Package: eiCR (via r-universe)

November 14, 2026

Type Package

Title Ecological Inference for RxC Tables using the Cressie-Read Power Divergence

Version 0.0.1

Description Estimates R by C vote transfer matrices (ecological contingency tables) from aggregate data by minimizing the Cressie-Read power measure. Acknowledgements: The authors wish to thank Consellería de Educación, Cultura y Universidades, Generalitat Valenciana (grant CIAICO/2023/031) for supporting this research.

License GPL (>= 2)

Encoding UTF-8

Imports stats

Suggests ggplot2, numDeriv, Rsolnp, scales

RoxygenNote 7.3.2

NeedsCompilation no

Author Jose M. Pavía [aut, cre] (ORCID: <https://orcid.org/0000-0002-0129-726X>), Antonio Forcina [aut] (ORCID: <https://orcid.org/0000-0001-5239-5495>)

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

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

Date/Publication 2026-11-13 18:30:02 UTC

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

RemoteRef HEAD

RemoteSha 6e36afa3815304c0d10e45e7664bf1f5aae019e5

Contents

ei_CR 2
plot.ei_CR 6

print.ei_CR 7
print.summary.ei_CR 8
summary.ei_CR 9

Index **11**

`ei_CR` *Ecological Inference for RxC Tables using the Cressie-Read power divergence*

Description

Estimates a matrix of transition probabilities between origin and destination categories by minimizing the Cressie–Read divergence between observed and expected aggregate counts. Optionally, standard errors and restricted confidence intervals are obtained. Moreover, unit-specific transition matrices may be also estimated by minimizing the Cressie–Read divergence from the global estimate, subject to the observed marginal constraints.

Usage

```
ei_CR(
  X,
  Y,
  census.change = "adjust1",
  P0 = NULL,
  lambda = 0,
  local = FALSE,
  se = FALSE,
  confidence = 0.95,
  max.iter = 5000,
  local.max.iter = 10000,
  method = "L-BFGS-B",
  control = list(),
  ...
)
```

Arguments

`X`	matrix (or data.frame) of order KxR with either the electoral results recorded in election 1 or the sum across columns (the margins of row options) of the K ecological tables.
`Y`	matrix (or data.frame) of order KxC with either the electoral results recorded in election 2 or the sum across rows (the margins of column options) of the K ecological tables.
`census.change`	A string character specifying how census changes between elections are handled; i.e. indicating how differences between the row totals of `X` and `Y` are considered. Available options include `"adjust1"`, `"adjust2"`, `"raw"`, `"regular"`,

	"ordinary", "enriched", "simultaneous", "semifull", "full", and "gold". See details
P0	Optional initial transition matrix. If NULL, an initial matrix is constructed automatically, respecting any structural-zero restrictions implied by census.change.
lambda	Numeric parameter controlling the Cressie–Read divergence. The value 0 corresponds to the Kullback–Leibler divergence.
local	Logical. If TRUE, estimates a separate transition matrix for each observational unit in addition to the global estimates, subject to its observed margins and the structural-zero restrictions.
se	Logical. If TRUE, computes standard errors for the global transition probabilities, vote transfers, and model parameters, together with restricted confidence intervals.
confidence	Confidence level for the restricted confidence intervals. Must be a single finite numeric value strictly between zero and one. Defaults to 0.95. Used only when se = TRUE.
max.iter	Maximum number of iterations allowed for the global optimization.
local.max.iter	Maximum number of iterations allowed for each initial unit-specific optimization.
method	Optimization method passed to the global optimizer.
control	A list of control parameters passed to stats::optim() for the global optimization (e.g., factr, pgtol or reltol, depending on method). See stats::optim() for details. If control contains maxit, it overrides max.iter. Defaults to list(), i.e., the optim defaults with maxit = max.iter.
...	Currently not used; reserved for future extensions.

Details

The function first preprocesses the input data according to census.change and then estimates a global transition matrix. When local = TRUE, unit-specific transition matrices are also estimated using the global estimate as a reference.

The census.change argument specifies how changes in the electoral census between the two elections are handled. Electors who enter the census are represented by entry (origin) categories and electors who leave it by exit (destination) categories. Depending on the option, these are supplied by the user, as the last columns of X (entries) or Y (exits), or are added by the preprocessing, as a new last column of X (net entries) or of Y (net exits), only when needed. Net entries and exits are computed following Pavía (2023). Transitions from entry to exit categories are not allowed (structural zeros).

"adjust1" The counts of X are proportionally rescaled in each unit to match the corresponding total of Y. No category is added.

"adjust2" The counts of Y are proportionally rescaled in each unit to match the corresponding total of X. No category is added.

"simultaneous" Intended for classical ecological inference problems, such as ecological studies of social or racial voting, and for simultaneous elections. The row sums of X and Y must coincide. No category is added.

`"raw"` Only the raw election results are available. Net entries and net exits are estimated from the data, and a column of net entries is added to `X` and/or a column of net exits to `Y` when they are non-null. When both are present, transfers between them are not allowed.

`"regular"` The last column of `X` contains, for each unit, the new (young) electors entitled to vote for the first time. Net exits, and possibly additional net entries, are computed and added as new columns when the totals of the two elections do not match. When net exits are present, transfers from new electors and from net entries to net exits are not allowed.

`"ordinary"` The last column of `Y` contains, for each unit, the electors who died between the two elections. Net entries, and possibly additional net exits, are computed and added as new columns when the totals of the two elections do not match. When net entries are present, transfers from net entries to deceased electors and to net exits are not allowed.

`"enriched"` Combines `"regular"` and `"ordinary"`: the last column of `X` contains the new (young) electors and the last column of `Y` the deceased electors. Other net entries and exits are computed and added as new columns when needed. Transfers from entry to exit categories are not allowed.

`"semifull"` The last column of `X` contains all new electors (young and immigrants) and the last column of `Y` all exits from the census (due to death or emigration). The row sums of `X` and `Y` must coincide. Transfers from new electors to exits are not allowed.

`"full"` As `"semifull"`, but the last two columns of `X` contain, respectively, the new young electors and the new immigrants entitled to vote. The row sums of `X` and `Y` must coincide. Transfers from both entry categories to exits are not allowed.

`"gold"` As `"full"`, but the last two columns of `Y` contain, respectively, the exits due to emigration and the exits due to death. The row sums of `X` and `Y` must coincide. Transfers from the two entry categories to the two exit categories are not allowed.

Deterministic bounds are computed for the transition probabilities using the observed margins. The `compound` bounds are obtained from the unit-level margins, whereas the `total` bounds are computed directly from the aggregated origin and destination margins. These bounds describe feasible values under the specified margins and structural constraints; they are not probability estimates or confidence intervals.

When se = TRUE, standard errors are computed for the global estimates using a sandwich (robust) estimator of the covariance matrix of the model parameters, which combines the Hessian of the objective function with the empirical variability of the unit-level gradient contributions. Standard errors for the transition probabilities are obtained through the transformation from model parameters to transition probabilities, and standard errors for vote transfers are obtained by scaling the corresponding transition-probability standard errors by the origin-category totals. Restricted confidence intervals are then constructed from Wald intervals and the deterministic bounds. The requested confidence level determines the normal critical value. Restricting the interval endpoints does not, by itself, guarantee exact frequentist coverage at the nominal confidence level.

Standard errors and confidence intervals are not currently computed for the unit-specific estimates.

Value

A list containing:

P Estimated global transition-probability matrix.

V Estimated global vote-transfer matrix.

V.agg Aggregated vote-transfer matrix from unit-specific estimates, or NULL if local = FALSE.

P.agg Transition-probability matrix obtained by normalizing V.agg by origin category, or NULL if local = FALSE.

P.local Array of unit-specific transition-probability estimates, or NULL if local = FALSE.

V.local Array of unit-specific vote-transfer estimates, or NULL if local = FALSE.

X.final, Y.final Preprocessed origin and destination data used for estimation.

bounds List containing deterministic bounds. The compound element contains bounds based on the units; the total element contains bounds based on aggregated margins.

standard_errors If se = TRUE, a list containing standard errors for P, V, and the model parameters beta; otherwise, NULL.

confidence_intervals If se = TRUE, a list containing the confidence level and lower and upper endpoints for P and V; otherwise, NULL if se = TRUE.

optimization List containing initialization values, estimated parameters, objective value, and convergence information for the global and, if applicable, local optimizations.

inputs List of the original function arguments.

Author(s)

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

Antonio Forcina, <forcinarosara@gmail.com>

References

Cressie, N. and Read, T.R.C. (1984). Multinomial goodness-of-fit tests. *Journal of the Royal Statistical Society: Series B*, 46(3), 440–464. doi:10.1111/j.25176161.1984.tb01318.x.

Pavia, J.M. (2023). Adjustment of initial estimates of voter transition probabilities to guarantee consistency and completeness. *SN Social Sciences*, 3, 75. doi:10.1007/s4354502300658y.

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 <- ei_CR(votes1, votes2)$P
```

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

Description

Plot method for objects obtained with ei_CR.

Usage

```
## S3 method for class 'ei_CR'
plot(
  x,
  margins = TRUE,
  digits = 2,
  row.names = NULL,
  col.names = NULL,
  size.numbers = 6,
  size.labels = 4,
  size.margins = 6,
  colour.cells = "steelblue3",
  colour.grid = "seashell",
  alpha = 0.5,
  which = NULL,
  ...,
  show.plot = TRUE
)
```

Arguments

`x`	An object output of the **ei_CR** function.
`margins`	A `TRUE/FALSE` argument informing whether 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, 6.
`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 value indicating whether 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 <- ei_CR(votes1, votes2)
p <- plot(example, show.plot = FALSE)
p
```

---

print.ei_CR *Print a summary of an output of the ei_CR function*

Description

Print method for objects obtained with the ei_CR function.

Usage

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

Arguments

x	An object output of the **ei_CR** 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 <- ei_CR(votes1, votes2)
print(example, digits = 1, margins = TRUE)
```

---

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

Description

Print method for summary.ei_CR objects

Usage

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

Arguments

x	An summary.ei_CR 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.ei_CR *Summarize a ei_CR output object*

Description

Summary method for objects obtained with the ei_CR function

Usage

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

Arguments

object	An object output of the **ei_CR** function.
...	Other arguments passed on to methods. Not currently used.

Value

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

prop.matrix	A matrix of order RxC with the estimated underlying probabilities/rates of the vote transitions from election 1 to election 2.
counts.matrix	A matrix of order RxC with the estimated vote transfers from election 1 to election 2.
row.margins	A vector of length R with the aggregate observed/adjusted distribution of proportions of votes in election 1.
col.margins	A vector of length C with the aggregate observed/adjusted distribution of proportions 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 <- ei_CR(votes1, votes2)
summary(example)
```

Index

ei_CR, 2

plot.ei_CR, 6
print.ei_CR, 7
print.summary.ei_CR, 8

stats::optim(), *3*
summary.ei_CR, 9