.election2` corresponds to electors who died in the interperiod election, (iii) other (net) entries and (net) exits are computed according to Pavia (2022). When both net entries and net exits are no null, constraints (12) to (15) of Pavia (2022) apply. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `semifull`: This value accounts for a scenario with two elections elapsed at least some months, where: (i) the column J1 = J of `votes.election1` totals new electors (young and immigrants) that have the right to vote for the first time in each polling unit and (ii) the column K2 = K of `votes.election2` corresponds to total exits of the census lists (due to death or emigration). In this scenario, the sum by rows of `votes.election1` and `votes.election2` must agree and constraint (15) of Pavia (2022) apply.
- `full`: This value accounts for a scenario with two elections elapsed at least some months, where J = J1, K = K2 and (i) the column J - 1 of `votes.election1` totals new (young) electors that have the right to vote for the first time, (ii) the column J of `votes.election1` measures new immigrants that have the right to vote and (iii) the column K of `votes.election2` corresponds to total exits of the census lists (due to death or emigration). In this scenario, the sum by rows of `votes.election1` and `votes.election2` must agree and constraints (13) and (15) of Pavia (2022) apply.
- `gold`: This value accounts for a scenario similar to `full`, where J = J1, K = K2 and total exits are separated out between exits due to emigration (column K - 1 of `votes.election2`) and death (column K of `votes.election2`). In this scenario, the sum by rows of `votes.election1` and `votes.election2` must agree. Constraints (12) to (15) of Pavia (2022) apply.

Value

A list with the following components

`VTM`
A matrix of order JxK (RxC) with the estimated proportions of the row-standardized vote transitions from election 1 to election 2. In `raw`, `regular`, `ordinary` and `enriched` scenarios, this matrix includes the row and the column corresponding to net entries and net exits (when they are present). When `local = TRUE` (default), this matrix is obtained as composition of the local solutions.

`VTM.votes`
A matrix of order JxK (RxC) with the estimated vote transfers from election 1 to election 2. In `raw`, `regular`, `ordinary` and `enriched` scenarios, this matrix includes the row and the column corresponding to net entries and net exits (when they are present). When `local = TRUE` (default), this matrix is obtained as aggregation of the local solutions.

`VTM.global`
A matrix of order JxK (RxC) with the estimated proportions of the row-standardized vote transitions from election 1 to election 2, attained directly from the global (whole electoral space) proportions. When `local = FALSE`. `VTM` and `VTM.global` coincide. In `raw`, `regular`, `ordinary` and `enriched` scenarios, this matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`VTM.votes.global`
A matrix of order JxK (RxC) with the estimated vote transfers from election 1 to election 2, attained directly from the global proportions. When `local = FALSE`, `VTM.votes` and `VTM.votes.global` coincide. In `raw`, `regular`, `ordinary` and `enriched` scenarios, this matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`VTM.lower`
A matrix of order JxK (RxC) with the estimated lower limits of the confidence intervals for the proportions of the row-standardized vote transitions from election 1 to election 2. In `raw`, `regular`, `ordinary` and `enriched` scenarios, this matrix includes the row and the column corresponding to net entries and net exits (when they are present). When `confidence = NULL` this is a `NULL` object.

`VTM.upper`
A matrix of order JxK (RxC) with the estimated upper limits of the confidence intervals for the proportions of the row-standardized vote transitions from election 1 to election 2. In `raw`, `regular`, `ordinary` and `enriched` scenarios, this matrix includes the row and the column corresponding to net entries and net exits (when they are present). When `confidence = NULL` this is a `NULL` object.

`VTM.crude.global`
A matrix of order JxK (RxC) with the (inconsistent) crude estimated proportions for the row-standardized vote transitions from election 1 to election 2 in the whole space attained in a 2x2 fashion before making them consistent using the iterative proportional fitting algorithm or the Thomsen iteratuve algortihm. In `raw`, `regular`, `ordinary` and `enriched` scenarios, this matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`VTM.units`
An array of order JxKxI (RxCxI) with the estimated proportions of the row-standardized vote transitions from election 1 to election 2 attained for each unit. When `local = FALSE`, this is a `NULL` object. In `raw`, `regular`, `ordinary` and `enriched` scenarios, each unit matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`VTM.votes.units`
An array of order JxKxI (RxCxI) with the estimated transfer of votes from elec-

tion 1 to election 2 attained for each unit. When `local = FALSE`, this is a `NULL` object. In `raw`, `regular`, `ordinary` and `enriched` scenarios, each unit matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`VTM.lower.units`
: An array of order JxKxI (RxCxI) with the estimated lower limits of the confidence intervals for the proportions of the row-standardized vote transitions from election 1 to election 2 corresponding to each unit. When either `local = FALSE` or `confidence = NULL`, this is a `NULL` object. In `raw`, `regular`, `ordinary` and `enriched` scenarios, each unit matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`VTM.upper.units`
: An array of order JxKxI (RxCxI) with the estimated upper limits of the confidence intervals for the proportions of the row-standardized vote transitions from election 1 to election 2 corresponding to each unit. When either `local = FALSE` or `confidence = NULL`, this is a `NULL` object. In `raw`, `regular`, `ordinary` and `enriched` scenarios, each unit matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`VTM.crude.units`
: An array of order JxKxI (RxCxI) with the (inconsistent) crude estimated proportions of the row-standardized vote transitions from election 1 to election 2 attained for each unit in a 2x2 fashion before making them consistent using the iterative proportional fitting algorithm or the Thomsen iterative algorithm. When `local = FALSE`, this is a NULL object. In `raw`, `regular`, `ordinary` and `enriched` scenarios, each unit matrix includes the row and the column corresponding to net entries and net exits (when they are present).

`correlations`
: A matrix of order JxK (Rxc) with the across units correlations between options for the proportions in the transformed scale.

`reference.outputs`
: A list with three components: `vjk.averages`, `vjk.by.reference` and `vjk.units.by.reference`. The first component `vjk.averages` is a JxKx8 array with eight different global solutions of transfer matrix of votes attained after combining with different weights each of the solutions obtained using the different combinations of a row and a column option as reference. The second component `vjk.by.reference` is a JxKx(J1·K1) array with the J1K1 different global solutions of transfer matrix of votes attained after choosing as reference all the possible combination of a row and a column option. The third component `vjk.units.by.reference` is a JxKxIx(J1·K1) array with the local solutions linked to `vjk.by.reference`. When either `method = "IPF"` or `reference` is not `NULL`, this is a `NULL` object.

`iter`
: A vector of either length 1 (when `reference` is different of `NULL`) or J1·K1 with the number of iterations needed by the Thomsen algorithm to reach convergence for each reference pair. When `method = "Thomsen"` this is a `NULL` object.

`inputs`
: A list containing all the objects with the values used as arguments by the function.

Note

This function somewhere builds on the .ado (STATA) functions written by Won-ho Park, in 2002.

Author(s)

Jose M. Pavia, <pavia@uv.es>

References

Achen, C.H. (2000). The Thomsen Estimator for Ecological Inference (Unpublished manuscript). University of Michigan.

Park, W.-H. (2008). Ecological Inference and Aggregate Analysis of Elections. PhD Dissertation. University of Michigan.

Pavia, J.M. (2022). Adjustment of initial estimates of voter transition probabilities to guarantee consistency and completeness.

Thomsen, S.R. (1987). Danish Elections 1920-79: a Logit Approach to Ecological Analysis and Inference. Politica, Aarhus, Denmark.

Examples

```
votes1 <- structure(list(P1 = c(16L, 4L, 13L, 6L, 1L, 16L, 6L, 17L, 48L, 14L),
                         P2 = c(8L, 3L, 0L, 5L, 1L, 4L, 7L, 6L, 28L, 8L),
                         P3 = c(38L, 11L, 11L, 3L, 13L, 39L, 14L, 34L, 280L, 84L),
                         P4 = c(66L, 5L, 18L, 39L, 30L, 57L, 35L, 65L, 180L, 78L),
                         P5 = c(14L, 0L, 5L, 2L, 4L, 21L, 6L, 11L, 54L, 9L),
                         P6 = c(8L, 2L, 5L, 3L, 0L, 7L, 7L, 11L, 45L, 17L),
                         P7 = c(7L, 3L, 5L, 2L, 3L, 17L, 7L, 13L, 40L, 8L)),
                         row.names = c(NA, 10L), class = "data.frame")
votes2 <- structure(list(C1 = c(2L, 1L, 2L, 2L, 0L, 4L, 0L, 4L, 19L, 14L),
                         C2 = c(7L, 3L, 1L, 7L, 2L, 5L, 3L, 10L, 21L, 6L),
                         C3 = c(78L, 7L, 28L, 42L, 28L, 84L, 49L, 85L, 260L, 100L),
                         C4 = c(56L, 14L, 20L, 7L, 19L, 54L, 22L, 50L, 330L, 91L),
                         C5 = c(14L, 3L, 6L, 2L, 3L, 14L, 8L, 8L, 45L, 7L)),
                         row.names = c(NA, 10L), class = "data.frame")
example <- ecolRxC(votes1, votes2, method = "IPF")$VTM
```

plot.ecolRxC *Graphical representation of a RxC ecological inference (vote transfer) matrix*

Description

Plot method for objects obtained with ecolRxC.

Usage

```
## S3 method for class 'ecolRxC'
plot(
  x,
  margins = TRUE,
  digits = 2,
  row.names = NULL,
  col.names = NULL,
  size.numbers = 6,
  size.labels = 4,
  size.margins = 4,
  colour.cells = ”cyan4”,
  colour.grid = ”cornsilk2”,
  alpha = 0.5,
  which = NULL,
  ...,
  show.plot = TRUE
)
```

Arguments

`x` An object output of the **ecolRxC** function.

`margins` A TRUE/FALSE argument informing if the margins of the matrix should be displayed. Default TRUE.

`digits` Integer indicating the number of decimal places to be shown. Default, 2.

`row.names` Names to be used for the rows of the matrix.

`col.names` Names to be used for the columns of the matrix.

`size.numbers` A reference number indicating the average font size to be used for the transfer numbers. Default, 6.

`size.labels` A number indicating the font size to be used for labels. Default, 4.

`size.margins` A number indicating the font size to be used for margin numbers. Default, 4.

`colour.cells` Background base colour for cells.

`colour.grid` Colour to be used for grid lines.

`alpha` A [0,1] number of colour transparency.

`which` A vector of integers informing the units for which the aggregate transfer matrix should be plotted. Default, NULL, the global matrix is shown.

`...` Other arguments passed on to methods. Not currently used.

`show.plot` A TRUE/FALSE indicating if the plot should be displayed as a side-effect. By default, TRUE.

Value

Invisibly returns the (ggplot) description of the plot, which is a list with components that contain the plot itself, the data, information about the scales, panels etc.

Note

ggplot2 is needed to be installed for this function to work.

Author(s)

Jose M. Pavia, <pavia@uv.es>

Examples

```
votes1 <- structure(list(P1 = c(16L, 4L, 13L, 6L, 1L, 16L, 6L, 17L, 48L, 14L),
                         P2 = c(8L, 3L, 0L, 5L, 1L, 4L, 7L, 6L, 28L, 8L),
                         P3 = c(38L, 11L, 11L, 3L, 13L, 39L, 14L, 34L, 280L, 84L),
                         P4 = c(66L, 5L, 18L, 39L, 30L, 57L, 35L, 65L, 180L, 78L),
                         P5 = c(14L, 0L, 5L, 2L, 4L, 21L, 6L, 11L, 54L, 9L),
                         P6 = c(8L, 2L, 5L, 3L, 0L, 7L, 7L, 11L, 45L, 17L),
                         P7 = c(7L, 3L, 5L, 2L, 3L, 17L, 7L, 13L, 40L, 8L)),
                         row.names = c(NA, 10L), class = "data.frame")
votes2 <- structure(list(C1 = c(2L, 1L, 2L, 2L, 0L, 4L, 0L, 4L, 19L, 14L),
                         C2 = c(7L, 3L, 1L, 7L, 2L, 5L, 3L, 10L, 21L, 6L),
                         C3 = c(78L, 7L, 28L, 42L, 28L, 84L, 49L, 85L, 260L, 100L),
                         C4 = c(56L, 14L, 20L, 7L, 19L, 54L, 22L, 50L, 330L, 91L),
                         C5 = c(14L, 3L, 6L, 2L, 3L, 14L, 8L, 8L, 45L, 7L)),
                         row.names = c(NA, 10L), class = "data.frame")
example <- ecolRxC(votes1, votes2, method = "IPF")
p <- plot(example, show.plot = FALSE)
p
```

`print.ecolRxC` *Print a summary of an output of the ecolRxC function*

Description

Print method for objects obtained with the ecolRxC function.

Usage

```
## S3 method for class 'ecolRxC'
print(x, ..., margins = TRUE, digits = 2)
```

Arguments

`x` An object output of the **ecolRxC** function.

`...` Other arguments passed on to methods. Not currently used.

`margins` A TRUE/FALSE argument informing if the margins of the transition matrix should be displayed. Default TRUE.

`digits` Integer indicating the number of decimal places to be shown. Default, 2.

Value

No return value, called for side effects.

Author(s)

Jose M. Pavia, <pavia@uv.es>

Examples

```
votes1 <- structure(list(P1 = c(16L, 4L, 13L, 6L, 1L, 16L, 6L, 17L, 48L, 14L),
                         P2 = c(8L, 3L, 0L, 5L, 1L, 4L, 7L, 6L, 28L, 8L),
                         P3 = c(38L, 11L, 11L, 3L, 13L, 39L, 14L, 34L, 280L, 84L),
                         P4 = c(66L, 5L, 18L, 39L, 30L, 57L, 35L, 65L, 180L, 78L),
                         P5 = c(14L, 0L, 5L, 2L, 4L, 21L, 6L, 11L, 54L, 9L),
                         P6 = c(8L, 2L, 5L, 3L, 0L, 7L, 7L, 11L, 45L, 17L),
                         P7 = c(7L, 3L, 5L, 2L, 3L, 17L, 7L, 13L, 40L, 8L)),
                         row.names = c(NA, 10L), class = "data.frame")
votes2 <- structure(list(C1 = c(2L, 1L, 2L, 2L, 0L, 4L, 0L, 4L, 19L, 14L),
                         C2 = c(7L, 3L, 1L, 7L, 2L, 5L, 3L, 10L, 21L, 6L),
                         C3 = c(78L, 7L, 28L, 42L, 28L, 84L, 49L, 85L, 260L, 100L),
                         C4 = c(56L, 14L, 20L, 7L, 19L, 54L, 22L, 50L, 330L, 91L),
                         C5 = c(14L, 3L, 6L, 2L, 3L, 14L, 8L, 8L, 45L, 7L)),
                         row.names = c(NA, 10L), class = "data.frame")
example <- ecolRxC(votes1, votes2, method = "IPF")
print(example, digits = 1, margins = TRUE)
```

---

print.summary.ecolRxC *Print a summary of a summary.ecolRxC object*

Description

Print method for summary.ecolRxC objects

Usage

```
## S3 method for class 'summary.ecolRxC'
print(x, ..., margins = TRUE, digits = 2)
```

Arguments

x	An summary.ecolRxC class object.
...	Other arguments passed on to methods. Not currently used.
margins	A TRUE/FALSE argument informing if the margins of the transition matrix should be displayed. Default TRUE.
digits	Integer indicating the number of decimal places to be shown. Default, 2.

Value

No return value, called for side effects.

`summary.ecolRxC` *Summarize a ecolRxC output object*

Description

Summary method for objects obtained with the ecolRxC function

Usage

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

Arguments

`object` An object output of the **ecolRxC** function.

`...` Other arguments passed on to methods. Not currently used.

Value

An object of class `"summary.ecolRxC"`. A list with four components:

`prop.matrix` A matrix of order JxK (RxC) with the estimated proportions of the row-standardized vote transitions from election 1 to election 2.

`counts.matrix` A matrix of order JxK (RxC) with the estimated vote transfers from election 1 to election 2.

`row.margins` A vector of length R with aggregate observed distribution of votes in election 1.

`col.margins` A vector of length C with aggregate observed distribution of votes in election 2.

Author(s)

Jose M. Pavia, <pavia@uv.es>

Examples

```
votes1 <- structure(list(P1 = c(16L, 4L, 13L, 6L, 1L, 16L, 6L, 17L, 48L, 14L),
                         P2 = c(8L, 3L, 0L, 5L, 1L, 4L, 7L, 6L, 28L, 8L),
                         P3 = c(38L, 11L, 11L, 3L, 13L, 39L, 14L, 34L, 280L, 84L),
                         P4 = c(66L, 5L, 18L, 39L, 30L, 57L, 35L, 65L, 180L, 78L),
                         P5 = c(14L, 0L, 5L, 2L, 4L, 21L, 6L, 11L, 54L, 9L),
                         P6 = c(8L, 2L, 5L, 3L, 0L, 7L, 7L, 11L, 45L, 17L),
                         P7 = c(7L, 3L, 5L, 2L, 3L, 17L, 7L, 13L, 40L, 8L)),
                         row.names = c(NA, 10L), class = "data.frame")
votes2 <- structure(list(C1 = c(2L, 1L, 2L, 2L, 0L, 4L, 0L, 4L, 19L, 14L),
                         C2 = c(7L, 3L, 1L, 7L, 2L, 5L, 3L, 10L, 21L, 6L),
                         C3 = c(78L, 7L, 28L, 42L, 28L, 84L, 49L, 85L, 260L, 100L),
                         C4 = c(56L, 14L, 20L, 7L, 19L, 54L, 22L, 50L, 330L, 91L),
                         C5 = c(14L, 3L, 6L, 2L, 3L, 14L, 8L, 8L, 45L, 7L)),
                         row.names = c(NA, 10L), class = "data.frame")
```

```
example <- ecolRxC(votes1, votes2, method = "IPF")
summary(example)
```

Index