

# Package: eiIT (via r-universe)

June 1, 2026

**Title** Ecological Inference via Information Theory

**Version** 0.0.1-1

**Description** Estimates  $R \times C$  transfer matrices from aggregated marginal data using a two-stage (GME+IPF) information-theoretic approach within a two-step (global+local) estimation procedure. The resulting matrices are consistent with observed row and column marginals across collections of subtables (e.g. precincts, polling stations, or districts). References: Golan, A., Judge, G., & Miller, D. (1996). Maximum Entropy Econometrics: Robust Estimation with Limited Data. Wiley. Judge, G., Miller, D.J., & Cho, W.K.T. (2004). An information theoretic approach to ecological estimation and inference. In G. King, O. Rosen, & M. A. Tanner (Eds.), Ecological Inference: New Methodological Strategies (pp. 162–187). Cambridge University Press. Mittelhammer, R., Judge, G., & Miller, D. (2000). Econometric Foundations. Cambridge University Press. Pavia, J.M. (2023) <[doi:10.1007/s43545-023-00658-y](https://doi.org/10.1007/s43545-023-00658-y)> Acknowledgements: The author wish to thank Conselleria de Economia, Hacienda y Administracion Publica (grant CIACIO/2023/031) for supporting this research.

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** stats, utils, nloptr

**Suggests** ggplot2, scales

**RoxygenNote** 7.3.2

**NeedsCompilation** no

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

**Maintainer** Jose M. Pavía <[jose.m.pavia@uv.es](mailto:jose.m.pavia@uv.es)>

**Repository** <https://cran.r-universe.dev>

**Date/Publication** 2026-06-01 08:40:07 UTC

**RemoteUrl** <https://github.com/cran/eiIT>

**RemoteRef** HEAD

**RemoteSha** 8b33bb740c90eb96b262bb431332c1b34d8eabfb

## Contents

ei_it . . . . .	2
plot.eiit . . . . .	9
print.eiit . . . . .	11
print.summary.eiit . . . . .	12
summary.eiit . . . . .	12
<b>Index</b>	<b>14</b>

---

ei_it	<i>Ecological Inference via Generalized Maximum Entropy</i>
-------	---

---

## Description

Estimates RxC transfer matrices from aggregated marginal data using a two-stage information-theoretic procedure combining Generalized Maximum Entropy (GME) and Iterative Proportional Fitting (IPF), together with a two-step estimation strategy based on global and local estimation. The resulting matrices are consistent with observed row and column marginals across collections of subtables (e.g. precincts, polling stations, or districts).

## Usage

```
ei_it(
  votes.election1,
  votes.election2,
  m.bound = NULL,
  census.changes = c("adjust", "raw", "regular", "ordinary", "enriched", "simultaneous",
    "semifull", "full", "gold"),
  init = "uniform",
  lambda = 100,
  ipf = TRUE,
  ipf.tol = 1e-06,
  ipf.max.iter = 1000,
  confidence = NULL,
  ci.method = "kkt",
  B = 500,
  optimizer.opts = list(),
  ...
)
```

**Arguments**`votes.election1`

data.frame (or matrix) of order  $I \times R1$  with the votes gained by the  $R1$  political options competing in election 1 (or origin) in the  $I$  units considered.  $R1$  could be smaller than  $R$  when some row options (such as new voters) are internally estimated.

`votes.election2`

data.frame (or matrix) of order  $I \times C2$  with the votes gained by the  $C2$  political options competing in election 2 (or destination) in the  $I$  (territorial) units considered.  $C2$  could be smaller than  $C$  when some column options (such as death voters) are internally estimated.

`m.bound`

Positive scalar. Half-width of the symmetric support interval for the reparametrised error term:

$$\varepsilon_{ik} = m \cdot (1 - 2v_{ik}), \quad v_{ik} \in [0, 1]$$

so  $\varepsilon_{ik} \in [-m, +m]$ . Smaller values imply more confidence in the data; larger values allow more measurement error. If NULL (default), set automatically to a conservative data-driven bound:  $0.5 * \max(Y) + 0.5 * (1 - \min(Y))$ , where  $Y$  is the row-standardized data version of `votes.election2`. See **Details**.

`census.changes`

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 (2023) 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".

`init`

Starting values for the transition probabilities. One of the following.

"uniform" (**default**) All transition probabilities initialised to  $1/C$ . This is the maximum-entropy prior and the standard choice in the GME literature.

"random" Random Dirichlet-like draw. Useful to check solution stability across starts.

• A numeric user-supplied starting matrix of shape  $R1 \times C2$  (or of shape  $R \times C$  if after accounting for new entries and exits of the election censuses the final matrix is of higher shape than  $R1 \times C2$ ). If rows do not sum to 1, there are internally row-standardised.

`lambda`

Non-negative scalar. Shrinkage intensity for local matrices,  $P_i$ . Controls the weight of the KL divergence penalty toward the global estimate:

$$\text{obj}_i = -H(p_i, v_i) + \lambda \cdot KL(p_i \parallel \hat{P})$$

`lambda = 0` gives fully free local estimates; larger values pull  $P_i$  toward  $\hat{P}$ , the first-step estimate. Default is 100.

`ipf`

Logical. If TRUE (default TRUE), applies an Iterative Proportional Fitting step after GME estimation to enforce exact consistency with the observed margins. The IPF solution minimises the KL divergence from the GME estimate subject to exact marginal constraints, and is therefore fully coherent with the information-theoretic framework of GME. Convergence is controlled by `ipf.tol` and `ipf.max.iter`.

<code>ipf.tol</code>	Positive scalar. Convergence tolerance for IPF, defined as the maximum absolute change in any cell of $P$ between iterations. Default $1e-06$ .
<code>ipf.max.iter</code>	Positive integer. Maximum number of IPF iterations. Default $1000$ .
<code>confidence</code>	Numeric in $(0, 1)$ or NULL. If NULL (default), no confidence intervals are computed. Otherwise, specifies the confidence level (e.g. $0.95$ for 95% their calculation). The method is controlled by <code>ci.method</code> .
<code>ci.method</code>	Character. Method used to compute confidence intervals when confidence is not NULL. One of: <b>"kkt" (default)</b> Asymptotic intervals for the global GME solution via Karush-Kuhn-Tucker multipliers. Computationally free (no additional optimisation required) but does not account for uncertainty introduced by the IPF projection and local composition steps. Appropriate when the residual GME inconsistency is small (typically below $0.02$ ) and local solutions do not deviate substantially from the global estimate. <b>"bootstrap"</b> Propagates the uncertainty of the global GME solution through the remaining stages of the procedure. Generates $B$ perturbed global matrices by sampling from the asymptotic distribution $\hat{P} \sim \mathcal{N}(\hat{P}_{GME}, \hat{\Sigma}_{KKT})$ , projects each through the IPF and local estimation stages, and computes empirical quantiles of the resulting final solutions. This approach captures the additional uncertainty introduced by the IPF projection and local composition steps.
<code>B</code>	Positive integer. Number of bootstrap replicates. Only used when <code>ci.method = "bootstrap"</code> . Default $500$ .
<code>optimizer.opts</code>	Named list of options passed to <code>nloptr::nloptr()</code> . Defaults to <code>list(algorithm = "NLOPT_LD_SLSQP", xtol_rel = 1e-10, ftol_rel = 1e-10, maxeval = 10000)</code> .
<code>...</code>	Other arguments to be passed to the function. Not currently used.

## Details

### A two-stage, two-step information-theoretic procedure:

The method proceeds in two steps. Each step consists of a GME and an IPF stage.

The **GME stage** finds the transition matrix  $\hat{P}$  that maximises the joint entropy of transition probabilities and reparametrised errors, subject to soft data constraints (allowing for measurement error in the observed margins) and Markov adding-up constraints.

The **IPF stage** (optional): projects  $\hat{P}$  onto the set of matrices exactly consistent with the observed margins by minimising the KL divergence from the GME solution. Both stages operate within the same information-theoretic framework: GME maximises entropy under noisy constraints; IPF minimises KL divergence under exact constraints. The combined estimator inherits the low-variance properties of GME and the exact consistency of IPF.

In the **first step** (global), the method estimates a global probability matrix consistent with the observed row and column marginals across a set of subtables (e.g. precincts, polling stations, or districts). By default, this corresponds to the most uniform matrix satisfying the aggregation constraints, although the estimation can also be centered around an initial matrix supplied by the user.

In the **second step** (local), local subtables are estimated with shrinkage toward the global estimate via a Kullback–Leibler penalty, in the first stage, and using IPF in the second stage. The final global solution is obtained as the composition (aggregation) of the local solutions.

### A two-stage, two-step information-theoretic estimation:

For  $I$  subtables with  $R$  row categories and  $C$  column categories, the GME estimator maximises the joint entropy of the transition probabilities and the reparametrised errors:

$$H(p, v) = - \sum_{j,k} p_{jk} \ln p_{jk} - \sum_{i,k} v_{ik} \ln v_{ik}$$

subject to the data constraints below, the Markov adding-up constraints  $\sum_k p_{jk} = 1$ , and non-negativity.

In the GME estimator the observed row margins  $x_{ij}$  and column margins  $y_{ik}$  satisfy (up to measurement error):

$$y_{ik} = \sum_j x_{ij} p_{jk} + \varepsilon_{ik}$$

The GME estimator allows for measurement error in the observed margins via the reparametrisation  $\varepsilon_{ik} = m(1 - 2v_{ik})$ ,  $v_{ik} \in [0, 1]$ . This implies that the solution  $\hat{P}$  will not in general satisfy the data constraints exactly:

$$\sum_j x_{.j} \hat{p}_{jk} \approx y_{.k}$$

The residual inconsistency is controlled by  $m$ . bound: smaller values force  $\hat{P}$  closer to exact consistency at the cost of less flexibility to absorb measurement error.

When `ipf = TRUE`, a post-hoc IPF (Iterative Proportional Fitting) step projects  $\hat{P}$  onto the set of matrices exactly consistent with the observed margins. IPF is embedded into the information-theoretic framework: it is the algorithm that solves

$$\min_P KL(P \parallel \hat{P}) \quad \text{s.t.} \quad \mathbf{x}^\top P = \mathbf{y}^\top, P \mathbf{1} = \mathbf{1}, P \geq 0$$

where  $\hat{P}$  is the GME solution used as the prior. The combined GME+IPF estimator can therefore be interpreted as a single entropy minimisation problem in two nested projections (Csiszár 1975; Deming and Stephan 1940).

For local matrices  $P_i$ , the same IPF projection is applied individually to each subtable using its own margins  $x_i$  and  $y_i$ , with  $\hat{P}_i$  (the GME local solution) as the prior.

### Choosing $m$ . bound:

The error support  $m$  bounds how far the linear prediction can deviate from the observed marginal. Practical guidelines:

- $m$ . bound = 1.0: maximally non-informative (full unit interval).
- $m$ . bound = 0.5: conservative default, recommended by Judge et al. (2004).
- $m$ . bound = 0.1: appropriate when data are precise (e.g. census counts).

The automatic rule (NULL) uses  $0.5 * \text{diff}(\text{range}(Y))$ , which adapts to the empirical spread of the column margins.

### Initialisation and the uniform prior:

The uniform initialisation is not arbitrary: it is the maximum-entropy distribution on the simplex and represents a state of complete prior ignorance about the transition structure. The GME objective is strictly concave and the feasible set is convex, so the problem has a unique global optimum that is invariant to the starting point. The "random" option is provided purely for numerical sanity checks.

**Local matrices and lambda:**

The first-step global estimator pools all subtables, imposing a single  $P$ . As heterogeneity across subtables is expected, local matrices  $P_i$  are estimated in a second pass. Each  $P_i$  is identified only by the  $C$  data constraints from subtable  $i$  plus  $R$  Markov constraints (a total of  $R \cdot C$  free parameters), which is typically under-identified without the KL penalty. lambda regularises this by shrinking  $P_i$  toward  $\hat{P}$ .

`census.changes:`

Census changes are usual between elections, this argument controls how differences across polling units are dealt with.

`adjust:` The default value. This is the simplest solution for handling discrepancies between the total number of counts for the first and second elections. This value implies that the  $R1$  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,  $R$  is equal to  $R1$  and  $C$  equal to  $C2$ .

`raw:` This argument accounts for a scenario with two elections elapsed at least some months where only the raw election data recorded in the  $I$  units, in which the electoral space under study is divided, are available and net entries and net exits are approximated from the available information. In this scenario, net exits and net entries are estimated according to Pavia (2023). When both net entries and exits are no null, constraint (15) of Pavia (2023) applies: no transfer between entries and exits are allowed. In this scenario,  $R$  could be equal to  $R1$  or  $R1 + 1$  and  $C$  equal to  $C2$  or  $C2 + 1$ .

`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  $C1$  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 (2023). When both net entries and exits are no null, constraints (13) and (15) of Pavia (2023) apply. In this scenario,  $R$  could be equal to  $R1$  or  $R1 + 1$  and  $C$  equal to  $C2$  or  $C2 + 1$ .

`ordinary:` This value accounts for a scenario with two elections elapsed at least some months where (i) the column  $C1$  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 (2023). When both net entries and net exits are no null, constraints (14) and (15) of Pavia (2023) apply. In this scenario,  $R$  could be equal to  $R1$  or  $R1 + 1$  and  $C$  equal to  $C2$  or  $C2 + 1$ .

`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  $R1$  of `votes.election1` corresponds to new (young) electors who have the right to vote for the first time, (ii) the column  $C2$  of `votes.election2` corresponds to electors who died in the interperiod election, (iii) other (net) entries and (net) exits are computed according to Pavia (2023). When both net entries and net exits are no null, constraints (12) to (15) of Pavia (2023) apply. In this scenario,  $R$  could be equal to  $R1$  or  $R1 + 1$  and  $C$  equal to  $C2$  or  $C2 + 1$ .

`semifull:` This value accounts for a scenario with two elections elapsed at least some months, where: (i) the column  $R1 = R$  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

$C2 = C$  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 (2023) apply.

**full:** This value accounts for a scenario with two elections elapsed at least some months, where  $R = R1$ ,  $C = C2$  and (i) the column  $R - 1$  of `votes.election1` totals new (young) electors that have the right to vote for the first time, (ii) the column  $R$  of `votes.election1` measures new immigrants that have the right to vote and (iii) the column  $C$  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 (2023) apply.

**gold:** This value accounts for a scenario similar to **full**, where  $R = R1$ ,  $C = C2$  where (i) the column  $R - 1$  of `votes_election1` totals new young electors that have the right to vote for the first time, (ii) the column  $R$  of `votes_election1` measures new immigrants that have the right to vote, and total exits are separated out between (iii) exits due to emigration (column  $C - 1$  of `votes.election2`) and (iv) deaths (column  $C$  of `votes.election2`). In this scenario, the sum by rows of `votes.election1` and `votes.election2` must agree. Constraints (12) to (15) of Pavia (2023) apply.

#### Confidence intervals:

When `confidence = TRUE`, two methods are available. The "kkt" method computes asymptotic standard errors for the global GME solution using the Karush-Kuhn-Tucker conditions, and is valid as an approximation when the IPF correction is small (residual inconsistency below  $\sim 0.02$ ) and the local composition step does not substantially alter the global estimate. The "bootstrap" method resamples subtables and repeats the full two-stage, two-step procedure, capturing uncertainty from all sources..

#### Value

An object of class "ei\_it", which is a list containing:

**VTM** A matrix of shape  $R \times C$  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).

**VTM.votes** A matrix of shape  $R \times C$  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).

**VTM.first** A matrix of shape  $R \times C$  with the first-step 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).

**VTM.votes.first** A matrix of shape  $R \times C$  with the first-step 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).

**VTM.gme** A matrix of shape  $R \times C$  with the first-stage (GME) estimated proportions of the probabilities 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).

- VTM.units An array of shape  $R \times C \times I$  with the estimated proportions of the row-standardized vote transitions from election 1 to election 2 attained for each unit. 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 shape  $R \times C \times I$  with the estimated transfer of votes from election 1 to election 2 attained for each unit. 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 A matrix of shape  $R \times C$  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 shape  $R \times C$  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.
- deterministic.bounds A list of two matrices of shape  $R \times C$  and two arrays of shape  $R \times C \times I$  containing for each vote transition the lower and upper allowed proportions given the observed aggregates.
- entropy Scalar. Value of the entropy objective at the first stage, first-step solution.
- convergence Character. Convergence message from the optimizer.
- m.bound The value of m.bound used (useful when auto-selected).
- inputs List containing all the objects with the values used as arguments by the function.

### Author(s)

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

### References

- Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3(1), 146–158.
- Deming, W. E., & Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics*, 11(4), 427–444.
- Golan, A., Judge, G., & Miller, D. (1996). *Maximum Entropy Econometrics: Robust Estimation with Limited Data*. Wiley.
- Judge, G., Miller, D.J., & Cho, W.K.T. (2004). An information theoretic approach to ecological estimation and inference. In G. King, O. Rosen, & M. A. Tanner (Eds.), *Ecological Inference: New Methodological Strategies* (pp. 162–187). Cambridge University Press.
- Mittelhammer, R., Judge, G., & Miller, D. (2000). *Econometric Foundations*. Cambridge University Press.
- 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_it(votes1, votes2)$VTM

```

plot.eiit

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

**Description**

Plot method for objects obtained with ei\_it.

**Usage**

```

## S3 method for class 'eiit'
plot(
  x,
  margins = TRUE,
  digits = 2,
  row.names = NULL,
  col.names = NULL,
  size.numbers = 6,
  size.labels = 4,
  size.margins = 4,
  colour.cells = "steelblue",
  colour.grid = "bisque2",
  alpha = 0.5,
  which = NULL,
  ...,
  show.plot = TRUE
)

```

**Arguments**

<code>x</code>	An object output of the <code>ei_it</code> function.
<code>margins</code>	A TRUE/FALSE argument informing if the margins of the matrix should be displayed. Default TRUE.
<code>digits</code>	Integer indicating the number of decimal places to be shown. Default, 2.
<code>row.names</code>	Names to be used for the rows of the matrix.
<code>col.names</code>	Names to be used for the columns of the matrix.
<code>size.numbers</code>	A reference number indicating the average font size to be used for the transfer numbers. Default, 6.
<code>size.labels</code>	A number indicating the font size to be used for labels. Default, 4.
<code>size.margins</code>	A number indicating the font size to be used for margin numbers. Default, 4.
<code>colour.cells</code>	Background base colour for cells.
<code>colour.grid</code>	Colour to be used for grid lines.
<code>alpha</code>	A [0,1] number of colour transparency.
<code>which</code>	A vector of integers informing the units for which the aggregate transfer matrix should be plotted. Default, NULL, the global matrix is shown.
<code>...</code>	Other arguments passed on to methods. Not currently used.
<code>show.plot</code>	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 <- ei_it(votes1, votes2)
p <- plot(example, show.plot = FALSE)
p

```

---

print.eiit

*Print a summary of an output of the ei\_it function*


---

### Description

Print method for objects obtained with the ei\_it function.

### Usage

```

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

```

### Arguments

x	An object output of the <b>ei_it</b> 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_it(votes1, votes2)
print(example, digits = 1, margins = TRUE)

```

---

```
print.summary.eiit
```

*Print a summary of a summary.eiit object*

---

### Description

Print method for summary.eiit objects

### Usage

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

### Arguments

x	An summary.eiit 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.eiit
```

*Summarize an eiit output object*

---

### Description

Summary method for objects obtained with the ei\_it function

### Usage

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

**Arguments**

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

**Value**

An object of class "summary.eiit". 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 <- ei_it(votes1, votes2)
summary(example)
```

# Index

`ei_it`, [2](#)

`nloptr::nloptr()`, [4](#)

`plot.eiit`, [9](#)

`print.eiit`, [11](#)

`print.summary.eiit`, [12](#)

`summary.eiit`, [12](#)