Package 'lphom'

February 20, 2024

Type Package

Title Ecological Inference by Linear Programming under Homogeneity

Version 0.3.5-4

Description Provides a bunch of algorithms based on linear programming for estimating, under the homogeneity hypothesis, RxC ecological contingency tables (or vote transition matrices) using mainly aggregate data (from voting units).
References:
Pavía and Romero (2022) <doi:10.1177/00491241221092725>.
Pavía (2023) <doi:10.1007/s43545-023-00658-y>.
Pavía and Romero (2024) <doi:10.1093/jrsssa/qnae013>.
Pavía (2024) A local convergent ecological inference algorithm for RxC tables.
Pavía and Penadés (2024). A bottom-up approach for ecological inference.
Romero, Pavía, Martín and Romero (2020) <doi:10.1080/02664763.2020.1804842>.
Acknowledgements:
The authors wish to thank Consellería de Educación, Universidades y Empleo, Generalitat Valenciana (grant AICO/2021/257) and Ministerio de Economía e Innovación (grant PID2021-128228NB-I00) for supporting this research.

License EPL | file LICENSE

Encoding UTF-8

Imports stats, lpSolve

Depends R (>= 3.5.0)

Suggests ggplot2, scales, Rsymphony (>= 0.1-30)

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2024-02-20 07:50:11 UTC

R topics documented:

`adjust2integers` *Integer-adjusting of outputs of the lphom-family functions*

Description

Takes as input an object generated with an algorithm of the lphom-family (lphom, tslphom, nslphom, tslphom_dual, nslphom_joint, ...) and returns as output an object of the same class as the input object with all their relevant estimated (local and global) transfer matrices of counts updated to their closest integer matrices. The rest of main components of the object are also accordingly updated.

Usage

```
adjust2integers(x, solver = "symphony", ...)
```

Arguments

`x`	An object output of a lphom family algorithm
`solver`	A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only `symphony` and `lp_solve` are allowed. By default, `symphony`. The package `Rsymphony` needs to be installed for the option `symphony` to be used.
`...`	Other arguments passed on the method. Not currently used.

Details

The updating of the matrices is performed using integer linear programming after imposing all the row- and column-constraints.

Value

An object of the same class and components as `x` with its components properly updated after adjusting the estimated count matrices in `x` using integer linear programming

Author(s)

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

References

...

Examples

```
mt.ts <- tslphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
mt.ts <- adjust2integers(mt.ts, solver = "lp_solve")
```

---

confidence_intervals_pjk

Confidence Intervals for lphom estimates

Description

Estimates confidence intervals for the (vote) transfer probabilities obtained with **lphom()**

Usage

```
confidence_intervals_pjk(lphom.object, level = 0.9, num.d = 11, B = 30)
```

Arguments

`lphom.object`	An object output of the **lphom()** function.
`level`	A number between 0 and 1 to be used as level of confidence for the intervals. By default 0.90
`num.d`	Number maximum of different disturbances, `d`, to be initially considered. Positive integer greater than or equal to 5. By default, 11.
`B`	Integer that determines the number of simulations to be performed for each disturbance value. By default, 30.

Value

A list with the following components

`TM.estimation`	Transfer matrix of probability point estimates.
`TM.lower`	Transfer matrix of lower values for the probability estimates.
`TM.upper`	Transfer matrix of upper values for the probability estimates.
`level`	Confidence level used when computing the confidence intervals.

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Romero, R, Pavia, JM, Martin, J and Romero G (2020). Assessing uncertainty of voter transitions estimated from aggregated data. Application to the 2017 French presidential election. *Journal of Applied Statistics*, 47(13-15), 2711-2736. doi:10.1080/02664763.2020.1804842

Martin, J (2020). Analisis de la incertidumbre en la estimacion de la movilidad electoral mediante el procedimiento plhom. PhD Dissertation.

Examples

```
# Do not run
# mt.lphom <- lphom(France2017P[, 1:8], France2017P[, 9:12], "raw", NULL, FALSE)
# set.seed(533423)
# confidence_intervals_pjk(mt.lphom, level = 0.90, num.d = 5, B = 8)
```

---

error_lphom *Global error of a lphom estimated table*

Description

Estimation of the error index (EI) of a RxC vote transfer matrix obtained with **lphom()**

Usage

```
error_lphom(
  lphom.object,
  upper.alfa = 0.1,
  show.plot = TRUE,
  num.d = 11,
  B = 30
)
```

Arguments

lphom.object	An object output of the **lphom()** function.
upper.alfa	Upper bound that will not be exceed by the EI estimate with a confidence 1 - alpha. By default, 0.10.
show.plot	TRUE/FALSE. Indicates whether the plot showing the relationship between EI and HETe estimated by simulation for the election under study should be displayed as a side-effect. By default, TRUE.
num.d	Number maximum of different disturbances, d, to be initially considered. Positive integer greater than or equal to 5. By default, 11.
B	Integer that determines the number of simulations to be performed for each disturbance value. By default, 30.

Value

A list with the following components

EI.estimate	Point estimate for EI.
EI.upper	Upper bound with confidence 1 - alpha of the EI estimate
figure	ggplot2 object describing the graphical representation of the relationship between EI and HETe.
equation	lm object of the adjustment between EI and HETe.
statistics	A four column matrix with the values of HET, HETe, EI and d associated with each simulated scenario.
TMs.real	Array with the simulated real transfer matrices associated with each scenario.
TMs.estimate	Array with the estimated transfer matrices associated with each scenario.

Note

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

See equation (12) in Romero et al. (2020) for a definition of the EI index.

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Romero, R, Pavia, JM, Martin, J and Romero G (2020). Assessing uncertainty of voter transitions estimated from aggregated data. Application to the 2017 French presidential election. *Journal of Applied Statistics*, 47(13-15), 2711-2736. doi:10.1080/02664763.2020.1804842

See Also

lphom confidence_intervals_pjk

Examples

```
mt.lphom <- lphom(France2017P[, 1:8], France2017P[, 9:12],
                  new_and_exit_voters = "raw", verbose = FALSE)
set.seed(253443)
example <- error_lphom(mt.lphom, upper.alfa = 0.10, show.plot = FALSE, num.d = 5, B = 8)
example$EI.estimate
```

`France2017D` *2017 French Presidential Election. Department official results.*

Description

Data frame containing the official results recorded in the first and second rounds of the 2017 French presidential election in the 107 territorial French departments and in an artificial department that groups the French electors living abroad.

Usage

```
data(France2017D)
```

Format

A table containing 108 observations and 13 variables:

ABSTENTION Number of people abstaining (NonVoters) in the first-round of 2017 Presidential Election.

BLANK_NULL Number of people voting either blank or null in the first-round of 2017 Presidential Election.

MACRON Number of votes gained at a national level by Emmanuel Macron in the first-round of 2017 Presidential Election.

LE_PEN Number of votes gained at a national level by Marine Le Pen in the first-round of 2017 Presidential Election.

FILLON Number of votes gained at a national level by Francois Fillon in the first-round of 2017 Presidential Election.

MELENCHON Number of votes gained at a national level by Jean-Luc Melenchon in the first-round of 2017 Presidential Election.

HAMON Number of votes gained at a national level by Benoit Hamon in the first-round of 2017 Presidential Election.

DUPONT.AIGNAN Number of votes gained at a national level by Nicolas Dupont-Aignan in the first-round of 2017 Presidential Election.

OTHERS Number of votes gained at a national level by the rest of candidates in the first-round of 2017 Presidential Election.

ABSTENTION2 Number of people abstaining (NonVoters) in the second-round of 2017 Presidential Election.

BLANK_NULL2 Number of people voting either blank or null in the second-round of 2017 Presidential Election.

MACRON2 Number of votes gained at a national level by Emmanuel Macron in the second-round of 2017 Presidential Election

LE_PEN2 Number of votes gained at a national level by Marine Le Pen in the second-round of 2017 Presidential Election

Source

Own elaboration from data available in https://www.conseil-constitutionnel.fr/, retrieved 3 March 2020.

`France2017P` *2017 French Presidential Election. Regional provisional results.*

Description

Data frame containing the provisional results of the first and second rounds of the 2017 French presidential election in the 12 French continental regions (Auvergne-Rhone-Alpes, Bourgogne-Franche-Comte, Brittany, Centre-Val de Loire, Grand Est, Hauts-de-France, Ile-de-France, Normandy, Nouvelle-Aquitaine, Occitanie, Pays de la Loire, Provence-Alpes-Cote d'Azur) plus an additional region that covers Corsica and the rest of French overseas regions.

Usage

```
data(France2017P)
```

Format

A table containing 13 observations and 12 variables:

ABSTENTION Number of people abstaining (NonVoters) and voting either blank or null in the first-round of 2017 Presidential Election.

MACRON Number of votes gained at a national level by Emmanuel Macron in the first-round of 2017 Presidential Election.

LE_PEN Number of votes gained at a national level by Marine Le Pen in the first-round of 2017 Presidential Election.

FILLON Number of votes gained at a national level by Francois Fillon in the first-round of 2017 Presidential Election.

MELENCHON Number of votes gained at a national level bu Jean-Luc Melenchon in the first-round of 2017 Presidential Election.

HAMON Number of votes gained at a national level by Benoit Hamon in the first-round of 2017 Presidential Election.

DUPONT Number of votes gained at a national level by Nicolas Dupont-Aignan in the first-round of 2017 Presidential Election.

OTHERS Number of votes gained at a national level by the rest of candidates in the first-round of 2017 Presidential Election.

ABSTENTION2 Number of people abstaining (NonVoters) in the second-round of 2017 Presidential Election.

BLANK_NULL Number of people voting either blank or null in the second-round of 2017 Presidential Election.

MACRON2 Number of votes gained at a national level by Emmanuel Macron in the second-round of 2017 Presidential Election

LE_PEN2 Number of votes gained at a national level by Marine Le Pen in the second-round of 2017 Presidential Election

Source

Own elaboration from data available in https://www.francetvinfo.fr/elections/resultats/, retrieved 7 May 2017.

`lclphom` *Implements lclphom algorithm*

Description

Estimates RxC (JxK) vote transfer matrices (ecological contingency tables) with lclphom

Usage

```
lclphom(
  votes_election1,
  votes_election2,
 new_and_exit_voters = c("raw", "regular", "ordinary", "enriched", "adjust1", "adjust2",
    "simultaneous", "semifull", "full", "fullreverse", "gold"),
  apriori = NULL,
  lambda = 0.5,
  uniform = TRUE,
  structural_zeros = NULL,
  integers = FALSE,
  iter.max = 1000,
  type.errors = "posterior",
  distance.local = c("abs", "max", "none"),
  verbose = TRUE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

`votes_election1`
: data.frame (or matrix) of order IxJ1 with the votes gained by (or the counts corresponding to) the J1 political options competing (available) on election 1 (or origin) in the I units considered. In general, the row marginals of the I tables corresponding to the units.

`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. In general, the column marginals of the I tables corresponding to the units.

`new_and_exit_voters`
: A character string indicating the level of information available in `votes_election1` and `votes_election2` regarding new entries and exits of the election censuses between the two elections. This argument allows, in addition to the options discussed in Pavia (2023), three more options. This argument admits eleven different values: `raw`, `regular`, `ordinary`, `enriched`, `adjust1`, `adjust2`, `simultaneous`, `semifull`, `full`, `fullreverse` and `gold`. Default, `raw`.

`apriori`
: data.frame (or matrix) of order J0xK0 with an initial estimate of the (row-standarized) global voter transition proportions/fractions, pjk0, between the first J0 (election) options of election 1 and the first K0 (election) options of election 2. This matrix can contain some missing values. When no a priori information is available `apriori` is a null object. Default, `NULL`.

`lambda`
: A number between 0 and 1, informing the relative weight the user assigns to the `apriori` information. Setting `lambda = 0` is equivalent to not having a priori information (i.e., `apriori = NULL`). Default, `0.5`.

`uniform`
: A `TRUE/FALSE` value that informs whether census exits impact all the electoral options in a (relatively) similar fashion in all iterations, including iteration 0 and when deriving units tables. If `uniform = TRUE` typically at least one of the equations among equations (6) to (11) of Pavia (2023) is included in the underlying model. This parameter has no effect in `simultaneous` scenarios. It also has not impact in `raw` and `regular` scenarios when no net exits are estimated by the function from the provided information. Default, `TRUE`.

`structural_zeros`
: Default `NULL`. A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2. For instance, when new_and_exit_voters is set to `"semifull"`, lphom implicitly states `structural_zeros = list(c(J1, K2))`.

`integers`
: A `TRUE/FALSE` value that indicates whether the problem is solved in integer values in both iterations, including iteration zero (lphom) and the rest of iterations, when deriving unit tables solutions. If `integers = TRUE`, the LP matrices are approximated to the closest integer solution solving the corresponding Integer Linear Program. Default, `FALSE`.

`iter.max`
: Maximum number of iterations to be performed. The process ends when either the number of iterations reaches `iter.max` or when there is no error reduction in any local unit between two consecutive iterations. By default, `1000`.

`type.errors` A string argument that indicates whether the errors (distance to homogeneity) to be computed for the temporary local solutions are calculated taking as reference the previous global matrix (the one that is used to derive the temporary local solution) or taking as reference the posterior global matrix (the one in which the temporary local solution is integrated). This argument admits two values: `previous` and `posterior`. Default, `posterior`.

`distance.local` A string argument that indicates whether the second step of the lphom_local algorithm should be performed to solve potential indeterminacies of local solutions. Default, `"abs"`. If `distance.local = "abs"` lphom_local selects in its second step the matrix closer to the temporary global solution under L_1 norm, among the first step compatible matrices. If `distance.local = "max"` lphom_local selects in its second step the matrix closer to the temporary global solution under L_Inf norm, among the first step compatible matrices. If `distance.local = "none"`, the second step of lphom_local is not performed.

`verbose` A `TRUE/FALSE` value that indicates if a summary of the results of the computations performed to estimate net entries and exits should be printed on the screen. Default, `TRUE`.

`solver` A character string indicating the linear programming solver to be used, only `lp_solve` and `symphony` are allowed. By default, `lp_solve`. The package `Rsymphony` needs to be installed for the option `symphony` to be used.

`integers.solver` A character string indicating the linear programming solver to be used for approximating the LP solution to the closest integer solution. Only `symphony` and `lp_solve` are allowed. By default, `symphony`. The package `Rsymphony` needs to be installed for the option `symphony` to be used. Only used when `integers = TRUE`.

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

Details

Description of the `new_and_exit_voters` argument in more detail.

- `raw`: The default value. This argument accounts for the most plausible scenario when estimating vote transfer matrices. 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. In this scenario, net exits and net entries are estimated according to equation (7) of Romero et al. (2020). When both net entries and exits are no null, constraint (15) of Pavia (2023) applies. If there are net exits and `uniform = TRUE` either constraints (6) or (8) and (15) of Pavia (2023) are imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `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 equation (7) of Romero et al. (2020), and (iii) we can (or not) assume that net exits impact equally all the first J1 - 1 options of election 1. When both net entries and exits are no null, constraints (13) and (15) of Pavia (2023) apply. If `uniform = TRUE` and there are net exits either constraints (8) or (11) of Pavia (2023), depending on whether there are or

not net entries, are also imposed. 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 period between elections, (ii) net entries and maybe other additional net exits are computed according to equation (7) of Romero et al. (2020), and (iii) we can assume (or not) that exits impact equally all the J1 options of election 1. When both net entries and exits are no null, constraints (14) and (15) of Pavia (2023) apply and if `uniform = TRUE` either constraints (8) and (9) or, without net entries, (6) and (7) of Pavia (2023) are also imposed. 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 somehow 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 equation (7) of Romero et al. (2020), and (iv) we can assume (or not) that exits impact equally all the J1 - 1 options of election 1. When both net entries and exits are no null, constraints (12) to (15) of Pavia (2023) apply and if `uniform = TRUE` constraints (10) and (11) of Pavia (2023) are also imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `adjust1`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the first election (the row-sums of `votes_election1`) are proportionally adjusted to match the corresponding census of the polling units in the second election (the row-sums of `votes_election2`). If `integers = TRUE`, each row in `votes_election1` is proportionally adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election2`.
- `adjust2`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the second election (the row-sums of `votes_election2`) are proportionally adjusted to match the corresponding census of the polling units in the first election (the row-sums of `votes_election1`). If `integers = TRUE`, each row in `votes_election2` is adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election1`.
- `simultaneous`: This is the value to be used in classical ecological inference problems, such as in ecological studies of racial voting, and in scenarios with two simultaneous elections. In this scenario, the sum by rows of `votes_election1` and `votes_election2` must coincide. Constraints defined by equations (8) and (9) of Romero et al. (2020) are not included in the model. In this case, the lphom function just implements the basic model defined, for instance, by equations (1) to (5) of Pavia (2024).
- `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 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 (2023) apply. Additionally, if `uniform = TRUE` constraints (8) of Pavia (2023) are also imposed.
- `full`: This value accounts for a scenario with two elections elapsed at least some months, where (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 (=J1) of votes_election1 measures new immigrants that have the right to vote and (iii) the column K (=K2) 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. Additionally, if uniform = TRUE constraints (11) of Pavia (2023) are also imposed.

- fullreverse: This value is somehow the mirror version of full. It 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 and (ii) where 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 and constraints (14) and (15) of Pavia (2023) apply. Additionally, if uniform = TRUE constraints (8) and (9) of Pavia (2023) are also imposed.
- gold: This value accounts for a scenario similar to full, where 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 (2023) apply and if uniform = TRUE constraints (10) and (11) of Pavia (2023) are also imposed.

Value

A list with the following components

VTM	A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated percentages of row-standardized vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census in all units, net entries are omitted (i.e., the number of rows of VTM is equal to J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census in all units, net exits are omitted (i.e., the number of rows of VTM is equal to K2) even when estimates for net exits different from zero are obtained.
VTM.votes	A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census, net entries are omitted (i.e., J = J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census, net exits are omitted (i.e., K = K2) even when estimates for net exits different from zero are obtained.
OTM	A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2.
HETe	The estimated heterogeneity index as defined in equation (15) of Pavia and Romero (2022).
VTM.complete	A matrix of order JxK with the estimated proportions of row-standardized vote transitions from election 1 to election 2, including in raw, regular, ordinary

and enriched scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.complete.votes
: A matrix of order JxK with the estimated vote transitions from election 1 to election 2, including in raw, regular, ordinary and enriched scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.prop.units
: An array of order JxKxI with the estimated proportions of vote transitions from election 1 to election 2 attained for each unit in the solution.

VTM.votes.units
: An array of order JxKxI with the estimated matrix of vote transitions from election 1 to election 2 attained for for each unit in the solution.

VTM.complete.last.iter
: A matrix of order JxK with the estimated proportions of vote transitions from election 1 to election 2, including in raw, regular, ordinary and enriched scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units, corresponding to the final iteration.

VTM.sequence
: Array of order JxKx(iter+1) (where iter is the efective number of iterations performed) of the intermediate estimated matrices corresponding to each iteration.

HETe.sequence
: Numeric vector of length iter+1 with the HETe coefficients corresponding to the matrices in VTM.sequence.

VTM.prop.units.last.iter
: An array of order JxKxI with the estimated proportions of vote transitions from election 1 to election 2 attained for each unit in the final iteration.

VTM.votes.units.last.iter
: An array of order JxKxI with the estimated matrix of vote transitions from election 1 to election 2 attained for each unit in the final iteration.

zeros
: A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2.

iter
: The real final number of iterations performed before ending the process.

iter.units
: A matrix of order Ix(iter+1) with the number of iteration corresponding to the solution selected for each unit in each iteration.

errors
: A vector of length I with the minimal error observed in the sequence for each unit. It corresponds to the unit-error associated with the solution linked with either VTM.prop.units or VTM.votes.units.

deterministic.bounds
: A list of two matrices of order JxK and two arrays of order JxKxI containing for each vote transition the lower and upper allowed proportions given the observed aggregates.

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

origin
: A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`destination` A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`EHet` A matrix of order IxK measuring in each spatial unit a distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results with the solution in each territorial unit for each option of election 2.

`solution_init` A list with the main outputs produced by **lphom()**.

- `VTM_init`: A matrix of order J'xK' with the estimated percentages of vote transitions from election 1 to election 2 initially obtained by **lphom()**.
- `VTM.votes_init`: A matrix of order J'xK' with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()**.
- `OTM_init`: A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2 initially obtained by **lphom()**.
- `HETe_init`: The estimated heterogeneity index defined in equation (10) of Romero et al. (2020).
- `EHet_init`: A matrix of order IxK measuring in each spatial unit the distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results, using the **lphom()** solution, in each territorial unit for each option of election 2.
- `VTM.complete_init`: A matrix of order JxK with the estimated proportions of vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in `raw`, `regular`, `ordinary` and `enriched` scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.
- `VTM.complete.votes_init`: A matrix of order JxK with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in `raw`, `regular`, `ordinary` and `enriched` scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

Author(s)

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

References

Pavia, JM, and Romero, R (2022). Improving estimates accuracy of voter transitions. Two new algorithms for ecological inference based on linear programming, *Sociological Methods & Research*. doi:10.1177/00491241221092725.

Pavia, JM. (2024). A local convergent ecological inference algorithm for RxC tables.

See Also

`lphom` `tslphom` `nslphom` `rslphom`

Other linear programing ecological inference functions: `lp_apriori()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `rslphom()`, `tslphom_dual()`, `tslphom_joint()`, `tslphom()`

Examples

```
mt.lc <- lclphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
mt.lc$VTM
mt.lc$HETe
mt.lc$solution_init$HETe_init
```

---

lphom *Implements lphom algorithm*

Description

Estimates RxC (JxK) vote transfer matrices (ecological contingency tables) with lphom

Usage

```
lphom(
  votes_election1,
  votes_election2,
 new_and_exit_voters = c("raw", "regular", "ordinary", "enriched", "adjust1", "adjust2",
    "simultaneous", "semifull", "full", "fullreverse", "gold"),
  apriori = NULL,
  lambda = 0.5,
  uniform = TRUE,
  structural_zeros = NULL,
  integers = FALSE,
  verbose = TRUE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

`votes_election1`

data.frame (or matrix) of order IxJ1 with the votes gained by (or the counts corresponding to) the J1 political options competing (available) on election 1 (or origin) in the I units considered. In general, the row marginals of the I tables corresponding to the units.

`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. In general, the column marginals of the I tables corresponding to the units.

`new_and_exit_voters`
: A character string indicating the level of information available in `votes_election1` and `votes_election2` regarding new entries and exits of the election censuses between the two elections. This argument allows, in addition to the options discussed in Pavia (2023), three more options. This argument admits eleven different values: `raw`, `regular`, `ordinary`, `enriched`, `adjust1`, `adjust2`, `simultaneous`, `semifull`, `full`, `fullreverse` and `gold`. Default, `raw`.

`apriori`
: data.frame (or matrix) of order J0xK0 with an initial estimate of the (row-standarized) global voter transition proportions/fractions, pjk0, between the first J0 (election) options of election 1 and the first K0 (election) options of election 2. This matrix can contain some missing values. When no a priori information is available `apriori` is a null object. Default, `NULL`.

`lambda`
: A number between 0 and 1, informing the relative weight the user assigns to the `apriori` information. Setting `lambda = 0` is equivalent to not having a priori information (i.e., `apriori = NULL`). Default, `0.5`.

`uniform`
: A `TRUE/FALSE` value that informs whether census exits affect all the electoral options in a (relatively) similar fashion. If `uniform = TRUE` typically at least one of the equations among equations (6) to (11) of Pavia (2022) is included in the underlying model. This parameter has never effect in `simultaneous` scenarios. It also has not impact in `raw` and `regular` scenarios when no net exits are estimated by the function from the provided information. Default, `TRUE`.

`structural_zeros`
: Default `NULL`. A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2. For instance, when new_and_exit_voters is set to `"semifull"`, lphom implicitly states `structural_zeros = list(c(J1, K2))`.

`integers`
: A `TRUE/FALSE` value that indicates whether the LP solution of counts (votes) must be approximate to the closest integer solution using ILP to generate the final solution. Default, `FALSE`.

`verbose`
: A `TRUE/FALSE` value that indicates if a summary of the results of the computations performed to estimate net entries and exits should be printed on the screen. Default, `TRUE`.

`solver`
: A character string indicating the linear programming solver to be used, only `lp_solve` and `symphony` are allowed. By default, `lp_solve`. The package `Rsymphony` needs to be installed for the option `symphony` to be used.

`integers.solver`
: A character string indicating the linear programming solver to be used for approximating the LP solution to the closest integer solution. Only `symphony` and `lp_solve` are allowed. By default, `symphony`. The package `Rsymphony` needs to be installed for the option `symphony` to be used. Only used when `integers = TRUE`.

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

Details

Description of the `new_and_exit_voters` argument in more detail.

- raw: The default value. This argument accounts for the most plausible scenario when estimating vote transfer matrices. 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. In this scenario, net exits and net entries are estimated according to equation (7) of Romero et al. (2020). When both net entries and exits are no null, constraint (15) of Pavia (2023) applies. If there are net exits and uniform = TRUE either constraints (6) or (8) and (15) of Pavia (2023) are imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- 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 equation (7) of Romero et al. (2020), and (iii) we can (or not) assume that net exits impact equally all the first J1 - 1 options of election 1. When both net entries and exits are no null, constraints (13) and (15) of Pavia (2023) apply. If uniform = TRUE and there are net exits either constraints (8) or (11) of Pavia (2023), depending on whether there are or not net entries, are also imposed. 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 period between elections, (ii) net entries and maybe other additional net exits are computed according to equation (7) of Romero et al. (2020), and (iii) we can assume (or not) that exits impact equally all the J1 options of election 1. When both net entries and exits are no null, constraints (14) and (15) of Pavia (2023) apply and if uniform = TRUE either constraints (8) and (9) or, without net entries, (6) and (7) of Pavia (2023) are also imposed. 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 somehow 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 equation (7) of Romero et al. (2020), and (iv) we can assume (or not) that exits impact equally all the J1 - 1 options of election 1. When both net entries and exits are no null, constraints (12) to (15) of Pavia (2023) apply and if uniform = TRUE constraints (10) and (11) of Pavia (2023) are also imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- adjust1: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the first election (the row-sums of votes_election1) are proportionally adjusted to match the corresponding census of the polling units in the second election (the row-sums of votes_election2). If integers = TRUE, each row in votes_election1 is proportionally adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in votes_election2.
- adjust2: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the second election (the row-sums of votes_election2) are proportionally adjusted to match the corresponding census of the polling units in the first election (the row-sums of votes_election1). If integers = TRUE, each row in votes_election2 is adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in votes_election1.

- simultaneous: This is the value to be used in classical ecological inference problems, such as in ecological studies of racial voting, and in scenarios with two simultaneous elections. In this scenario, the sum by rows of votes_election1 and votes_election2 must coincide. Constraints defined by equations (8) and (9) of Romero et al. (2020) are not included in the model. In this case, the lphom function just implements the basic model defined, for instance, by equations (1) to (5) of Pavia (2024).
- 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 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 (2023) apply. Additionally, if uniform = TRUE constraints (8) of Pavia (2023) are also imposed.
- full: This value accounts for a scenario with two elections elapsed at least some months, where (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 (=J1) of votes_election1 measures new immigrants that have the right to vote and (iii) the column K (=K2) 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. Additionally, if uniform = TRUE constraints (11) of Pavia (2023) are also imposed.
- fullreverse: This value is somehow the mirror version of full. It 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 and (ii) where 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 and constraints (14) and (15) of Pavia (2023) apply. Additionally, if uniform = TRUE constraints (8) and (9) of Pavia (2023) are also imposed.
- gold: This value accounts for a scenario similar to full, where 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 (2023) apply and if uniform = TRUE constraints (10) and (11) of Pavia (2023) are also imposed.

Value

A list with the following components

VTM — A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated percentages of row-standardized vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census in all units, net entries are omitted (i.e., the number of rows of VTM is equal to J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census in all units, net exits are omitted (i.e., the number of rows of VTM is equal to K2) even when estimates for net exits different from zero are obtained.

`VTM.votes` A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated vote transitions from election 1 to election 2. In `raw`, `regular`, `ordinary` and `enriched` scenarios when the percentage of net entries is small, less than 1% of the census, net entries are omitted (i.e., J = J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census, net exits are omitted (i.e., K = K2) even when estimates for net exits different from zero are obtained.

`OTM` A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2.

`HETe` The estimated heterogeneity index defined in equation (11) of Romero et al. (2020).

`VTM.complete` A matrix of order JxK with the estimated proportions of 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) even when they are really small, less than 1%.

`VTM.complete.votes` A matrix of order JxK with the estimated 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) even when they are really small, less than 1%.

`deterministic.bounds` A list of two matrices of order JxK containing for each vote transition the lower and upper proportions allowed given the observed aggregates.

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

`origin` A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`destination` A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`EHet` A matrix of order IxK measuring in each spatial unit a distance to the homogeneity hypothesis. That is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results in each territorial unit for each option of election 2.

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Romero, R, Pavia, JM, Martin, J and Romero G (2020). Assessing uncertainty of voter transitions estimated from aggregated data. Application to the 2017 French presidential election. *Journal of Applied Statistics*, 47(13-15), 2711-2736. doi:10.1080/02664763.2020.1804842

See Also

tslphom nslphom lclphom rslphom

Other linear programing ecological inference functions: lclphom(), lp_apriori(), lphom_dual(), lphom_joint(), nslphom_dual(), nslphom_joint(), nslphom(), rslphom(), tslphom_dual(), tslphom_joint(), tslphom()

Examples

```
lphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
```

---

lphom_dual *Implements lphom_dual algorithm*

Description

Estimates RxC vote transfer matrices (ecological contingency tables) with lphom_dual

Usage

```
lphom_dual(
  votes_election1,
  votes_election2,
  integers = FALSE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

votes_election1	data.frame (or matrix) of order IxJ with the counts to be initially mapped to rows. When estimating vote transfer matrices, the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.
votes_election2	data.frame (or matrix) of order IxK with the counts to be initially mapped to columns. When estimating vote transfer matrices, the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.
integers	A TRUE/FALSE value that indicates whether the LP solution of counts (votes) must be approximate to the closest integer solution using ILP. Default, FALSE.
solver	A character string indicating the linear programming solver to be used, only lp_solve and symphony are allowed. By default, lp_solve. The package Rsymphony needs to be installed for the option symphony to be used.

integers.solver
: A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only symphony and lp_solve are allowed. By default, symphony. The package Rsymphony needs to be installed for the option symphony to be used. Only used when integers = TRUE.

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

Value

A list with the following components

VTM.votes.w
: The matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2, attained weighting the two dual solutions using as weights the corresponding HTEe estimates.

VTM.votes.a
: The matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2, attained simple averaging the two dual solutions.

HETe.w
: Estimated heterogeneity index associated to the VTM.votes.w solution.

HETe.a
: Estimated heterogeneity index associated to the VTM.votes.a solution.

VTM12.w
: The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the VTM.votes.w solution.

VTM21.w
: The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the VTM.votes.w solution.

VTM12.a
: The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the VTM.votes.a solution.

VTM21.a
: The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the VTM.votes.a solution.

lphom.object.12
: The output of the lphom function attained solving the problem X –> Y. That is, mapping votes_election1 to rows and votes_election2 to columns.

lphom.object.21
: The output of the lphom function attained solving the problem Y –> X. That is, mapping votes_election2 to rows and votes_election1 to columns.

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

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM and Romero, R (2024). Symmetry estimating RxC vote transfer matrices from aggregate data. *Journal of the Royal Statistical Society, Series A – Statistics in Society*, forthcoming. doi:10.1093/jrsssa/qnae013

See Also

lphom tslphom_dual nslphom_dual lphom_joint tslphom_joint nslphom_joint

Other linear programing ecological inference functions: lclphom(), lp_apriori(), lphom_joint(), lphom(), nslphom_dual(), nslphom_joint(), nslphom(), rslphom(), tslphom_dual(), tslphom_joint(), tslphom()

Examples

```
x <- France2017P[, 1:8]
y <- France2017P[, 9:12]
y[,1] <- y[,1]  - (rowSums(y) - rowSums(x))
mt <- lphom_dual(x, y)
mt$VTM.votes.w
mt$HETe.w
```

---

lphom_joint *Implements the lphom_joint algorithm*

Description

Estimates RxC vote transfer matrices (ecological contingency tables) with lphom_joint

Usage

```
lphom_joint(
  votes_election1,
  votes_election2,
  integers = FALSE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

votes_election1

data.frame (or matrix) of order IxJ with the counts to be initially mapped to rows. When estimating vote transfer matrices, the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.

votes_election2

data.frame (or matrix) of order IxK with the counts to be initially mapped to columns. When estimating vote transfer matrices, the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.

integers — A TRUE/FALSE value that indicates whether the LP solution of counts (votes) must be approximate to the closest integer solution using ILP. Default, FALSE.

solver — A character string indicating the linear programming solver to be used, only lp_solve and symphony are allowed. By default, lp_solve. The package Rsymphony needs to be installed for the option symphony to be used.

integers.solver — A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only symphony and lp_solve are allowed. By default, symphony. The package Rsymphony needs to be installed for the option symphony to be used. Only used when integers = TRUE.

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

Value

A list with the following components

VTM.votes — A matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2.

HETe — The estimated heterogeneity index associated to the VTM.votes solution.

VTM12 — The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the VTM.votes solution.

VTM21 — The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the VTM.votes solution.

EHet12 — A matrix of order IxK measuring in each unit a distance to the homogeneity hypothesis. That is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results in each territorial unit for each option of election two. The matrix Eik.

EHet21 — A matrix of order IxJ measuring in each unit a distance to the homogeneity hypothesis. That is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results in each territorial unit for each option of election one. The matrix Eij.

deterministic.bounds — A list of two matrices of order JxK containing for each vote transition the lower and upper proportions allowed given the observed aggregates.

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

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM and Romero, R (2024). Symmetry estimating RxC vote transfer matrices from aggregate data. *Journal of the Royal Statistical Society, Series A – Statistics in Society*, forthcoming. doi:10.1093/jrsssa/qnae013

See Also

lphom lphom_dual tslphom_dual nslphom_dual tslphom_joint nslphom_joint

Other linear programing ecological inference functions: lclphom(), lp_apriori(), lphom_dual(), lphom(), nslphom_dual(), nslphom_joint(), nslphom(), rslphom(), tslphom_dual(), tslphom_joint(), tslphom()

Examples

```
x <- France2017P[, 1:8]
y <- France2017P[, 9:12]
y[,1] <- y[,1]  - (rowSums(y) - rowSums(x))
mt <- lphom_joint(x, y)
mt$VTM.votes
mt$HETe
```

---

lp_apriori *Implements lp_apriori models*

Description

Adjusts an initial J0xK0 vote transfer matrix (ecological contingency table) to guarantee (i) congruency with aggregate results and (ii) completeness.

Usage

```
lp_apriori(
  votes_election1,
  votes_election2,
  apriori,
  weights = "constant",
  new_and_exit_voters = "raw",
  uniform = TRUE,
  solver = "lp_solve",
  integers = TRUE,
  integers.solver = "symphony",
  ...
)
```

Arguments

votes_election1
: data.frame (or matrix) of order IxJ1 (or vector of length J1) with the votes gained by (or the numbers corresponding to) the J1 political options competing on election 1 (or origin) in the I territorial units considered.

votes_election2
: data.frame (or matrix) of order IxK2 (or vector of length K2) with the votes gained by (or the numbers corresponding to) the K2 political options competing on election 2 (or destination) in the I territorial units considered.

`apriori` data.frame (or matrix) of order J0xK0 with an initial estimate of the (row-standarized) voter transition proportions/fractions, pjk0, between the first J0 election options of election 1 and the first K0 election options of election 2. It could be also a data.frame (matrix) of counts. This matrix can contain some missing values.

`weights` Either a numeric matrix (or data.frame) of order J0xK0 of weights, wjk, or a character string indicating the structure of weights to be used. As character string this argument admits seven different values: `constant`, `x`, `xy`, `expected`, `counts`, `sqrt`, or `sd`. Default, `constant` (i.e., wjk = 1). The wjk coefficients measure the (relative) degree of confidence we have in the a priori values pjk0. Everything else constant, the greater a weight wjk the closer the estimated pjk and the pjk0 proportions will be. As numeric matrix, this matrix can contain some missing values, usually located in the same cells than the missing values of `apriori`.

`new_and_exit_voters` A character string indicating the level of information available in `votes_election1` and `votes_election2` regarding new entries and exits of the election censuses between the two elections. This argument allows, in addition to the options discussed in Pavia (2023), three more options. This argument admits eleven different values: `raw`, `regular`, `ordinary`, `enriched`, `adjust1`, `adjust2`, `simultaneous`, `semifull`, `full`, `fullreverse` and `gold`. Default, `raw`.

`uniform` A TRUE/FALSE value that indicates if census exits affect all the electoral options in a (relatively) similar fashion; depending on the scenario any equation(s) among equations (6) to (11) of Pavia (2023) could be used in the underlying model. Default, `TRUE`.

`solver` A character string indicating the linear programming solver to be used, only `lp_solve` and `symphony` are allowed. By default, `lp_solve`. The package `Rsymphony` needs to be installed for the option `symphony` to be used.

`integers` A TRUE/FALSE value that indicates whether the LP solution of counts (votes) must be approximate to the closest integer solution using ILP to generate the final solution. Default, TRUE.

`integers.solver` A character string indicating the linear programming solver to be used for approximating to the closest integer solution. Only `symphony` and `lp_solve` are allowed. By default, `symphony`. The package `Rsymphony` needs to be installed for the option `symphony` to be used. Only used when `integers = TRUE`.

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

Details

Description of the `new_and_exit_voters` argument in more detail.

- `raw`: The default value. This argument accounts for the most plausible scenario when estimating vote transfer matrices. 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. In this scenario, net exits and net entries are estimated according to equation (7) of Romero et al. (2020). When both net entries and exits are no

null, constraint (15) of Pavia (2023) applies. If there are net exits and `uniform = TRUE` either constraints (6) or (8) and (15) of Pavia (2023) are imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.

- `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 equation (7) of Romero et al. (2020), and (iii) we can (or not) assume that net exits impact equally all the first J1 - 1 options of election 1. When both net entries and exits are no null, constraints (13) and (15) of Pavia (2023) apply. If `uniform = TRUE` and there are net exits either constraints (8) or (11) of Pavia (2023), depending on whether there are or not net entries, are also imposed. 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 period between elections, (ii) net entries and maybe other additional net exits are computed according to equation (7) of Romero et al. (2020), and (iii) we can assume (or not) that exits impact equally all the J1 options of election 1. When both net entries and exits are no null, constraints (14) and (15) of Pavia (2023) apply and if `uniform = TRUE` either constraints (8) and (9) or, without net entries, (6) and (7) of Pavia (2023) are also imposed. 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 somehow 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 equation (7) of Romero et al. (2020), and (iv) we can assume (or not) that exits impact equally all the J1 - 1 options of election 1. When both net entries and exits are no null, constraints (12) to (15) of Pavia (2023) apply and if `uniform = TRUE` constraints (10) and (11) of Pavia (2023) are also imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.

- `adjust1`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the first election (the row-sums of `votes_election1`) are proportionally adjusted to match the corresponding census of the polling units in the second election (the row-sums of `votes_election2`). If `integers = TRUE`, each row in `votes_election1` is proportionally adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election2`.

- `adjust2`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the second election (the row-sums of `votes_election2`) are proportionally adjusted to match the corresponding census of the polling units in the first election (the row-sums of `votes_election1`). If `integers = TRUE`, each row in `votes_election2` is adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election1`.

- `simultaneous`: This is the value to be used in classical ecological inference problems, such as in ecological studies of racial voting, and in scenarios with two simultaneous elections. In this scenario, the sum by rows of `votes_election1` and `votes_election2` must coincide. Constraints defined by equations (8) and (9) of Romero et al. (2020) are not included in the

model. In this case, the lphom function just implements the basic model defined, for instance, by equations (1) to (5) of Pavia (2024).

- `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 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 (2023) apply. Additionally, if `uniform` = `TRUE` constraints (8) of Pavia (2023) are also imposed.
- `full`: This value accounts for a scenario with two elections elapsed at least some months, where (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 (=J1) of `votes_election1` measures new immigrants that have the right to vote and (iii) the column K (=K2) 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. Additionally, if `uniform` = `TRUE` constraints (11) of Pavia (2023) are also imposed.
- `fullreverse`: This value is somehow the mirror version of `full`. It 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 and (ii) where 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 and constraints (14) and (15) of Pavia (2023) apply. Additionally, if `uniform` = `TRUE` constraints (8) and (9) of Pavia (2023) are also imposed.
- `gold`: This value accounts for a scenario similar to `full`, where 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 (2023) apply and if `uniform` = `TRUE` constraints (10) and (11) of Pavia (2023) are also imposed.

Value

A list with the following components

`VTM` A matrix of order JxK with the estimated percentages of row-standardized vote transitions from election 1 to election 2. In `raw`, `regular`, `ordinary` and `enriched` scenarios when the percentage of net entries is small, less than 1% of the census, net entries are omitted (i.e., J = J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census, net exits are omitted (i.e., K = K2) even when estimates for net exits different from zero are obtained.

`VTM.votes` A matrix of order JxK with the estimated vote transitions from election 1 to election 2. In `raw`, `regular`, `ordinary` and `enriched` scenarios when the percentage of net entries is small, less than 1% of the census, net entries are omitted (i.e., J = J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less

than 1% of the census, net exits are omitted (i.e., K = K2) even when estimates for net exits different from zero are obtained.

`weights` A matrix of order JxK with the weights used to adjust the a priori vote transitions from election 1 to election 2.

`OTM` A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2.

`VTM.complete` A matrix of order JxK with the estimated proportions of 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) even when they are really small.

`VTM.complete.votes` A matrix of order JxK with the estimated 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) even when they are really small.

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

`origin` A vector with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`destination` A vector with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

Author(s)

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

References

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

See Also

`lphom` `tslphom` `nslphom` `lclphom`

Other linear programing ecological inference functions: `lclphom()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `rslphom()`, `tslphom_dual()`, `tslphom_joint()`, `tslphom()`

Examples

```
P0 <- matrix(c(.75, .02, .15, .08, .01, .01, .97, .01,
               .01, .01, .01, .97, .01, .10, .80, .09,
               .20, .30, .30, .20, .10, .10, .50, .30,
               .10, .30, NA, NA, .25, .20, NA, NA),
             byrow = TRUE, 8, 4)
mt <- lp_apriori(France2017P[, 1:8], France2017P[, 9:12], P0, integers = FALSE)
```

nslphom *Implements nslphom algorithm*

Description

Estimates RxC (JxK) vote transfer matrices (ecological contingency tables) with nslphom

Usage

```
nslphom(
  votes_election1,
  votes_election2,
 new_and_exit_voters = c("raw", "regular", "ordinary", "enriched", "adjust1", "adjust2",
    "simultaneous", "semifull", "full", "fullreverse", "gold"),
  apriori = NULL,
  lambda = 0.5,
  uniform = TRUE,
  iter.max = 10,
  min.first = FALSE,
  structural_zeros = NULL,
  integers = FALSE,
  distance.local = c("abs", "max", "none"),
  verbose = TRUE,
  solver = "lp_solve",
  integers.solver = "symphony",
  burnin = 0,
  tol = 10^-5,
  ...
)
```

Arguments

votes_election1
: data.frame (or matrix) of order IxJ1 with the votes gained by (or the counts corresponding to) the J1 political options competing (available) on election 1 (or origin) in the I units considered. In general, the row marginals of the I tables corresponding to the units.

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. In general, the column marginals of the I tables corresponding to the units.

new_and_exit_voters
: A character string indicating the level of information available in votes_election1 and votes_election2 regarding new entries and exits of the election censuses

between the two elections. This argument allows, in addition to the options discussed in Pavia (2023), three more options. This argument admits eleven different values: raw, regular, ordinary, enriched, adjust1, adjust2, simultaneous, semifull, full, fullreverse and gold. Default, raw.

apriori — data.frame (or matrix) of order J0xK0 with an initial estimate of the (row-standarized) global voter transition proportions/fractions, pjk0, between the first J0 (election) options of election 1 and the first K0 (election) options of election 2. This matrix can contain some missing values. When no a priori information is available apriori is a null object. Default, NULL.

lambda — A number between 0 and 1, informing the relative weight the user assigns to the apriori information. Setting lambda = 0 is equivalent to not having a priori information (i.e., apriori = NULL). Default, 0.5.

uniform — A TRUE/FALSE value that informs whether census exits impact all the electoral options in a (relatively) similar fashion in all iterations, including iteration 0 and when deriving units tables. If uniform = TRUE typically at least one of the equations among equations (6) to (11) of Pavia (2023) is included in the underlying model. This parameter has no effect in simultaneous scenarios. It also has not impact in raw and regular scenarios when no net exits are estimated by the function from the provided information. Default, TRUE.

iter.max — Maximum number of iterations to be performed. The process ends when either the number of iterations reaches iter.max or when the maximum variation between two consecutive estimates of the probability transfer matrix is less than tol. By default, 10.

min.first — A TRUE/FALSE value. If min.first = FALSE, the matrix associated with the minimum HETe after performing iter.max iterations is taken as solution. If min.first = TRUE, the associated matrix to the instant in which the first decrease of HETe occurs is taken as solution. The process stops at that moment. In this last scenario (when min.first = TRUE), burnin = 0 is forced and iter.max is at least 100. Default, FALSE.

structural_zeros — Default NULL. A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2. For instance, when new_and_exit_voters is set to "semifull", lphom implicitly states structural_zeros = list(c(J1, K2)).

integers — A TRUE/FALSE value that indicates whether the problem is solved in integer values in both iterations, including iteration zero (lphom) and the rest of iterations, when deriving unit tables solutions. If integers = TRUE, the LP matrices are approximated to the closest integer solution solving the corresponding Integer Linear Program. Default, FALSE.

distance.local — A string argument that indicates whether the second step of the lphom_local algorithm should be performed to solve potential indeterminacies of local solutions. Default, "abs". If distance.local = "abs" lphom_local selects in its second step the matrix closer to the temporary global solution under L_1 norm, among the first step compatible matrices. If distance.local = "max" lphom_local selects in its second step the matrix closer to the temporary global solution under L_Inf norm, among the first step compatible matrices. If distance.local = "none", the second step of lphom_local is not performed.

verbose — A TRUE/FALSE value that indicates if a summary of the results of the computations performed to estimate net entries and exits should be printed on the screen. Default, TRUE.

solver — A character string indicating the linear programming solver to be used, only lp_solve and symphony are allowed. By default, lp_solve. The package Rsymphony needs to be installed for the option symphony to be used.

integers.solver — A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only symphony and lp_solve are allowed. By default, symphony. The package Rsymphony needs to be installed for the option symphony to be used. Only used when integers = TRUE.

burnin — Number of initial solutions to be discarded before determining the final solution. By default, 0.

tol — Maximum deviation allowed between two consecutive iterations. The process ends when the maximum variation between two proportions for the estimation of the transfer matrix between two consecutive iterations is less than tol or the maximum number of iterations, iter.max, has been reached. By default, 0.00001.

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

Details

Description of the new_and_exit_voters argument in more detail.

- raw: The default value. This argument accounts for the most plausible scenario when estimating vote transfer matrices. 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. In this scenario, net exits and net entries are estimated according to equation (7) of Romero et al. (2020). When both net entries and exits are no null, constraint (15) of Pavia (2023) applies. If there are net exits and uniform = TRUE either constraints (6) or (8) and (15) of Pavia (2023) are imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- 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 equation (7) of Romero et al. (2020), and (iii) we can (or not) assume that net exits impact equally all the first J1 - 1 options of election 1. When both net entries and exits are no null, constraints (13) and (15) of Pavia (2023) apply. If uniform = TRUE and there are net exits either constraints (8) or (11) of Pavia (2023), depending on whether there are or not net entries, are also imposed. 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 period between elections, (ii) net entries and maybe other additional net exits are computed according to equation (7) of Romero et al. (2020), and (iii) we can assume (or not) that exits impact equally all the J1 options of election 1. When both net entries and exits are no null, constraints (14) and (15) of Pavia (2023) apply and if uniform = TRUE either constraints (8) and (9) or,

without net entries, (6) and (7) of Pavia (2023) are also imposed. 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 somehow 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 equation (7) of Romero et al. (2020), and (iv) we can assume (or not) that exits impact equally all the J1 - 1 options of election 1. When both net entries and exits are no null, constraints (12) to (15) of Pavia (2023) apply and if `uniform = TRUE` constraints (10) and (11) of Pavia (2023) are also imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `adjust1`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the first election (the row-sums of `votes_election1`) are proportionally adjusted to match the corresponding census of the polling units in the second election (the row-sums of `votes_election2`). If `integers = TRUE`, each row in `votes_election1` is proportionally adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election2`.
- `adjust2`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the second election (the row-sums of `votes_election2`) are proportionally adjusted to match the corresponding census of the polling units in the first election (the row-sums of `votes_election1`). If `integers = TRUE`, each row in `votes_election2` is adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election1`.
- `simultaneous`: This is the value to be used in classical ecological inference problems, such as in ecological studies of racial voting, and in scenarios with two simultaneous elections. In this scenario, the sum by rows of `votes_election1` and `votes_election2` must coincide. Constraints defined by equations (8) and (9) of Romero et al. (2020) are not included in the model. In this case, the lphom function just implements the basic model defined, for instance, by equations (1) to (5) of Pavia (2024).
- `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 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 (2023) apply. Additionally, if `uniform = TRUE` constraints (8) of Pavia (2023) are also imposed.
- `full`: This value accounts for a scenario with two elections elapsed at least some months, where (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 (=J1) of `votes_election1` measures new immigrants that have the right to vote and (iii) the column K (=K2) 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. Additionally, if `uniform = TRUE` constraints (11) of Pavia (2023) are also imposed.
- `fullreverse`: This value is somehow the mirror version of `full`. It 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 and (ii) where 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 and constraints (14) and (15) of Pavia (2023) apply. Additionally, if uniform = TRUE constraints (8) and (9) of Pavia (2023) are also imposed.

- gold: This value accounts for a scenario similar to full, where 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 (2023) apply and if uniform = TRUE constraints (10) and (11) of Pavia (2023) are also imposed.

Value

A list with the following components

VTM — A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated percentages of row-standardized vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census in all units, net entries are omitted (i.e., the number of rows of VTM is equal to J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census in all units, net exits are omitted (i.e., the number of rows of VTM is equal to K2) even when estimates for net exits different from zero are obtained.

VTM.votes — A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census, net entries are omitted (i.e., J = J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census, net exits are omitted (i.e., K = K2) even when estimates for net exits different from zero are obtained.

OTM — A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2.

HETe — The estimated heterogeneity index as defined in equation (15) of Pavia and Romero (2022).

VTM.complete — A matrix of order JxK with the estimated proportions of row-standardized vote transitions from election 1 to election 2, including in regular and raw scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.complete.votes — A matrix of order JxK with the estimated vote transitions from election 1 to election 2, including in regular and raw scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.sequence — Array of order JxKx(iter+1) (where iter is the efective number of iterations performed) of the estimated matrices corresponding to each iteration.

`HETe.sequence` Numeric vector of length `iter+1` with the `HETe` coefficients corresponding to the matrices in `VTM.sequence`.

`VTM.prop.units` An array of order JxKxI with the estimated proportions of vote transitions from election 1 to election 2 attained for each unit in the selected iteration.

`VTM.votes.units`
An array of order JxKxI with the estimated matrix of vote transitions from election 1 to election 2 attained for for each unit in the selected iteration.

`zeros` A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2.

`iter` The real final number of iterations performed before ending the process.

`iter.min` Number of the iteration associated to the selected `VTM` solution.

`deterministic.bounds`
A list of two matrices of order JxK and two arrays of order JxKxI containing for each vote transition the lower and upper allowed proportions given the observed aggregates.

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

`origin` A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`destination` A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`EHet` A matrix of order IxK measuring in each spatial unit a distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results with the solution in each territorial unit for each option of election 2.

`solution_init` A list with the main outputs produced by **lphom()**.

- `VTM_init`: A matrix of order JxK with the estimated percentages of vote transitions from election 1 to election 2 initially obtained by **lphom()**.
- `VTM.votes_init`: A matrix of order JxK with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()**.
- `OTM_init`: A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2 initially obtained by **lphom()**.
- `HETe_init`: The estimated heterogeneity index defined in equation (10) of Romero et al. (2020).
- `EHet_init`: A matrix of order IxK measuring in each spatial unit the distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results, using the **lphom()** solution, in each territorial unit for each option of election 2.
- `VTM.complete_init`: A matrix of order J'xK' with the estimated proportions of vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in `regular` and `raw` scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

- VTM.complete.votes_init: A matrix of order J'xK' with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in regular and raw scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM, and Romero, R (2022). Improving estimates accuracy of voter transitions. Two new algorithms for ecological inference based on linear programming, *Sociological Methods & Research*. doi:10.1177/00491241221092725.

See Also

lphom tslphom lclphom rslphom

Other linear programing ecological inference functions: lclphom(), lp_apriori(), lphom_dual(), lphom_joint(), lphom(), nslphom_dual(), nslphom_joint(), rslphom(), tslphom_dual(), tslphom_joint(), tslphom()

Examples

```
mt.ns <- nslphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
mt.ns$VTM
mt.ns$HETe
mt.ns$solution_init$HETe_init
```

---

nslphom_dual *Implements the nslphom_dual algorithm*

Description

Estimates RxC vote transfer matrices (ecological contingency tables) with nslphom_dual

Usage

```
nslphom_dual(
  votes_election1,
  votes_election2,
  iter.max = 10,
  min.first = FALSE,
  integers = FALSE,
  solver = "lp_solve",
  integers.solver = "symphony",
```

```
  tol = 10^-5,
  ...
)
```

Arguments

`votes_election1`
: data.frame (or matrix) of order IxJ with the counts to be initially mapped to rows. When estimating vote transfer matrices, the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. The sum by rows of `votes_election1` and `votes_election2` must coincide.

`votes_election2`
: data.frame (or matrix) of order IxK with the counts to be initially mapped to columns. When estimating vote transfer matrices, the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. The sum by rows of `votes_election1` and `votes_election2` must coincide.

`iter.max`
: Maximum number of iterations to be performed in each dual linear program. The process ends independently in each system when either the number of iterations reaches iter.max or when the maximum variation between two consecutive estimates of the probability transfer matrix is less than `tol`. By default, 10.

`min.first`
: A `TRUE/FALSE` value. If `FALSE`, the matrix associated with the minimum `HETe` after performing `iter.max` iterations is taken as solution. If `TRUE`, the associated matrix to the instant in which the first decrease of `HETe` occurs is taken as solution. The process stops at that moment. In this last scenario (when `min.first = TRUE`), `iter.max` is is forced to be at least 100. Default, `FALSE`.

`integers`
: A `TRUE/FALSE` value that indicates whether the problem is solved in integer values in each iteration: zero (lphom) and intermediate and final (including unit) solutions. If `TRUE`, the initial LP matrices are approximated in each iteration to the closest integer solution solving the corresponding Integer Linear Program. Default, `FALSE`.

`solver`
: A character string indicating the linear programming solver to be used, only `lp_solve` and `symphony` are allowed. By default, `lp_solve`. The package `Rsymphony` needs to be installed for the option `symphony` to be used.

`integers.solver`
: A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only `symphony` and `lp_solve` are allowed. By default, `symphony`. The package `Rsymphony` needs to be installed for the option `symphony` to be used. Only used when `integers = TRUE`.

`tol`
: Maximum deviation allowed between two consecutive iterations. The process ends when the maximum variation between two proportions for the estimation of the transfer matrix between two consecutive iterations is less than `tol` or the maximum number of iterations, `iter.max`, has been reached. By default, 0.00001.

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

Value

A list with the following components

`VTM.votes.w` The matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2, attained weighting the two dual solutions using as weights the corresponding HTEe estimates.

`VTM.votes.units.w` The array of order JxKxI with the local estimated cross-distributions of votes of elections 1 and 2 by unit, attained weighting the two dual solutions using as weights the corresponding HTEe estimates.

`VTM.votes.a` The matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2, attained simple averaging the two dual solutions.

`VTM.votes.units.a` The matrix of order JxKxI with the estimated cross-distributions of votes of elections 1 and 2 by unit, attained weighting the two dual solutions using as weights the corresponding HTEe estimates.

`HETe.w` Estimated heterogeneity index associated to the `VTM.votes.w` solution.

`HETe.a` Estimated heterogeneity index associated to the `VTM.votes.a` solution.

`VTM12.w` The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the `VTM.votes.w` solution.

`VTM21.w` The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the `VTM.votes.w` solution.

`VTM12.a` The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the `VTM.votes.a` solution.

`VTM21.a` The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the `VTM.votes.a` solution.

`nslphom.object.12` The output of the `nslphom` function attained solving the problem X –> Y, that is, mapping `votes_election1` to rows and `votes_election2` to columns.

`nslphom.object.21` The output of the `nslphom` function attained solving the problem Y –> X, that is, mapping `votes_election2` to rows and `votes_election1` to columns.

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

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM and Romero, R (2024). Symmetry estimating RxC vote transfer matrices from aggregate data. *Journal of the Royal Statistical Society, Series A – Statistics in Society*, forthcoming. doi:10.1093/jrsssa/qnae013

See Also

nslphom lphom_dual tslphom_dual lphom_joint tslphom_joint nslphom_joint

Other linear programing ecological inference functions: lclphom(), lp_apriori(), lphom_dual(), lphom_joint(), lphom(), nslphom_joint(), nslphom(), rslphom(), tslphom_dual(), tslphom_joint(), tslphom()

Examples

```
x <- France2017P[, 1:8]
y <- France2017P[, 9:12]
y[,1] <- y[,1]  - (rowSums(y) - rowSums(x))
mt <- nslphom_dual(x, y)
mt$VTM.votes.w
mt$HETe.w
```

`nslphom_joint` *Implements the nslphom_joint algorithm*

Description

Estimates RxC vote transfer matrices (ecological contingency tables) with nslphom_joint

Usage

```
nslphom_joint(
  votes_election1,
  votes_election2,
  iter.max = 10,
  min.first = FALSE,
  integers = FALSE,
  solver = "lp_solve",
  integers.solver = "symphony",
  tol = 0.001,
  ...
)
```

Arguments

`votes_election1`

data.frame (or matrix) of order IxJ with the counts to be initially mapped to rows. When estimating vote transfer matrices, the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. The sum by rows of `votes_election1` and `votes_election2` must coincide.

votes_election2
: data.frame (or matrix) of order IxK with the counts to be initially mapped to columns. When estimating vote transfer matrices, the votes gained by the K political options competing on election 2 (or destination) in the I territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.

iter.max
: Maximum number of iterations to be performed. The process ends independently when either the number of iterations reaches iter.max or when the maximum variation between two consecutive estimates of both ways probability transfer matrices are less than tol. By default, 10.

min.first
: A TRUE/FALSE value. If FALSE, the matrix associated with the minimum HETe after performing iter.max iterations is taken as solution. If TRUE, the associated matrix to the instant in which the first decrease of HETe occurs is taken as solution. The process stops at that moment. In this last scenario (when min.first = TRUE), iter.max is is forced to be at least 100. Default, FALSE.

integers
: A TRUE/FALSE value that indicates whether the problem is solved in integer values in each iteration: zero (lphom) and intermediate and final (including unit) solutions. If TRUE, the initial LP matrices are approximated in each iteration to the closest integer solution solving the corresponding Integer Linear Program. Default, FALSE.

solver
: A character string indicating the linear programming solver to be used, only lp_solve and symphony are allowed. By default, lp_solve. The package Rsymphony needs to be installed for the option symphony to be used.

integers.solver
: A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only symphony and lp_solve are allowed. By default, symphony. The package Rsymphony needs to be installed for the option symphony to be used. Only used when integers = TRUE.

tol
: Maximum deviation allowed between two consecutive iterations. The process ends when the maximum variation between the estimated cross-distributions of votes between two consecutive iterations is less than tol or the maximum number of iterations, iter.max, has been reached. By default, 0.001.

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

Value

A list with the following components

VTM.votes
: A matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2.

HETe
: The estimated heterogeneity index associated to the VTM.votes solution.

VTM12
: The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the VTM.votes solution.

VTM21
: The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the VTM.votes solution.

`VTM.votes.units`
: An array of order JxKxI with the estimated matrix of cross-distributions of votes of elections 1 and 2 attained for each unit in iteration of the solution.

`iter`
: The real final number of iterations performed before ending the process.

`iter.min`
: Number of the iteration associated to the selected `VTM.votes` solution.

`EHet12`
: A matrix of order IxK measuring in each unit a distance to the homogeneity hypothesis. That is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results in each territorial unit for each option of election two. The matrix Eik.

`EHet21`
: A matrix of order IxJ measuring in each unit a distance to the homogeneity hypothesis. That is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results in each territorial unit for each option of election one. The matrix Eij.

`deterministic.bounds`
: A list of two matrices of order JxK and two arrays of order JxKxI containing for each vote transition the lower and upper allowed proportions given the observed aggregates.

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

`solution_init`
: A list with the main outputs produced by **lphom_joint()**.

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM and Romero, R (2024). Symmetry estimating RxC vote transfer matrices from aggregate data. *Journal of the Royal Statistical Society, Series A – Statistics in Society*, forthcoming. doi:10.1093/jrsssa/qnae013

See Also

`nslphom` `lphom_dual` `tslphom_dual` `nslphom_dual` `lphom_joint` `tslphom_joint`

Other linear programing ecological inference functions: `lclphom()`, `lp_apriori()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom()`, `rslphom()`, `tslphom_dual()`, `tslphom_joint()`, `tslphom()`

Examples

```
x <- France2017P[, 1:8]
y <- France2017P[, 9:12]
y[,1] <- y[,1]  - (rowSums(y) - rowSums(x))
mt <- nslphom_joint(x, y, iter.max = 3)
mt$VTM.votes
mt$HETe
```

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

Description

Plot method for objects obtained with an algorithm of the lphom-family (lphom, tslphom, nslphom, tslphom_dual, nslphom_joint, ...).

Usage

```
## S3 method for class 'lphom'
plot(
  x,
  complete = FALSE,
  margins = TRUE,
  digits = 2,
  row.names = NULL,
  col.names = NULL,
  size.numbers = 6,
  size.labels = 4,
  size.margins = 4,
  colour.cells = "deeppink3",
  colour.grid = "blanchedalmond",
  alpha = 0.5,
  which = NULL,
  ...,
  type = "w",
  show.plot = TRUE
)
```

Arguments

`x`	An object output of a `lphom` family algorithm.
`complete`	A `TRUE/FALSE` argument informing if the complete matrix should be displayed. In `raw`, `regular`, `ordinary` and `enriched` scenarios the plot includes the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units. Default, `FALSE`.
`margins`	A `TRUE/FALSE`` argument informing if the margins of the matrix should be displayed. Defa
`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.
type	A character string indicating the solution (transfer matrix) to be plotted. Only valid for **_dual** algorithms. type = "w" stands for the weighted solution and type = "a" for the simple average solution. Default w.
show.plot	A TRUE/FALSE argument 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

```
mt.ns <- nslphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
p <- plot(mt.ns, show.plot = FALSE)
p
```

---

print.lphom *Print a summary of a lphom-family object*

Description

Print method for objects obtained with an algorithm of the lphom-family (lphom, tslphom, nslphom, tslphom_dual, nslphom_joint, ...).

Usage

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

Arguments

x	An object output of a lphom family algorithm.
...	Other arguments passed on to methods. Not currently used.
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.

Author(s)

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

Examples

```
mt.ns <- nslphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
print(mt.ns, digits = 2, margins = TRUE)
```

---

print.summary.lphom *Print a summary of a lphom-family object*

Description

Print method for summary.lphom objects

Usage

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

Arguments

x	An summary.lphom class object.
...	Other arguments passed on to methods. Not currently used.
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.

`rslphom` *Implements rslphom algorithm*

Description

Estimates RxC (JxK) vote transfer matrices (ecological contingency tables) with rslphom

Usage

```
rslphom(
  votes_election1,
  votes_election2,
  emphasis = 0.995,
 new_and_exit_voters = c("raw", "regular", "ordinary", "enriched", "adjust1", "adjust2",
    "simultaneous", "semifull", "full", "fullreverse", "gold"),
  apriori = NULL,
  lambda = 0.5,
  uniform = TRUE,
  structural_zeros = NULL,
  integers = FALSE,
  distance.local = c("abs", "max", "none"),
  save.local.by.emphasis = FALSE,
  verbose = TRUE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

`votes_election1`
: data.frame (or matrix) of order IxJ1 with the votes gained by (or the counts corresponding to) the J1 political options competing (available) on election 1 (or origin) in the I units considered. In general, the row marginals of the I tables corresponding to the units.

`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. In general, the column marginals of the I tables corresponding to the units.

`emphasis`
: A numerical vector of values between 0 and 1 informing of the weights/emphasis to be used to promote each unit when estimating its transfer matrix. Default, 0.995. When the length of `emphasis` is one, only a weight (a level of emphasis) is analyzed. When the length of `emphasis` is higher than one, as many as different weights/emphasis as the length of emphasis are tried in the estimation of the transfer matrix of each unit. In each unit, the local solution selected corresponds to the transfer matrix with lower expected error.

new_and_exit_voters
: A character string indicating the level of information available in votes_election1 and votes_election2 regarding new entries and exits of the election censuses between the two elections. This argument allows, in addition to the options discussed in Pavia (2023), three more options. This argument admits eleven different values: raw, regular, ordinary, enriched, adjust1, adjust2, simultaneous, semifull, full, fullreverse and gold. Default, raw.

apriori
: data.frame (or matrix) of order J0xK0 with an initial estimate of the (row-standarized) global voter transition proportions/fractions, pjk0, between the first J0 (election) options of election 1 and the first K0 (election) options of election 2. This matrix can contain some missing values. When no a priori information is available apriori is a null object. Default, NULL.

lambda
: A number between 0 and 1 informing the relative weight the user assigns to the apriori information. Setting lambda = 0 is equivalent to not having a priori information (i.e., apriori = NULL). Default, 0.5.

uniform
: A TRUE/FALSE value that informs whether census exits impact all the electoral options in a (relatively) similar fashion in all iterations, including iteration 0 and when deriving units tables. If uniform = TRUE typically at least one of the equations among equations (6) to (11) of Pavia (2023) is included in the underlying model. This parameter has no effect in simultaneous scenarios. It also has not impact in raw and regular scenarios when no net exits are estimated by the function from the provided information. Default, TRUE.

structural_zeros
: Default NULL. A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2. For instance, when new_and_exit_voters is set to "semifull", lphom implicitly states structural_zeros = list(c(J1, K2)).

integers
: A TRUE/FALSE value that indicates whether the problem is solved in integer values in all the steps, including lphom intermediate solutions and unit solutions. If integers = TRUE, the LP matrices are approximated to the closest integer solution solving the corresponding Integer Linear Program. Default, FALSE.

distance.local
: A string argument that indicates whether the second step of the lphom_local algorithm should be performed to solve potential indeterminacies of local solutions. Default, "abs". If distance.local = "abs" lphom_local selects in its second step the matrix closer to the temporary global solution under L_1 norm, among the first step compatible matrices. If distance.local = "max" lphom_local selects in its second step the matrix closer to the temporary global solution under L_Inf norm, among the first step compatible matrices. If distance.local = "none", the second step of lphom_local is not performed.

save.local.by.emphasis
: A TRUE/FALSE value that indicates if the estimated matrices obtained in each unit for each value of emphasis should be saved. Default, FALSE.

verbose
: A TRUE/FALSE value that indicates if a summary of the results of the computations performed to estimate net entries and exits should be printed on the screen. Default, TRUE.

`solver` A character string indicating the linear programming solver to be used, only `lp_solve` and `symphony` are allowed. By default, `lp_solve`. The package `Rsymphony` needs to be installed for the option `symphony` to be used.

`integers.solver` A character string indicating the linear programming solver to be used for approximating the LP solution to the closest integer solution. Only `symphony` and `lp_solve` are allowed. By default, `symphony`. The package `Rsymphony` needs to be installed for the option `symphony` to be used. Only used when `integers = TRUE`.

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

Details

Description of the `new_and_exit_voters` argument in more detail.

- `raw`: The default value. This argument accounts for the most plausible scenario when estimating vote transfer matrices. 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. In this scenario, net exits and net entries are estimated according to equation (7) of Romero et al. (2020). When both net entries and exits are no null, constraint (15) of Pavia (2023) applies. If there are net exits and `uniform = TRUE` either constraints (6) or (8) and (15) of Pavia (2023) are imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `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 equation (7) of Romero et al. (2020), and (iii) we can (or not) assume that net exits impact equally all the first J1 - 1 options of election 1. When both net entries and exits are no null, constraints (13) and (15) of Pavia (2023) apply. If `uniform = TRUE` and there are net exits either constraints (8) or (11) of Pavia (2023), depending on whether there are or not net entries, are also imposed. 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 period between elections, (ii) net entries and maybe other additional net exits are computed according to equation (7) of Romero et al. (2020), and (iii) we can assume (or not) that exits impact equally all the J1 options of election 1. When both net entries and exits are no null, constraints (14) and (15) of Pavia (2023) apply and if `uniform = TRUE` either constraints (8) and (9) or, without net entries, (6) and (7) of Pavia (2023) are also imposed. 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 somehow 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 equation (7) of Romero et al. (2020), and (iv) we can assume (or not) that exits impact equally all the J1 - 1 options of election 1. When both net entries and exits are no null, constraints (12) to (15) of Pavia (2023) apply and if `uniform = TRUE` constraints (10) and (11) of Pavia (2023)

are also imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.

- `adjust1`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the first election (the row-sums of `votes_election1`) are proportionally adjusted to match the corresponding census of the polling units in the second election (the row-sums of `votes_election2`). If `integers = TRUE`, each row in `votes_election1` is proportionally adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election2`.
- `adjust2`: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the second election (the row-sums of `votes_election2`) are proportionally adjusted to match the corresponding census of the polling units in the first election (the row-sums of `votes_election1`). If `integers = TRUE`, each row in `votes_election2` is adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in `votes_election1`.
- `simultaneous`: This is the value to be used in classical ecological inference problems, such as in ecological studies of racial voting, and in scenarios with two simultaneous elections. In this scenario, the sum by rows of `votes_election1` and `votes_election2` must coincide. Constraints defined by equations (8) and (9) of Romero et al. (2020) are not included in the model. In this case, the lphom function just implements the basic model defined, for instance, by equations (1) to (5) of Pavia (2024).
- `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 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 (2023) apply. Additionally, if `uniform = TRUE` constraints (8) of Pavia (2023) are also imposed.
- `full`: This value accounts for a scenario with two elections elapsed at least some months, where (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 (=J1) of `votes_election1` measures new immigrants that have the right to vote and (iii) the column K (=K2) 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. Additionally, if `uniform = TRUE` constraints (11) of Pavia (2023) are also imposed.
- `fullreverse`: This value is somehow the mirror version of `full`. It 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 and (ii) where 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 and constraints (14) and (15) of Pavia (2023) apply. Additionally, if `uniform = TRUE` constraints (8) and (9) of Pavia (2023) are also imposed.
- `gold`: This value accounts for a scenario similar to `full`, where 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 (2023) apply and if uniform = TRUE constraints (10) and (11) of Pavia (2023) are also imposed.

Value

A list with the following components

VTM — A matrix of order J'xK' (where J'=J-1 or J and K'= K-1 or K) with the estimated percentages of row-standardized vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census in all units, net entries are omitted (i.e., the number of rows of VTM is equal to J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census in all units, net exits are omitted (i.e., the number of rows of VTM is equal to K2) even when estimates for net exits different from zero are obtained.

VTM.votes — A matrix of order J'xK' (where J'=J-1 or J and K'= K-1 or K) with the estimated vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census, net entries are omitted (i.e., J = J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census, net exits are omitted (i.e., K = K2) even when estimates for net exits different from zero are obtained.

OTM — A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2.

HETe — The estimated heterogeneity index as defined in equation (15) of Pavia and Romero (2022).

VTM.complete — A matrix of order JxK with the estimated proportions of row-standardized vote transitions from election 1 to election 2, including in raw, regular, ordinary and enriched scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.complete.votes — A matrix of order JxK with the estimated vote transitions from election 1 to election 2, including in raw, regular, ordinary and enriched scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.prop.units — An array of order JxKxI with the estimated proportions of vote transitions from election 1 to election 2 attained for each unit after adjusting the **lphom()** initial estimate.

VTM.votes.units — An array of order JxKxI with the estimated matrix of vote transitions from election 1 to election 2 attained for each unit after adjusting the **lphom()** initial estimate.

VTM.sequence — Array of order JxKxlength(emphasis) with the global estimated matrices corresponding to each weight.

`zeros` A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2.

`errors` A matrix of order Ixlength(emphasis) with the expected errors for each unit and weight. The solution determined by `VTM.prop.units` or `VTM.votes.units` is the one obtained combining the unit solutions corresponding to the minimum observed errors.

`VTM.prop.units.by.emphasis`
An array of order JxKxIxlength(emphasis) with the estimated proportions of vote transitions from election 1 to election 2 attained in each unit for each weight. This is a `NULL` array if `save.local.by.emphasis = FALSE`.

`deterministic.bounds`
A list of two matrices of order JxK and two arrays of order JxKxI containing for each vote transition the lower and upper allowed proportions given the observed aggregates.

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

`origin` A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`destination` A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`EHet` A matrix of order IxK measuring in each spatial unit a distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results with the solution in each territorial unit for each option of election 2.

`solution_init` A list with the main outputs produced by **lphom()**.

- `VTM_init`: A matrix of order J'xK' with the estimated percentages of vote transitions from election 1 to election 2 initially obtained by **lphom()** with the raw data, without promoting any unit.
- `VTM.votes_init`: A matrix of order J'xK' with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()** with the raw data, without promoting any unit.
- `OTM_init`: A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2 initially obtained by **lphom()** with the raw data, without promoting any unit.
- `HETe_init`: The estimated heterogeneity index defined in equation (10) of Romero et al. (2020).
- `EHet_init`: A matrix of order IxK measuring in each spatial unit the distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results, using the **lphom()** solution with the raw data, without promoting any unit, in each territorial unit for each option of election 2.
- `VTM.complete_init`: A matrix of order JxK with the estimated proportions of vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in `raw`, `regular`, `ordinary` and `enriched` scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

- `VTM.complete.votes_init`: A matrix of order JxK with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in `raw`, `regular`, `ordinary` and `enriched` scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

Author(s)

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

References

Pavia, JM, and Romero, R (2022). Improving estimates accuracy of voter transitions. Two new algorithms for ecological inference based on linear programming, *Sociological Methods & Research*. doi:10.1177/00491241221092725.

Pavia, JM, and Penades, A (2024). A bottom-up approach for ecological inference.

See Also

`lphom` `tslphom` `nslphom` `lclphom`

Other linear programing ecological inference functions: `lclphom()`, `lp_apriori()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `tslphom_dual()`, `tslphom_joint()`, `tslphom()`

Examples

```
mt.rs <- rslphom(France2017P[, 1:8] , France2017P[, 9:12], emphasis = 0.5)
mt.rs$VTM
```

`summary.lphom` *Summarize a lphom-family object*

Description

Summary method for objects obtained with an algorithm of the lphom-family (lphom, tslphom, nslphom, tslphom_dual, nslphom_joint, ...).

Usage

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

Arguments

`object`	An object output of a `lphom` family algorithm.
`...`	Other arguments passed on to methods. Not currently used.

Value

An object of class "summary.lphom".

Author(s)

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

Examples

```
mt.ns <- nslphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
summary(mt.ns)
```

---

tslphom *Implements tslphom algorithm*

Description

Estimates RxC (JxK) vote transfer matrices (ecological contingency tables) with tslphom

Usage

```
tslphom(
  votes_election1,
  votes_election2,
 new_and_exit_voters = c("raw", "regular", "ordinary", "enriched", "adjust1", "adjust2",
    "simultaneous", "semifull", "full", "fullreverse", "gold"),
  apriori = NULL,
  lambda = 0.5,
  uniform = TRUE,
  structural_zeros = NULL,
  integers = FALSE,
  distance.local = c("abs", "max", "none"),
  verbose = TRUE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

votes_election1

data.frame (or matrix) of order IxJ1 with the votes gained by (or the counts corresponding to) the J1 political options competing (available) on election 1 (or origin) in the I units considered. In general, the row marginals of the I tables corresponding to the units.

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. In general, the column marginals of the I tables corresponding to the units.

new_and_exit_voters
: A character string indicating the level of information available in votes_election1 and votes_election2 regarding new entries and exits of the election censuses between the two elections. This argument allows, in addition to the options discussed in Pavia (2023), three more options. This argument admits eleven different values: raw, regular, ordinary, enriched, adjust1, adjust2, simultaneous, semifull, full, fullreverse and gold. Default, raw.

apriori
: data.frame (or matrix) of order J0xK0 with an initial estimate of the (row-standarized) global voter transition proportions/fractions, pjk0, between the first J0 (election) options of election 1 and the first K0 (election) options of election 2. This matrix can contain some missing values. When no a priori information is available apriori is a null object. Default, NULL.

lambda
: A number between 0 and 1, informing the relative weight the user assigns to the apriori information. Setting lambda = 0 is equivalent to not having a priori information (i.e., apriori = NULL). Default, 0.5.

uniform
: A TRUE/FALSE value that informs whether census exits impact all the electoral options in a (relatively) similar fashion in all iterations, including iteration 0 and when deriving units tables. If uniform = TRUE typically at least one of the equations among equations (6) to (11) of Pavia (2023) is included in the underlying model. This parameter has no effect in simultaneous scenarios. It also has not impact in raw and regular scenarios when no net exits are estimated by the function from the provided information. Default, TRUE.

structural_zeros
: Default NULL. A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2. For instance, when new_and_exit_voters is set to "semifull", lphom implicitly states structural_zeros = list(c(J1, K2)).

integers
: A TRUE/FALSE`` value that indicates whether the problem is solved in integer values in b tegers = TRUE, the LP matrices are approximated to the closest integer solution solving

distance.local
: A string argument that indicates whether the second step of the lphom_local algorithm should be performed to solve potential indeterminacies of local solutions. Default, "abs". If distance.local = "abs" lphom_local selects in its second step the matrix closer to the temporary global solution under L_1 norm, among the first step compatible matrices. If distance.local = "max" lphom_local selects in its second step the matrix closer to the temporary global solution under L_Inf norm, among the first step compatible matrices. If distance.local = "none", the second step of lphom_local is not performed.

verbose
: A TRUE/FALSE value that indicates if a summary of the results of the computations performed to estimate net entries and exits should be printed on the screen. Default, TRUE.

`solver` A character string indicating the linear programming solver to be used, only `lp_solve` and `symphony` are allowed. By default, `lp_solve`. The package `Rsymphony` needs to be installed for the option `symphony` to be used.

`integers.solver` A character string indicating the linear programming solver to be used for approximating the LP solution to the closest integer solution. Only `symphony` and `lp_solve` are allowed. By default, `symphony`. The package `Rsymphony` needs to be installed for the option `symphony` to be used. Only used when `integers = TRUE`.

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

Details

Description of the `new_and_exit_voters` argument in more detail.

- `raw`: The default value. This argument accounts for the most plausible scenario when estimating vote transfer matrices. 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. In this scenario, net exits and net entries are estimated according to equation (7) of Romero et al. (2020). When both net entries and exits are no null, constraint (15) of Pavia (2023) applies. If there are net exits and `uniform = TRUE` either constraints (6) or (8) and (15) of Pavia (2023) are imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.
- `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 equation (7) of Romero et al. (2020), and (iii) we can (or not) assume that net exits impact equally all the first J1 - 1 options of election 1. When both net entries and exits are no null, constraints (13) and (15) of Pavia (2023) apply. If `uniform = TRUE` and there are net exits either constraints (8) or (11) of Pavia (2023), depending on whether there are or not net entries, are also imposed. 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 period between elections, (ii) net entries and maybe other additional net exits are computed according to equation (7) of Romero et al. (2020), and (iii) we can assume (or not) that exits impact equally all the J1 options of election 1. When both net entries and exits are no null, constraints (14) and (15) of Pavia (2023) apply and if `uniform = TRUE` either constraints (8) and (9) or, without net entries, (6) and (7) of Pavia (2023) are also imposed. 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 somehow 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 equation (7) of Romero et al. (2020), and (iv) we can assume (or not) that exits impact equally all the J1 - 1 options of election 1. When both net entries and exits are no null, constraints (12) to (15) of Pavia (2023) apply and if `uniform = TRUE` constraints (10) and (11) of Pavia (2023)

are also imposed. In this scenario, J could be equal to J1 or J1 + 1 and K equal to K2 or K2 + 1.

- adjust1: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the first election (the row-sums of votes_election1) are proportionally adjusted to match the corresponding census of the polling units in the second election (the row-sums of votes_election2). If integers = TRUE, each row in votes_election1 is proportionally adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in votes_election2.
- adjust2: This value accounts for a scenario with two elections elapsed at least some months where the census in each of the I polling units of the second election (the row-sums of votes_election2) are proportionally adjusted to match the corresponding census of the polling units in the first election (the row-sums of votes_election1). If integers = TRUE, each row in votes_election2 is adjusted to the closest integer vector whose sum is equal to the sum of the corresponding row in votes_election1.
- simultaneous: This is the value to be used in classical ecological inference problems, such as in ecological studies of racial voting, and in scenarios with two simultaneous elections. In this scenario, the sum by rows of votes_election1 and votes_election2 must coincide. Constraints defined by equations (8) and (9) of Romero et al. (2020) are not included in the model. In this case, the lphom function just implements the basic model defined, for instance, by equations (1) to (5) of Pavia (2024).
- 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 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 (2023) apply. Additionally, if uniform = TRUE constraints (8) of Pavia (2023) are also imposed.
- full: This value accounts for a scenario with two elections elapsed at least some months, where (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 (=J1) of votes_election1 measures new immigrants that have the right to vote and (iii) the column K (=K2) 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. Additionally, if uniform = TRUE constraints (11) of Pavia (2023) are also imposed.
- fullreverse: This value is somehow the mirror version of full. It 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 and (ii) where 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 and constraints (14) and (15) of Pavia (2023) apply. Additionally, if uniform = TRUE constraints (8) and (9) of Pavia (2023) are also imposed.
- gold: This value accounts for a scenario similar to full, where 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 (2023) apply and if uniform = TRUE constraints (10) and (11) of Pavia (2023) are also imposed.

Value

A list with the following components

VTM — A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated percentages of row-standardized vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census in all units, net entries are omitted (i.e., the number of rows of VTM is equal to J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census in all units, net exits are omitted (i.e., the number of rows of VTM is equal to K2) even when estimates for net exits different from zero are obtained.

VTM.votes — A matrix of order J'xK' (where J'=J-1 or J and K'=K-1 or K) with the estimated vote transitions from election 1 to election 2. In raw, regular, ordinary and enriched scenarios when the percentage of net entries is small, less than 1% of the census, net entries are omitted (i.e., J = J1) even when estimates for net entries different from zero are obtained. Likewise, in the same scenarios when the percentage of net exits is small, less than 1% of the census, net exits are omitted (i.e., K = K2) even when estimates for net exits different from zero are obtained.

OTM — A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2.

HETe — The estimated heterogeneity index as defined in equation (15) of Pavia and Romero (2022).

VTM.complete — A matrix of order JxK with the estimated proportions of row-standardized vote transitions from election 1 to election 2, including in raw, regular, ordinary and enriched scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.complete.votes — A matrix of order JxK with the estimated vote transitions from election 1 to election 2, including in raw, regular, ordinary and enriched scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

VTM.prop.units — An array of order JxKxI with the estimated proportions of vote transitions from election 1 to election 2 attained for each unit after adjusting the **lphom()** initial estimate.

VTM.votes.units — An array of order JxKxI with the estimated matrix of vote transitions from election 1 to election 2 attained for each unit after adjusting the **lphom()** initial estimate.

zeros — A list of vectors of length two, indicating the election options for which no transfer of votes are allowed between election 1 and election 2.

`deterministic.bounds`
: A list of two matrices of order JxK and two arrays of order JxKxI containing for each vote transition the lower and upper allowed proportions given the observed aggregates.

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

`origin`
: A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`destination`
: A matrix with the final data used as votes of the origin election after taking into account the level of information available regarding to new entries and exits of the election censuses between the two elections.

`EHet`
: A matrix of order IxK measuring in each spatial unit a distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results with the solution in each territorial unit for each option of election 2.

`solution_init`
: A list with the main outputs produced by **lphom()**.

- `VTM_init`: A matrix of order J'xK' with the estimated percentages of vote transitions from election 1 to election 2 initially obtained by **lphom()**.
- `VTM.votes_init`: A matrix of order J'xK' with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()**.
- `OTM_init`: A matrix of order KxJ with the estimated percentages of the origin of the votes obtained for the different options of election 2 initially obtained by **lphom()**.
- `HETe_init`: The estimated heterogeneity index defined in equation (10) of Romero et al. (2020).
- `EHet_init`: A matrix of order IxK measuring in each spatial unit the distance to the homogeneity hypothesis, that is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results, using the **lphom()** solution, in each territorial unit for each option of election 2.
- `VTM.complete_init`: matrix of order JxK with the estimated proportions of vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in `raw`, `regular`, `ordinary` and `enriched` scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.
- `VTM.complete.votes_init`: A matrix of order JxK with the estimated vote transitions from election 1 to election 2 initially obtained by **lphom()**, including in `raw`, `regular`, `ordinary` and `enriched` scenarios the row and the column corresponding to net_entries and net_exits even when they are really small, less than 1% in all units.

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM, and Romero, R (2022). Improving estimates accuracy of voter transitions. Two new algorithms for ecological inference based on linear programming, *Sociological Methods & Research*. doi:10.1177/00491241221092725.

See Also

`lphom` `nslphom` `lclphom` `rslphom`

Other linear programing ecological inference functions: `lclphom()`, `lp_apriori()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `rslphom()`, `tslphom_dual()`, `tslphom_joint()`

Examples

```
mt.ts <- tslphom(France2017P[, 1:8] , France2017P[, 9:12], new_and_exit_voters= "raw")
mt.ts$VTM
mt.ts$HETe
mt.ts$solution_init$HETe_init
```

`tslphom_dual` *Implements the tslphom_dual algorithm*

Description

Estimates RxC vote transfer matrices (ecological contingency tables) with tslphom_dual

Usage

```
tslphom_dual(
  votes_election1,
  votes_election2,
  integers = FALSE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

`votes_election1`

data.frame (or matrix) of order IxJ with the counts to be initially mapped to rows. When estimating vote transfer matrices, the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. The sum by rows of `votes_election1` and `votes_election2` must coincide.

votes_election2
: data.frame (or matrix) of order IxK with the counts to be initially mapped to columns. When estimating vote transfer matrices, the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.

integers
: A TRUE/FALSE value that indicates whether the problem is solved in integer values in both iterations: zero (lphom) and final (including unit) solutions. If TRUE, the LP matrices are approximated to the closest integer solution solving the corresponding Integer Linear Program. Default, FALSE.

solver
: A character string indicating the linear programming solver to be used, only lp_solve and symphony are allowed. By default, lp_solve. The package Rsymphony needs to be installed for the option symphony to be used.

integers.solver
: A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only symphony and lp_solve are allowed. By default, symphony. The package Rsymphony needs to be installed for the option symphony to be used. Only used when integers = TRUE.

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

Value

A list with the following components

VTM.votes.w
: The matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2, attained weighting the two dual solutions using as weights the corresponding HTEe estimates.

VTM.votes.units.w
: The array of order JxKxI with the local estimated cross-distributions of votes of elections 1 and 2 by unit, attained weighting the two dual solutions using as weights the corresponding HTEe estimates.

VTM.votes.a
: The matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2, attained simple averaging the two dual solutions.

VTM.votes.units.a
: The matrix of order JxKxI with the estimated cross-distributions of votes of elections 1 and 2 by unit, attained weighting the two dual solutions using as weights the corresponding HTEe estimates.

HETe.w
: Estimated heterogeneity index associated to the VTM.votes.w solution.

HETe.a
: Estimated heterogeneity index associated to the VTM.votes.a solution.

VTM12.w
: The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the VTM.votes.w solution.

VTM21.w
: The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the VTM.votes.w solution.

VTM12.a
: The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the VTM.votes.a solution.

`VTM21.a` The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the `VTM.votes.a` solution.

`tslphom.object.12`
The output of the `tslphom` function attained solving the problem X –> Y, that is, mapping `votes_election1` to rows and `votes_election2` to columns.

`tslphom.object.21`
The output of the `tslphom` function attained solving the problem Y –> X, that is, mapping `votes_election2` to rows and `votes_election1` to columns.

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

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM and Romero, R (2024). Symmetry estimating RxC vote transfer matrices from aggregate data. *Journal of the Royal Statistical Society, Series A – Statistics in Society*, forthcoming. doi:10.1093/jrsssa/qnae013

See Also

`tslphom` `lphom_dual` `nslphom_dual` `lphom_joint` `tslphom_joint` `nslphom_joint`

Other linear programing ecological inference functions: `lclphom()`, `lp_apriori()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `rslphom()`, `tslphom_joint()`, `tslphom()`

Examples

```
x <- France2017P[, 1:8]
y <- France2017P[, 9:12]
y[,1] <- y[,1]  - (rowSums(y) - rowSums(x))
mt <- tslphom_dual(x, y)
mt$VTM.votes.w
mt$HETe.w
```

---

tslphom_joint *Implements the tslphom_joint algorithm*

Description

Estimates RxC vote transfer matrices (ecological contingency tables) with tslphom_joint

Usage

```
tslphom_joint(
  votes_election1,
  votes_election2,
  integers = FALSE,
  solver = "lp_solve",
  integers.solver = "symphony",
  ...
)
```

Arguments

votes_election1	data.frame (or matrix) of order IxJ with the counts to be initially mapped to rows. When estimating vote transfer matrices, the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.
votes_election2	data.frame (or matrix) of order IxK with the counts to be initially mapped to columns. When estimating vote transfer matrices, the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. The sum by rows of votes_election1 and votes_election2 must coincide.
integers	A TRUE/FALSE value that indicates whether the problem is solved in integer values in both iterations: zero (lphom) and final (including unit) solutions. If TRUE, the LP matrices are approximated to the closest integer solution solving the corresponding Integer Linear Program. Default, FALSE.
solver	A character string indicating the linear programming solver to be used, only lp_solve and symphony are allowed. By default, lp_solve. The package Rsymphony needs to be installed for the option symphony to be used.
integers.solver	A character string indicating the linear programming solver to be used to approximate to the closest integer solution, only symphony and lp_solve are allowed. By default, symphony. The package Rsymphony needs to be installed for the option symphony to be used. Only used when integers = TRUE.
...	Other arguments to be passed to the function. Not currently used.

Value

A list with the following components

VTM.votes	A matrix of order JxK with the estimated cross-distribution of votes of elections 1 and 2.
HETe	The estimated heterogeneity index associated to the VTM.votes solution.
VTM12	The matrix of order JxK with the estimated row-standardized proportions of vote transitions from election 1 to election 2 associated to the VTM.votes solution.

`VTM21` The matrix of order KxJ with the estimated row-standardized proportions of vote transitions from election 2 to election 1 associated to the `VTM.votes` solution.

`VTM.votes.units`
An array of order JxKxI with the estimated matrix of cross-distributions of votes of elections 1 and 2 attained for each unit after congruently adjusting the **lphom_joint()** initial estimate.

`EHet12` A matrix of order IxK measuring in each unit a distance to the homogeneity hypothesis. That is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results in each territorial unit for each option of election two. The matrix Eik.

`EHet21` A matrix of order IxJ measuring in each unit a distance to the homogeneity hypothesis. That is, the differences under the homogeneity hypothesis between the actual recorded results and the expected results in each territorial unit for each option of election one. The matrix Eij.

`deterministic.bounds`
A list of two matrices of order JxK and two arrays of order JxKxI containing for each vote transition the lower and upper allowed proportions given the observed aggregates.

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

`solution_init` A list with the main outputs produced by **lphom_joint()**.

Author(s)

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

Rafael Romero <rromero@eio.upv.es>

References

Pavia, JM and Romero, R (2024). Symmetry estimating RxC vote transfer matrices from aggregate data. *Journal of the Royal Statistical Society, Series A – Statistics in Society*, forthcoming. doi:10.1093/jrsssa/qnae013

See Also

`tslphom` `lphom_dual` `tslphom_dual` `nslphom_dual` `lphom_joint` `nslphom_joint`

Other linear programing ecological inference functions: `lclphom()`, `lp_apriori()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `rslphom()`, `tslphom_dual()`, `tslphom()`

Examples

```
x <- France2017P[, 1:8]
y <- France2017P[, 9:12]
y[,1] <- y[,1]  - (rowSums(y) - rowSums(x))
mt <- tslphom_joint(x, y)
mt$VTM.votes
mt$HETe
```

Index