Package ‘migest’

October 5, 2019

Type Package

Title Methods for the Indirect Estimation of Bilateral Migration

Version 1.8.1

Maintainer Guy J. Abel <g.j.abel@gmail.com>

Description Indirect methods for estimating bilateral migration flows in the presence of partial or missing data, including the estimation of bilateral migration flows from changes in bilateral migrant stock tables (e.g. Abel (2013) <doi:10.4054/DemRes.2013.28.18>).

URL https://github.com/gjabel/migest/

BugReports https://github.com/gjabel/migest/issues

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports mipfp, dplyr, purrr, tidyr, stringr, magrittr, stats, utils

NeedsCompilation no

Author Guy J. Abel [aut, cre] (<https://orcid.org/0000-0002-4893-5687>)

Repository CRAN

Date/Publication 2019-10-05 14:50:02 UTC

R topics documented:

`migest-package` *Methods for the Indirect Estimation of Bilateral Migration*

Description

The migest package contains a collection of R functions for indirect methods to estimate bilateral migration flows in the presence of partial or missing data. Methods might be relevant to other categorical data situations on non-migration data, where for example, marginal totals are known and only auxiliary bilateral data is available.

Details

Package:	migest
Type:	Package
License:	GPL-2

The estimation methods in this package can be grouped as 1) functions for origin-destination matrices (`cm2` and `ipf2`) and 2) functions for origin-destination matrices categorized by a further set of characteristics, such as ethnicity, employment or health status (`cm3`, `ipf3` and `ipf3_qi`). Each of these routines are based on indirect estimation methods where marginal totals are known, and a

Poisson regression (log-linear) model is assumed.

The flow from stock functions, `ffs_demo` is a wrapper for a combination of some of these estimation routines with further adjustments for changes in foreign born stocks over a period. The demo files, `demo(cfplot_reg2)`, `demo(cfplot_reg)` and `demo(cfplot_nat)`, produce circular migration flow plots for migration estimates from Abel(2017) and Abel and Sander (2014), which were derived using the `ffs_demo` function.

Blog posts with some additional details of the implementation of functions in the package can be found at http://gjabel.wordpress.com/category/r/migest/

Github repo: http://github.com/gjabel/migest

Author(s)

Guy J. Abel

References

Abel, G. J. (2018). Estimates of Global Bilateral Migration Flows by Gender between 1960 and 2015. *International Migration Review*.

Abel, G. J. (2013). Estimating Global Migration Flow Tables Using Place of Birth. *Demographic Research* 28, (18) 505-546

Abel, G. J. (2005) *The Indirect Estimation of Elderly Migrant Flows in England and Wales* (MS.c. Thesis). University of Southampton

Abel, G. J. and Sander, N. (2014). Quantifying Global International Migration Flows. *Science*, 343 (6178) 1520-1522

Raymer, J., G. J. Abel, and P. W. F. Smith (2007). Combining census and registration data to estimate detailed elderly migration flows in England and Wales. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170 (4), 891–908.

Willekens, F. (1999). Modelling Approaches to the Indirect Estimation of Migration Flows: From Entropy to EM. *Mathematical Population Studies* 7 (3), 239–78.

---

birth_mat *Calculate Births for Each Element of Place of Birth - Place of Residence Stock Matrix*

Description

This function is predominantly intended to be used within the ffs routines in the migest package.

Usage

```
birth_mat(b_por = NULL, m2 = NULL, non_negative = TRUE)
```

Arguments

`b_por` Vector of numberic values for births in each place of residence

`m2` Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1.

`non_negative` Adjust birth matrix calculation to ensure all deductions from `m2` will result in positive population counts. On rare occasions when working with international stock data the number of births can exceed the increase in the number of native born population.

Value

Matrix of place of birth by place of residence for new-born's

Author(s)

Guy J. Abel

See Also

`ipf3_qi`, `ffs_diff`

`block_matrix` *Create a Block Matrix with Non-Uniform Block Sizes.*

Description

Creates a `matrix` with differing size blocks

Usage

```
block_matrix(x = NULL, b = NULL, byrow = FALSE, dimnames = NULL)
```

Arguments

`x` Vector of numbers to identify each block.

`b` Numeric value for the size of the blocks within the matrix ordered depending on `byrow`

`byrow` Logical value. If `FALSE` (the default) the blocks are filled by columns, otherwise the blocks in the matrix are filled by rows.

`dimnames` Character string of name attribute for the basis of the block matrix. If `NULL` a vector of the same length of `b` provides the basis of row and column names.#'

Value

Returns a `matrix` with block sizes determined by the `b` argument. Each block is filled with the same value taken from `x`.

Author(s)

Guy J. Abel

See Also

stripe_matrix, block_sum, ipf2_block

Examples

```
block_matrix(x = 1:16, b = c(2,3,4,2))
block_matrix(x = 1:25, b = c(2,3,4,2,1))
```

---

block_sum   *Sum of Selected Block in a Block Matrix*

Description

Returns of a sum of a block within a matrix. This function is predominantly intended to be used within the ipf2_block routine.

Usage

```
block_sum(block = NULL, m = NULL, block_id = NULL)
```

Arguments

block	Numeric value of block to summed. To be matched against the matrix in block_id.
m	Matrix of all blocks combined.
block_id	Matrix of the same dimensions of m used to identify blocks.

Value

Returns a numeric value of the sum of a single block.

Author(s)

Guy J. Abel

See Also

block_matrix, stripe_matrix, ipf2_block

Examples

```
m <- matrix(data = 100:220, nrow = 11, ncol = 11)
b <- block_matrix(x = 1:16, b = c(2, 3, 4, 2))
block_sum(block = 1, m = m, block_id = b)
block_sum(block = 4, m = m, block_id = b)
block_sum(block = 16, m = m, block_id = b)
```

cm2 *Conditional Maximization Routine for the Indirect Estimation of Origin-Destination Migration Flow Table with Known Margins.*

Description

The cm2 function finds the maximum likelihood estimates for parameters in the log-linear model:

$$\log y_{ij} = \log \alpha_i + \log \beta_j + \log m_{ij}$$

as introduced by Willekens (1999). The α_i and β_j represent background information related to the characteristics of the origin and destinations respectively. The m_{ij} factor represents auxiliary information on migration flows, which imposes its interaction structure onto the estimated flow matrix.

Usage

```
cm2(row_tot = NULL, col_tot = NULL, m = matrix(data = 1, nrow =
  length(row_tot), ncol = length(col_tot)), tol = 1e-06, maxit = 500,
  verbose = TRUE, rtot = row_tot, ctot = col_tot)
```

Arguments

`row_tot` Vector of origin totals to constrain the sum of the imputed cell rows.

`col_tot` Vector of destination totals to constrain the sum of the imputed cell columns.

`m` Matrix of auxiliary data. By default set to 1 for all origin-destination combinations.

`tol` Numeric value for the tolerance level used in the parameter estimation.

`maxit` Numeric value for the maximum number of iterations used in the parameter estimation.

`verbose` Logical value to indicate the print the parameter estimates at each iteration. By default `FALSE`.

`rtot` Depreciated. Use `row_tot`

`ctot` Depreciated. Use `col_tot`

Value

Parameter estimates are obtained using the EM algorithm outlined in Willekens (1999). This is equivalent to a conditional maximization of the likelihood, as discussed by Raymer et. al. (2007). It also provides identical indirect estimates to those obtained from the `ipf2` routine.

The user must ensure that the row and column totals are equal in sum. Care must also be taken to allow the dimension of the auxiliary matrix (`m`) to equal those provided in the row (`row_tot`) and column (`col_tot`) arguments.

Returns a `list` object with

`N` Origin-Destination matrix of indirect estimates

`theta` Collection of parameter estimates

Author(s)

Guy J. Abel

References

Raymer, J., G. J. Abel, and P. W. F. Smith (2007). Combining census and registration data to estimate detailed elderly migration flows in England and Wales. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170 (4), 891–908.

Willekens, F. (1999). Modelling Approaches to the Indirect Estimation of Migration Flows: From Entropy to EM. *Mathematical Population Studies* 7 (3), 239–78.

See Also

ipf2

Examples

```
## with Willekens (1999) data
dn <- LETTERS[1:2]
y <- cm2(row_tot = c(18, 20), col_tot = c(16, 22),
         m = matrix(c(5, 1, 2, 7), ncol = 2, dimnames = list(orig = dn, dest = dn)))
y

## with all elements of offset equal (independence fit)
y <- cm2(row_tot = c(18, 20), col_tot = c(16, 22))
y

## with bigger matrix
dn <- LETTERS[1:4]
y <- cm2(row_tot = c(250, 100, 140, 110), col_tot = c(150, 150, 180, 120),
        m = matrix(data = c(0, 100, 30, 70, 50, 0, 45, 5, 60, 35, 0, 40, 20, 25, 20, 0),
                 nrow = 4, ncol = 4, dimnames = list(orig = dn, dest = dn), byrow = TRUE))

# display with row and col totals
round(addmargins(y$n))
```

cm3 — *Conditional Maximization Routine for the Indirect Estimation of Origin-Destination-Migrant Type Migration Flow Tables with Known Origin and Destination Margins.*

Description

The cm3 function finds the maximum likelihood estimates for parameters in the log-linear model:

$$\log y_{ijk} = \log \alpha_i + \log \beta_j + \log m_{ijk}$$

as introduced by Abel (2005). The α_i and β_j represent background information related to the characteristics of the origin and destinations respectively. The m_{ijk} factor represents auxiliary information on origin-destination migration flows by a migrant characteristic (such as age, sex, disability, household type, economic status, etc.). This method is useful for combining data from detailed data collection processes (such as a Census) with more up-to-date information on migration inflows and outflows (where details on movements by migrant characteristics are not known).

Usage

```
cm3(row_tot = NULL, col_tot = NULL, m = NULL, tol = 1e-06,
  maxit = 500, verbose = TRUE)
```

Arguments

`row_tot`	Vector of origin totals to constrain the sum of the imputed cell rows.
`col_tot`	Vector of destination totals to constrain the sum of the imputed cell columns.
`m`	Array of auxiliary data. By default set to 1 for all origin-destination-migrant typology combinations.
`tol`	Numeric value for the tolerance level used in the parameter estimation.
`maxit`	Numeric value for the maximum number of iterations used in the parameter estimation.
`verbose`	Logical value to indicate the print the parameter estimates at each iteration. By default `FALSE`.

Value

Parameter estimates were obtained using the conditional maximization of the likelihood, as discussed by Abel (2005) and Raymer et. al. (2007).

The user must ensure that the row and column totals are equal in sum. Care must also be taken to allow the row and column dimension of the auxiliary matrix (`m`) to equal those provided in the row and column totals.

Returns a `list` object with

`N`	Origin-Destination matrix of indirect estimates
`theta`	Collection of parameter estimates

Author(s)

Guy J. Abel

References

Abel, G. J. (2005) *The Indirect Estimation of Elderly Migrant Flows in England and Wales* (MS.c. Thesis). University of Southampton

Raymer, J., G. J. Abel, and P. W. F. Smith (2007). Combining census and registration data to estimate detailed elderly migration flows in England and Wales. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170 (4), 891–908.

See Also

cm2, ipf3

Examples

```
## over two tables
dn <- LETTERS[1:2]
y <- cm3(row_tot = c(18, 20) * 2, col_tot = c(16, 22) * 2,
         m = array(c(5, 1, 2, 7, 4, 2, 5, 9), dim = c(2, 2, 2),
                   dimnames = list(orig = dn, dest = dn, type = c("ILL", "HEALTHY"))))
# display with row, col and table totals
y

## over three tables
y <- cm3(row_tot = c(170, 120, 410), col_tot = c(500, 140, 60),
         m = array(c(5, 1, 2, 7,  4, 2, 5, 9,  5, 4, 3, 1), dim = c(2, 2, 3),
                dimnames = list(orig = dn, dest = dn, type = c("0--15", "15-60", ">60"))),
                   verbose = FALSE)
# display with row, col and table totals
y
```

cm_net *Conditional Maximisation Routine for the Indirect Estimation of Origin-Destination-Type Migration Flow Tables with Known Net Migration Totals.*

Description

The cm_net function finds the maximum likelihood estimates for fitted values in the log-linear model:

$$\log y_{ij} = \log \alpha_i + \log \alpha_i^{-1} + \log m_{ij}$$

Usage

```
cm_net(net_tot = NULL, m = NULL, tol = 1e-06, maxit = 500,
  verbose = TRUE)
```

Arguments

net_tot	Vector of net migration totals to constrain the sum of the imputed cell columns. Elements must sum to zero.
m	Array of auxiliary data. By default set to 1 for all origin-destination-migrant typologies combinations.
tol	Numeric value for the tolerance level used in the parameter estimation.
maxit	Numeric value for the maximum number of iterations used in the parameter estimation.
verbose	Logical value to indicate the print the parameter estimates at each iteration. By default FALSE.

Value

Conditional maximisation routine set up using the partial likelihood derivatives. The argument `net_tot` takes the known net migration totals. The user must ensure that the net migration totals sum globally to zero.

Returns a `list` object with

`mu`	Array of indirect estimates of origin-destination matrices by migrant characteristic
`it`	Iteration count
`tol`	Tolerance level at final iteration

Author(s)

Guy J. Abel

Examples

```
m <- matrix(data = 1:16, nrow = 4)
# m[lower.tri(m)] <- t(m)[lower.tri(m)]
addmargins(m)
sum_net(m)

y <- cm_net(net_tot = c(30, 40, -15, -55), m = m)
addmargins(y$n)
sum_net(y$n)

dn <- LETTERS[1:4]
m <- matrix(data = c(0, 100, 30, 70, 50, 0, 45, 5, 60, 35, 0, 40, 20, 25, 20, 0),
            nrow = 4, ncol = 4,
            dimnames = list(orig = dn, dest = dn), byrow = TRUE)
addmargins(m)
sum_net(m)

y <- cm_net(net_tot = c(-100, 125, -75, 50), m = m)
addmargins(y$n)
sum_net(y$n)
```

`death_mat` *Calculate Deaths for Each Element of Place of Birth - Place of Residence Stock Matrix*

Description

This function is predominantly intended to be used within the ffs routines in the migest package.

Usage

```
death_mat(d_por = NULL, m1 = NULL, method = "proportion",
  m2 = NULL, b_por = NULL)
```

Arguments

`d_por` Vector of numberic values for deaths in each place of residence.

`m1` Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1. Used to distribute deaths proportionally to each migrant stock population. For use when `method = "accounting"`

`method` Character string of either `"proportion"` or `"accounting"` to choose method to distrubte deaths. The `"proportion"` method assumes the mortality rate in each place of birth sub-group (native born and all foreign born stocks) is the same. The `"accounting"` method ensures that the the deaths by place of birth matches that implied by demographic accounting. Still needs to be explored fully.

`m2` Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1. Used to distribute deaths proportionally to each migrant stock population.

`b_por` Vector of numberic values for births in each place of residence. For use when `method = "accounting"`.

Value

Matrix of place of death by place of residence

Author(s)

Guy J. Abel

See Also

`ipf3_qi`, `ffs_diff`

`ffs_demo` *Estimation of Bilateral Migrant Flows from Bilateral Migrant Stocks Using Demographic Accounting Approaches*

Description

Estimates migrant transitions flows between two sequential migrant stock tables. Replaces old `ffs`.

Usage

```
ffs_demo(m1 = NULL, m2 = NULL, b_por = NULL, d_por = NULL,
  m = NULL, stayer_assumption = TRUE,
  match_pob_tot_method = "rescale", birth_non_negative = TRUE,
  death_method = "proportion", ...)
```

Arguments

`m1` Matrix of migrant stock totals at time *t*. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*

`m2` Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1.

`b_por` Vector of the number of births between time *t* and *t*+1 in each region.

`d_por` Vector of the number of deaths between time *t* and *t*+1 in each region.

`m` Matrix of auxiliary data. By default set to 1 for all origin-destination combinations.

`stayer_assumption`
Logical value to indicate wheather to use `ipf3` or `ipf3_qi` to estimate flows. By default uses `ipf3_qi`, i.e. is set to `TRUE`. The `ipf` function is useful for replicating method of Azoze and Raferty.

`match_pob_tot_method`
Character string passed to `method` argument in `match_pob_tot` to ensure place of birth margins in stock tables match.

`birth_non_negative`
Logical value passed to `non_negative` argument in `birth_mat`.

`death_method` Character string passed to `method` argument in `death_mat`.

`...` Additional arguments passes to `ipf3_qi` or `ipf3`.

Value

Estimates migrant transitions flows between two sequential migrant stock tables using various methods. See the example section for possible variations on estimation methods.

Returns a `list` object with:

`mu` Array of indirect estimates of origin-destination matrices by place of birth.

`it` Iteration count.

`tol` Tolerance level at final iteration.

`y` Array of indirect estimates of origin-destination matrices by place of birth with additional rows and columns for births, deaths and moves to other regions.

`...` Slots to record which estimation method was used (as set by arguments above)

`od_flow` Matrix of estimated origin-destination flows

Author(s)

Guy J. Abel

References

Abel, G. J. (2018). Estimates of Global Bilateral Migration Flows by Gender between 1960 and 2015. *International Migration Review* Forthcoming.

Abel, G. J. and Sander, N. (2014). Quantifying Global International Migration Flows. *Science*, 343 (6178) 1520-1522

Abel, G. J. (2013). Estimating Global Migration Flow Tables Using Place of Birth. *Demographic Research* 28, (18) 505-546

See Also

ipf3_qi, ffs_diff, ffs_rates

Examples

```
##
## without births and deaths over period
##
# data as in papers
s1 <- matrix(data = c(1000, 100, 10, 0, 55, 555, 50, 5, 80, 40, 800, 40, 20, 25, 20, 200),
             nrow = 4, ncol = 4, byrow = TRUE)
s2 <- matrix(data = c(950, 100, 60, 0, 80, 505, 75, 5, 90, 30, 800, 40, 40, 45, 0, 180),
             nrow = 4, ncol = 4, byrow = TRUE)
b <- d <- rep(0, 4)
reg <- LETTERS[1:4]
dimnames(s1) <- dimnames(s2) <- list(pob = reg, por = reg)
names(b) <- names(d) <- reg
s1; s2; b; d

# demographic research and science paper example
ffs_demo(m1 = s1, m2 = s2, b_por = b, d_por = d)

# international migration review paper example
s1[,] <- c(100, 20, 10, 20, 10, 55, 40, 25, 10, 25, 140, 20, 0, 10, 65, 200)
s2[,] <- c(70, 25, 10, 40, 30, 60, 55, 45, 10, 10, 140, 0, 10, 15, 50, 180)
ffs_demo(m1 = s1, m2 = s2, b_por = b, d_por = d)

# international migration review supp. material example
dm <- matrix(data = c(0, 5, 50, 500, 5, 0, 45, 495, 50, 45, 0, 450, 500, 495, 450, 0),
             nrow = 4, ncol = 4, byrow = TRUE)
dimnames(dm) <- list(orig = reg, dest = reg)
ffs_demo(m1 = s1, m2 = s2, b_por = b, d_por = d, m = dm)

##
## with births and deaths over period
##
# demographic research paper example
s1[,] <- c(1000, 55, 80, 20, 100, 555, 40, 25, 10, 50, 800, 20, 0, 5, 40, 200)
s2[,] <- c(1060, 45, 70, 30, 60, 540, 75, 30, 10, 40, 770, 20, 10, 0, 70, 230)
b[] <- c(80, 20, 40, 60)
d[] <- c(70, 30, 50, 10)
ffs_demo(m1 = s1, m2 = s2, b_por = b, d_por = d, match_pob_tot_method = "open-dr")
# makes more sense to use this method
ffs_demo(m1 = s1, m2 = s2, b_por = b, d_por = d, match_pob_tot_method = "open")

# science paper  supp. material example
b[] <- c(80, 20, 60, 60)
ffs_demo(m1 = s1, m2 = s2, b_por = b, d_por = d)

# international migration review supp. material example
s1[,] <- c(100, 20, 10, 20, 10, 55, 40, 25, 10, 25, 140, 20, 0, 10, 65, 200)
s2[,] <- c(75, 20, 30, 30, 25, 45, 40, 30, 5, 30, 150, 20, 0, 15, 60, 230)
```

```
b[] <- c(10, 50, 25, 60)
d[] <- c(30, 10, 40, 10)
ffs_demo(m1 = s1, m2 = s2, b_por = b, d_por = d)
```

ffs_diff *Estimation of Bilateral Migrant Flows from Bilateral Migrant Stocks Using Stock Differencing Approaches*

Description

Estimates migrant transitions flows between two sequential migrant stock tables using differencing approaches commonly used by economists.

Usage

```
ffs_diff(m1, m2, decrease = "return", include_native_born = FALSE)
```

Arguments

m1	Matrix of migrant stock totals at time *t*. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*
m2	Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1.
decrease	How to treat decreases in bilateral stocks over the *t* to *t*+1 period (so as to avoid a negative bilateral flow estimates). See details for possible options. Defualt is return
include_native_born	Logcial value to indicate wheather to include diagonal elements of m1 and m2. Default of FALSE - not include.

Value

Estimates migrant transitions flows between two sequential migrant stock tables.

When decrease = "zero" all decreases in migrant stocks over there period are set to zero, following the approach of Bertoli and Fernandez-Huertas Moraga (2015)

When decrease = "return" all decreases in migrant stocks are assumed to correspond to return flows back to their place of birth, following the approach of Beine and Parsons (2015) #' @references Beine, Michel, Simone Bertoli, and Jesús Fernández-Huertas Moraga. (2016). A Practitioners' Guide to Gravity Models of International Migration. *The World Economy* 39(4):496–512.

Author(s)

Guy J. Abel

See Also

ffs_demo, ffs_rates

Examples

```
s1 <- matrix(data = c(100, 10, 10, 0, 20, 55, 25, 10, 10, 40, 140, 65, 20, 25, 20, 200),
             nrow = 4, ncol = 4, byrow = TRUE)
s2 <- matrix(data = c(75, 25, 5, 15, 20, 45, 30, 15, 30, 40, 150, 35, 10, 50, 5, 200),
             nrow = 4, ncol = 4, byrow = TRUE)
reg <- LETTERS[1:4]
dimnames(s1) <- dimnames(s2) <- list(pob = reg, por = reg)
s1; s2

ffs_diff(m1 = s1, m2 = s2, decrease = "zero")
ffs_diff(m1 = s1, m2 = s2, decrease = "return")
```

`ffs_rates` *Estimation of Bilateral Migrant Flows from Bilateral Migrant Stocks Using Rates Approaches*

Description

Estimates migrant transitions flows between two sequential migrant stock tables using approached based on rates.

Usage

```
ffs_rates(m1 = NULL, m2 = NULL, M = NULL, method = "dennett")
```

Arguments

`m1`	Matrix of migrant stock totals at time *t*. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*
`m2`	Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1.
`M`	Numeric value for the global sum of migration flows, used for `dennett` approach.
`method`	Method to estimate flows. Can take values `dennett` or `rogers-von-rabenau`. See detials section for more information. Uses `dennett` as default.

Value

Estimates migrant transitions flows based on migration rates.

When `method = "dennett"` migration are derived from the matrix supplied to `m1`. Dennett uses bilateral migrant stocks at begining of period. Rates then multiplied by global migration flows supplied in `M`.

When `method = "rogers-von-rabenau"` a matrix of growth rates are derived from the changes in inital poplations stock `m1` to obtain `m2`;

$$P^{t+1} = gP^t$$

and then multiplied by the corresponding populations at risk in `m1`. Can result in negative flows.

#' @references Dennett, A. (2015). Estimating an Annual Time Series of Global Migration Flows - An Alternative Methodology for Using Migrant Stock Data. *Global Dynamics: Approaches from Complexity Science*, 125–142. https://doi.org/10.1002/9781118937464.ch7

Rogers, A., & Von Rabenau, B. (1971). Estimation of interregional migration streams from place-of-birth-by-residence data. *Demography*, 8(2), 185–194.

Author(s)

Guy J. Abel

See Also

`ffs_demo`, `ffs_rates`

Examples

```
s1 <- matrix(data = c(100, 10, 10, 0, 20, 55, 25, 10, 10, 40, 140, 65, 20, 25, 20, 200),
             nrow = 4, ncol = 4, byrow = TRUE)
s2 <- matrix(data = c(75, 25, 5, 15, 20, 45, 30, 15, 30, 40, 150, 35, 10, 50, 5, 200),
             nrow = 4, ncol = 4, byrow = TRUE)
reg <- LETTERS[1:4]
dimnames(s1) <- dimnames(s2) <- list(pob = reg, por = reg)
s1; s2

# calculate total migration flows for dennett approach
n <- colSums(s2) - colSums(s1)

ffs_rates(m1 = s1, M =  sum(abs(n)), method = "dennett" )
ffs_rates(m1 = s1, m2 = s2, method = "rogers-von-rabenau" )
```

ipf2 — *Iterative Proportional Fitting Routine for the Indirect Estimation of Origin-Destination Migration Flow Table with Known Margins.*

Description

The `ipf2` function finds the maximum likelihood estimates for fitted values in the log-linear model:

$$\log y_{ij} = \log \alpha_i + \log \beta_j + \log m_{ij}$$

where m_{ij} is a set of prior estimates for y_{ij} and itself is no more complex than the one being fitted.

Usage

```
ipf2(row_tot = NULL, col_tot = NULL, m = matrix(1, length(row_tot),
  length(col_tot)), tol = 1e-05, maxit = 500, verbose = FALSE)
```

Arguments

`row_tot` Vector of origin totals to constrain the sum of the imputed cell rows.

`col_tot` Vector of destination totals to constrain the sum of the imputed cell columns.

`m` Matrix of auxiliary data. By default set to 1 for all origin-destination combinations.

`tol` Numeric value for the tolerance level used in the parameter estimation.

`maxit` Numeric value for the maximum number of iterations used in the parameter estimation.

`verbose` Logical value to indicate the print the parameter estimates at each iteration. By default `FALSE`.

Value

Iterative Proportional Fitting routine set up in a similar manner to Agresti (2002, p.343). This is equivalent to a conditional maximization of the likelihood, as discussed by Willekens (1999), and hence provides identical indirect estimates to those obtained from the `cm2` routine.

The user must ensure that the row and column totals are equal in sum. Care must also be taken to allow the dimension of the auxiliary matrix (`m`) to equal those provided in the row and column totals.

If only one of the margins is known, the function can still be run. The indirect estimates will correspond to the log-linear model without the α_i term if (`row_tot = NULL`) or without the β_j term if (`col_tot = NULL`)

Returns a `list` object with

`mu` Origin-Destination matrix of indirect estimates

`it` Iteration count

`tol` Tolerance level at final iteration

Author(s)

Guy J. Abel

References

Agresti, A. (2002). *Categorical Data Analysis* 2nd edition. Wiley.

Willekens, F. (1999). Modelling Approaches to the Indirect Estimation of Migration Flows: From Entropy to EM. *Mathematical Population Studies* 7 (3), 239–78.

See Also

`cm2`, `ipf3`

Examples

```
## with Willekens (1999) data
dn <- LETTERS[1:2]
y <- ipf2(row_tot = c(18, 20), col_tot = c(16, 22),
          m = matrix(c(5, 1, 2, 7), ncol = 2,
                     dimnames = list(orig = dn, dest = dn)))
round(addmargins(y$mu),2)

## with all elements of offset equal
y <- ipf2(row_tot = c(18, 20), col_tot = c(16, 22))
round(addmargins(y$mu),2)

## with bigger matrix
dn <- LETTERS[1:3]
y <- ipf2(row_tot = c(170, 120, 410), col_tot = c(500, 140, 60),
          m = matrix(c(50, 10, 220, 120, 120, 30, 545, 0, 10), ncol = 3,
                     dimnames = list(orig = dn, dest = dn)))
# display with row and col totals
round(addmargins(y$mu))

## only one margin known
dn <- LETTERS[1:2]
y <- ipf2(row_tot = c(18, 20), col_tot = NULL,
          m = matrix(c(5, 1, 2, 7), ncol = 2,
                     dimnames = list(orig = dn, dest = dn)))
round(addmargins(y$mu))
```

ipf2_block

Iterative Proportional Fitting Routine for the Indirect Estimation of Origin-Destination-Type Migration Flow Tables with Known Origin and Destination Margins and Block Diagonal Elements.

Description

The ipf2.b function finds the maximum likelihood estimates for fitted values in the log-linear model:

$$\log y_{pq} = \log \alpha_p + \log \beta_q + \log \lambda_{ij} I(p \in i, q \in j) + \log m_{pq}$$

where m_{pq} is a prior estimate for y_{pq} and is no more complex than the matrices being fitted. The $\lambda_{ij} I(p \in i, q \in j)$ term ensures a saturated fit on the block the (i,j) block.

Usage

```
ipf2_block(row_tot = NULL, col_tot = NULL, block_tot = NULL,
  block = NULL, m = NULL, tol = 1e-05, maxit = 500,
  verbose = TRUE, ...)
```

Arguments

row_tot	Vector of origin totals to constrain the sum of the imputed cell rows.
col_tot	Vector of destination totals to constrain the sum of the imputed cell columns.
block_tot	Matrix of block totals to constrain the sum of the imputed cell blocks.
block	Matrix of block structure corresponding to block_tot.
m	Matrix of auxiliary data. By default set to 1 for all origin-destination combinations.
tol	Numeric value for the tolerance level used in the parameter estimation.
maxit	Numeric value for the maximum number of iterations used in the parameter estimation.
verbose	Logical value to indicate the print the parameter estimates at each iteration. By default FALSE.
...	Additional arguments passes to block_matrix.

Value

Iterative Proportional Fitting routine set up using the partial likelihood derivatives. The arguments row_tot and col_tot take the row-table and column-table specific known margins. The block_tot take the totals over the blocks in the matrix defined with b. Diagonal values can be added by the user, but care must be taken to ensure resulting diagonals are feasible given the set of margins.

The user must ensure that the row and column totals in each table sum to the same value. Care must also be taken to allow the dimension of the auxiliary matrix (m) equal those provided in the row and column totals.

Returns a list object with

mu	Array of indirect estimates of origin-destination matrices by migrant characteristic
it	Iteration count
tol	Tolerance level at final iteration

Author(s)

Guy J. Abel

See Also

block_matrix, stripe_matrix, block_sum

Examples

```
y <- ipf2_block(row_tot= c(30,20,30,10,20,5,0,10,5,5,5,10),
                col_tot = c(45,10,10,5,5,10,50,5,10,0,0,0),
                block_tot = matrix(data = c(0,0 ,50,0, 35,0,25,0, 10,10,0,0, 10,10,0,0),
                              nrow = 4, byrow = TRUE),
                block = block_matrix(x = 1:16, b = c(2,3,4,3)))
addmargins(y$mu)
```

ipf2_stripe *Iterative Proportional Fitting Routine for the Indirect Estimation of Origin-Destination-Type Migration Flow Tables with Known Origin and Destination Margins and Stripe Elements.*

Description

The ipf2.b function finds the maximum likelihood estimates for fitted values in the log-linear model:

$$\log y_{pq} = \log \alpha_p + \log \beta_q + \log \lambda_{ij} I(p \in i, q \in j) + \log m_{pq}$$

where m_{pq} is a prior estimate for y_{pq} and is no more complex than the matrices being fitted. The $\lambda_{ij} I(p \in i, q \in j)$ term ensures a saturated fit on the block the (i, j) block.

Usage

```
ipf2_stripe(row_tot = NULL, col_tot = NULL, stripe_tot = NULL,
  stripe = NULL, m = NULL, tol = 1e-05, maxit = 500,
  verbose = TRUE, ...)
```

Arguments

row_tot Vector of origin totals to constrain the sum of the imputed cell rows.

col_tot Vector of destination totals to constrain the sum of the imputed cell columns.

stripe_tot Matrix of stripe totals to constrain the sum of the imputed cell blocks.

stripe Matrix of stripe stucture corresponding to stripe_tot.

m Matrix of auxiliary data. By default set to 1 for all origin-destination combinations.

tol Numeric value for the tolerance level used in the parameter estimation.

maxit Numeric value for the maximum number of iterations used in the parameter estimation.

verbose Logical value to indicate the print the parameter estimates at each iteration. By default FALSE.

... Additional arguments passes to stripe_matrix.

Value

Iterative Proportional Fitting routine set up using the partial likelihood derivatives. The arguments row_tot and col_tot take the row-table and column-table specific known margins. The stripe_tot take the totals over the stripes in the matrix defined with b. Diagonal values can be added by the user, but care must be taken to ensure resulting diagonals are feasible given the set of margins. The user must ensure that the row and column totals in each table sum to the same value. Care must also be taken to allow the dimension of the auxiliary matrix (m) equal those provided in the row and column totals. Returns a list object with `mu` Array of indirect estimates of origin-destination matrices by migrant characteristic

`it` Iteration count

`tol` Tolerance level at final iteration

Author(s)

Guy J. Abel

See Also

`stripe_matrix`, `block_matrix`, `block_sum`

Examples

```
y <- ipf2_stripe(row_tot = c(85, 70, 35, 30, 60, 55, 65),
 stripe_tot = matrix(c(15,20,50,
                35,10,25,
                5 ,0 ,30,
                10,10,10,
                30,30,0,
                15,30,10,
                35,25,5 ), ncol = 3, byrow = TRUE),
 stripe = stripe_matrix(x = 1:21, s = c(2,2,3), byrow = TRUE))
 addmargins(y$mu)
```

ipf3 *Iterative Proportional Fitting Routine for the Indirect Estimation of Origin-Destination-Migrant Type Migration Flow Tables with Known Origin and Destination Margins.*

Description

The `ipf3` function finds the maximum likelihood estimates for fitted values in the log-linear model:

$$\log y_{ijk} = \log \alpha_i + \log \beta_j + \log \lambda_k + \log \gamma_{ik} + \log \kappa_{jk} + \log m_{ijk}$$

where m_{ijk} is a set of prior estimates for y_{ijk} and is no more complex than the matrices being fitted.

Usage

```
ipf3(row_tot = NULL, col_tot = NULL, m = NULL, tol = 1e-05,
  maxit = 500, verbose = TRUE)
```

Arguments

row_tot Vector of origin totals to constrain the sum of the imputed cell rows.

col_tot Vector of destination totals to constrain the sum of the imputed cell columns.

m Array of auxiliary data. By default set to 1 for all origin-destination-migrant typologies combinations.

tol Numeric value for the tolerance level used in the parameter estimation.

maxit Numeric value for the maximum number of iterations used in the parameter estimation.

verbose Logical value to indicate the print the parameter estimates at each iteration. By default FALSE.

Value

Iterative Proportional Fitting routine set up in a similar manner to Agresti (2002, p.343). The arguments row_tot and col_tot take the row-table and column-table specific known margins.

The user must ensure that the row and column totals in each table sum to the same value. Care must also be taken to allow the dimension of the auxiliary matrix (m) to equal those provided in the row and column totals.

Returns a list object with

mu Array of indirect estimates of origin-destination matrices by migrant characteristic

it Iteration count

tol Tolerance level at final iteration

Author(s)

Guy J. Abel

References

Abel, G. J. (2013). Estimating Global Migration Flow Tables Using Place of Birth. *Demographic Research* 28, (18) 505-546

Agresti, A. (2002). *Categorical Data Analysis* 2nd edition. Wiley.

See Also

ipf3_qi, ipf2

Examples

```
## create row-table and column-table specific known margins.
dn <- LETTERS[1:4]
P1 <- matrix(c(1000, 100,  10,   0,
               55,   555,  50,   5,
               80,    40, 800 , 40,
               20,    25,  20, 200),
             nrow = 4, ncol = 4, byrow = TRUE,
             dimnames = list(pob = dn, por = dn))
P2 <- matrix(c(950, 100,  60,   0,
                80, 505,  75,   5,
                90,  30, 800,  40,
                40,  45,   0, 180),
             nrow = 4, ncol = 4, byrow = TRUE,
             dimnames = list(pob = dn, por = dn))
# display with row and col totals
addmargins(P1)
addmargins(P2)

# run ipf
y <- ipf3(row_tot = t(P1), col_tot = P2)
# display with row, col and table totals
round(addmargins(y$mu), 1)
# origin-destination flow table
round(sum_od(y$mu), 1)

## with alternative offset term
dis <- array(c(1, 2, 3, 4, 2, 1, 5, 6, 3, 4, 1, 7, 4, 6, 7, 1), c(4, 4, 4))
y <- ipf3(row_tot = t(P1), col_tot = P2, m = dis)
# display with row, col and table totals
round(addmargins(y$mu), 1)
# origin-destination flow table
round(sum_od(y$mu), 1)
```

ipf3_qi *Iterative Proportional Fitting Routine for the Indirect Estimation of Origin-Destination-Migrant Type Migration Flow Tables with Known Origin and Destination Margins and Diagonal Elements.*

Description

This function is predominantly intended to be used within the ffs routine.

Usage

```
ipf3_qi(row_tot = NULL, col_tot = NULL, diag_count = NULL,
  m = NULL, speed = TRUE, tol = 1e-05, maxit = 500,
  verbose = TRUE)
```

Arguments

row_tot Vector of origin totals to constrain the sum of the imputed cell rows.

col_tot Vector of destination totals to constrain the sum of the imputed cell columns.

diag_count Array with counts on diagonal to constrain diagonal elements of the indirect estimates too. By default these are taken as their maximum possible values given the relevant margins totals in each table. If user specifies their own array of diagonal totals, values on the non-diagonals in the array can take any positive number (they are ultimately ignored).

`m` Array of auxiliary data. By default set to 1 for all origin-destination-migrant typologies combinations.

`speed` Speeds up the IPF algorithm by minimizing sufficient statistics.

`tol` Numeric value for the tolerance level used in the parameter estimation.

`maxit` Numeric value for the maximum number of iterations used in the parameter estimation.

`verbose` Logical value to indicate the print the parameter estimates at each iteration. By default `FALSE`.

Details

The `ipf3` function finds the maximum likelihood estimates for fitted values in the log-linear model:

$$\log y_{ijk} = \log \alpha_i + \log \beta_j + \log \lambda_k + \log \gamma_{ik} + \log \kappa_{jk} + \log \delta_{ijk} I(i=j) + \log m_{ijk}$$

where m_{ijk} is a set of prior estimates for y_{ijk} and is no more complex than the matrices being fitted. The $\delta_{ijk} I(i=j)$ term ensures a saturated fit on the diagonal elements of each (i,j) matrix.

Value

Iterative Proportional Fitting routine set up using the partial likelihood derivatives illustrated in Abel (2013). The arguments `row_tot` and `col_tot` take the row-table and column-table specific known margins. By default the diagonal values are taken as their maximum possible values given the relevant margins totals in each table. Diagonal values can be added by the user, but care must be taken to ensure resulting diagonals are feasible given the set of margins.

The user must ensure that the row and column totals in each table sum to the same value. Care must also be taken to allow the dimension of the auxiliary matrix (`m`) equal those provided in the row and column totals.

Returns a `list` object with

`mu` Array of indirect estimates of origin-destination matrices by migrant characteristic

`it` Iteration count

`tol` Tolerance level at final iteration

Author(s)

Guy J. Abel

References

Abel, G. J. (2013). Estimating Global Migration Flow Tables Using Place of Birth. *Demographic Research* 28, (18) 505-546

See Also

`ipf3`, `ffs_demo`

Examples

```
## create row-table and column-table specific known margins.
dn <- LETTERS[1:4]
P1 <- matrix(c(1000, 100,  10,   0,
               55,   555,  50,   5,
               80,    40, 800 , 40,
               20,    25,  20, 200),
             nrow = 4, ncol = 4, byrow = TRUE,
             dimnames = list(pob = dn, por = dn))
P2 <- matrix(c(950, 100,  60,   0,
                80, 505,  75,   5,
                90,  30, 800,  40,
                40,  45,   0, 180),
             nrow = 4, ncol = 4, byrow = TRUE,
             dimnames = list(pob = dn, por = dn))
# display with row and col totals
addmargins(P1)
addmargins(P2)

# run ipf
y <- ipf3_qi(row_tot = t(P1), col_tot = P2)
# display with row, col and table totals
round(addmargins(y$mu), 1)
# origin-destination flow table
round(sum_od(y$mu), 1)

## with alternative offset term
dis <- array(c(1, 2, 3, 4, 2, 1, 5, 6, 3, 4, 1, 7, 4, 6, 7, 1), c(4, 4, 4))
y <- ipf3_qi(row_tot = t(P1), col_tot = P2, m = dis)
# display with row, col and table totals
round(addmargins(y$mu), 1)
# origin-destination flow table
round(sum_od(y$mu), 1)
```

ipf_seed *Quickly Create IPF Seed*

Description

This function is predominantly intended to be used within the ipf routines in the migest package.

Usage

```
ipf_seed(m = NULL, R = NULL, n_dim = NULL, dn = NULL)
```

Arguments

m	Matrix, Array or NULL to build seed. If NULL seed will be 1 for all elements.
R	Number of rows, columns and possibly n_dimensions for seed matrix or array.

n_dim Numeric integer for the number of n_dimensions - 2 for matrix, 3 or more for an array

dn Vector of character strings for n_dimension names

Value

An array or matrix

Author(s)

Guy J. Abel

See Also

ipf3_qi, ffs_diff

Examples

```
ipf_seed(m = NULL, R = 4, n_dim = 2)
ipf_seed(m = NULL, R = 5, n_dim = 3, dn = LETTERS[1:5])
ipf_seed(m = matrix(1:4, nrow = 2), n_dim = 3, dn = LETTERS[1:2])
```

---

match_pob_tot *Adjust Migrant Stock Tables to Have Matching Place of Birth Totals*

Description

This function is predominantly intended to be used within the ffs routines in the migest package.

Usage

```
match_pob_tot(m1, m2, method = "rescale")
```

Arguments

m1 Matrix of migrant stock totals at time *t*. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1.

m2 Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1.

method Character string matching either rescale, open, open-dr. The rescale method ensure flow estimates closely match the net migration totals implied by the changes in population totals, births and deaths - as introduced in the Science paper. The open-dr method allows for moves in and out of the global system - as introduced in the Demographic Research paper. The open method is a slight improvement over open-dr - the calculation of the moves and in and out use more sensible weights.

Value

Returns a `list` object with:

`m1_adj`	Matrix of adjusted `m1` where rows (place of births) match `m2_adj`.
`m2_adj`	Matrix of adjusted `m2` where rows (place of births) match `m1_adj`.
`in_mat`	Matrix of estimated inflows into the system.
`out_mat`	Matrix of estimated outflows from the system.

Author(s)

Guy J. Abel

References

Abel, G. J. (2018). Estimates of Global Bilateral Migration Flows by Gender between 1960 and 2015. *International Migration Review* Forthcoming.

Abel, G. J. and Sander, N. (2014). Quantifying Global International Migration Flows. *Science*, 343 (6178) 1520-1522

See Also

`ipf3_qi`, `ffs_diff`

`multi_comp` *Multiplicative Component Description of Origin-Destination Migration Flow Tables*

Description

Multiplicative component descriptions of *n*-dimension flow tables based on total reference coding system.

Usage

```
multi_comp(m)
```

Arguments

`m`	`matrix` or `array` of migration flows

Value

`matrix` or `array` of multiplicative components of ‘m‘. When output is an array the total for each table of origin-destination flows is used.

Examples

```
n <- LETTERS[1:4]
m0 <- matrix(data = c(0, 100, 30, 70, 50, 0, 45, 5, 60, 35, 0, 40, 20, 25, 20, 0),
             nrow = 4, ncol = 4, dimnames = list(orig = n, dest = n), byrow = TRUE)
addmargins(m0)
multi_comp(m = m0)
```

---

multi_comp2 *Multiplicative component descriptions of origin-destination flow tables based on total reference coding system.*

Description

Multiplicative component descriptions of origin-destination flow tables based on total reference coding system.

Usage

```
multi_comp2(m)
```

Arguments

m	`matrix` of migration flows

Value

`matrix` of multiplicative components of ‘m‘. When output is an array the total for each table of origin-destination flows is used.

Examples

```
n <- LETTERS[1:2]
m0 <- array(c(5, 1, 2, 7, 4, 2, 5, 9), dim = c(2, 2, 2),
            dimnames = list(orig = n, dest = n, type = c("ILL", "HEALTHY")))
addmargins(m0)
multi_comp(m = m0)
multi_comp2(m = m0)
```

net_param *Estimate Parameters for Net Migration Scaling.*

Description

This function is predominantly intended to be used within the cm_net routine.

Usage

```
net_param(m, region, net_tot)
```

Arguments

m	Matrix of origin-destination flows, where the first and second dimensions correspond to origin and destination respectively.
region	Integer value corresponding to the region that the net migration sum is desired. Will return sums for all regions by default.
net_tot	Vector of net migration totals to constrain the sum of the imputed cell columns. Elements must sum to zero.

Value

Vector of two values corresponding to the roots for the quadratic equation.

Author(s)

Guy J. Abel

net_scale *Scale Migration Flows in Origin-Destination*

Description

This function is predominantly intended to be used within the cm_net routine.

Usage

```
net_scale(m, region = NULL, alpha)
```

Arguments

m	Matrix of origin-destination flows, where the first and second dimensions correspond to origin and destination respectively.
region	Integer corresponding to row (column) in a square matrix for the region where scaling is to be applied
alpha	Numeric value of the scaling factor

Value

Matrix scaled in region(s) by value of alpha, where alpha applied to destination flows and inverse of alpha applied to origin flows

Author(s)

Guy J. Abel

---

net_sum *Extract Net Migration from an Origin-Destination Migration Flow Matrix.*

Description

Sums each regions flows (from origin rows and destination columns) to obtain net migration sums.

Usage

```
net_sum(m, region = 1:dim(m)[1])
```

Arguments

m	Matrix of origin-destination flows, where the first and second dimensions correspond to origin and destination respectively.
region	Integer value corresponding to the region that the net migration sum is desired. Will return sums for all regions by default.

Value

Returns a numeric value of the sum of a single block.

Author(s)

Guy J. Abel

See Also

block_sum, sum_od

Examples

```
m <- matrix(data = 1:16, nrow = 4, ncol = 4)
net_sum(m)
```

`quadratic_eqn` *Solve Quadratic Equation*

Description

General function to solve classic quadratic equation:

$$ax^2 + bx + c = 0$$

Usage

```
quadratic_eqn(a, b, c)
```

Arguments

a	Numeric value for quadratic term of x.
b	Numeric value for multiplicative term of x.
c	Numeric value for constant term.

Value

Vector of two values corresponding to the roots for the quadratic equation.

Author(s)

Guy J. Abel

Source

Adapted from https://rpubs.com/kikihatzistavrou/80124

Examples

```
quadratic_eqn(a = 2, b = 4, c = -6)
```

rc9 *Rogers-Castro Migration Schedule*

Description

Provides the Rogers-Castro schedule,

$$M(x) = a_1 \exp[-\alpha_1 x] + a_2 \exp[\alpha_2 (x - \mu_2) - \exp[\lambda_2 (x - \mu_2)]] + c$$

for a given set of parameters and ages.

Usage

```
rc9(x, param = NULL, scaled = TRUE)
```

Arguments

`x`	Vector of numbers
`param`	List with names matching the parameters in the age schedule
`scaled`	Scale estimates to sum to one across all ages, x.

Value

Returns the M(x) values from the Rogers-Castro schedule of age specific migration rate. The age range for the calculation can take any sequence of positive numbers, such as ages in single or 5-year intervals. The `param` argument must be a list with correct names for each parameter. See for example the `rc9.fund` object for an example of the naming convention.

Author(s)

Guy J. Abel

References

Rogers, A., and L. J. Castro. (1981). Model Migration Schedules. *IIASA Research Report 81* RR-81-30

See Also

`rc9.fund`

Examples

```
# single age groups
x <- 1:100
m <- rc9(x, param = rc9.fund)
plot(x, m, type="l")

# 5 year age groups
m <- rc9(x, param = rc9.fund)
plot(x, m, type="l")
```

---

rc9.fund *Fundamental Parameters for Rogers-Castro Migration Schedule*

Description

Set of fundamental parameters for the Rogers-Castro migration age schedule, as suggested in Rogers and Castro (1981).

Usage

```
rc9.fund
```

Format

A `list` of the parameters required by the `rc9` function:

$$a_1 = 0.02$$

$$\alpha_1 = 0.1$$

$$a_2 = 0.06$$

$$\alpha_2 = 0.1$$

$$\mu_2 = 20$$

$$\lambda_2 = 0.4$$

$$c = 0.003$$

Source

Rogers, A., and L. J. Castro. (1981). Model Migration Schedules. *IIASA Research Report 81* RR-81-30

Examples

```
# check format
str(rc9.fund)

# single age groups
x <- 1:100
m <- rc9(x, param = rc9.fund)
plot(x, m, type="l")

# alter to see the effect of mu2
p1 <- rc9.fund
p1$mu2 <- 30
m1 <- rc9(x, param = p1)
plot(x, m, type="l")
lines(x, m1, lty=2)
```

---

rescale_integer_sum *Rescale Integer Vector to a Set sum*

Description

For when you want to rescale a set of numbers to sum to a given value and do not want all rescaled values to be integers.

Usage

```
rescale_integer_sum(x, tot)
```

Arguments

x	Vector of numeric values
tot	Numeric integer value to rescale sum to.

Value

Vector or integer values that sum to to tot

Author(s)

Guy J. Abel

See Also

ipf3_qi, ffs_diff

Examples

```
x <- rnorm(n = 10, mean = 5, sd = 20)
y <- rescale_integer_sum(x, tot = 10)
y
sum(y)

for(i in 1:10){
  y <- rescale_integer_sum(x = rpois(n = 10, lambda = 10), tot = 1000)
  print(sum(y))
}
```

`rescale_nb` *Rescale Native Born Populations to Match Differences in Births and Deaths over Period*

Description

This function is predominantly intended to be used within the ffs routines in the migest package. Adjustment to ensure that global differences in stocks match the global demographic changes from births and deaths.

Usage

```
rescale_nb(m1, m2, b, d, verbose = FALSE)
```

Arguments

`m1` Matrix of migrant stock totals at time *t*. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*

`m2` Matrix of migrant stock totals at time *t*+1. Rows in the matrix correspond to place of birth and columns to place of residence at time *t*+1.

`b` Vector of the number of births between time *t* and *t*+1 in each region.

`d` Vector of the number of deaths between time *t* and *t*+1 in each region.

`verbose` Logical value to indicate the print the parameter estimates at each iteration. By default `FALSE`.

Value

List with adjusted `m1` and `m2`.

Author(s)

Guy J. Abel

See Also

`ipf3_qi`, `ffs_diff`

Examples

```
dn <- LETTERS[1:4]
P1 <- matrix(data = c(1000, 100, 10, 0, 55, 555, 50, 5, 80, 40, 800, 40, 20, 25, 20, 200),
             nrow = 4, ncol = 4, dimnames = list(pob = dn, por = dn), byrow = TRUE)
P2 <- matrix(data = c(950, 100, 60, 0, 80, 505, 75, 5, 90, 30, 800, 40, 40, 45, 0, 180),
             nrow = 4, ncol = 4, dimnames = list(pob = dn, por = dn), byrow = TRUE)
# display with row and col totals
addmargins(A = P1)
addmargins(A = P2)

# births and deaths
b <- rep(x = 10, 4)
d <- rep(x = 5, 4)
# no change in stocks, but 20 more deaths than births...
sum(P2 - P1) - sum(b - d)
# rescale
y <- rescale_nb(m1 = P1, m2 = P2, b = b, d = d)
y
sum(y$m1_adj - y$m2_adj) - sum(b - d)

# check for when extra is positive and odd
d[1] <- 31
d
sum(P2 - P1) - sum(b - d)
# rescale
y <- rescale_nb(m1 = P1, m2 = P2, b = b, d = d)
sum(y$m1_adj - y$m2_adj) - sum(b - d)
```

rescale_net *Rescale Net Migration Total to a Global Zero Sum*

Description

Modify a set of net migration (or any numbers) so that they sum to zero.

Usage

```
rescale_net(x, method = "no-switches", w = rep(1, length(x)),
  integer_result = TRUE)
```

Arguments

x	Vector of net migration values
method	Method used to adjust net migration values of x to obtain a global zero sum. By default method="no-switches". Can also take values method="switches". See details for explanation on each method.
w	Weights used in rescaling method
integer_result	Logical operator to indicate if output shoud be integers, default is TRUE.

Value

Rescales net migration for a number of regions in vector `x` to sum to zero. When `method="no-switches"` rescaling of values are done for the positive and negative values separately, to ensure the final global sum is zero. When `method="switches"` the mean of the unscaled net migration is subtracted from each value.

Author(s)

Guy J. Abel

References

Abel, G. J. (2018). Non-zero trajectories for long-run net migration assumptions in global population projection models. *Demographic Research* 38, (54) 1635–1662

Examples

```
# net migration in regions countries (does not add up to zero)
x <- c(-200, -30, -5, 0, 10, 20, 60, 80)
x
sum(x)
# rescale
y1 <- rescale_net(x)
y1
sum(y1)
# rescale without integer restriction
y2 <- rescale_net(x, integer_result = FALSE)
y2
sum(y2)
# rescale allowing switching of signs (small negative value becomes positive)
y3 <- rescale_net(x, method = "switches")
y3
sum(y3)
```

`stripe_matrix` *Create a Stripped Matrix with Non-Uniform Block Sizes.*

Description

Create a Stripped Matrix with Non-Uniform Block Sizes.

Usage

```
stripe_matrix(x = NULL, s = NULL, byrow = FALSE, dimnames = NULL)
```

Arguments

x	Vector of numbers to identify each stripe.
s	Vector of values for the size of the stripes, order depending on byrow
byrow	Logical value. If FALSE (the default) the stripes are filled by columns, otherwise the stripes in the matrix are filled by rows.
dimnames	Character string of name attribute for the basis of the stripped matrix. If NULL a vector of the same length of s provides the basis of row and column names.

Value

Returns a matrix with stripe sizes determined by the s argument. Each stripe is filled with the same value taken from x.

Author(s)

Guy J. Abel

See Also

block_matrix, block_sum, ipf2_stripe

Examples

```
stripe_matrix(x = 1:44, s = c(2,3,4,2), dimnames = LETTERS[1:4], byrow = TRUE)
```

sum_net *Extract Net Migration from an Origin-Destination Migration Flow Matrix.*

Description

Sums each regions flows (from origin rows and destination columns) to obtain net migration sums.

Usage

```
sum_net(m, region = 1:dim(m)[1])
```

Arguments

m	Matrix of origin-destination flows, where the first and second dimensions correspond to origin and destination respectively.
region	Integer value corresponding to the region that the net migration sum is desired. Will return sums for all regions by default.

Value

Returns a numeric value of the sum of a single block.

Author(s)

Guy J. Abel

See Also

block_sum, sum_od

Examples

```
m <- matrix(data = 1:16, nrow = 4, ncol = 4)
sum_net(m)
```

---

sum_od *Extract a Classic Origin-Destination Migration Flow Matrix.*

Description

Extract a classic origin-destination migration flow matrix from a more detailed dis-aggregation of flows stored in an (`array`) object.

Usage

```
sum_od(y)
```

Arguments

y	Array of origin-destination matrices, where the first and second dimensions correspond to origin and destination respectively. Higher dimension(s) refer to additional migrant characteristic(s).

Value

Matrix from summing over the first and second dimension. Set diagonals to zero.

Returns a `matrix` object of origin-destination flows

Author(s)

Guy J. Abel

Examples

```
dn <- LETTERS[1:2]
y <- cm3(row_tot = c(18, 20) * 2, col_tot = c(16, 22) * 2,
         m = array(c(5, 1, 2, 7, 4, 2, 5, 9), dim = c(2, 2, 2),
                   dimnames = list(orig = dn, dest = dn, type = c("ILL", "HEALTHY"))))
round(addmargins(y$n))
round(addmargins(sum_od(y$n)))
```

Index