# Package 'lphom'

September 9, 2021

**Type** Package

**Title** Ecological Inference by Linear Programming under Homogeneity

**Version** 0.3.0-7

**Description** Provides a bunch of algorithms based on linear programming for estimating RxC ecological contingency tables (or vote transition matrices) using exclusively aggregate results from voting units under the homogeneity hypothesis. References:
Romero, Pavia, Martin and Romero (2020) <doi:10.1080/02664763.2020.1804842>.
Pavia and Romero (2021a) <doi:10.31124/advance.14716638.v1>.
Pavia and Romero (2021b) Symmetry estimating R×C vote transfer matrices from aggregate data.

**License** EPL | file LICENSE

**Encoding** UTF-8

**Imports** stats

**Depends** R (>= 3.5.0), lpSolve, Rsymphony

**Suggests** ggplot2

**LazyData** true

**RoxygenNote** 7.1.1

**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** 2021-09-09 13:50:01 UTC

## R topics documented:

---

`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.

## See Also

`lphom` `error_lphom`

## Examples

```
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 exceed by the EI estimate with a confidence 1 - alpha. By default, 0.10. |
| `show.plot` | TRUE/FALSE. Indicates whether the graphical representation describing 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.

## 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], "raw", NULL, 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 of the first and second rounds of the 2017 French presidential election in the 107 territorial French departments plus 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_NULL2** 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 for Emmanuel Macron in the first-round of 2017 Presidential Election.

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

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

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

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

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

**OTHERS** Number of votes gained at a national level for 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 for Emmanuel Macron in the second-round of 2017 Presidential Election

**LE_PEN2** Number of votes gained at a national level for 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 for Emmanuel Macron in the first-round of 2017 Presidential Election.

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

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

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

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

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

**OTHERS** Number of votes gained at a national level for 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 for Emmanuel Macron in the second-round of 2017 Presidential Election

**LE_PEN2** Number of votes gained at a national level for 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 vote transfer matrices (ecological contingency tables) with lclphom

## Usage

```
lclphom(
  votes_election1,
  votes_election2,
  new_and_exit_voters = c("regular", "raw", "simultaneous", "full", "gold"),
  structural_zeros = NULL,
  iter.max = 1000,
  uniform = TRUE,
  distance.local = c("abs", "max", "none"),
  counts = FALSE,
  solver = "lp_solve",
  verbose = FALSE
)
```

## Arguments

`votes_election1`
: data.frame (or matrix) of order IxJ (likely of final order IxJ-1 in `regular` and `raw` scenarios) with the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered.

`votes_election2`
: data.frame (or matrix) of order IxK (likely of final order IxK-1 in `regular` and `raw` scenarios) with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered.

`new_and_exit_voters`
: A character string indicating the level of information available regarding new entries and exits of the election censuses between the two elections. This argument captures the different options discussed on Section 3 of Romero et al. (2020). This argument admits five values: `regular`, `raw`, `simultaneous`, `full` and `gold`. Default, `regular`.

`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 `"regular"`, lphom implicitly

states structural_zeros = list(c(J, K)) in case exits and/or entries are computed because the sum by rows of `votes_election1` and `votes_election2` does not coincide.

`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.

`uniform` A TRUE/FALSE value that indicates if census exits affects all the electoral options in a (relatively) similar fashion in each voting unit: equation (13) of Pavia and Romero (2021). Default, TRUE.

`distance.local` A string argument that indicates whether the second step of the lphom_local algorithm should be performed in order to resolve potential indeterminacies of local solutions. Default, `"abs"`. If `distance.local = "abs"` lphom_local selects in its the second step the matrix closer to the temporary global solution under L_1 norm, among the first step compatible matrices. If `distance.local = "abs"` lphom_local selects in its the 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.

`counts` A TRUE/FALSE value that indicates whether the problem is solved in integer values (counts) in each iteration, including zero (lphom) and local solutions. Initial LP matrices are approximate 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 `symphony` and `lp_solve` are allowed. By default, `lp_solve`.

`verbose` A TRUE/FALSE value that indicates if the main outputs of the function should be printed on the screen. Default, FALSE.

## Details

Description of the `new_and_exit_voters` argument in more detail.

- `regular`: The default value. This argument accounts for the most plausible scenario. A scenario with two elections elapsed at least some months. In this scenario, (i) the column *J* of `votes_election1` corresponds to new young electors who have the right to vote for the first time and (ii) net exits (basically a consequence of mortality), and eventually net entries, are computed according equation (7) of Romero et al. (2020), and (iii) we assume net exits affect equally all the first *J-1* options of election 1, hence (8) and (9) constraints of Romero et al. (2020) are imposed.
- `raw`: This value accounts for a scenario with two elections where only the raw election data recorded in the *I* territorial units, in which the area under study is divided, are available. In this scenario, net exits (basically deaths) and net entries (basically new young voters) are estimated according to equation (7) of Romero et al. (2020). Constraints defined by equations (8) and (9) of Romero et al. (2020) are imposed. In this scenario, when net exits and/or net entries are negligible (such as between the first- and second-round of French Presidential elections), they are omitted in the outputs.
- `simultaneous`: This value accounts for either a scenario with two simultaneous elections or a classical ecological inference problem. 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.

- `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* of votes_election1 measures new immigrants that have the right to vote; and (iii) the column *K* of votes_election2 corresponds to total exits of the census lists (due to death or emigration). In this scenario, the sum by rows of `votes_election1` and `votes_election2` must agree and constraints (8) and (9) of Romero et al. (2020) are 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. The same restrictions as in the above scenario apply but for both columns *K-1* and *K* of the vote transition probability matrix

## 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.

`VTM.votes` A matrix of order JxK with the estimated 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.

`HETe` The estimated heterogeneity index as defined in equation (15) of Pavia and Romero (2021).

`VTM.complete` A matrix of order J'xK' 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 J'xK' 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.prop.units` An array of order J'xK'xI 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 J'xK'xI 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 J'xK' with the estimated proportions of 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, corresponding to the final iteration

`VTM.sequence` Array of order J'xK'x(iter+1) (where `iter` is the efective number of iterations performed) of the intermediated 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 J'xK'xI 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 J'xK'xI 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 error associated with the solution linked with `VTM_units.iter`.

`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 two.

`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()**.
- `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 two.
- `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.

## Author(s)

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

Rafael Romero <rromero@eio.upv.es>

## See Also

lphom tslphom nslphom

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

## Examples

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

---

# lphom *Implements lphom algorithm*

---

## Description

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

## Usage

```
lphom(
  votes_election1,
  votes_election2,
  new_and_exit_voters = c("regular", "raw", "simultaneous", "full", "gold"),
  structural_zeros = NULL,
  counts = FALSE,
  verbose = FALSE,
  solver = "lp_solve"
)
```

## Arguments

votes_election1
: data.frame (or matrix) of order IxJ (likely of final order IxJ-1 in `regular` and `raw` scenarios) with the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered.

votes_election2
: data.frame (or matrix) of order IxK (likely of final order IxK-1 in `regular` and `raw` scenarios) with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered.

`new_and_exit_voters`
: A character string indicating the level of information available regarding new entries and exits of the election censuses between the two elections. This argument captures the different options discussed on Section 3 of Romero et al. (2020). This argument admits five values: `regular`, `raw`, `simultaneous`, `full` and `gold`. Default, `regular`.

`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 `"regular"`, lphom implicitly states structural_zeros = list(c(J, K)) in case exits and/or entries are computed because the sum by rows of `votes_election1` and `votes_election2` does not coincide.

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

`verbose`
: A TRUE/FALSE value that indicates if the main outputs of the function should be printed on the screen. 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`.

## Details

Description of the `new_and_exit_voters` argument in more detail.

- `regular`: The default value. This argument accounts for the most plausible scenario. A scenario with two elections elapsed at least some months. In this scenario, (i) the column *J* of `votes_election1` corresponds to new young electors who have the right to vote for the first time and (ii) net exits (basically a consequence of mortality), and eventually net entries, are computed according equation (7) of Romero et al. (2020), and (iii) we assume net exits affect equally all the first *J-1* options of election 1, hence (8) and (9) constraints of Romero et al. (2020) are imposed.
- `raw`: This value accounts for a scenario with two elections where only the raw election data recorded in the *I* territorial units, in which the area under study is divided, are available. In this scenario, net exits (basically deaths) and net entries (basically new young voters) are estimated according to equation (7) of Romero et al. (2020). Constraints defined by equations (8) and (9) of Romero et al. (2020) are imposed. In this scenario, when net exits and/or net entries are negligible (such as between the first- and second-round of French Presidential elections), they are omitted in the outputs.
- `simultaneous`: This value accounts for either a scenario with two simultaneous elections or a classical ecological inference problem. 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.
- `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* of votes_election1 measures new immigrants that have the right to vote; and (iii) the column *K* of votes_election2 corresponds to total exits of the census lists (due to death or emigration). In this scenario, the sum by rows of `votes_election1`

and votes_election2 must agree and constraints (8) and (9) of Romero et al. (2020) are 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. The same restrictions as in the above scenario apply but for both columns *K-1* and *K* of the vote transition probability matrix

## 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. |
| VTM.votes | A matrix of order JxK with the estimated 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. |
| HETe | The estimated heterogeneity index defined in equation (11) of Romero et al. (2020). |
| VTM.complete | A matrix of order J'xK' 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 J'xK' 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. |
| 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 two. |

## 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`

Other linear programing ecological inference functions: `lclphom()`, `lphom_dual()`, `lphom_joint()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `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,
  counts = FALSE,
  solver = "lp_solve"
)
```

## Arguments

| | |
|---|---|
| `votes_election1` | data.frame (or matrix) of order IxJ with the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. In general, the counts to be initially mapped to columns. The sum by rows of `votes_election1` and `votes_election2` must coincide. |
| `votes_election2` | data.frame (or matrix) of order IxK with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. In general, the counts to be initially mapped to columns. The sum by rows of `votes_election1` and `votes_election2` must coincide. |
| `counts` | A TRUE/FALSE value that indicates whether the linked LP solution of 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`. |

## 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 (2021). Symmetry estimating R×C vote transfer matrices from aggregate data, mimeo.

## See Also

lphom tslphom_dual nslphom_dual lphom_joint tslphom_joint nslphom_joint

Other linear programing ecological inference functions: lclphom(), lphom_joint(), lphom(), nslphom_dual(), nslphom_joint(), nslphom(), 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,
  counts = FALSE,
  solver = "lp_solve"
)
```

## Arguments

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

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

`counts`
: A TRUE/FALSE value that indicates whether the linked LP solution of 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`.

## 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. |
| 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 (2021). Symmetry estimating R×C vote transfer matrices from aggregate data, mimeo.

## See Also

lphom lphom_dual tslphom_dual nslphom_dual tslphom_joint nslphom_joint

Other linear programing ecological inference functions: lclphom(), lphom_dual(), lphom(), nslphom_dual(), nslphom_joint(), nslphom(), 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
```

`nslphom` *Implements nslphom algorithm*

## Description

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

## Usage

```
nslphom(
  votes_election1,
  votes_election2,
  new_and_exit_voters = c("regular", "raw", "simultaneous", "full", "gold"),
  structural_zeros = NULL,
  iter.max = 10,
  min.first = FALSE,
  uniform = TRUE,
  distance.local = c("abs", "max", "none"),
  counts = FALSE,
  solver = "lp_solve",
  burnin = 0,
  verbose = FALSE,
  tol = 10^-5
)
```

## Arguments

`votes_election1`
: data.frame (or matrix) of order IxJ (likely of final order IxJ-1 in `regular` and `raw` scenarios) with the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered.

`votes_election2`
: data.frame (or matrix) of order IxK (likely of final order IxK-1 in `regular` and `raw` scenarios) with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered.

`new_and_exit_voters`
: A character string indicating the level of information available regarding new entries and exits of the election censuses between the two elections. This argument captures the different options discussed on Section 3 of Romero et al. (2020). This argument admits five values: `regular`, `raw`, `simultaneous`, `full` and `gold`. Default, `regular`.

`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 `"regular"`, lphom implicitly states structural_zeros = list(c(J, K)) in case exits and/or entries are computed

because the sum by rows of votes_election1 and votes_election2 does not coincide.

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 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), burnin = 0 is forced and iter.max is at least 100. Default, FALSE.

uniform — A TRUE/FALSE value that indicates if census exits affects all the electoral options in a (relatively) similar fashion in each voting unit: equation (13) of Pavia and Romero (2021). Default, TRUE.

distance.local — A string argument that indicates whether the second step of the lphom_local algorithm should be performed in order to resolve potential indeterminacies of local solutions. Default, "abs". If distance.local = "abs" lphom_local selects in its the 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 the 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.

counts — A TRUE/FALSE value that indicates whether the problem is solved in integer values (counts) 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.

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

verbose — A TRUE/FALSE value that indicates if the main outputs of the function should be printed on the screen. Default, FALSE.

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 has been reached. By default, 0.00001.

## Details

Description of the new_and_exit_voters argument in more detail.

- regular: The default value. This argument accounts for the most plausible scenario. A scenario with two elections elapsed at least some months. In this scenario, (i) the column *J* of votes_election1 corresponds to new young electors who have the right to vote for the first

time and (ii) net exits (basically a consequence of mortality), and eventually net entries, are computed according equation (7) of Romero et al. (2020), and (iii) we assume net exits affect equally all the first *J-1* options of election 1, hence (8) and (9) constraints of Romero et al. (2020) are imposed.

- `raw`: This value accounts for a scenario with two elections where only the raw election data recorded in the *I* territorial units, in which the area under study is divided, are available. In this scenario, net exits (basically deaths) and net entries (basically new young voters) are estimated according to equation (7) of Romero et al. (2020). Constraints defined by equations (8) and (9) of Romero et al. (2020) are imposed. In this scenario, when net exits and/or net entries are negligible (such as between the first- and second-round of French Presidential elections), they are omitted in the outputs.
- `simultaneous`: This value accounts for either a scenario with two simultaneous elections or a classical ecological inference problem. 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.
- `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* of votes_election1 measures new immigrants that have the right to vote; and (iii) the column *K* of votes_election2 corresponds to total exits of the census lists (due to death or emigration). In this scenario, the sum by rows of `votes_election1` and `votes_election2` must agree and constraints (8) and (9) of Romero et al. (2020) are 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. The same restrictions as in the above scenario apply but for both columns *K-1* and *K* of the vote transition probability matrix

## 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.

`VTM.votes` A matrix of order JxK with the estimated 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.

`HETe` The estimated heterogeneity index as defined in equation (15) of Pavia and Romero (2021).

`VTM.complete` A matrix of order J'xK' 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 J'xK' 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 J'xK'x(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 J'xK'xI 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 J'xK'xI 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.

`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 two.

`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()**.
- `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 two.
- `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.

## Author(s)

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

Rafael Romero <rromero@eio.upv.es>

## References

Pavia, JM, and Romero, R (2021). Improving estimates accuracy of voter transitions. Two new algorithms for ecological inference based on linear programming. doi: 10.31124/advance.14716638.v1.

## See Also

lphom tslphom lclphom

Other linear programing ecological inference functions: lclphom(), lphom_dual(), lphom_joint(), lphom(), nslphom_dual(), nslphom_joint(), 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,
  counts = FALSE,
  solver = "lp_solve",
  tol = 10^-5
)
```

## Arguments

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

`votes_election2`
: data.frame (or matrix) of order IxK with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. In general, the counts to be initially mapped to columns. 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.

`counts`
: A TRUE/FALSE value that indicates whether the problem is solved in integer values (counts) 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`.

`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 has been reached. By default, 0.00001.

## 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 (2021). Symmetry estimating R×C vote transfer matrices from aggregate data, mimeo.

## See Also

nslphom lphom_dual tslphom_dual lphom_joint tslphom_joint nslphom_joint

Other linear programing ecological inference functions: lclphom(), lphom_dual(), lphom_joint(), lphom(), nslphom_joint(), nslphom(), 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,
  counts = FALSE,
  solver = "lp_solve",
  tol = 0.001
)
```

## Arguments

| | |
|---|---|
| votes_election1 | data.frame (or matrix) of order IxJ with the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. In general, the counts to be initially mapped to columns. The sum by rows of votes_election1 and votes_election2 must coincide. |
| votes_election2 | data.frame (or matrix) of order IxK with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. In general, the counts to be initially mapped to columns. 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. |
| counts | A TRUE/FALSE value that indicates whether the problem is solved in integer values (counts) 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`.

`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 has been reached. By default, 0.001.

## 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.

`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 (2021). Symmetry estimating R×C vote transfer matrices from aggregate data, mimeo.

## See Also

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

Other linear programing ecological inference functions: `lclphom()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom()`, `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)
mt$VTM.votes
mt$HETe
```

---

`tslphom` *Implements tslphom algorithm*

---

## Description

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

## Usage

```
tslphom(
  votes_election1,
  votes_election2,
  new_and_exit_voters = c("regular", "raw", "simultaneous", "full", "gold"),
  structural_zeros = NULL,
  uniform = TRUE,
  distance.local = c("abs", "max", "none"),
  counts = FALSE,
  verbose = FALSE,
  solver = "lp_solve"
)
```

## Arguments

`votes_election1`
: data.frame (or matrix) of order IxJ (likely of final order IxJ-1 in `regular` and `raw` scenarios) with the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered.

`votes_election2`
: data.frame (or matrix) of order IxK (likely of final order IxK-1 in `regular` and `raw` scenarios) with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered.

`new_and_exit_voters`
: A character string indicating the level of information available regarding new entries and exits of the election censuses between the two elections. This argument captures the different options discussed on Section 3 of Romero et al. (2020). This argument admits five values: `regular`, `raw`, `simultaneous`, `full` and `gold`. Default, `regular`.

`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 `"regular"`, lphom implicitly states structural_zeros = list(c(J, K)) in case exits and/or entries are computed because the sum by rows of `votes_election1` and `votes_election2` does not coincide.

`uniform`
: A TRUE/FALSE value that indicates if census exits affects all the electoral options in a (relatively) similar fashion in each voting unit: equation (13) of Pavia and Romero (2021). Default, TRUE.

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

`counts`
: A TRUE/FALSE value that indicates whether the problem is solved in integer values (counts), 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.

`verbose`
: A TRUE/FALSE value that indicates if the main outputs of the function should be printed on the screen. 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`.

## Details

Description of the `new_and_exit_voters` argument in more detail.

- `regular`: The default value. This argument accounts for the most plausible scenario. A scenario with two elections elapsed at least some months. In this scenario, (i) the column *J* of `votes_election1` corresponds to new young electors who have the right to vote for the first time and (ii) net exits (basically a consequence of mortality), and eventually net entries, are computed according equation (7) of Romero et al. (2020), and (iii) we assume net exits affect equally all the first *J-1* options of election 1, hence (8) and (9) constraints of Romero et al. (2020) are imposed.
- `raw`: This value accounts for a scenario with two elections where only the raw election data recorded in the *I* territorial units, in which the area under study is divided, are available. In this scenario, net exits (basically deaths) and net entries (basically new young voters) are estimated according to equation (7) of Romero et al. (2020). Constraints defined by equations (8) and (9) of Romero et al. (2020) are imposed. In this scenario, when net exits and/or net entries are negligible (such as between the first- and second-round of French Presidential elections), they are omitted in the outputs.
- `simultaneous`: This value accounts for either a scenario with two simultaneous elections or a classical ecological inference problem. 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.

- 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* of votes_election1 measures new immigrants that have the right to vote; and (iii) the column *K* of votes_election2 corresponds to total exits of the census lists (due to death or emigration). In this scenario, the sum by rows of votes_election1 and votes_election2 must agree and constraints (8) and (9) of Romero et al. (2020) are 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. The same restrictions as in the above scenario apply but for both columns *K-1* and *K* of the vote transition probability matrix

## 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. |
| VTM.votes | A matrix of order JxK with the estimated 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. |
| HETe | The estimated heterogeneity index as defined in equation (15) of Pavia and Romero (2021). |
| VTM.complete | A matrix of order J'xK' 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 J'xK' 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.prop.units | An array of order J'xK'xI 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 J'xK'xI 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. |
| 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 two.

`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()**.
- `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 two.
- `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.

## Author(s)

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

Rafael Romero <rromero@eio.upv.es>

## References

Pavia, JM, and Romero, R (2021). Improving estimates accuracy of voter transitions. Two new algorithms for ecological inference based on linear programming. doi: 10.31124/advance.14716638.v1.

## See Also

`lphom` `nslphom` `lclphom`

Other linear programing ecological inference functions: `lclphom()`, `lphom_dual()`, `lphom_joint()`, `lphom()`, `nslphom_dual()`, `nslphom_joint()`, `nslphom()`, `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,
  counts = FALSE,
  solver = "lp_solve"
)
```

## Arguments

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

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

`counts`
: A TRUE/FALSE value that indicates whether the problem is solved in integer values (counts), 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`.

## 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 (2021). Symmetry estimating R×C vote transfer matrices from aggregate data, mimeo.

## See Also

tslphom lphom_dual nslphom_dual lphom_joint tslphom_joint nslphom_joint

Other linear programing ecological inference functions: lclphom(), lphom_dual(), lphom_joint(), lphom(), nslphom_dual(), nslphom_joint(), nslphom(), 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,
  counts = FALSE,
  solver = "lp_solve"
)
```

## Arguments

| | |
|---|---|
| votes_election1 | data.frame (or matrix) of order IxJ with the votes gained by the *J* political options competing on election 1 (or origin) in the *I* territorial units considered. In general, the counts to be initially mapped to columns. The sum by rows of votes_election1 and votes_election2 must coincide. |
| votes_election2 | data.frame (or matrix) of order IxK with the votes gained by the *K* political options competing on election 2 (or destination) in the *I* territorial units considered. In general, the counts to be initially mapped to columns. The sum by rows of votes_election1 and votes_election2 must coincide. |
| counts | A TRUE/FALSE value that indicates whether the problem is solved in integer values (counts), 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`.

## 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.

`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 (2021). Symmetry estimating R×C vote transfer matrices from aggregate data, mimeo.

## See Also

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

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