Package ‘lefko3’

July 22, 2021

Type Package

Title Historical and Ahistorical Population Projection Matrix Analysis

Version 3.6.0

Date 2021-07-21


Encoding UTF-8

License GPL (>= 2)

Imports Rcpp (>= 1.0.5), glmmTMB, lme4, MASS, MuMIn, pscl, SparseM, stats, stringr, VGAM

LinkingTo Rcpp, RcppArmadillo

LazyData true

RoxygenNote 7.1.1

Suggests knitr, popbio, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Richard P. Shefferson [aut, cre] (<https://orcid.org/0000-0002-5234-3131>), Johan Ehrlen [aut] (<https://orcid.org/0000-0001-8539-8967>)

Maintainer Richard P. Shefferson <cdorm@ecc.u-tokyo.ac.jp>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2021-07-22 04:40:10 UTC
## R topics documented:

- `aflefko2` ......................................................... 3
- `cond_hnpmp` ................................................... 9
- `cypdata` ......................................................... 11
- `cypvert` .......................................................... 14
- `elasticity3` ..................................................... 16
- `elasticity3.lefkoMat` .......................................... 18
- `elasticity3.list` ................................................ 21
- `elasticity3.matrix` .............................................. 24
- `flefko2` .......................................................... 27
- `flefko3` .......................................................... 33
- `historicalize3` .................................................. 40
- `image3` ........................................................... 47
- `image3.lefkoElas` ................................................ 49
- `image3.lefkoMat` ................................................ 52
- `image3.lefkoSens` ................................................ 54
- `image3.list` ....................................................... 57
- `image3.matrix` .................................................... 59
- `lambda3` .......................................................... 61
- `lambda3.lefkoMat` ............................................... 63
- `lambda3.matrix` ................................................... 66
- `lathyrus` ........................................................ 69
- `lefko3` .............................................................. 72
- `lmean` ............................................................. 73
- `modelsearch` ..................................................... 76
- `overwrite` ........................................................ 83
- `projection3` ....................................................... 85
- `repvalue3` ......................................................... 88
- `repvalue3.lefkoMat` ............................................. 90
- `repvalue3.matrix` ............................................... 94
- `rlefko2` .......................................................... 96
- `rlefko3` .......................................................... 101
- `sensitivity3` ....................................................... 107
- `sensitivity3.lefkoMat` .......................................... 109
- `sensitivity3.list` ............................................... 112
- `sensitivity3.matrix` ............................................. 115
- `sf_create` ......................................................... 117
- `sf_distrib` ....................................................... 121
- `slambda3` ........................................................ 124
- `stablestage3` ..................................................... 127
- `stablestage3.lefkoMat` ......................................... 129
- `stablestage3.matrix` ............................................ 133
- `summary.lefkoCondMat` ........................................ 135
- `summary.lefkoElas` ............................................. 138
- `summary.lefkoMat` ............................................... 139
- `summary.lefkoMod` ............................................. 141
- `supplemental` ................................................... 142
Function `aflefko2()` returns ahistorical age x stage MPMs corresponding to the patches and years given, including the associated component transition and fecundity matrices, data frame detailing the characteristics of ahistorical stages and the exact age-stage combinations corresponding to rows and columns in estimated matrices, and a data frame characterizing the patch and year combinations corresponding to these matrices. Unlike `rlefko2()` and `rlefko3()`, this function does not currently distinguish populations.

Usage

```r
aflefko2(
    year = "all",
    patch = "all",
    stageframe,
    supplement = NA,
    repmatrix = NA,
    overwrite = NA,
    data = NA,
    modelsuite = NA,
    surv_model = NA,
    obs_model = NA,
    size_model = NA,
    repst_model = NA,
    fec_model = NA,
    jsurv_model = NA,
    jobs_model = NA,
    jsize_model = NA,
    jrepst_model = NA,
    paramnames = NA,
    inda = 0,
    indb = 0,
    indc = 0,
    surv_dev = 0,
    obs_dev = 0,
    size_dev = 0,
    repst_dev = 0,
    fec_dev = 0,
    jsurv_dev = 0,
    jobs_dev = 0,
)```
jsize_dev = 0,  
jrepst_dev = 0,  
repmod = 1,  
yearcol = "year2",  
patchcol = "patchid",  
year.as.random = FALSE,  
patch.as.random = FALSE,  
final_age = 10,  
continue = TRUE,  
randomseed = NA,  
negfec = FALSE,  
reduce = FALSE,  
err_check = FALSE
)

Arguments

year  A variable corresponding to year or observation time, or a set of such values, given in values associated with the year term used in linear model development. Can also equal all, in which case matrices will be estimated for all years. Defaults to all.

patch  A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to all if matrices should be estimated for all patches. Defaults to all.

stageframe  A stageframe object that includes information on the size, observation status, propagule status, immaturity status, and maturity status of each ahistorical stage. Should also incorporate bin widths if size is continuous.

supplement  An optional data frame of class lefkoSD that provides supplemental data that should be incorporated into the MPM. Three kinds of data may be integrated this way: transitions to be estimated via the use of proxy transitions, transition overwrites from the literature or supplemental studies, and transition multipliers for fecundity. This data frame should be produced using the supplemental() function. Can be used in place of or in addition to an overwrite table (see overwrite below) and a reproduction matrix (see repmatrix below).

repmatrix  A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in supplement. If left blank, then aflefko2() will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in supplement above. Must be the dimensions of an ahistorical matrix.

overwrite  An optional data frame developed with the overwrite() function describing transitions to be overwritten either with given values or with other estimated transitions. Note that this function supplements overwrite data provided in supplement.
The original historical demographic data frame used to estimate vital rates (class hfvdata). The original data frame is required in order to initialize years and patches properly.

modelsuite An optional lefkomod object holding the vital rate models. If given, then surv_model, obs_model, size_model, repst_model, fec_model, jsurv_model, jobs_model, jsize_model, jrepst_model, paramnames, yearcol, and patchcol are not required. No models should include size or reproductive status in time $t-1$.

surv_model A linear model predicting survival probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any survival probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

obs_model A linear model predicting sprouting or observation probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any observation probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

size_model A linear model predicting size. This can be a model of class glm or glmer, both of which require a predicted poisson variable under a log link, or a model of class lm or lmer, in which a Gaussian response is assumed. If given, then will overwrite any size model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

repst_model A linear model predicting reproduction probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

fec_model A linear model predicting fecundity. This can be a model of class glm or glmer, and requires a predicted poisson variable under a log link. If given, then will overwrite any fecundity model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

jsurv_model A linear model predicting juvenile survival probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile survival probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

jobs_model A linear model predicting juvenile sprouting or observation probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile observation probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

jsize_model A linear model predicting juvenile size. This can be a model of class glm or glmer, both of which require a predicted poisson variable under a log link, or a model of class lm or lmer, in which a Gaussian response is assumed. If given, then will overwrite any juvenile size model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$. 
jrepst_model  A linear model predicting reproduction probability of a mature individual that
was immature in the previous year. This can be a model of class glm or glmer,
and requires a predicted binomial variable under a logit link. If given, then will
overwrite any reproduction probability model given in modelsuite. This model
must have been developed in a modeling exercise testing only the impacts of
time $t$.

paramnames  A dataframe with two columns, the first showing the general model terms that
will be used in matrix creation, and the second showing the equivalent terms
used in modeling. Only required if modelsuite is not supplied.

inda  A numeric value to use for individual covariate a. Defaults to 0.
indb  A numeric value to use for individual covariate b. Defaults to 0.
indc  A numeric value to use for individual covariate c. Defaults to 0.
surv_dev  A numeric value to be added to the y-intercept in the linear model for survival
probability.
obs_dev  A numeric value to be added to the y-intercept in the linear model for observation
probability.
size_dev  A numeric value to be added to the y-intercept in the linear model for size.
reps_dev  A numeric value to be added to the y-intercept in the linear model for probability
of reproduction.
fec_dev  A numeric value to be added to the y-intercept in the linear model for fecundity.
jsurv_dev  A numeric value to be added to the y-intercept in the linear model for juvenile
survival probability.
jobs_dev  A numeric value to be added to the y-intercept in the linear model for juvenile
observation probability.
jsize_dev  A numeric value to be added to the y-intercept in the linear model for juvenile
size.
jrepst_dev  A numeric value to be added to the y-intercept in the linear model for juvenile
reproduction probability.
repsmod  A scalar multiplier of fecundity. Defaults to 1.
yearcol  The variable name or column number corresponding to year in time $t$ in the
dataset. Not needed if a modelsuite is supplied.
patchcol  The variable name or column number corresponding to patch in the dataset. Not
needed if a modelsuite is supplied.
year.as.random  A logical term indicating whether coefficients for missing patches within vital
rate models should be estimated as random intercepts. Defaults to FALSE, in
which case missing time step coefficients are set to 0.
patch.as.random  A logical term indicating whether coefficients for missing patches within vital
rate models should be estimated as random intercepts. Defaults to FALSE, in
which case missing patch coefficients are set to 0.
final_age  The final age to model in the matrix, where the first age will be age 0.
continue  A logical value designating whether to allow continued survival of individuals
going past the final age, using the demographic characteristics of the final age.
randomseed  A numeric value used as a seed to generate random estimates for missing time step and patch coefficients, if either year.as.random or patch.as.random is set to TRUE. Defaults to set.seed() default.

negfec  A logical value denoting whether fecundity values estimated to be negative should be reset to 0. Defaults to FALSE.

reduce  A logical value denoting whether to remove ahistorical stages associated solely with 0 transitions. These are only removed in cases where the associated row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.

err_check  A logical value indicating whether to add matrices of vital rate probabilities associated with each matrix. Defaults to FALSE.

Value

If all inputs are properly formatted, then this function will return an object of class lefkoMat. Output includes:

A  A list of full projection matrices in order of sorted patches and years.

U  A list of survival transition matrices sorted as in A.

F  A list of fecundity matrices sorted as in A.

hstages  Null for ahistorical matrices.

agestages  A data frame showing the stage number and stage name corresponding to ahstages, as well as the associated age, of each actual row in each age-by-stage matrix.

ahstages  A data frame detailing the characteristics of associated ahistorical stages, in the form of a modified stageframe that includes status as an entry stage through reproduction.

labels  A data frame giving the patch and year of each matrix in order. In aflefko2(), only one population may be analyzed at once, and so pop = NA

matrixqc  A short vector describing the number of non-zero elements in U and F matrices, and the number of annual matrices.

modelqc  This is the qc portion of the modelsuite input.

prob_out  An optional element only added if err_check = TRUE. This is a list of vital rate probability matrices, with 4 columns in the order of survival, observation probability, reproduction probability, and size transition probability.

Notes

This function will yield incorrect estimates if the models utilized incorporate state in time \( t-1 \). Only use models developed testing for ahistorical effects.

The default behavior of this function is to estimate fecundity with regards to transitions specified via associated fecundity multipliers in either supplement or repmatrix. If both of these fields are left empty, then fecundity will be estimated at full for all transitions leading from reproductive stages to immature and propagule stages. However, if a supplement is provided and a repmatrix is not, or if repmatrix is set to 0, then only fecundity transitions noted in the supplement will be set to non-zero values. To use the default behavior of setting all reproductive stages to reproduce
at full fecundity into immature and propagule stages but also incorporate given or proxy survival transitions, input those given and proxy transitions through the overwrite option.

The reproduction matrix (field repmatrix) may only be supplied as ahistorical. If provided as historical, then flefko2() will fail and produce an error.

Users may at times wish to estimate MPMs using a dataset incorporating multiple patches or subpopulations. Should the aim of analysis be a general MPM that does not distinguish these patches or subpopulations, the patchcol variable should be left to NA, which is the default.

Input options including multiple variable names must be entered in the order of variables in time \( t+1, t, \) and \( t-1. \) Rearranging the order WILL lead to erroneous calculations, and will probably also lead to fatal errors.

Using the err_check option will produce a matrix of 4 columns, each characterizing a different vital rate. The product of each row yields an element in the associated \( \$U \) matrix. The number and order of elements in each column of this matrix matches the associated matrix in column vector format. Use of this option is generally for the purposes of debugging code.

**Examples**

data(lathyrus)

```r
sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8)
stagevector <- c("Sd", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr", "Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r", "Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
revecvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)
levecvector <- c(0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
minima <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
maxima <- c(NA, 1, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = revecvector, obsstatus = levecvector, matstatus = propvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propvector, minage = minima, maxage = maxima)

```
cond_hmpm

cond_hmpm

Extract Conditional Ahistorical Matrices from Historical MPM

Description

Function cond_hmpm() takes historical MPMs and decomposes them into ahistorical matrices conditional upon stage in time t-1. In effect, the function takes each historical matrix within a lefkoMat object, and forms one ahistorical matrix for each stage in time t-1.

Usage

cond_hmpm(hmpm, matchoice = NULL, err_check = NULL)

Arguments

hmpm A historical matrix projection model of class lefkoMat.
matchoice A character denoting whether to use A, U, or F matrices. Defaults to A matrices.
err_check A logical value denoting whether to include a data frame of element equivalence from the conditional matrices to the original matrices. Used only for debugging purposes. Defaults to FALSE.
Value

A `lefkoCondMat` object, with the following elements:

Acond  A multi-level list holding the conditional A matrices derived from the input `lefkoMat` object. The top level of the list corresponds to each historical matrix in turn, and the lower level corresponds to each stage in time $t-1$, with individual conditional matrices named for the latter.

hstages  A data frame matrix showing the pairing of ahistorical stages used to create historical stage pairs.

ahstages  A data frame detailing the characteristics of associated ahistorical stages.

labels  A data frame showing the patch and year of each input full A matrix in order.

Examples

data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 0, 0, 1, 2, 2, 2, 0, 0)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0)
immvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0)
propvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
cypsup3r <- supplemental(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3",
"SL", "SL", "SL", "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"),
stage2 = c("SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL",
"SL", "SL", "SL", "SL", "rep", "rep"),
stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "SL", "P3",
"P3", "P3", "SL", "SL", "SL", "all", "all"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
"D", "XSm", "Sm", "D", "XSm", "Sm", "NA"),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
"XSm", "XSm", "XSm", "XSm", "NA"),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
"XSm", "XSm", "XSm", "XSm", "NA"),
```r
givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.25, 0.4, 0.4, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5), type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1), stageframe = cypframe_raw, historical = TRUE)
cypmatrix3r <- rlefko3(data = cypraw_v1, stageframe = cypframe_raw, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), size = c("size3added", "size2added", "size1added"), supplement = cypsupp3r, yearcol = "year2", patchcol = "patchid", indivcol = "individ")
cypcondmats <- cond_hmpm(cypmatrix3r)
summary(cypcondmats)
```

### cypdata

**Demographic Dataset of Cypripedium candidum Population, in Horizontal Format**

#### Description

A dataset containing the states and fates of *Cypripedium candidum* (white lady’s slipper orchids), family Orchidaceae, from a population in Illinois, USA, resulting from monitoring that occurred annually between 2004 and 2009.

#### Usage

```r
data(cypdata)
```

#### Format

A data frame with 77 individuals and 27 variables. Each row corresponds to an unique individual, and each variable from `size.04` on refers to the state of the individual in a particular year.

- **plantid** A numeric variable giving a unique number to each individual.
- **patch** A variable referring to patch within the population.
- **censor** A variable coding for whether the data point is valid. An entry of 1 means that it is so.
- **Inf2.04** Number of double inflorescences in 2004.
- **Inf.04** Number of inflorescences in 2004.
- **Veg.04** Number of stems without inflorescences in 2004.
- **Pod.04** Number of fruits in 2004.
- **Inf2.05** Number of double inflorescences in 2005.
- **Inf.05** Number of inflorescences in 2005.
Veg.05 Number of stems without inflorescences in 2005.
Pod.05 Number of fruits in 2005.
Inf2.06 Number of double inflorescences in 2006.
Inf.06 Number of inflorescences in 2006.
Veg.06 Number of stems without inflorescences in 2006.
Pod.06 Number of fruits in 2006.
Inf2.07 Number of double inflorescences in 2007.
Inf.07 Number of inflorescences in 2007.
Veg.07 Number of stems without inflorescences in 2007.
Pod.07 Number of fruits in 2007.
Inf2.08 Number of double inflorescences in 2008.
Inf.08 Number of inflorescences in 2008.
Veg.08 Number of stems without inflorescences in 2008.
Pod.08 Number of fruits in 2008.
Inf2.09 Number of double inflorescences in 2009.
Inf.09 Number of inflorescences in 2009.
Veg.09 Number of stems without inflorescences in 2009.
Pod.09 Number of fruits in 2009.

Source

Examples

# Cypripedium example using blocksize
data(cypdata)

sizevector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XS", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repsvector, obsstatus = obsvector, matstatus = matvector, propstatus = propvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "P1"),
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
type =c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = cypframe_raw, historical = FALSE)

cymatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cymatrix2r$A[[intersect(which(cymatrix2r$labels$patch == "A"),
which(cymatrix2r$labels$year2 == 2004))]]

lambda3(cymatrix2r)

# Cypripedium example using partial repeat patterns with blocksize and part
# explicit variable name cast
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
"XLg")
retpvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = c("Inf.04", "Inf.05", "Inf.06", "Inf.07", "Inf.08", "Inf.09"),
cypvert

Demographic Dataset of Cypripedium candidum Population, in Vertical Format

Description

A dataset containing the states and fates of Cypripedium candidum (white lady’s slipper orchids), family Orchidaceae, from a population in Illinois, USA, resulting from monitoring that occurred annually between 2004 and 2009. Same dataset as cypdata, but arranged in an ahistorical vertical format.

Usage

data(cypvert)

Format

A data frame with 77 individuals, 331 rows, and 12 variables. Each row corresponds to a specific two-year transition for a specific individual. Variable codes are similar to those for cypdata, but use .2 to identify time \( t \) and .3 to identify time \( t+1 \).

plntid  A numeric variable giving a unique number to each individual.
patch  A variable referring to patch within the population.
censor  A variable coding for whether the data point is valid. An entry of 1 means that it is so.
year2 Year in time $t$.
Inf2.2 Number of double inflorescences in time $t$.
Inf.2 Number of inflorescences in time $t$.
Veg.2 Number of stems without inflorescences in time $t$.
Pod.2 Number of fruits in time $t$.
Inf2.3 Number of double inflorescences in time $t+1$.
Inf.3 Number of inflorescences in time $t+1$.
Veg.3 Number of stems without inflorescences in time $t+1$.
Pod.3 Number of fruits in time $t+1$.

Source

Examples
```r
data(cypvert)
sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)
propvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypframe_raw
```
```r
cypraw_v2 <- historicalize3(data = cypvert, patchidcol = "patch",
individcol = "plantid", year2col = "year2", sizea2col = "Inf2.2",
sizea3col = "Inf.2.3", sizeb2col = "Inf.2.2", sizeb3col = "Inf.3",
sizec2col = "Veg.2", sizec3col = "Veg.3", repstr2a2col = "Inf2.2",
repstr3a2col = "Inf2.3", repstrb2col = "Inf.2", repstrb3col = "Inf.3",
feca2col = "Pod.2", feca3col = "Pod.3", repstrrel = 2,
stageassign = cypframe_raw, stagesize = "sizeadded", censorcol = "censor",
censor = FALSE, NAs0 = TRUE, NRasRep = TRUE, reduce = TRUE)
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
...
elasticity3

Estimate Elasticity of Population Growth Rate to Matrix Elements

Description

elasticity3() is a generic function that returns the elasticity of the population growth rate to the elements of the matrices in a matrix population model. Currently, this function estimates both deterministic and stochastic elasticities, where the growth rate is $\lambda$ in the former case and the log of the stochastic $\lambda$ in the latter case. This function is made to handle very large and sparse matrices supplied as lefkoMat objects, as lists of matrices, and as individual matrices.

Usage

elasticity3(mats, ...)

Arguments

mats A lefkoMat object, or population projection matrix, for which the stable stage distribution is desired.

... Other parameters.

Value

The value returned depends on the class of the mats argument.

See Also

elasticity3.lefkoMat()
elasticity3.matrix()
elasticity3.list()
elasticity3

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 0, 0, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 0)
immvector <- c(0, 0, 0, 0, 0, 0, 0)
propvector <- c(0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sdl", "Sdl", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
givenrate = c(0.345, 0.345, 0.054, 0.054, 0, 0, 0),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, 0),
type = c(1, 1, 1, 1, 3, 3, 1),
type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"),
supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
elasticity3(ehrlen3mean)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
elasticity3.lefkoMat

Estimate Elasticity of Population Growth Rate of a lefkoMat Object

Description

elasticity3.lefkoMat() returns the elasticities of population growth rate to elements of all $A$ matrices in an object of class lefkoMat. If deterministic, then $\lambda$ is taken as the population growth rate. If stochastic, then stochastic $\lambda$, or the stochastic growth rate, is taken as the population growth rate. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.
Usage

```r
## S3 method for class 'lefkoMat'
elasticity3(
  mats,
  stochastic = FALSE,
  steps = 10000,
  time_weights = NA,
  sparse = "auto",
  ...
)
```

Arguments

- **mats**
  - An object of class `lefkoMat`.
- **stochastic**
  - A logical value determining whether to conduct a deterministic (FALSE) or stochastic (TRUE) elasticity analysis. Defaults to FALSE.
- **steps**
  - The number of times to project forward in stochastic simulation. Defaults to 10,000.
- **time_weights**
  - Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among times.
- **sparse**
  - A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
- **...**
  - Other parameters.

Value

This function returns an object of class `lefkoElas`, which is a list with 8 elements. The first, `h_elasmats`, is a list of historical elasticity matrices (NULL if an ahMPM is used as input). The second, `ah_elasmats`, is a list of either ahistorical elasticity matrices if an ahMPM is used as input, or, if an hMPM is used as input, then the result is a list of elasticity matrices in which historical elasticities have been summed by the stage in times \( t \) and \( t+1 \) to produce historically-corrected elasticity matrices, which are equivalent in dimension to ahistorical elasticity matrices but reflect the effects of stage in time \( t-1 \). The third element, `h_stages`, is a data frame showing historical stage pairs (NULL if ahMPM used as input). The fourth element, `agestages`, shows age-stage combinations in the order used in age-by-stage MPMs, if supplled. The fifth element, `ah_stages`, is a data frame showing the order of ahistorical stages. The last 3 elements are the A, U, and F portions of the input.

Notes

Deterministic elasticities are estimated as eqn. 9.72 in Caswell (2001, Matrix Population Models). Stochastic elasticities are estimated as eqn. 14.99 in Caswell (2001). Note that stochastic elasticities are of the stochastic \( \lambda \), while stochastic sensitivities are with regard to the log of the stochastic \( \lambda \).

See Also

```r
elasticity3()
```
elasticity3.matrix()
elasticity3.list()
summary.lefkoElas()

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
                repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
                propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
                patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
                juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
                fecacol = "Intactseed88", deadacol = "Dead1988",
                nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
                censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
                          stage2 = c("Sd", "Sd", "Sd", "Sdl", "rep", "rep", "Sdl"),
                          stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
                          eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
                          eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
                          eststage1 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
                          givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
                          multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
                          type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1),
                          stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
                  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
                  yearcol = "year2", individcol = "individ")

elasticity3(ehrlen3, stochastic = TRUE)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)
elasticity3.list

Estimate Elasticity of Population Growth Rate of a List of Matrices

Description

elasticity3.list() returns the elasticities of lambda to elements of a single matrix. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.
Usage

```r
## S3 method for class 'list'
elasticity3(
mats,
    stochastic = FALSE,
    steps = 10000,
    time_weights = NA,
    historical = FALSE,
    sparse = "auto",
    ...
)
```

Arguments

- `mats` A list of objects of class `matrix`.
- `stochastic` A logical value determining whether to conduct a deterministic (FALSE) or stochastic (TRUE) elasticity analysis. Defaults to FALSE.
- `steps` The number of times to project forward in stochastic simulation. Defaults to 10,000.
- `time_weights` Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among times.
- `historical` A logical value denoting whether the input matrices are historical. Defaults to FALSE.
- `sparse` A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
- `...` Other parameters.

Value

This function returns an object of class `lefkoElas`, which is a list with 8 elements. The first, `h_elasmats`, is a list of historical elasticity matrices, though in the standard list case it returns a NULL value. The second, `ah_elasmats`, is a list of ahistorical elasticity matrices. The third element, `h_stages`, the fourth element, `agestages`, and the fifth element, `ah_stages`, are set to NULL. The last 3 elements are the original A matrices in element A, followed by NULL values for the U and F elements.

Notes

Deterministic elasticities are estimated as eqn. 9.72 in Caswell (2001, Matrix Population Models). Stochastic elasticities are estimated as eqn. 14.99 in Caswell (2001). Note that stochastic elasticities are of stochastic $\lambda$, while stochastic sensitivities are with regard to the log of the stochastic $\lambda$.

See Also

- `elasticity3()`
- `elasticity3.lefkoMat()`
- `elasticity3.matrix()`
Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)

binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propvector)


lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"), stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"), stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"), eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"), eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"), eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"), givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA), type = c(1, 1, 1, 3, 3, 1, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1), stageframe = lathframe, historical = TRUE)

ehrle3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all", stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3, yearcol = "year", individcol = "individ")

elasticity3(ehrlen3$A, stochastic = TRUE)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
                          repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                          propstatus = propvector, immstatus = immvector, indataset = indataset,
                          binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
                          patchidcol = "patch", individcol = "plantid", blocksize = 4,
                          sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
                          repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
                          stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
                          NRasRep = TRUE)

cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
                          stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep", "rep"),
                          eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
                          eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", NA, NA),
                          givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA, NA),
                          multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
                          type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
                          stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
                          year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
                          size = c("size3added", "size2added"), supplement = cypsupp2r,
                          yearcol = "year2", patchcol = "patchid", individcol = "individ")

elasticity3(cypmatrix2r$A)

---

**elasticity3.matrix**  
*Estimate Elasticity of Population Growth Rate of a Single Matrix*

**Description**

`elasticity3.matrix()` returns the elasticities of lambda to elements of a single matrix. Because this handles only one matrix, the elasticities are inherently deterministic and based on the dominant eigen value as the best metric of the population growth rate. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.
Usage

```r
## S3 method for class 'matrix'
elasticity3(mats, sparse = "auto", ...)
```

Arguments

- `mats`: An object of class `matrix`.
- `sparse`: A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
- `...`: Other parameters.

Value

This function returns a single elasticity matrix.

See Also

- `elasticity3()`
- `elasticity3.lefkoMat()`
- `elasticity3.list()`

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "VSm", "Sb", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
proproxyvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
                         repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
                         immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
                         propstatus = proproxyvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
                         patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
                         juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
                         fecacol = "Intactseed88", deadacol = "Dead1988",
                         nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
                         censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
                          stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
                         ...)
```
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all", stages = c("stage3", "stage2", "stage1"), yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
elasticity3(ehrlen3mean$A[1,1])

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0)
propvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, propstatus = propvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)

Cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"), stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL", "rep", "rep"), eststage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "NA", NA), eststage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL", "rep", "rep"), givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5), type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3), stageframe = cypframe_raw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, 
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), 
size = c("size3added", "size2added"), supplement = cypsupp2r, 
yearcol = "year2", patchcol = "patchid", indivcol = "individ")

elasticity3(cypmatrix2r$A[1][])

---

**flefko2**

*Create Function-based Ahistorical Matrix Projection Model*

**Description**

Function `flefko2()` returns ahistorical MPMs corresponding to the patches and years given, including the associated component transition and fecundity matrices, a data frame detailing the characteristics of the ahistorical stages used, and a data frame characterizing the patch and year combinations corresponding to these matrices. Unlike `rlefko2()` and `rlefko3()`, this function does not currently distinguish populations.

**Usage**

```r
flefko2(
  year = "all",
  patch = "all",
  stageframe, 
  supplement = NA,
  repmatrix = NA,
  overwrite = NA,
  data = NA,
  modelsuite = NA,
  surv_model = NA,
  obs_model = NA,
  size_model = NA,
  repst_model = NA,
  fec_model = NA,
  jsurv_model = NA,
  jobs_model = NA,
  jsize_model = NA,
  jrepst_model = NA,
  paramnames = NA,
  inda = 0,
  indb = 0,
  indc = 0,
  surv_dev = 0,
  obs_dev = 0,
  size_dev = 0,
  repst_dev = 0,
)```
flecko2

fec_dev = 0,
jsurv_dev = 0,
jobs_dev = 0,
jsize_dev = 0,
jreps_dev = 0,
repmod = 1,
yearcol = NA,
patchcol = NA,
year.as.random = FALSE,
patch.as.random = FALSE,
randomseed = NA,
negfec = FALSE,
reduce = FALSE,
err_check = FALSE)

Arguments

year A variable corresponding to year or observation time, or a set of such values, given in values associated with the year term used in linear model development. Can also equal all, in which case matrices will be estimated for all years. Defaults to all.

patch A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to all if matrices should be estimated for all patches. Defaults to all.

stageframe A stageframe object that includes information on the size, observation status, propagule status, immaturity status, and maturity status of each ahistorical stage. Should also incorporate bin widths if size is continuous.

supplement An optional data frame of class lefkoSD that provides supplemental data that should be incorporated into the MPM. Three kinds of data may be integrated this way: transitions to be estimated via the use of proxy transitions, transition overwrites from the literature or supplemental studies, and transition multipliers for fecundity. This data frame should be produced using the supplemental() function. Can be used in place of or in addition to an overwrite table (see overwrite below) and a reproduction matrix (see repmatrix below).

repmatrix A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in supplement. If left blank, then flefko2() will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in supplement above. Must be the dimensions of an ahistorical matrix.

overwrite An optional data frame developed with the overwrite() function describing transitions to be overwritten either with given values or with other estimated
transitions. Note that this function supplements overwrite data provided in supplement.

data
The original historical demographic data frame used to estimate vital rates (class hfvdframe). The original data frame is required in order to initialize years and patches properly.

modelsuite
An optional lefkoMod object holding the vital rate models. If given, then surv_model, obs_model, size_model, repst_model, fec_model, jsurv_model, jobs_model, jsize_model, jrepst_model, paramnames, yearcol, and patchcol are not required. No models should include size or reproductive status in time \( t-1 \).

surv_model
A linear model predicting survival probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any survival probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time \( t \).

obs_model
A linear model predicting sprouting or observation probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any observation probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time \( t \).

size_model
A linear model predicting size. This can be a model of class glm or glmer, both of which require a predicted poisson variable under a log link, or a model of class lm or lmer, in which a Gaussian response is assumed. If given, then will overwrite any size model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time \( t \).

repst_model
A linear model predicting reproduction probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time \( t \).

fec_model
A linear model predicting fecundity. This can be a model of class glm or glmer, and requires a predicted poisson variable under a log link. If given, then will overwrite any fecundity model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time \( t \).

jsurv_model
A linear model predicting juvenile survival probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile survival probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time \( t \).

jobs_model
A linear model predicting juvenile sprouting or observation probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile observation probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time \( t \).

jsize_model
A linear model predicting juvenile size. This can be a model of class glm or glmer, both of which require a predicted poisson variable under a log link, or a model of class lm or lmer, in which a Gaussian response is assumed. If given,
then will overwrite any juvenile size model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

**jrepst_model** A linear model predicting reproduction probability of a mature individual that was immature in the previous year. This can be a model of class glm or glmmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in modelsuite. This model must have been developed in a modeling exercise testing only the impacts of time $t$.

**paramnames** A dataframe with two columns, the first showing the general model terms that will be used in matrix creation, and the second showing the equivalent terms used in modeling. Only required if modelsuite is not supplied.

**inda** A numeric value to use for individual covariate a. Defaults to 0.

**indb** A numeric value to use for individual covariate b. Defaults to 0.

**indc** A numeric value to use for individual covariate c. Defaults to 0.

**surv_dev** A numeric value to be added to the y-intercept in the linear model for survival probability.

**obs_dev** A numeric value to be added to the y-intercept in the linear model for observation probability.

**size_dev** A numeric value to be added to the y-intercept in the linear model for size.

**repst_dev** A numeric value to be added to the y-intercept in the linear model for probability of reproduction.

**fec_dev** A numeric value to be added to the y-intercept in the linear model for fecundity.

**jsurv_dev** A numeric value to be added to the y-intercept in the linear model for juvenile survival probability.

**jobs_dev** A numeric value to be added to the y-intercept in the linear model for juvenile observation probability.

**jsize_dev** A numeric value to be added to the y-intercept in the linear model for juvenile size.

**jrepst_dev** A numeric value to be added to the y-intercept in the linear model for juvenile reproduction probability.

**repmad** A scalar multiplier of fecundity. Defaults to 1.

**yearcol** The variable name or column number corresponding to year in time $t$ in the dataset. Not needed if a modelsuite is supplied.

**patchcol** The variable name or column number corresponding to patch in the dataset. Not needed if a modelsuite is supplied.

**year.as.random** A logical term indicating whether coefficients for missing patches within vital rate models should be estimated as random intercepts. Defaults to FALSE, in which case missing time step coefficients are set to 0.

**patch.as.random** A logical term indicating whether coefficients for missing patches within vital rate models should be estimated as random intercepts. Defaults to FALSE, in which case missing patch coefficients are set to 0.
A numeric value used as a seed to generate random estimates for missing time step and patch coefficients, if either year.as.random or patch.as.random is set to TRUE. Defaults to \texttt{set.seed()} default.

A logical value denoting whether fecundity values estimated to be negative should be reset to 0. Defaults to FALSE.

A logical value denoting whether to remove ahistorical stages associated solely with 0 transitions. These are only removed in cases where the associated row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.

A logical value indicating whether to add matrices of vital rate probabilities associated with each matrix. Defaults to FALSE.

If all inputs are properly formatted, then this function will return either an object of class \texttt{llefkoMat}. Output includes:

- \textbf{A}: A list of full projection matrices in order of sorted patches and years. All matrices output in the \texttt{matrix} class.
- \textbf{U}: A list of survival transition matrices sorted as in \textbf{A}. All matrices output in the \texttt{matrix} class.
- \textbf{F}: A list of fecundity matrices sorted as in \textbf{A}. All matrices output in the \texttt{matrix} class.
- \textbf{hstages}: Set to NA for ahistorical matrices.
- \textbf{agestages}: Set to NA. Only used in output to function \texttt{afleko2}().
- \textbf{ahstages}: A data frame detailing the characteristics of associated ahistorical stages, in the form of a modified stageframe that includes status as an entry stage through reproduction.
- \textbf{labels}: A data frame giving the patch and year of each matrix in order. In \texttt{fleko2()}, only one population may be analyzed at once, and so \texttt{pop = NA}.
- \textbf{matrixqc}: A short vector describing the number of non-zero elements in \textbf{U} and \textbf{F} matrices, and the number of annual matrices.
- \textbf{modelqc}: This is the \texttt{qc} portion of the modelsuite input.
- \textbf{prob_out}: An optional element only added if \texttt{err_check = TRUE}. This is a list of vital rate probability matrices, with 4 columns in the order of survival, observation probability, reproduction probability, and size transition probability.

This function will yield incorrect estimates if the models utilized incorporate state in time $t-1$. Only use models developed testing for ahistorical effects.

The default behavior of this function is to estimate fecundity with regards to transitions specified via associated fecundity multipliers in either supplement or repmatrix. If both of these fields are left empty, then fecundity will be estimated at full for all transitions leading from reproductive stages to immature and propagule stages. However, if a supplement is provided and a repmatrix is not, or if repmatrix is set to 0, then only fecundity transitions noted in the supplement will be
set to non-zero values. To use the default behavior of setting all reproductive stages to reproduce at full fecundity into immature and propagule stages but also incorporate given or proxy survival transitions, input those given and proxy transitions through the overwrite option.

The reproduction matrix (field `repmatrix`) may only be supplied as ahistorical. If provided as historical, then `flefko2()` will fail and produce an error.

Users may at times wish to estimate MPMs using a dataset incorporating multiple patches or subpopulations. Should the aim of analysis be a general MPM that does not distinguish these patches or subpopulations, the `patchcol` variable should be left to NA, which is the default.

Input options including multiple variable names must be entered in the order of variables in time $t+1$ and $t$. Rearranging the order WILL lead to erroneous calculations, and will probably also lead to fatal errors.

Using the `err_check` option will produce a matrix of 4 columns, each characterizing a different vital rate. The product of each row yields an element in the associated $SU$ matrix. The number and order of elements in each column of this matrix matches the associated matrix in column vector format. Use of this option is generally for the purposes of debugging code.

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9)
stagevector <- c("Sd", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
                 "Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
                 "Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
reppvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
proppvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
                          repstatus = reppvector, obsstatus = obsvector, matstatus = matvector,
                          immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
                          propstatus = proppvector)
lathvertln <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
                           patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
                           juvcol = "Seedling1988", sizeacol = "lnVol88", repstracol = "Intactseed88",
                           fecacol = "Intactseed88", deadacol = "Dead1988",
                           nonobsacol = "Dormant1988", stageassign = lathframeln,
                           stagesize = "sizea", censorcol = "Missing1988", censorkeep = NA,
                           NAas0 = TRUE, censor = TRUE)
```
lathvertln$feca2 <- round(lathvertln$feca2)
lathvertln$feca1 <- round(lathvertln$feca1)
lathvertln$feca3 <- round(lathvertln$feca3)

lathmodelsln2 <- modelsearch(lathvertln, historical = FALSE,
                              approach = "mixed", suite = "main",
                              vitalrates = c("surv", "obs", "size", "reps", "fec"),
                              juvestimate = "Sdl",
                              bestfit = "AICc&k", sizedist = "gaussian", fecdist = "poisson",
                              indiv = "individ", patch = "patchid", year = "year2",
                              year.as.random = TRUE, patch.as.random = TRUE, show.model.tables = TRUE,
                              quiet = TRUE)

# Here we use supplemental to provide overwrite and reproductive info
lathsupp2 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl"),
                           stage2 = c("Sd", "Sdl", "rep", "rep"),
                           givenrate = c(0.345, 0.054, NA, NA),
                           multiplier = c(NA, NA, 0.345, 0.054),
                           type = c(1, 1, 3, 3), stageframe = lathframeln, historical = FALSE)

lathmat2ln <- flefko2(year = "all", patch = "all", stageframe = lathframeln,
                       modelsuite = lathmodelsln2, data = lathvertln, supplement = lathsupp2,
                       patchcol = "patchid", yearcol = "year2", year.as.random = FALSE,
                       patch.as.random = FALSE, reduce = FALSE)
summary(lathmat2ln)

---

**flefko3**

*Create Function-based Historical Matrix Projection Model*

**Description**

Function `flefko3()` returns function-based historical MPMs corresponding to the patches and years given, including the associated component transition and fecundity matrices, data frames detailing the characteristics of the ahistorical stages used and historical stage pairs created, and a data frame characterizing the patch and year combinations corresponding to these matrices. Unlike `rlefko3()`, this function currently does not distinguish populations within the same dataset.

**Usage**

```r
flefko3(
  year = "all",
  patch = "all",
  stageframe,
  supplement = NA,
  repmatrix = NA,
  overwrite = NA,
  data = NA,
)```
modelsuite = NA,
surv_model = NA,
obs_model = NA,
size_model = NA,
repest_model = NA,
fec_model = NA,
jsurv_model = NA,
jobs_model = NA,
jsize_model = NA,
jrepest_model = NA,
paramnames = NA,
inda = 0,
indb = 0,
indc = 0,
surv_dev = 0,
obs_dev = 0,
size_dev = 0,
repest_dev = 0,
fec_dev = 0,
jsurv_dev = 0,
jobs_dev = 0,
jsize_dev = 0,
jrepest_dev = 0,
repmod = 1,
yearcol = NA,
patchcol = NA,
year.as.random = FALSE,
patch.as.random = FALSE,
randomseed = NA,
negfec = FALSE,
format = "ehrlen",
reduce = FALSE,
err_check = FALSE
}

Arguments

year A variable corresponding to year or observation time, or a set of such values, given in values associated with the year term used in linear model development. Can also equal all, in which case matrices will be estimated for all years. Defaults to all.

patch A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to all if matrices should be estimated for all patches. Defaults to all.

stageframe A stageframe object that includes information on the size, observation status, propagule status, immaturity status, and maturity status of each ahistorical stage. Should also incorporate bin widths if size is continuous.

supplement An optional data frame of class lefkoSD that provides supplemental data that
should be incorporated into the MPM. Three kinds of data may be integrated this way: transitions to be estimated via the use of proxy transitions, transition overwrites from the literature or supplemental studies, and transition multipliers for fecundity. This data frame should be produced using the `supplemental()` function. Can be used in place of or in addition to an overwrite table (see overwrite below) and a reproduction matrix (see repmatrix below).

**repmatrix**
A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equalling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in supplement. If left blank, then `flefko3()` will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in supplement above. May be the dimensions of either a historical or an ahistorical matrix. If the latter, then all stages will be used in time \( t-1 \) for each suggested ahistorical transition.

**overwrite**
An optional data frame developed with the `overwrite()` function describing transitions to be overwritten either with given values or with other estimated transitions. Note that this function supplements overwrite data provided in supplement.

**data**
The historical vertical demographic data frame used to estimate vital rates (class `hfvdata`), which is required to initialize years and patches properly.

**modelsuite**
An optional `lefkoMod` object holding the vital rate models. If given, then `surv_model`, `obs_model`, `size_model`, `repst_model`, `fec_model`, `jsurv_model`, `jobs_model`, `jsize_model`, `jrepst_model`, `paramnames`, `yearcol`, and `patchcol` are not required. One or more of these models should include size or reproductive status in time \( t-1 \).

**surv_model**
A linear model predicting survival probability. This can be a model of class `glm` or `glmer`, and requires a predicted binomial variable under a logit link. If given, then will overwrite any survival probability model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times \( t \) and \( t-1 \).

**obs_model**
A linear model predicting sprouting or observation probability. This can be a model of class `glm` or `glmer`, and requires a predicted binomial variable under a logit link. If given, then will overwrite any observation probability model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times \( t \) and \( t-1 \).

**size_model**
A linear model predicting size. This can be a model of class `glm` or `glmer`, both of which require a predicted poisson variable under a log link, or a model of class `lm` or `lmer`, in which a Gaussian response is assumed. If given, then will overwrite any size model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times \( t \) and \( t-1 \).

**repst_model**
A linear model predicting reproduction probability. This can be a model of class `glm` or `glmer`, and requires a predicted binomial variable under a logit link.
link. If given, then will overwrite any reproduction probability model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times $t$ and $t-1$.

**fec_model**
A linear model predicting fecundity. This can be a model of class glm or glmer, and requires a predicted poisson variable under a log link. If given, then will overwrite any fecundity model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times $t$ and $t-1$.

**jsurv_model**
A linear model predicting juvenile survival probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile survival probability model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times $t$ and $t-1$.

**jobs_model**
A linear model predicting juvenile sprouting or observation probability. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile observation probability model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times $t$ and $t-1$.

**jsize_model**
A linear model predicting juvenile size. This can be a model of class glm or glmer, both of which require a predicted poisson variable under a log link, or a model of class lm or lmer, in which a Gaussian response is assumed. If given, then will overwrite any juvenile size model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times $t$ and $t-1$.

**jrepst_model**
A linear model predicting reproduction probability of a mature individual that was immature in the previous year. This can be a model of class glm or glmer, and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in modelsuite. This model must have been developed in a modeling exercise testing the impacts of times $t$ and $t-1$.

**paramnames**
A dataframe with two columns, the first showing the general model terms that will be used in matrix creation, and the second showing the equivalent terms used in modeling. Only required if modelsuite is not supplied.

**inda**
A numeric value to use for individual covariate a. Defaults to 0.

**indb**
A numeric value to use for individual covariate b. Defaults to 0.

**indc**
A numeric value to use for individual covariate c. Defaults to 0.

**surv_dev**
A numeric value to be added to the y-intercept in the linear model for survival probability.

**obs_dev**
A numeric value to be added to the y-intercept in the linear model for observation probability.

**size_dev**
A numeric value to be added to the y-intercept in the linear model for size.

**repest_dev**
A numeric value to be added to the y-intercept in the linear model for probability of reproduction.

**fec_dev**
A numeric value to be added to the y-intercept in the linear model for fecundity.

**jsurv_dev**
A numeric value to be added to the y-intercept in the linear model for juvenile survival probability.
jobs_dev  A numeric value to be added to the y-intercept in the linear model for juvenile observation probability.

jsize_dev  A numeric value to be added to the y-intercept in the linear model for juvenile size.

jrepst_dev A numeric value to be added to the y-intercept in the linear model for juvenile reproduction probability.

repmode  A scalar multiplier of fecundity. Defaults to 1.

yearcol The variable name or column number corresponding to year in time t in the dataset. Not needed if modelsuite is supplied.

patchcol The variable name or column number corresponding to patch in the dataset. Not needed if modelsuite is supplied.

year.as.random A logical term indicating whether coefficients for missing patches within vital rate models should be estimated as random intercepts. Defaults to FALSE, in which case missing time step coefficients are set to 0.

patch.as.random A logical term indicating whether coefficients for missing patches within vital rate models should be estimated as random intercepts. Defaults to FALSE, in which case missing patch coefficients are set to 0.

randomseed A numeric value used as a seed to generate random estimates for missing time step and patch coefficients, if either year.as.random or patch.as.random is set to TRUE. Defaults to set.seed() default.

negfec A logical value denoting whether fecundity values estimated to be negative should be reset to 0. Defaults to FALSE.

format A string indicating whether to estimate matrices in ehr1en format or deVries format. The latter adds one extra prior stage to account for the prior state of newborns. Defaults to ehr1en format.

reduce A logical value denoting whether to remove historical stages associated solely with 0 transitions. These are only removed in cases where the associated row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.

err_check A logical value indicating whether to append matrices of vital rate probabilities associated with each matrix. These matrices are developed internally and can be used for erroc checking. Defaults to FALSE.

Value

If all inputs are properly formatted, then this function will return an object of class lefkoMat, which is a list with the following elements:

A A list of full projection matrices in order of sorted patches and years. All matrices output in the matrix class.

U A list of survival transition matrices sorted as in A. All matrices output in the matrix class.

F A list of fecundity matrices sorted as in A. All matrices output in the matrix class.
hstages  A data frame matrix showing the pairing of ahistorical stages used to create historical stage pairs.

agestages  Set to NA. Only used in output to function aflefko2().

ahstages  A data frame detailing the characteristics of associated ahistorical stages, in the form of a modified stageframe that includes status as an entry stage through reproduction.

labels  A data frame showing the patch and year of each matrix in order. In flefko3(), only one population may be analyzed at once, and so pop = NA.

matrixqc  A short vector describing the number of non-zero elements in U and F matrices, and the number of annual matrices.

modelqc  This is the qc portion of the modelsuite input.

prob_out  An optional element only added if err_check = TRUE. This is a list of vital rate probability matrices, with 4 columns in the order of survival, observation probability, reproduction probability, and size transition probability.

Notes

The default behavior of this function is to estimate fecundity with regards to transitions specified via associated fecundity multipliers in either supplement or repmatrix. If both of these fields are left empty, then fecundity will be estimated at full for all transitions leading from reproductive stages to immature and propagule stages. However, if a supplement is provided and a repmatrix is not, or if repmatrix is set to 0, then only fecundity transitions noted in the supplement will be set to non-zero values. To use the default behavior of setting all reproductive stages to reproduce at full fecundity into immature and propagule stages but incorporate given or proxy survival transitions, input those given and proxy transitions through the overwrite option.

The reproduction matrix (field repmatrix) may be supplied as either historical or ahistorical. If provided as ahistorical, then flefko3() will assume that all historical transitions involving stages noted for times $t$ and $t+1$ should be set to the respective fecundity multipliers noted.

Users may at times wish to estimate MPMs using a dataset incorporating multiple patches or subpopulations. Should the aim of analysis be a general MPM that does not distinguish these patches or subpopulations, the patchcol variable should be left to NA, which is the default.

Input options including multiple variable names must be entered in the order of variables in time $t+1$, $t$, and $t-1$. Rearranging the order WILL lead to erroneous calculations, and will probably also lead to fatal errors.

Using the err_check option will produce a matrix of 4 columns, each characterizing a different vital rate. The product of each row yields an element in the associated $U$ matrix. The number and order of elements in each column of this matrix matches the associated matrix in column vector format. Use of this option is generally for the purposes of debugging code.

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8,
9) stagevector <- c("Sd", "SdL", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", 
"Sz4nr", "Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r", 
"Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
obsvector <- c(0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
matvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
imvvector <- c(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector, 
repstatus = repvector, obsstatus = obsvector, matstatus = matvector, 
imstatus = immvector, indataset = indataset, binhalfwidth = binvec, 
propstatus = propvector)
lathvertln <- verticalize3(lathyrus, noyears = 4, firstyear = 1988, 
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9, 
juvcol = "Seedling1988", sizeacol = "lnVol88", repstracol = "Intactseed88", 
fecacol = "Intactseed88", deadacol = "Dead1988", 
onobscol = "Dormant1988", stageassign = lathframeln, stagesize = "sizea", 
censorcol = "Missing1988", censorkeep = NA, NAas0 = TRUE, censor = TRUE)
lathvertln$feca2 <- round(lathvertln$feca2)
lathvertln$feca1 <- round(lathvertln$feca1)
lathvertln$feca3 <- round(lathvertln$feca3)
lathomodelsln3 <- modelsearch(lathvertln, historical = TRUE, 
approach = "mixed", suite = "main", 
vitalrates = c("surv", "obs", "size", "repst", "fec"), juvestimate = "Sdl", 
bestfit = "AICc&k", sizedist = "gaussian", fecdist = "poisson", 
div = "individ", patch = "patchid", year = "year2", year.as.random = TRUE, 
patch.as.random = TRUE, show.model.tables = TRUE, quiet = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "SdL", "SdL", "mat", "Sd", "SdL"), 
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep"), 
stage1 = c("Sd", "rep", "Sd", "rep", "Sd", "mat", "mat"), 
eststage3 = c(NA, NA, NA, NA, NA, NA, NA), 
eststage2 = c(NA, NA, NA, NA, NA, NA, NA), 
eststage1 = c(NA, NA, NA, NA, NA, NA, NA), 
givenrate = c(0.345, 0.345, 0.054, 0.054, 0.054, 0.054, NA, NA, NA), 
multiplier = c(NA, NA, NA, NA, NA, NA, NA), 
type = c(1, 1, 1, 1, 1, 1, 1), 
type_t12 = c(1, 2, 1, 2, 1, 1, 1), 
stageframe = lathframeln, historical = TRUE)
lathmat3ln <- flefko3(year = "all", patch = "all", stageframe = lathframeln, 
modelsuite = lathomodelsln3, data = lathvertln, supplement = lathsupp3, 
patchcol = "patchid", yearcol = "year2", year.as.random = FALSE, 
patch.as.random = FALSE, reduce = FALSE)
historicalize3 Create Historical Vertical Data Frame From Ahistorical Vertical Data Frame

Description

historicalize3() returns a vertically formatted demographic data frame organized to create historical projection matrices, given a vertically but ahistorically formatted data frame. This data frame is in standard \texttt{lefko3} format and can be used in all functions in the package.

Usage

```r
historicalize3(data, 
  popidcol = 0, 
  patchidcol = 0, 
  individcol, 
  year2col = 0, 
  year3col = 0, 
  xcol = 0, 
  ycol = 0, 
  sizea2col = 0, 
  sizea3col = 0, 
  sizeb2col = 0, 
  sizeb3col = 0, 
  sizec2col = 0, 
  sizec3col = 0, 
  repstra2col = 0, 
  repstra3col = 0, 
  repstrb2col = 0, 
  repstrb3col = 0, 
  feca2col = 0, 
  feca3col = 0, 
  fecb2col = 0, 
  fecb3col = 0, 
  indcova2col = 0, 
  indcova3col = 0, 
  indcovb2col = 0, 
  indcovb3col = 0, 
  indcovc2col = 0, 
  indcovc3col = 0, 
  alive2col = 0, 
  alive3col = 0,
```
dead2col = 0,
dead3col = 0,
obsv2col = 0,
obsv3col = 0,
nonobs2col = 0,
nonobs3col = 0,
repstrrel = 1,
fecrel = 1,
stage2col = 0,
stage3col = 0,
juv2col = 0,
juv3col = 0,
stageassign = NA,
stagesize = NA,
censor = FALSE,
censorcol = 0,
censorkeep = 0,
spacing = NA,
NAas0 = FALSE,
NRasRep = FALSE,
reduce = TRUE
)

Arguments

data

The horizontal data file.

popidcol

A variable name or column number corresponding to the identity of the population for each individual.

patchidcol

A variable name or column number corresponding to the identity of the patch for each individual, if patches have been designated within populations.

individcol

A variable name or column number corresponding to the unique identity of each individual.

year2col

A variable name or column number corresponding to the year or time in time t.

year3col

A variable name or column number corresponding to the year or time in time t+1.

xcol

A variable name or column number corresponding to the x coordinate of each individual in Cartesian space.

ycol

A variable name or column number corresponding to the y coordinate of each individual in Cartesian space.

sizea2col

A variable name or column number corresponding to the primary size entry in time t.

sizea3col

A variable name or column number corresponding to the primary size entry in time t+1.

sizeb2col

A variable name or column number corresponding to the secondary size entry in time t.
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sizeb3col</td>
<td>A variable name or column number corresponding to the secondary size entry in time ( t+1 ).</td>
</tr>
<tr>
<td>sizec2col</td>
<td>A variable name or column number corresponding to the tertiary size entry in time ( t ).</td>
</tr>
<tr>
<td>sizec3col</td>
<td>A variable name or column number corresponding to the tertiary size entry in time ( t+1 ).</td>
</tr>
<tr>
<td>repstra2col</td>
<td>A variable name or column number corresponding to the production of reproductive structures, such as flowers, in time ( t ). This can be binomial or count data, and is used to in analysis of the probability of reproduction.</td>
</tr>
<tr>
<td>repstra3col</td>
<td>A variable name or column number corresponding to the production of reproductive structures, such as flowers, in time ( t+1 ). This can be binomial or count data, and is used to in analysis of the probability of reproduction.</td>
</tr>
<tr>
<td>repstrb2col</td>
<td>A second variable name or column number corresponding to the production of reproductive structures, such as flowers, in time ( t ). This can be binomial or count data.</td>
</tr>
<tr>
<td>repstrb3col</td>
<td>A second variable name or column number corresponding to the production of reproductive structures, such as flowers, in time ( t+1 ). This can be binomial or count data.</td>
</tr>
<tr>
<td>feca2col</td>
<td>A variable name or column number corresponding to fecundity in time ( t ). This may represent egg counts, fruit counts, seed production, etc.</td>
</tr>
<tr>
<td>feca3col</td>
<td>A variable name or column number corresponding to fecundity in time ( t+1 ). This may represent egg counts, fruit counts, seed production, etc.</td>
</tr>
<tr>
<td>fecb2col</td>
<td>A second variable name or column number corresponding to fecundity in time ( t ). This may represent egg counts, fruit counts, seed production, etc.</td>
</tr>
<tr>
<td>fecb3col</td>
<td>A second variable name or column number corresponding to fecundity in time ( t+1 ). This may represent egg counts, fruit counts, seed production, etc.</td>
</tr>
<tr>
<td>indcova2col</td>
<td>A variable name or column number corresponding to an individual covariate to be used in analysis, in time ( t ).</td>
</tr>
<tr>
<td>indcova3col</td>
<td>A variable name or column number corresponding to an individual covariate to be used in analysis, in time ( t+1 ).</td>
</tr>
<tr>
<td>indcovb2col</td>
<td>A second variable name or column number corresponding to an individual covariate to be used in analysis, in time ( t ).</td>
</tr>
<tr>
<td>indcovb3col</td>
<td>A second variable name or column number corresponding to an individual covariate to be used in analysis, in time ( t+1 ).</td>
</tr>
<tr>
<td>indcovic2col</td>
<td>A third variable name or column number corresponding to an individual covariate to be used in analysis, in time ( t ).</td>
</tr>
<tr>
<td>indcovic3col</td>
<td>A third variable name or column number corresponding to an individual covariate to be used in analysis, in time ( t+1 ).</td>
</tr>
<tr>
<td>alive2col</td>
<td>A variable name or column number that provides information on whether an individual is alive in time ( t ). If used, living status must be designated as binomial (living = 1, dead = 0).</td>
</tr>
<tr>
<td>alive3col</td>
<td>A variable name or column number that provides information on whether an individual is alive in time ( t+1 ). If used, living status must be designated as binomial (living = 1, dead = 0).</td>
</tr>
</tbody>
</table>
dead2col A variable name or column number that provides information on whether an individual is dead in time $t$. If used, dead status must be designated as binomial (dead = 1, living = 0).

dead3col A variable name or column number that provides information on whether an individual is dead in time $t+1$. If used, dead status must be designated as binomial (dead = 1, living = 0).

obs2col A variable name or column number providing information on whether an individual is in an observable stage in time $t$. If used, observation status must be designated as binomial (observed = 1, not observed = 0).

obs3col A variable name or column number providing information on whether an individual is in an observable stage in time $t+1$. If used, observation status must be designated as binomial (observed = 1, not observed = 0).

nonobs2col A variable name or column number providing information on whether an individual is in an unobservable stage in time $t$. If used, observation status must be designated as binomial (not observed = 1, observed = 0).

nonobs3col A variable name or column number providing information on whether an individual is in an unobservable stage in time $t+1$. If used, observation status must be designated as binomial (not observed = 1, observed = 0).

repstrrel This is a scalar multiplier to make the variable represented by repstrb2col equivalent to the variable represented by repstra2col. This can be useful if two reproductive status variables have related but unequal units, for example if repstrb2col refers to one-flowered stems while repstra2col refers to two-flowered stems.

fecrel This is a scalar multiplier that makes the variable represented by fecb2col equivalent to the variable represented by feca2col. This can be useful if two fecundity variables have related but unequal units.

stage2col Optional variable name or column number corresponding to life history stage in time $t$.

stage3col Optional variable name or column number corresponding to life history stage in time $t+1$.

juv2col A variable name or column number that marks individuals in immature stages in time $t$. The `historicalize3()` function assumes that immature individuals are identified in this variable marked with a number equal to or greater than 1, and that mature individuals are marked as 0 or NA.

juv3col A variable name or column number that marks individuals in immature stages in time $t+1$. The `historicalize3()` function assumes that immature individuals are identified in this variable marked with a number equal to or greater than 1, and that mature individuals are marked as 0 or NA.

stageassign The stageframe object identifying the life history model being operationalized. Note that if stage2col is provided, then this stageframe is not utilized in stage designation.

stagesize A variable name or column number describing which size variable to use in stage estimation. Defaults to NA, and can also take sizea, sizeb, sizec, or sizeadded, depending on which size variable is chosen.
A logical variable determining whether the output data should be censored using the variable defined in censorcol. Defaults to FALSE.

censorcol

A variable name or column number corresponding to a censor variable within the dataset, used to distinguish between entries to use and those to discard from analysis, or to designate entries with special issues that require further attention.

censorkeep

The value of the censoring variable identifying data that should be included in analysis. Defaults to 0, but may take any value including NA.

spacing

The spacing at which density should be estimated, if density estimation is desired and x and y coordinates are supplied. Given in the same units as those used in the x and y coordinates given in xcol and ycol. Defaults to NA.

NAas0

If TRUE, then all NA entries for size and fecundity variables will be set to 0. This can help increase the sample size analyzed by modelsearch(), but should only be used when it is clear that this substitution is biologically realistic. Defaults to FALSE.

NRasRep

If set to TRUE, then this function will treat non-reproductive but mature individuals as reproductive during stage assignment. This can be useful when a matrix is desired without separation of reproductive and non-reproductive but mature stages of the same size. Only used if stageassign is set to a stageframe. Defaults to FALSE.

reduce

A logical variable determining whether unused variables and some invariant state variables should be removed from the output dataset. Defaults to TRUE.

Value

If all inputs are properly formatted, then this function will output a historical vertical data frame (class hfvdata), meaning that the output data frame will have three consecutive years of size and reproductive data per individual per row. This data frame is in standard format for all functions used in lefko3, and so can be used without further modification. Note that determination of state in times *t*-1 and *t*+1 gives preference to condition in time *t* within the input dataset. Conflicts in condition in input datasets that have both times *t* and *t*+1 listed per row are resolved by using condition in time *t*.

Variables in this data frame include the following:

rowid

Unique identifier for the row of the data frame.

popid

Unique identifier for the population, if given.

patchid

Unique identifier for patch within population, if given.

individ

Unique identifier for the individual.

year2

Year or time at time t.

firstseen

Year or time of first observation.

lastseen

Year or time of last observation.

obsage

Observed age in time t, assuming first observation corresponds to age = 0.

obslifespan

Observed lifespan, given as lastseen - firstseen + 1.

xpos1,xpos2,xpos3

X position in Cartesian space in times t-1, t, and t+1, respectively, if provided.
ypos1, ypos2, ypos3
Y position in Cartesian space in times t-1, t, and t+1, respectively, if provided.

sizea1, sizea2, sizea3
Main size measurement in times t-1, t, and t+1, respectively.

sizeb1, sizeb2, sizeb3
Secondary size measurement in times t-1, t, and t+1, respectively.

sizec1, sizec2, sizec3
Tertiary measurement in times t-1, t, and t+1, respectively.

size1added, size2added, size3added
Sum of primary, secondary, and tertiary size measurements in times t-1, t, and t+1, respectively.

repstra1, repstra2, repstra3
Main numbers of reproductive structures in times t-1, t, and t+1, respectively.

repstrb1, repstrb2, repstrb3
Secondary numbers of reproductive structures in times t-1, t, and t+1, respectively.

repstr1added, repstr2added, repstr3added
Sum of primary and secondary reproductive structures in times t-1, t, and t+1, respectively.

fec1, fec2, fec3
Main numbers of offspring in times t-1, t, and t+1, respectively.

fecb1, fecb2, fecb3
Secondary numbers of offspring in times t-1, t, and t+1, respectively.

fec1added, fec2added, fec3added
Sum of primary and secondary fecundity in times t-1, t, and t+1, respectively.

censor1, censor2, censor3
Censor state values in times t-1, t, and t+1, respectively.

juvgiven1, juvgiven2, juvgiven3
Binomial variable indicating whether individual is juvenile in times t-1, t, and t+1. Only given if juvcol is provided.

obsstatus1, obsstatus2, obsstatus3
Binomial observation state in times t-1, t, and t+1, respectively.

repstatus1, repstatus2, repstatus3
Binomial reproductive state in times t-1, t, and t+1, respectively.

fecstatus1, fecstatus2, fecstatus3
Binomial offspring production state in times t-1, t, and t+1, respectively.

matstatus1, matstatus2, matstatus3
Binomial maturity state in times t-1, t, and t+1, respectively.

alive1, alive2, alive3
Binomial state as alive in times t-1, t, and t+1, respectively.

density
Density of individuals per unit designated in spacing. Only given if spacing is not NA.
Notes

In some datasets on species with unobservable stages, observation status (obsstatus) might not be inferred properly if a single size variable is used that does not yield sizes greater than 0 in all cases in which individuals were observed. Such situations may arise, for example, in plants when leaf number is the dominant size variable used, but individuals occasionally occur with inflorescences but no leaves. In this instances, it helps to mark related variables as sizeb and sizec, because observation status will be interpreted in relation to all 3 size variables. Further analysis can then utilize only a single size variable, of the user’s choosing. Similar issues can arise in reproductive status (repstatus).

Warnings that some individuals occur in state combinations that do not match any stages in the stageframe used to assign stages are common when first working with a dataset. Typically, these situations can be identified as NoMatch entries in stage3, although such entries may crop up in stage1 and stage2, as well. In rare cases, these warnings will arise with no concurrent NoMatch entries, which indicates that the input dataset contained conflicting state data at once suggesting that the individual is in some stage but is also dead. The latter is removed if the conflict occurs in time t or time t-1, as only living entries are allowed in these times.

Care should be taken to avoid variables with negative values indicating size, fecundity, or reproductive or observation status. Negative values can be interpreted in different ways, typically reflecting estimation through other algorithms rather than actual measured data. Variables holding negative values can conflict with data management algorithms in ways that are difficult to predict.

Examples

data(cypvert)
sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 1, 1, 1, 1)
matvector <- c(0, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
                          repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                          propstatus = propvector, immstatus = immvector, indataset = indataset,
                          binhalfwidth = binvec)
cypframe_raw

historicalize3(data = cypvert, patchidcol = "patch",
                individcol = "plantid", year2col = "year2", sizea2col = "Inf2.2",
                sizea3col = "Inf2.3", sizeb2col = "Inf.2", sizeb3col = "Inf.3",
                sizec2col = "Veg.2", sizec3col = "Veg.3", repstra2col = "Inf2.2",
                repstra3col = "Inf2.3", repstrb2col = "Inf.2", repstrb3col = "Inf.3",
                feca2col = "Pod.2", feca3col = "Pod.3", repstrel = 2,
                stageassign = cypframe_raw, stagesize = "sizeadded", censorcol = "censor",
censor = FALSE, NAas0 = TRUE, NRasRep = TRUE, reduce = TRUE)

cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
  type =c(1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
  stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v2, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2"),
  size = c("size3added", "size2added"), supplement = cypsupp2r,
  yearcol = "year2", patchcol = "patchid", individcol = "individ")

cypmatrix2r$A[intersect(which(cypmatrix2r$labels$patch == "A"),
  which(cypmatrix2r$labels$year2 == 2004))]

lambda3(cypmatrix2r)

---

image3 Create Matrix Image

Description

Function image3() is a generic function that creates matrix plots. It acts as a wrapper for the
image() function in package SparseM, conducting all necessary conversions and automating image production across all or just specific matrices.

Usage

image3(mats, ...)

Arguments

mats A lefkoMat object, or a single projection matrix, for which the dominant eigenvalue is desired.

... Other parameters

Value

Produces a single matrix image, or a series of images, depending on the input. Non-zero elements appear as red space, while zero elements appear as white space.
See Also

image3.lefkoMat()
image3.matrix()

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sdl", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

image3(ehrlen3, used = 1, type = "U")

# Cypripedium example
rm(list = ls(all = TRUE))
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
matvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0)
immvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
reppstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
reppstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "D", 
"XSm", "Sm", "SD", "P1"),
stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep", 
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, 0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cypframe_raw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("sizeadded", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")
image3(cypmatrix2r, used = 1, type = "U")

# Description

Function image3.lefkoElas plots matrix images for elasticity matrices supplied within lefkoElas objects. This function operates as a wrapper the image() function in package SparseM, conducting
all necessary conversions and automating image production across all or just specific matrices.

Usage

```r
## S3 method for class 'lefkolas'
image3(mats, used = "all", type = "a", ...)
```

Arguments

- `mats`: A lefkolas object.
- `used`: A numeric value or vector designating the matrices to plot. Can also take the value "all", which plots all matrices. Defaults to "all".
- `type`: Character value indicating whether to plot "a"historical or "h"istorical elasticity matrices. Defaults to "a"historical, but will plot a historical elasticity matrix image if no ahistorical elasticity matrix exists.
- `...`: Other parameters.

Value

Plots a matrix image, or series of matrix images, denoting non-zero elements as red space and zero elements as white space.

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
 repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
 immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
 propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
parchicocol = "SUBPLOT", individcol = "GENET", blocksize = 9,
 nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
 stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
...
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe,
year = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
supplement = lathsupp3, yearcol = "year2", indivcol = "individ")
ehrlen_elas <- elasticity3(ehrlen3)
image3(ehrlen_elas, used = 1, type = "h")

# Cypripedium example
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5, NA, NA, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = lathframe, historical = TRUE)
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsupp2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cypelas <- elasticity3(cypmatrix2r)

image3(cypelas, used = 1, type = "a")

---

image3.lefkoMat  Create Matrix Image(s) for lefkoMat Object

Description

Function image3.lefkoMat plots matrix images for matrices supplied within lefkoMat objects. This function operates as a wrapper for the image() function in package SparseM, conducting all necessary conversions and automating image production across all or just specific matrices.

Usage

## S3 method for class 'lefkoMat'
image3(mats, used = "all", type = "A", ...)

Arguments

<table>
<thead>
<tr>
<th>mats</th>
<th>A lefkoMat object.</th>
</tr>
</thead>
<tbody>
<tr>
<td>used</td>
<td>A numeric value or vector designating the matrices to plot. Can also take the value &quot;all&quot;, which plots all matrices. Defaults to &quot;all&quot;.</td>
</tr>
<tr>
<td>type</td>
<td>Character value indicating whether to plot A, U, or F matrices. Defaults to &quot;A&quot;.</td>
</tr>
<tr>
<td>...</td>
<td>Other parameters.</td>
</tr>
</tbody>
</table>

Value

Plots a matrix image, or series of matrix images, denoting non-zero elements as red space and zero elements as white space.

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repevector <- c(0, 0, 0, 0, 1, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 183, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)
lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
 censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sdl", "Sdl", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

# Cypripedium example
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, 
patchidcol = "patch", individcol = "plantid", blocksize = 4, 
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", 
repsracoal = "Inf.04", repstrbcocol = "Inf2.04", fecacol = "Pod.04", 
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, 
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info 
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", 
"XSm", "Sm", "SD", "P1"), 
"rep"), 
eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA), 
eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA), 
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA), 
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), 
type =c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1), 
stageframe = cypframe_raw, historical = FALSE)

cymatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, 
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), 
size = c("size3added", "size2added"), supplement = cypsupp2r, 
yearcol = "year2", patchcol = "patchid", indivcol = "individ")

image3(cymatrix2r, used = 1, type = "U")

---

### image3.lefkoSens

Create Matrix Image(s) for lefkoSens Object

**Description**

Function `image3.lefkoSens` plots matrix images for sensitivity matrices supplied within `lefkoSens` objects. This function operates as a wrapper the `image()` function in package SparseM, conducting all necessary conversions and automating image production across all or just specific matrices.

**Usage**

```r
## S3 method for class 'lefkoSens'
image3(mats, used = "all", type = "a", ...)
```

**Arguments**

- `mats`: A `lefkoSens` object.
- `used`: A numeric value or vector designating the matrices to plot. Can also take the value "all", which plots all matrices. Defaults to "all".
type

Character value indicating whether to plot "a" historical or "h" historical sensitivity matrices. Defaults to "a" historical, but will plot a historical sensitivity matrix image if no ahistorical sensitivity matrix exists.

Other parameters.

Value

Plots a matrix image, or series of matrix images, denoting non-zero elements as red space and zero elements as white space.

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
reppstatus = reppvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe,
year = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
supplement = lathsupp3, yearcol = "year2", indivcol = "individ"
ehrren_sensi  <- sensitivity3(ehrren3)

image3(ehrren_sensi, used = 1, type = "h")

# Cypripedium example
data(cypdata)

sizevector  <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector  <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector  <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector  <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector  <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector  <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propropvector  <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset  <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec  <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw  <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, propstatus = propropvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec)

cypraw_v1  <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r  <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"), stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL", "rep", "rep"), eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA), eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA), givenrate = c(0, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA), type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3), stageframe = cypframe_raw, historical = FALSE)

cympmatrix2r  <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), size = c("size3added", "size2added"), supplement = cypsupp2r, yearcol = "year2", patchcol = "patchid", individcol = "individ")

cypsens  <- sensitivity3(cympmatrix2r)
image3(cypsens, used = 1, type = "a")
Description

Function `image3.matrix` plots matrix images for matrices contained in a list of matrices. This function operates as a wrapper for the `image()` function in package `SparseM`, conducting all necessary conversions and automating image production across all or just specific matrices.

Usage

```r
## S3 method for class 'list'
image3(mats, used = "all", ...)
```

Arguments

- `mats`: A list class object.
- `used`: A numeric vector of projection matrices within `mats` to represent as matrix images. Can also take the text value "all", which will produce images of all matrices. Defaults to "all".
- `...`: Other parameters.

Value

Plots a matrix image, or series of matrix images, denoting non-zero elements as red space and zero elements as white space.

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
revector <- c(0, 0, 0, 0, 1, 0)
obsevector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propropvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = revector, obsstatus = obsevector, matstatus = matvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propropvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
```
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sd1", "Sd1", "Sd", "Sd1"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep"),
stage1 = c("Sd", "rep", "Sd", "rep", "all", "all"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054),
type = c(1, 1, 1, 1, 3, 3), type_t12 = c(1, 2, 1, 2, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"),
supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")

image3(ehrlen3$A, used = 1)

# Cypripedium example
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reepvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypraw_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = reepvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypraw_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cyupsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep", "rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA)
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cypframe_raw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")
image3(cypmatrix2r$A, used = 1)

---

**image3.matrix**

Create a Matrix Image for a Single Matrix

**Description**

Function `image3.matrix` plots a matrix image for a single matrix. This function operates as a wrapper for the `image()` function in package `SparseM`, conducting all necessary conversions and automating image production across all or just specific matrices.

**Usage**

```r
# S3 method for class 'matrix'
image3(mats, ...)
```

**Arguments**

- `mats`: A matrix class object.
- `...`: Other parameters.

**Value**

Plots a matrix image, or series of matrix images, denoting non-zero elements as red space and zero elements as white space.

**Examples**

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
```
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immedstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

image3(ehrlen3$U[[1]])

# Cypripedium example
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
lambda3

sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 3),
stageframe = cypframe_raw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")

image3(cypmatrix2r$U[[1]])

---

**lambda3**

Estimate Dominant Eigenvalue and Deterministic Population Growth Rate

**Description**

lambda3() is a generic function that returns the dominant eigenvalue of a matrix, and set of dominant eigenvalues of a set of matrices. It can handle very large and sparse matrices supplied as lefkoMat objects or as individual matrices, and can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

**Usage**

lambda3(mats, ...)

**Arguments**

- **mats** A lefkoMat object, or a single projection matrix, for which the dominant eigenvalue is desired.
- **...** Other parameters.

**Value**

The value returned depends on the class of the mats argument.
See Also

- `lambda3.lefkoMat()`
- `lambda3.matrix()`
- `slambda3()`

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 1, 1, 1, 1, 1, 1)
immvector <- c(1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
                        repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                        immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
                        propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
                         patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
                         juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
                         fecacol = "Intactseed88", deadacol = "Dead1988",
                         nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
                         censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "mat"),
                           stage2 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "Sdl"),
                           stage1 = c("Sd", "rep", "Sdl", "rep", "Sdl"),
                           eststage3 = c(NA, NA, NA, NA, NA, "mat"),
                           eststage2 = c(NA, NA, NA, NA, NA, "Sdl"),
                           eststage1 = c(NA, NA, NA, NA, NA, "NotAlive"),
                           givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
                           multiplier = c(NA, NA, NA, 0.345, 0.054, NA),
                           type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1),
                           stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
                   stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
                   yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
lambda3(ehrlen3mean)

# Cypripedium example
```
```
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "SD", "P1"),
stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep", "rep"),
eststage3 = c(NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")

lambda3(cypmatrix2r)
```
Description

`lambda3.lefkoMat()` returns the dominant eigenvalues of all projection matrices supplied within `lefkoMat` objects. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

Usage

```r
## S3 method for class 'lefkoMat'
lambda3(mats, sparse = "auto", ...)
```

Arguments

- `mats`: An object of class `lefkoMat`.
- `sparse`: A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
- `...`: Other parameters.

Value

This function returns the dominant eigenvalue of each $A$ matrix in `mats`. The output includes a data frame showing the population, patch, and lambda estimate for each $A$ matrix. Row names correspond to the order of the matrix within the $A$ element of `mats`.

Notes

The `sparse` option allows the function to utilize underlying methods of either dense or sparse matrix manipulation in order to speed up processing time and prevent memory shortages. Under the auto setting, the function will determine whether the matrix is sparse and act accordingly. For extremely large, sparse matrices, the user may simply set `sparse = "yes"` to save time further and force the use of sparse format in calculations.

See Also

- `lambda3()`
- `lambda3.matrix()`
- `slambda3()`

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 0)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
```
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA, NA, NA, NA, NA, NA),
multiplier = c(NO, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
lambda3(ehrlen3mean)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
"Xlg")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 0.5, 1, 1, 2.5, 3)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, 
  patchidcol = "patch", individcol = "plantid", blocksize = 4, 
  sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", 
  repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", 
  stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, 
  NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", 
  "XSm", "Sm", "SD", "P1"), 
  "rep"), 
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA), 
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), 
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), 
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), 
  type = c(1, 1, 1, 1, 1, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 1, 1, 3, 3), 
  stageframe = cypframe_raw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, 
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), 
  size = c("size3added", "size2added"), supplement = cypsupp2r, 
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

lambda3(cypmatrix2r)

---

**lambda3.matrix**

**Estimate Deterministic Population Growth Rate of Single Projection Matrix**

**Description**

`lambda3.matrix()` returns the dominant eigenvalue of a single projection matrix. This function can handle large and sparse matrices, so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

**Usage**

```r
## S3 method for class 'matrix'
lambda3(mats, sparse = "auto", ...)
```

**Arguments**

- **mats**: A population projection matrix of class `matrix`.
- **sparse**: A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
- **...**: Other parameters.
Value

This function returns the dominant eigenvalue of the matrix.

Notes

The `sparse` option allows the function to utilize underlying methods of either dense or sparse matrix manipulation in order to speed up processing time and prevent memory shortages. Under the `auto` setting, the function will determine whether the matrix is sparse and act accordingly. For extremely large sparse matrices, the user may simply set `sparse = "yes"` to save time further and force the use of sparse format in calculations.

See Also

`lambda3()`
`lambda3.lefkoMat()`
`slambda3()`

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)
lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
```

lambda3.matrix

multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrln3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")

ehrln3mean <- lmean(ehrlen3mean$A[[1]])
lambdanemo3(ehrlen3mean$A[[1]])

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 1, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 1, 1, 1, 1)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 2.5, 7)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypraw_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
type =c(1, 1, 1, 1, 1, 3, 3),
stageframe = cypraw_v1, historical = FALSE)

cyppmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypraw_v1, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cyppsupp2r,
Description

A dataset containing the states and fates of *Lathyrus vernus* (spring vetch), family Fabaceae, from a population in Sweden monitored annually from 1988 to 1991 in six study plots.

Usage

data(lathyrus)

Format

A data frame with 1119 individuals and 34 variables. Each row corresponds to a unique individual, and each variable from `Volume88` on refers to the state of the individual in a given year.

- **SUBPLOT**  A variable referring to patch within the population.
- **GENET**  A numeric variable giving a unique number to each individual.
- **Volume88**  Aboveground volume in cubic mm in 1988.
- **lnVol88**  Natural logarithm of `Volume88`.
- **FCODE88**  Equals 1 if flowering and 0 if not flowering in 1988.
- **Flow88**  Number of flowers in 1988.
- **Intactseed88**  Number of intact mature seeds produced in 1988. Not always an integer, as in some cases seed number was estimated via linear modeling.
- **Dead1988**  Marked as 1 if known to be dead in 1988.
- **Dormant1988**  Marked as 1 if known to be alive but vegetatively dormant in 1988.
- **Missing1988**  Marked as 1 if not found in 1988.
- **Seedling1988**  Marked as 1, 2, or 3 if observed as a seedling in year $t$. Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1988, 2 = likely that plant is a seedling in 1988, 3 = probable that plant is a seedling in 1988.
- **Volume89**  Aboveground volume in cubic mm in 1989.
- **lnVol89**  Natural logarithm of `Volume89`.
- **FCODE89**  Equals 1 if flowering and 0 if not flowering in 1989.
- **Flow89**  Number of flowers in 1989.
- **Intactseed89**  Number of intact mature seeds produced in 1989. Not always an integer, as in some cases seed number was estimated via linear modeling.
- **Dead1989**  Marked as 1 if known to be dead in 1989.
**Dormant1989**  Marked as 1 if known to be alive but vegetatively dormant in 1989.

**Missing1989**  Marked as 1 if not found in 1989.

**Seedling1989**  Marked as 1, 2, or 3 if observed as a seedling in year \( t \). Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1989, 2 = likely that plant is a seedling in 1989, 3 = probable that plant is a seedling in 1989.

**Volume90**  Aboveground volume in mm\(^3\) in 1990.

**lnVol90**  Natural logarithm of \( \text{Volume90} \).

**FCODE90**  Equals 1 if flowering and 0 if not flowering in 1990.

**Flow90**  Number of flowers in 1990.

**Intactseed90**  Number of intact mature seeds produced in 1990. Not always an integer, as in some cases seed number was estimated via linear modeling.

**Dead1990**  Marked as 1 if known to be dead in 1990.

**Dormant1990**  Marked as 1 if known to be alive but vegetatively dormant in 1990.

**Missing1990**  Marked as 1 if not found in 1990.

**Seedling1990**  Marked as 1, 2, or 3 if observed as a seedling in year \( t \). Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1990, 2 = likely that plant is a seedling in 1990, 3 = probable that plant is a seedling in 1990.

**Volume91**  Aboveground volume in mm\(^3\) in 1991.

**lnVol91**  Natural logarithm of \( \text{Volume91} \).

**FCODE91**  Equals 1 if flowering and 0 if not flowering in 1991.

**Flow91**  Number of flowers in 1991.

**Intactseed91**  Number of intact mature seeds produced in 1991. Not always an integer, as in some cases seed number was estimated via linear modeling.

**Dead1991**  Marked as 1 if known to be dead in 1991.

**Dormant1991**  Marked as 1 if known to be alive but vegetatively dormant in 1991.

**Missing1991**  Marked as 1 if not found in 1991.

**Seedling1991**  Marked as 1, 2, or 3 if observed as a seedling in year \( t \). Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1991, 2 = likely that plant is a seedling in 1991, 3 = probable that plant is a seedling in 1991.

**Source**


**Examples**

```R
# Lathyrus example using blocksize - when repeated patterns exist in variable # order
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
```
lathyrus <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sdl", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "Sdl", "Sd1"),
eststage3 = c(NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA),
multiplier = c(NA, NA, NA, NA, NA),
type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
ehrlen3mean$A[[1]]

lambda3(ehrlen3mean)

# Lathyrus example without blocksize - when no repeated patterns exist in
# variable order and all variables names are specified
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "Sdl", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
This package creates population matrix projection models (MPMs) for use in population ecological analyses. Its specialty is the estimation of historical MPMs, which are 2-dimensional matrices comprising 3 time steps of demographic information. The package constructs both function-based and raw MPMs for both standard ahistorical (i.e. 2 time interval) and historical analyses.
Details

The lefko3 package provides six categories of functions:
1. Data transformation and handling functions
2. Functions determining population characteristics from vertical data
3. Model building and selection
4. Matrix / integral projection model creation functions
5. Population dynamics analysis functions
6. Functions describing, summarizing, or visualizing MPMs and derived structures

lefko3 also includes example datasets complete with sample code.

Author(s)

Richard P. Shefferson <cdorm@g.ecc.u-tokyo.ac.jp>
Johan Ehrlen

References


---

**lmean**

*Estimate Mean Projection Matrices*

**Description**

`lmean()` estimates mean projection matrices as element-wise arithmetic means.

**Usage**

`lmean(mats, matsout = "all")`

**Arguments**

- `mats`: A `lefkoMat` object.
- `matsout`: A string identifying which means to estimate. Option "pop" indicates population-level only, "patch" indicates patch-level only, and "all" indicates that both patch- and population-level means should be estimated. Defaults to "all".
Value

Yields a `lefkoMat` object with the following characteristics:

A
A list of full mean projection matrices in order of sorted populations, patches, and years. These are typically estimated as the sums of the associated mean U and F matrices. All matrices output in the `matrix` class.

U
A list of mean survival-transition matrices sorted as in A. All matrices output in the `matrix` class.

F
A list of mean fecundity matrices sorted as in A. All matrices output in the `matrix` class.

hstages
A data frame showing the pairing of ahistorical stages used to create historical stage pairs. Given if the MPM is historical.

ahstages
A data frame detailing the characteristics of associated ahistorical stages.

labels
A data frame detailing the order of population, patch, and year of each mean matrix. If `pop`, `patch`, or `year2` are NA in the original `labels` set, then these will be re-labeled as A, 1, or 1, respectively.

matrixqc
A short vector describing the number of non-zero elements in U and F mean matrices, and the number of annual matrices.

Examples

```r
# Lathyrus example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)
lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"));
```
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
ehrlen3mean$A[[1]]

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframeraw <- sf_create(sizes = sizevector, stagenames = stagevector,
repsstatutre = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframeraw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5, NA, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cypframeraw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, 
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")
cyp2mean <- lmean(cypmatrix2r)
cyp2mean

---

### Description

Function `modelsearch()` returns both a best-fit model for each vital rate, and a model table showing all models tested. The final output can be used as input in other functions within this package.

### Usage

```r
modelsearch(
data, 
historical = TRUE, 
approach = "mixed", 
suite = "size", 
bestfit = "AICc&k", 
vitalrates = c("surv", "size", "fec"), 
surv = c("alive3", "alive2", "alive1"), 
obs = c("obsstatus3", "obsstatus2", "obsstatus1"), 
size = c("sizea3", "sizea2", "sizeal"), 
replst = c("replstatus3", "replstatus2", "replstatus1"), 
fec = c("feca3", "feca2", "feca1"), 
stage = c("stage3", "stage2", "stage1"), 
indiv = "individ", 
patch = NA,
year = "year2", 
sizedist = "gaussian", 
fecdist = "gaussian", 
size.zero = FALSE, 
size.trunc = FALSE, 
fec.zero = FALSE, 
fec.trunc = FALSE, 
patch.as.random = TRUE, 
year.as.random = TRUE, 
juvestimate = NA, 
juvsize = FALSE, 
jsize.zero = FALSE, 
jsize.trunc = FALSE, 
fectime = 2,
```
censor = NA,
age = NA,
indcova = NA,
indcovb = NA,
indcovc = NA,
show.model.tables = TRUE,
global.only = FALSE,
quiet = FALSE
)

Arguments

data

The vertical dataset to be used for analysis. This dataset should be of class hfvdata, but can also be a data frame formatted similarly to the output format provided by functions verticalize3() or historicalize3(), as long as all needed variables are properly designated.

historical

A logical variable denoting whether to assess the effects of state in time t-1 in addition to state in time t. Defaults to TRUE.

approach

The statistical approach to be taken for model building. The default is mixed, which uses the mixed model approach utilized in packages lme4 and glmmTMB. Other options include glm, which uses lm, glm, glm.nb, and related functions in packages MASS, stats, and pscl.

suite

This describes the global model for each vital rate estimation and has the following possible values: full, includes main effects and all two-way interactions of size and reproductive status; main, includes main effects only of size and reproductive status; size, includes only size (also interactions between size in historical model); rep, includes only reproductive status (also interactions between status in historical model); cons, all vital rates estimated only as y-intercepts. If approach = "glm" and year.as.random = FALSE, then year is also included as a fixed effect, and, in the case of full, included in two-way interactions. Defaults to size.

bestfit

A variable indicating the model selection criterion for the choice of best-fit model. The default is AICc&k, which chooses the best-fit model as the model with the lowest AICc or, if not the same model, then the model that has the lowest degrees of freedom among models with \( \Delta AICc \leq 2.0 \). Alternatively, AICc may be chosen, in which case the best-fit model is simply the model with the lowest AICc value.

vitalrates

A vector describing which vital rates will be estimated via linear modeling, with the following options: surv, survival probability; obs, observation probability; size, overall size; repst, probability of reproducing; and fec, amount of reproduction (overall fecundity). Defaults to c("surv","size","fec").

surv

A vector indicating the variable names coding for status as alive or dead in times t+1, t, and t-1, respectively. Defaults to c("alive3","alive2","alive1").

obs

A vector indicating the variable names coding for observation status in times t+1, t, and t-1, respectively. Defaults to c("obsstatus3","obsstatus2","obsstatus1").

size

A vector indicating the variable names coding for size in times t+1, t, and t-1, respectively. Defaults to c("sizea3","sizea2","sizea1").
repst  A vector indicating the variable names coding for reproductive status in times $t+1$, $t$, and $t-1$, respectively. Defaults to c("repstatus3","repstatus2","repstatus1").

fec  A vector indicating the variable names coding for fecundity in times $t+1$, $t$, and $t-1$, respectively. Defaults to c("feca3","feca2","feca1").

stage  A vector indicating the variable names coding for stage in times $t+1$, $t$, and $t-1$. Defaults to c("stage3","stage2","stage1").

indiv  A variable indicating the variable name coding individual identity. Defaults to individ.

patch  A variable indicating the variable name coding for patch, where patches are defined as permanent subgroups within the study population. Defaults to NA.

year  A variable indicating the variable coding for observation time in time $t$. Defaults to year2.

sizedist  The probability distribution used to model size. Options include gaussian for the Normal distribution (default), poisson for the Poisson distribution, and negbin for the negative binomial distribution.

cedist  The probability distribution used to model fecundity. Options include gaussian for the Normal distribution (default), poisson for the Poisson distribution, and negbin for the negative binomial distribution.

size.zero  A logical variable indicating whether size distribution should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to FALSE.

size.trunc  A logical variable indicating whether size distribution is zero-truncated. Defaults to FALSE. Cannot be TRUE if size.zero = TRUE.

ced.zero  A logical variable indicating whether fecundity distribution should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to FALSE.

ced.trunc  A logical variable indicating whether fecundity distribution is zero-truncated. Defaults to FALSE. Cannot be TRUE if ced.zero = TRUE.

patch.as.random  If set to TRUE and approach = "lme4", then patch is included as a random factor. If set to FALSE and approach = "glm", then patch is included as a fixed factor. All other combinations of logical value and approach lead to patch not being included in modeling. Defaults to TRUE.

year.as.random  If set to TRUE and approach = "lme4", then year is included as a random factor. If set to FALSE, then year is included as a fixed factor. All other combinations of logical value and approach lead to year not being included in modeling. Defaults to TRUE.

juvestimate  An optional variable denoting the stage name of the juvenile stage in the vertical dataset. If not NA, and stage is also given (see below), then vital rates listed in vitalrates other than fec will also be estimated from the juvenile stage to all adult stages. Defaults to NA, in which case juvenile vital rates are not estimated.

juvsize  A logical variable denoting whether size should be used as a term in models involving transition from the juvenile stage. Defaults to FALSE, and is only used if juvestimate does not equal NA.
**values**

This function yields an object of class `lefkoMod`, which is a list in which the first 9 elements are the best-fit models for survival, observation status, size, reproductive status, fecundity, juvenile survival, juvenile observation, juvenile size, and juvenile transition to reproduction, respectively, followed by 9 elements corresponding to the model tables for each of these vital rates, in order, followed by a single character element denoting the criterion used for model selection, and ending on a quality control vector:

- **survival_model** Best-fit model of the binomial probability of survival from time $t$ to time $t+1$. Defaults to 1.
- **observation_model** Best-fit model of the binomial probability of observation in time $t+1$ given survival to that time. Defaults to 1.
- **size_model** Best-fit model of size in time $t+1$ given survival to and observation in that time. Defaults to 1.
- **repstatus_model** Best-fit model of the binomial probability of reproduction in time $t+1$, given survival to and observation in that time. Defaults to 1.
fecundity_model
Best-fit model of fecundity in time \( t+1 \) given survival to, and observation and reproduction in that time. Defaults to 1.

juv_survival_model
Best-fit model of the binomial probability of survival from time \( t \) to time \( t+1 \) of an immature individual. Defaults to 1.

juv_observation_model
Best-fit model of the binomial probability of observation in time \( t+1 \) given survival to that time of an immature individual. Defaults to 1.

juv_size_model
Best-fit model of size in time \( t+1 \) given survival to and observation in that time of an immature individual. Defaults to 1.

juv_reproduction_model
Best-fit model of the binomial probability of reproduction in time \( t+1 \), given survival to and observation in that time of an individual that was immature in time \( t \). This model is technically not a model of reproduction probability for individuals that are immature, rather reproduction probability here is given for individuals that are mature in time \( t+1 \) but immature in time \( t \). Defaults to 1.

survival_table
Full dredge model table of survival probability.

observation_table
Full dredge model table of observation probability.

size_table
Full dredge model table of size.

restatus_table
Full dredge model table of reproduction probability.

fecundity_table
Full dredge model table of fecundity.

juv_survival_table
Full dredge model table of immature survival probability.

juv_observation_table
Full dredge model table of immature observation probability.

juv_size_table
Full dredge model table of immature size.

juv_reproduction_table
Full dredge model table of immature reproduction probability.

criterion
Character variable denoting the criterion used to determine the best-fit model.

cq
Data frame with three variables: 1) Name of vital rate, 2) number of individuals used to model that vital rate, and 3) number of individual transitions used to model that vital rate.

Notes
The mechanics governing model building are fairly robust to errors and exceptions. The function attempts to build global models, and simplifies models automatically should model building fail. Model building proceeds through the functions \( \text{lm()} \) (GLM with Gaussian response), \( \text{glm()} \) (GLM with Poisson or binomial response), \( \text{glm.nb()} \) (GLM with negative binomial response), \( \text{zeroinfl()} \) (zero-inflated Poisson or negative binomial response), \( \text{lmer()} \) (mixed model with Gaussian response), \( \text{glmer()} \) (mixed model with binomial or Poisson response), and \( \text{glmmTMB()} \)
(mixed model with negative binomial, or zero-truncated or zero-inflated Poisson or negative binomial response). See documentation related to these functions for further information. Any response term that is invariable in the dataset will lead to a best-fit model for that response represented by a single constant value.

Exhaustive model building and selection proceeds via the `dredge()` function in package `MuMIn`. This function is verbose, so that any errors and warnings developed during model building, model analysis, and model selection can be found and dealt with. Interpretations of errors during global model analysis may be found in documentation in for the functions and packages mentioned. Package `MuMIn` is used for model dredging (see `dredge()`), and errors and warnings during dredging can be interpreted using the documentation for that package. Errors occurring during dredging lead to the adoption of the global model as the best-fit, and the user should view all logged errors and warnings to determine the best way to proceed. The `quiet = TRUE` option can be used to silence dredge warnings, but users should note that automated model selection can be viewed as a black box, and so care should be taken to ensure that the models run make biological sense, and that model quality is prioritized.

Exhaustive model selection through dredging works best with larger datasets and fewer tested parameters. Setting `suite = "full"` may initiate a dredge that takes a dramatically long time, particularly if the model is historical, individual covariates are used, or a zero-inflated distribution is assumed. In such cases, the number of models built and tested will run at least in the millions. Small datasets will also increase the error associated with these tests, leading to adoption of simpler models overall. We do not yet offer a parallelization option for function `modelsearch()`, but plan to offer one in the future to speed this process up for particularly large global models.

Care must be taken to build models that test the impacts of state in time \( t-1 \) for historical models, and that do not test these impacts for ahistorical models. Ahistorical matrix modeling particularly will yield biased transition estimates if historical terms from models are ignored. This can be dealt with at the start of modeling by setting `historical = FALSE` for the ahistorical case, and `historical = TRUE` for the historical case.

This function handles generalized linear models (GLMs) under zero-inflated distributions using the `zeroinfl()` function, and zero-truncated distributions using the `vglm()` function. Model dredging may fail with these functions, leading to the global model being accepted as the best-fit model. However, model dredges of mixed models work for all distributions. We encourage the use of mixed models in all cases.

The negative binomial and truncated negative binomial distributions use the quadratic structure emphasized in Hardin and Hilbe (2018, 4th Edition of Generalized Linear Models and Extensions). The truncated negative binomial distribution may fail to predict size probabilities correctly when dispersion is near that expected of the Poisson distribution. To prevent this problem, we have integrated a cap on the overdispersion parameter. However, when using this distribution, please check the matrix column sums to make sure that they do not predict survival greater than 1.0. If they do, then please use either the negative binomial distribution or the zero-truncated Poisson distribution.

**Examples**

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8,
```

9) stagevector <- c("Sd", "Sdl", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr", "Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r", "Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propvector)
lathvertln$feca2 <- round(lathvertln$feca2)
lathvertln$feca1 <- round(lathvertln$feca1)
lathvertln$feca3 <- round(lathvertln$feca3)
lathmodelsln3 <- modelsearch(lathvertln, historical = TRUE, approach = "glm", vitalrates = c("surv", "obs", "size", "repst", "fec"), juvestimate = "Sdl", bestfit = "AICc&k", sizedist = "gaussian", fecdist = "poisson", indiv = "individ", patch = "patchid", year = "year2", year.as.random = TRUE, patch.as.random = TRUE, show.model.tables = TRUE, quiet = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl"), stage2 = c("Sd", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl"), stage1 = c("Sd", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl"), eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA), eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA), eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA), givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA), type = c(1, 1, 1, 1, 1, 1, 1, 1, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1, 1, 1), stageframe = lathframeln, historical = TRUE)
lathmat3ln <- flefko3(year = "all", patch = "all", stageframe = lathframeln, modelsuite = lathmodelsln3, data = lathvertln, supplement = lathsupp3, patchcol = "patchid", yearcol = "year2", year.as.random = FALSE, patch.as.random = FALSE, reduce = FALSE)
Create Overwrite Table for MPM Development

**Description**

`overwrite()` returns a data frame describing which particular transitions within an ahistorical or historical projection matrix to overwrite with either given rates and probabilities, or other estimated transitions.

**Usage**

```r
overwrite(
  stage3,
  stage2,
  stage1 = NA,
  eststage3 = NA,
  eststage2 = NA,
  eststage1 = NA,
  givenrate = NA,
  type = NA,
  type_t12 = NA
)
```

**Arguments**

- `stage3` The name of the stage in time \( t+1 \) in the transition to be replaced.
- `stage2` The name of the stage in time \( t \) in the transition to be replaced.
- `stage1` The name of the stage in time \( t-1 \) in the transition to be replaced. Only needed if a historical matrix is to be produced. Use `rep` if all reproductive stages are to be used, and leave empty or use `all` if all stages in stageframe are to be used.
- `eststage3` The name of the stage to replace `stage3`. Only needed if a transition will be replaced by another estimated transition.
- `eststage2` The name of the stage to replace `stage2`. Only needed if a transition will be replaced by another estimated transition.
- `eststage1` The name of the stage to replace `stage1`. Only needed if a transition will be replaced by another estimated transition, and the matrix to be estimated is historical.
- `givenrate` A fixed rate or probability to replace for the transition described by `stage3`, `stage2`, and `stage1`. 
**type**

A vector denoting the kind of transition between times $t$ and $t+1$ to be replaced. This should be entered as 1, S, or s for the replacement of a survival transition; or 2, F, or f for the replacement of a fecundity transition. If empty or not provided, then defaults to 1 for survival transition.

**type_t12**

An optional vector denoting the kind of transition between times $t-1$ and $t$. Only necessary if a historical MPM in deVries format is desired. This should be entered as 1, S, or s for a survival transition; or 2, F, or f for a fecundity transition. Defaults to 1 for survival transition, with impacts only on the construction of deVries-format hMPMs.

**Value**

A data frame that puts the above vectors together and can be used as input in `fleuko3()`, `fleuko2()`, `rleuko3()`, `rleuko2()`, and `afleuko2()`.

Variables in this data frame include the following:

- **stage3**: Stage at time $t+1$ in the transition to be replaced.
- **stage2**: Stage at time $t$ in the transition to be replaced.
- **stage1**: Stage at time $t-1$ in the transition to be replaced.
- **eststage3**: Stage at time $t+1$ in the transition to replace the transition designated by `stage3`, `stage2`, and `stage1`.
- **eststage2**: Stage at time $t$ in the transition to replace the transition designated by `stage3`, `stage2`, and `stage1`.
- **eststage1**: Stage at time $t-1$ in the transition to replace the transition designated by `stage3`, `stage2`, and `stage1`.
- **givenrate**: A constant to be used as the value of the transition.
- **convtype**: Designates whether the transition from time $t$ to time $t+1$ is a survival-transition probability (1) or a fecundity rate (2).
- **convtype_t12**: Designates whether the transition from time $t-1$ to time $t$ is a survival transition probability (1), a fecundity rate (2).

**Examples**

cypover2r <- overwrite(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm"),
                      stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL"),
                      eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm"),
                      eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm"),
                      givenrate = c(0.1, 0.2, 0.2, 0.2, 0.25, 0.4, NA, NA, NA),
)
cypover2r
cypover3r <- overwrite(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3", "SL", "SL",
                                  "SL", "SL", "D", "XSm", "Sm", "D", "XSm", "Sm"),
                      stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL",
cypover3r
projection3

Estimate Stochastic Population Growth Rate

Description

Function `projection3()` projects the population forward in time by a user-defined number of time steps. Projections may be deterministic or stochastic. If deterministic, then projections will be cyclical if matrices exist covering multiple times for each population or patch. If stochastic, then annual matrices will be shuffled within patches and populations.

Usage

```r
projection3(
  mpm,
  times = 10000L,
  stochastic = FALSE,
  standardize = FALSE,
  growthonly = TRUE,
  integeronly = FALSE,
  start_vec = NULL,
  tweights = NULL
)
```

Arguments

- **mpm**: A matrix projection model of class `lefkoMat`, or a list of full matrix projection matrices.
- **times**: Number of iterations to random samples. Defaults to 10,000.
- **stochastic**: A logical value denoting whether to conduct a stochastic projection or a deterministic / cyclical projection.
- **standardize**: A logical value denoting whether to re-standardize the population size to 1.0 at each time step. Defaults to FALSE.
growthonly  A logical value indicating whether to produce only the projected population size at each time step, or a vector showing the stage distribution followed by the reproductive value vector followed by the full population size at each time step. Defaults to TRUE.

integeronly  A logical value indicating whether to round the number of individuals projected in each stage at each time step to the nearest integer. Defaults to FALSE.

start_vec  An optional numeric vector denoting the starting stage distribution for the projection. Defaults to a single individual of each stage.

tweights  An optional numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among times.

Value

A list with two elements:

projection  A list of matrices showing the total number of individuals per stage per time step, or showing the former with the projected stage distribution and reproductive value per stage per time step followed by the total population size per time step (all row-bound in order).

labels  A data frame showing the order of populations and patches in item projection.

Projections are run both at the patch level and at the population level. Population level estimates will be noted at the end of the data frame with 0 entries for patch designation.

Notes

Weightings given in tweights do not need to sum to 1. Final weightings used will be based on the proportion per element of the sum of elements in the user-supplied vector.

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathrepm <- matrix(0, 7, 7)
lathrepm[1, 6] <- 0.345
lathrepm[2, 6] <- 0.054
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep"),
stage1 = c("Sd", "rep", "Sd", "rep", "all", "all"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054),
type = c(1, 1, 1, 1, 3), type_t12 = c(1, 2, 1, 2, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe,
year = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
repmatrix = lathrepm, supplement = lathsupp3, yearcol = "year2",
indivcol = "individ")
lathproj <- projection3(ehrlen3, stochastic = TRUE)

# Cypripedium example
rm(list = ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
obsvvector <- c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1)
matvvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0)
proppvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvvector, obsstatus = obsvvector, matstatus = matvvector,
propstatus = propvvector, immstatus = immvvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
rep_cyp_raw <- matrix(0, 11, 11)
rep_cyp_raw[1:2,7:11] <- 0.5

cypsupp3r <- supplemental(stage3 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.4, 0.4, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3),
type_t12 = c(1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = cypframe_raw, historical = TRUE)

cypmatrix3r <- rlefko3(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added", "size1added"),
repmatrix = rep_cyp_raw, supplement = cypsupp3r, yearcol = "year2",
patchcol = "patchid", individcol = "individ")

cypstoch <- projection3(cypmatrix3r, stochastic = TRUE)

---

**Description**

`repvalue3()` is a generic function that estimates returns the reproductive values of stages in a population projection matrix or a set of matrices. The specifics of estimation vary with the class of input object. This function is made to handle very large and sparse matrices supplied as `lefkoMat` objects or as individual matrices, and can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

**Usage**

`repvalue3(mats, ...)`

**Arguments**

- **mats**: A `lefkoMat` object, or population projection matrix.
- **...**: Other parameters.
Value

The value returned depends on the class of the mats argument. See related functions for details.

See Also

repvalue3.lefkoMat()
repvalue3.matrix()

Examples

# Lathyrus deterministic example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repsstatus = reppvector, obsstatus = obsvector, matstatus = matvector,
immsstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propropsstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
feacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
repvalue3(ehrlen3mean)
# Cypripedium stochastic example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1)
binec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
                          repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                          propstatus = propvector, immstatus = immvector, indataset = indataset,
                          binhalfwidth = binec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
                          patchidcol = "patch", individcol = "plantid", blocksize = 4,
                          sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
                          repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
                          stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
                          NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SM", "D", 
"XSm", "Sm", "SD", "P1"),
stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "Lg", "rep", 
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", individcol = "individ")

repvalue3(cyfmatrix2r, stochastic = TRUE)
Description

repvalue3.lefkoMat() returns the reproductive values for stages in a set of population projection matrices provided as a lefkoMat object. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

Usage

## S3 method for class 'lefkoMat'
repvalue3(
mats,  
stochastic = FALSE,  
times = 10000,  
tweights = NA,  
seed = NA,  
sparse = "auto",  
...)

Arguments

mats An object of class lefkoMat object.
stochastic A logical value indicating whether to use deterministic (FALSE) or stochastic (TRUE) analysis. Defaults to FALSE.
times An integer variable indicating number of times to project if using stochastic analysis. Defaults to 10000.
tweights An optional vector indicating the probability weighting to use for each matrix in stochastic simulations. If not given, then defaults to equal weighting.
seed A number to use as a random number seed.
sparse A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
...

Other parameters.

Value

This function returns the asymptotic reproductive value vectors if deterministic analysis chosen, and long-run mean reproductive value vectors if stochastic analysis was chosen.

The output depends on whether the lefkoMat object used as input is ahistorical or historical, and whether the analysis is deterministic or stochastic. If ahistorical, then a single data frame is output, which includes the number of the matrix within the $A$ element of the input lefkoMat object, followed by the stage id (numeric and assigned through sf_create()), the stage name, and the estimated reproductive value (rep_value). Reproductive values are scaled by the first non-zero value.

If a historical matrix is used as input, then two data frames are output into a list object. The $hist$ element contains a data frame in which the stable stage distribution is given in terms of across-year stage pairs. The structure includes the matrix number, the numeric stage designations for stages in
times $t$ and $t-1$, respectively, followed by the respective stage names, and ending with the estimated reproductive value for that stage within its matrix (`rep_value`). The `$ahist` element is a data frame showing the reproductive values of the basic stages in the associated stageframe. The reproductive values in this second data frame are estimated via the approach developed in Ehrlen (2000), in which each ahistorical stage’s reproductive value is the average of the RVs summed by stage at time $t$ weighted by the proportion of that stage pair within the historical stable stage distribution associated with the matrix. Both historical and ahistorical reproductive values are scaled to the first non-zero reproductive value in each case.

In addition to the data frames noted above, stochastic analysis will result in the additional output of a list of matrices containing the actual projected reproductive value vectors across all projected times, in the order of population-patch combinations in the `lefkoMat` input.

Notes

In stochastic analysis, the projected mean reproductive value vector is the arithmetic mean across the final projected 1000 times if the simulation is at least 2000 projected times long. If between 500 and 2000 projected times long, then only the final 200 are used, and if fewer than 500 times are used, then all are used. Note that because reproductive values in stochastic simulations can change greatly in the initial portion of the run, we encourage a minimum 2000 projected times per simulation, with 10000 preferred.

See Also

`repvalue3()`

`repvalue3.matrix()`

Examples

```r
# Lathyrus deterministic example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
revector <- c(0, 0, 0, 0, 1, 0, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 1, 1, 1, 1, 1, 1)
immvector <- c(1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 1, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 0, 0)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = revector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)
lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988")
```
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 3, 3, 1, 1),
type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
repvalue3(ehrlen3mean)

# Cypripedium stochastic example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repevector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repevector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
"rep"),
Estimate Reproductive Value Vector for a Single Population Projection Matrix

Description

repvalue3.matrix() returns the reproductive values for stages in a population projection matrix. The function makes no assumptions about whether the matrix is ahistorical and simply provides standard reproductive values corresponding to each row, meaning that the overall reproductive values of basic life history stages in a historical matrix are not provided (the repvalue3.lefkoMat() function estimates these on the basis of stage description information provided in the lefkoMat object used as input in that function). This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

Usage

## S3 method for class 'matrix'
repvalue3(mats, sparse = "auto", ...)

Arguments

mats  A population projection matrix.
sparse  A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".

Value

This function returns a vector data frame characterizing the reproductive values for stages of a population projection matrix. This is given as the left eigenvector associated with largest real part of the dominant eigenvalue, divided by the first non-zero element of the left eigenvector.
repvalue3.matrix

See Also

repvalue3()
repvalue3.lefkoMat()

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = reppvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sdl", "Sdl", "Sdl", "Sdl", "mat"),
stage1 = c("Sd", "rep", "Sdl", "rep", "Sdl"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrle3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

ehrle3mean <- lmean(ehrlen3)
repvalue3(ehrle3mean$A[[1]])

# Cyripedium example
rm(list=ls(all=TRUE))
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector, repsstatus = repsvector, obsstatus = obsvector, matstatus = matvector, propstatus = propvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cympsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "rep", "rep"), eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), givenrate = c(0.10, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1), stageframe = cypframe_raw, historical = FALSE)

cymatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), size = c("sizeadded", "size2added"), supplement = cympsupp2r, yearcol = "year2", patchcol = "patchid", individcol = "individ")

repvalue3(cymatrix2r$A[[1]])
the population, patch, and year associated with each matrix.

Usage

rlefko2(
  data,
  stageframe,
  year = "all",
  pop = NA,
  patch = NA,
  censor = FALSE,
  stages = NA,
  alive = c("alive3", "alive2"),
  size = c("sizea3", "sizea2"),
  repst = c("repsstatus3", "repsstatus2"),
  matst = c("matstatus3", "matstatus2"),
  fec = c("feca3", "feca2"),
  supplement = NA,
  repmatrix = NA,
  overwrite = NA,
  yearcol = NA,
  popcol = NA,
  patchcol = NA,
  indivcol = NA,
  censorcol = NA,
  censorkeep = 0,
  reduce = FALSE
)

Arguments

data A vertical demographic data frame, with variables corresponding to the naming conventions in 

stageframe A stageframe object that includes information on the size, observation status, propagule status, immaturity status, and maturity status of each ahistorical stage. Should also incorporate bin widths if size is continuous.

year A variable corresponding to year or observation time, or a set of such values, given in values associated with the year term used in linear model development. Can also equal all, in which case matrices will be estimated for all years. Defaults to all.

pop A variable designating which populations will have matrices estimated. Should be set to specific population names, or to all if all populations should have matrices estimated.

patch A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to all if matrices should be estimated for all patches. Defaults to all.
censor If TRUE, then data will be removed according to the variable set in censorcol, such that only data with censor values equal to 1 will remain. Defaults to FALSE.

stages An optional vector denoting the names of the variables within the main vertical dataset coding for the stages of each individual in times $t+1$, $t$, and $t-1$. The names of stages in these variables should match those used in the stageframe exactly. If left blank, then rlefko3() will attempt to infer stages by matching values of alive, size, repst, and matst to characteristics noted in the associated stageframe.

alive A vector of names of binomial variables corresponding to status as alive (1) or dead (0) in times $t+1$, $t$, and $t-1$, respectively.

size A vector of names of variables coding size in times $t+1$ and $t$, respectively. Defaults to c("sizea3","sizea2").

repst A vector of names of variables coding reproductive status in times $t+1$ and $t$, respectively. Defaults to c("repstatus3","repstatus2").

matst A vector of names of variables coding maturity status in times $t+1$ and $t$, respectively. Defaults to c("matstatus3","matstatus2"). Must be supplied if stages is not provided.

cac A vector of names of variables coding fecundity in times $t+1$ and $t$, respectively. Defaults to c("feca3","feca2").

supplement An optional data frame of class lefkoSD that provides supplemental data that should be incorporated into the MPM. Three kinds of data may be integrated this way: transitions to be estimated via the use of proxy transitions, transition overwrites from the literature or supplemental studies, and transition multipliers for fecundity. This data frame should be produced using the supplemental() function. Can be used in place of or in addition to an overwrite table (see overwrite below) and a reproduction matrix (see repmatrix below).

repmatrix A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in supplement. If left blank, then rlefko2() will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in supplement above. Must be the dimensions of an ahistorical matrix.

overwrite An optional data frame developed with the overwrite() function describing transitions to be overwritten either with given values or with other estimated transitions. Note that this function supplements overwrite data provided in supplement.

yearcol The variable name or column number corresponding to time in time $t$ in the dataset.

popcol The variable name or column number corresponding to the identity of the population.
patchcol  The variable name or column number corresponding to patch in the dataset.
indivcol  The variable name or column number coding individual identity.
censorcol  The variable name or column number denoting the censor status. Only needed if censor = TRUE.
censorkeep  The value of the censor variable denoting data elements to keep. Defaults to 0.
reduce  A logical value denoting whether to remove historical stages associated with only zero transitions. These are removed only if all row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.

Value

If all inputs are properly formatted, then this function will return an object of class lefkoMat. This includes:

A  A list of full projection matrices in order of sorted populations, patches, and years. All matrices output in the matrix class.
U  A list of survival transition matrices sorted as in A. All matrices output in the matrix class.
F  A list of fecundity matrices sorted as in A. All matrices output in the matrix class.
hstages  Set to NA for ahistorical matrices.
agstages  Set to NA. Only used in output to function afleko2().
ahstages  A data frame detailing the characteristics of associated ahistorical stages, in the form of a modified stageframe that includes status as an entry stage through reproduction.
labels  A data frame giving the population, patch, and year of each matrix in order.
matrixqc  A short vector describing the number of non-zero elements in U and F matrices, and the number of annual matrices.
dataqc  A vector showing the numbers of individuals and rows in the vertical dataset used as input.

Notes

The default behavior of this function is to estimate fecundity with regards to transitions specified via associated fecundity multipliers in either supplement or repmatrix. If both of these fields are left empty, then fecundity will be estimated at full for all transitions leading from reproductive stages to immature and propagule stages. However, if a supplement is provided and a repmatrix is not, or if repmatrix is set to 0, then only fecundity transitions noted in the supplement will be set to non-zero values. To use the default behavior of setting all reproductive stages to reproduce at full fecundity into immature and propagule stages but also incorporate given or proxy survival transitions, input those given and proxy transitions through the overwrite option.

The reproduction matrix (field repmatrix) may only be supplied as ahistorical. If provided as historical, then rlefko2() will fail and produce an error.

Users may at times wish to estimate MPMs using a dataset incorporating multiple patches or sub-populations. Should the aim of analysis be a general MPM that does not distinguish these patches or subpopulations, the patchcol variable should be left to NA, which is the default.
Input options including multiple variable names must be entered in the order of variables in time \( t+1 \) and \( t \). Rearranging the order WILL lead to erroneous calculations, and will probably also lead to fatal errors.

**Examples**

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
                        repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                        immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
                        propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
                        patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
                        juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
                        fecacol = "Intactseed88", deadacol = "Dead1988", nonobsacol = "Dormant1988",
                        stageassign = lathframe, stagesize = "sizea", censorcol = "Missing1988",
                        censorkeep = NA, censor = TRUE)

lathsupp2 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl"),
                          stage2 = c("Sd", "Sd", "rep", "rep"),
                          givenrate = c(0, 0.345, 0.054, NA, NA),
                          multiplier = c(NA, NA, 0.345, 0.054),
                          type = c(1, 1, 3, 3), stageframe = lathframe, historical = FALSE)

ehrld2 <- rlefko2(data = lathvert, stageframe = lathframe, year = "all",
                  stages = c("stage3", "stage2"), supplement = lathsupp2, yearcol = "year2",
                  indivcol = "individ")

summary(ehrlen2)
```

```r
# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
                "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1)

```

immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cypmatrix2r$A[[1]]

rlefko3

Create Raw Historical Matrix Projection Model

Description

rlefko3() returns raw historical MPMs, including the associated component transition and fecun-
dity matrices, data frames describing the ahistorical stages used and the historical paired stages, and a
data frame describing the population, patch, and year associated with each matrix.

Usage

rlefko3(
data,
stageframe,
year = "all",
pop = NA,
patch = NA,
censor = FALSE,
stages = NA,
alive = c("alive3", "alive2", "alive1"),
size = c("sizea3", "sizea2", "sizea1"),
reptst = c("repstatus3", "repstatus2", "repstatus1"),
matst = c("matstatus3", "matstatus2", "matstatus1"),
fec = c("feca3", "feca2", "feca1"),
supplement = NA,
repmatrix = NA,
overwrite = NA,
yearcol = NA,
popcol = NA,
patchcol = NA,
indivcol = NA,
censorcol = NA,
censorkeep = 0,
format = "ehrilen",
reduce = FALSE,
err_check = FALSE
)

Arguments

data A vertical demographic data frame, with variables corresponding to the naming conventions in `verticalize3()`.

stageframe A stageframe object that includes information on the size, observation status, propagule status, immaturity status, and maturity status of each ahistorical stage. Should also incorporate bin widths if size is continuous.

year A variable corresponding to year or observation time, or a set of such values, given in values associated with the year term used in linear model development. Can also equal all, in which case matrices will be estimated for all years. Defaults to all.

pop A variable designating which populations will have matrices estimated. Should be set to specific population names, or to all if all populations should have matrices estimated.

patch A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to all if matrices should be estimated for all patches. Defaults to all.

censor If TRUE, then data will be removed according to the variable set in censorcol, such that only data with censor values equal to 1 will remain. Defaults to FALSE.

stages An optional vector denoting the names of the variables within the main vertical dataset coding for the stages of each individual in times $t+1$, $t$, and $t-1$. The
names of stages in these variables should match those used in the stageframe exactly. If left blank, then rlefko3() will attempt to infer stages by matching values of alive, size, repst, and matst to characteristics noted in the associated stageframe.

alive A vector of names of binomial variables corresponding to status as alive (1) or dead (0) in times t+1, t, and t-1, respectively.

size A vector of names of variables coding size in times t+1, t, and t-1, respectively. Defaults to c("sizea3","sizea2","sizea1").

repst A vector of names of variables coding reproductive status in times t+1, t, and t-1, respectively. Defaults to c("repstatus3","repstatus2","repstatus1").

matst A vector of names of variables coding maturity status in times t+1, t, and t-1, respectively. Defaults to c("matstatus3","matstatus2","matstatus1"). Must be supplied if stages is not provided.

fec A vector of names of variables coding fecundity in times t+1, t, and t-1, respectively. Defaults to c("feca3","feca2","feca1").

supplement An optional data frame of class lefkoSD that provides supplemental data that should be incorporated into the MPM. Three kinds of data may be integrated this way: transitions to be estimated via the use of proxy transitions, transition overwrites from the literature or supplemental studies, and transition multipliers for fecundity. This data frame should be produced using the supplemental() function. Can be used in place of or in addition to an overwrite table (see overwrite below) and a reproduction matrix (see repmatrix below).

repmatrix A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in supplement. If left blank, then rlefko3() will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in supplement above. May be the dimensions of either a historical or an ahistorical matrix. If the latter, then all stages will be used in time t-1 for each suggested ahistorical transition.

overwrite An optional data frame developed with the overwrite() function describing transitions to be overwritten either with given values or with other estimated transitions. Note that this function supplements overwrite data provided in supplement.

yearcol The variable name or column number corresponding to time in time t in the dataset.

popcol The variable name or column number corresponding to the identity of the population.

patchcol The variable name or column number corresponding to patch in the dataset.

indivcol The variable name or column number coding individual identity.
The variable name or column number denoting the censor status. Only needed if censor = TRUE.

censorkeep
The value of the censor variable denoting data elements to keep. Defaults to 0.

format
A string indicating whether to estimate matrices in ehrlen format or deVries format. The latter adds one extra prior stage to account for the prior state of newborns. Defaults to ehrlen format.

reduce
A logical value denoting whether to remove historical stages associated exclusively with zero transitions. These are removed only if all row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.

err_check
A logical value indicating whether to append extra information used in matrix calculation within the output list. Used for development debugging purposes.

Value
If all inputs are properly formatted, then this function will return either an object of class lefkoMat. Output includes:

A
A list of full projection matrices in order of sorted populations, patches, and years. All matrices output in the matrix class.

U
A list of survival transition matrices sorted as in A. All matrices output in the matrix class.

F
A list of fecundity matrices sorted as in A. All matrices output in the matrix class.

hstages
A data frame matrix showing the pairing of ahistorical stages used to create historical stage pairs.

agestages
Set to NA. Only used in output to function aflefko2().

ahstages
A data frame detailing the characteristics of associated ahistorical stages, in the form of a modified stageframe that includes status as an entry stage through reproduction.

labels
A data frame giving the population, patch, and year of each matrix in order.

matrixqc
A short vector describing the number of non-zero elements in U and F matrices, and the number of annual matrices.

dataqc
A vector showing the numbers of individuals and rows in the vertical dataset used as input.

Notes
The default behavior of this function is to estimate fecundity with regards to transitions specified via associated fecundity multipliers in either supplement or repmatrix. If both of these fields are left empty, then fecundity will be estimated at full for all transitions leading from reproductive stages to immature and propagule stages. However, if a supplement is provided and a repmatrix is not, or if repmatrix is set to 0, then only fecundity transitions noted in the supplement will be set to non-zero values. To use the default behavior of setting all reproductive stages to reproduce at full fecundity into immature and propagule stages but incorporate given or proxy survival transitions, input those given and proxy transitions through the overwrite option.
The reproduction matrix (field \texttt{repmatrix}) may be supplied as either historical or ahistorical. If provided as ahistorical, then \texttt{rlefko3()} will assume that all historical transitions involving stages noted for times \(t\) and \(t+1\) should be set to the respective fecundity multipliers noted.

Users may at times wish to estimate MPMs using a dataset incorporating multiple patches or sub-populations. Should the aim of analysis be a general MPM that does not distinguish these patches or subpopulations, the \texttt{patchcol} variable should be left to \texttt{NA}, which is the default.

Input options including multiple variable names must be entered in the order of variables in time \(t+1, t, \) and \( t \). Rearranging the order WILL lead to erroneous calculations, and will probably also lead to fatal errors.

\textbf{Examples}

\begin{verbatim}
# Lathyrus example
data(lathyrus)

tvec <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevec <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")

sizevector <- c(tvec[1], 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
                       repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                       immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
                       propstatus = propvector)
lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
                         patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
                         juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
                         fecacol = "Intactseed88", deadacol = "Dead1988", nonobsacol = "Dormant1988",
                         stageassign = lathframe, stagesize = "sizea", censorcol = "Missing1988",
                         censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
                           stage2 = c("Sd", "Sd", "Sdl", "Sdl", "rep", "rep", "Sdl"),
                           stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sdl"),
                           eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
                           eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
                           eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
                           givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
                           multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
                           type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
                           stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
                   stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
                   yearcol = "year2", individcol = "individ")
\end{verbatim}
summary(ehrlen3)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0)
proppvector <- c(1, 0, 0, 0, 0, 1, 1, 1, 1, 1)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

Cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, propstatus = proppvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = Cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)

cyper_supp3r <- supplemental(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"), stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep", "rep"), stage1 = c("SD", "rep", "rep", "SD", "P1", "P2", "P3", "SL", "P3", "P3", "SL", "SL", "all", "all"), eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm", NA, NA), eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm", NA, NA), eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm", NA, NA), givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.25, 0.4, 0.4, NA, NA, NA, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5), type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3), type_t12 = c(1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1), stageframe = Cypframe_raw, historical = TRUE)

cyprmatrix3r <- rlefko3(data = cypraw_v1, stageframe = Cypframe_raw, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), size = c("size3added", "size2added", "size1added"),
sensitivity3

supplement = cypsupp3r, yearcol = "year2", patchcol = "patchid",
indivcol = "individ"

summary(cypmatrix3r)

describe

---

**Description**

`sensitivity3()` is a generic function that returns the sensitivity of the population growth rate to the elements of the matrices in a matrix population model. Currently, this function estimates both deterministic and stochastic sensitivities, where the growth rate is $\lambda$ in the former case and the log of the stochastic $\lambda$ in the latter case. This function is made to handle very large and sparse matrices supplied as `lefkoMat` objects, as lists of matrices, and as individual matrices.

**Usage**

`sensitivity3(mats, ...)`

**Arguments**

- `mats` A `lefkoMat` object, or population projection matrix, for which the stable stage distribution is desired.
- `...` Other parameters

**Value**

The value returned depends on the class of the `mats` argument.

**See Also**

`sensitivity3.lefkoMat()`
`sensitivity3.matrix()`
`sensitivity3.list()`

**Examples**

```r
# Lathyrus example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
```
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
                        repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                        immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
                        propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
                         patchicol = "SUBPLOT", individcol = "GENET", blocksize = 9,
                         juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
                         fecacol = "Intactseed88", deadacol = "Dead1988",
                         nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
                         censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sd", "Sdl", "mat"),
                           stage2 = c("Sd", "Sd", "Sdl", "rep", "rep", "Sdl"),
                           stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
                           eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
                           eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
                           eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
                           givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
                           multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
                           type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
                           stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
                    stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
                    yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
sensitivity3(ehrlen3mean)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
                          repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
                          propstatus = propvector, immstatus = immvector, indataset = indataset,
                          binhalfwidth = binvec)
sensitivity3.lefkoMat

Estimate Sensitivity of Population Growth Rate of a lefkoMat Object

Description

sensitivity3.lefkoMat() returns the sensitivities of population growth rate to elements of all $A$ matrices in an object of class lefkoMat. If deterministic, then $\lambda$ is taken as the population growth rate. If stochastic, then the log of stochastic $\lambda$, or the log stochastic growth rate, is taken as the population growth rate. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

Usage

## S3 method for class 'lefkoMat'
sensitivity3(
  mats,
  stochastic = FALSE,
  steps = 10000,
  time_weights = NA,
  sparse = "auto",
  ...
)
Arguments

mats An object of class lefkoMat.
stochastic A logical value determining whether to conduct a deterministic (FALSE) or stochastic (TRUE) sensitivity analysis. Defaults to FALSE.
steps The number of times to project forward in stochastic simulation. Defaults to 10,000.
time_weights Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among times.
sparse A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
... Other parameters.

Value

This function returns an object of class lefkoSens, which is a list of 8 elements. The first, h_sensmats, is a list of historical sensitivity matrices (NULL if an ahMPM is used as input). The second, ah_elasmats, is a list of either ahistorical sensitivity matrices if an ahMPM is used as input, or, if an hMPM is used as input, then the result is a list of ahistorical matrices based on the equivalent historical dependencies assumed in the input historical matrices. The third element, h_stages, is a data frame showing historical stage pairs (NULL if ahMPM used as input). The fourth element, ages, show the order of age-stage combinations, if age-by-stage MPMs have been supplied. The fifth element, ah_stages, is a data frame showing the order of ahistorical stages. The last 3 elements are the A, U, and F portions of the input.

Notes

Deterministic sensitivities are estimated as eqn. 9.14 in Caswell (2001, Matrix Population Models). Stochastic sensitivities are estimated as eqn. 14.97 in Caswell (2001). Note that stochastic sensitivities are of the log of the stochastic $\lambda$.

See Also

sensitivity3()
sensitivity3.matrix()
sensitivity3.list()

Examples

# Lathyrus example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagvector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sdl", "Sdl", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA,
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1),
type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")
sensitivity3(ehrlen3, stochastic = TRUE)

# Cyripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
"XLg")
repsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cpdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizea0col = "Inf2.04", sizeb0col = "Inf.04", sizec0col = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)
cyps supp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"), stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL", "rep", "rep"), eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA), eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA), givenrate = c(0.10, 0.20, 0.20, 0.25, 0.40, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA), type = c(1, 1, 1, 1, 1, 1, 1, 1), stageframe = cypframe_raw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), size = c("size3added", "size2added"), supplement = cyps supp2r, yearcol = "year2", patchcol = "patchid", individcol = "individ")
sensitivity3(cypmatrix2r)

**sensitivity3.list** Estimate Sensitivity of Population Growth Rate of a List of Matrices

**Description**

`sensitivity3.list()` returns the sensitivities of population growth rate to elements of matrices supplied in a list. The sensitivity analysis can be deterministic or stochastic, but if the latter then at least two A matrices must be included in the list. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

**Usage**

```r
## S3 method for class 'list'
sensitivity3(
  mats,
  stochastic = FALSE,
  steps = 10000,
  time_weights = NA,
  historical = FALSE,
  sparse = "auto",
  ...)
```

Arguments

mats  An object of class matrix.

stochastic  A logical value determining whether to conduct a deterministic (FALSE) or stochastic (TRUE) sensitivity analysis. Defaults to FALSE.

steps  The number of times to project forward in stochastic simulation. Defaults to 10,000.

time_weights  Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among times.

historical  A logical value indicating whether matrices are historical. Defaults to FALSE.

sparse  A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".

...  Other parameters.

Value

This function returns an object of class lefkoSens, which is a list of 8 elements. The first, h_sensmats, is a list of historical sensitivity matrices (NULL if an ahMPM is used as input). The second, ah_elasmats, is a list of ahistorical sensitivity matrices if an ahMPM is used as input (NULL if an hMPM is used as input). The third element, h_stages, the fourth element, agestages, and the fifth element, ah_stages, are NULL. The last 3 elements include the original A matrices supplied (as the A element), followed by NULLs for the U and F elements.

Notes

Deterministic sensitivities are estimated as eqn. 9.14 in Caswell (2001, Matrix Population Models). Stochastic sensitivities are estimated as eqn. 14.97 in Caswell (2001). Note that stochastic sensitivities are with regard to the log of the stochastic λ.

Currently, this function does not estimate equivalent ahistorical stochastic sensitivities for input historical matrices, due to the lack of guidance input on the order of stages (such guidance is provided within lefkoMat objects).

See Also

sensitivity3()
sensitivity3.lefkoMat()
sensitivity3.matrix()

Examples

# Lathyrus example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
imstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 1),
type_t12 = c(1, 1, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")

sensitivity3(ehrlen3$A)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XS", "M", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 1, 1, 1, 1)

ecf <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
imstatus = immvector, indataset = indataset, binhalfwidth = binvec)
sensitivity3.matrix  

**Estimate Sensitivity of Population Growth Rate of a Single Matrix**

**Description**

`sensitivity3.matrix()` returns the sensitivities of $\lambda$ to elements of a single matrix. Because this handles only one matrix, the sensitivities are inherently deterministic and based on the dominant eigen value as the best metric of the population growth rate. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

**Usage**

```r
## S3 method for class 'matrix'
sensitivity3(mats, sparse = "auto", ...)
```

**Arguments**

- **mats**: An object of class `matrix`.
- **sparse**: A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
- **...**: Other parameters.
This function returns a single deterministic sensitivity matrix.

See Also

sensitivity3()
sensitivity3.lefkoMat()
sensitivity3.list()

Examples

# Lathyrus example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 0)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immsstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)
lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA,NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")
ehrlen3mean <- lmean(ehrlen3)
sensitivity3(ehrlen3mean$A[[1]])

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
immvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
indataset <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
propstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypdata <- cypmatrix2r$A[[1]]
sensitivity3(cypdata)
cypframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
propstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cypdata <- cypmatrix2r$A[[1]]
sensitivity3(cypdata)
sf_create() returns a data frame describing each ahistorical life history stage in the life history model. This data frame can be used as input into MPM creation functions such as flefko3(), where it determines how each stage is treated during matrix estimation.

Usage

```
sf_create(
  sizes,
  stagenames = NA,
  repstatus = 1,
  obsstatus = 1,
  propstatus = NA,
  immstatus = NA,
  matstatus = 1,
  minage = NA,
  maxage = NA,
  indataset = NA,
  binhalfwidth = 0.5,
  comments = NA,
  ipmbins = 100,
  roundsize = 5
)
```

Arguments

- **sizes**: A numeric vector of the typical or representative size of each life history stage.
- **stagenames**: An optional vector of stage names, in the same order as elements in sizes. If an IPM or function-based matrix with many stages is desired, then two stages that occur within the dataset and represent the lower and upper size limits of the IPM must be marked as ipm in this vector. These stages must be mature stages, and should have all characteristics other than size equal. If two or more groups of stages, each with its own characteristics, are to be developed for an IPM, then an even number of stages with two stages marking the minimum and maximum size of each group should be marked, with all other characteristics equal within each group.
- **repstatus**: A vector denoting the binomial reproductive status of each life history stage. Defaults to 1.  
- **obsstatus**: A vector denoting the binomial observation status of each life history stage. Defaults to 1, but may be changed for unobservable stages. 
- **propstatus**: A vector denoting whether each life history stage is a propagule. Such stages are generally only used in fecundity estimation. Defaults to NA. 
- **immstatus**: A vector denoting whether each stage is immature. Must be composed of binomial values if given. Defaults to NA. 
- **matstatus**: A vector denoting whether each stage is mature. Must be composed of binomial values if given. Defaults to 1 for all stages defined in sizes.
minage | An optional vector denoting the minimum age at which a stage can occur. Only used in age x stage matrix development. Defaults to NA.
maxage | An optional vector denoting the maximum age at which a stage should occur. Only used in age x stage matrix development. Defaults to NA.
indataset | A vector designating which stages are found within the dataset. While `rlefko2()` and `rlefko3()` can use all stages in the input dataset, `flefko3()` and `flefko2()` can only handle size-classified stages with non-overlapping combinations of size and reproductive status, plus one immature stage. Stages that do not actually exist within the dataset should be marked as 0 in this vector.
binhalfwidth | A numeric vector giving the half-width of size bins. Required to classify individuals appropriately within size classes. Defaults to 0.5 for all sizes.
comments | An optional vector of text entries holding useful text descriptions of all stages.
ipmbins | If an IPM is desired, then this parameter sets the number of stages to create for that IPM. This number is in addition to any stages that are not size-classified. Defaults to 100, and numbers greater than this yield a warning about the loss of statistical power and increasing chance of matrix over-parameterization resulting from increasing numbers of stages.
roundsize | This parameter sets the precision of size classification, and equals the number of digits used in rounding sizes. Defaults to 5.

### Value

A data frame of class `stageframe`, which includes information on the stage name, size, reproductive status, observation status, propagule status, immaturity status, maturity status, presence within the core dataset, counts of similarly sized stages, raw bin half-width, and the minimum, center, and maximum of each size bin, as well as its width. If minimum and maximum ages were specified, then these are also included. Also includes an empty string variable that can be used to describe stages meaningfully. This object can be used as the `stageframe` input for `flefko3()` `flefko2()`, `rlefko3()`, and `rlefko2()`.

Variables in this data frame include the following:

- stage | The unique names of the stages to be analyzed.
- size | The typical or representative size at which each stage occurs.
- repstatus | A binomial variable showing whether each stage is reproductive.
- obsstatus | A binomial variable showing whether each stage is observable.
- propstatus | A binomial variable showing whether each stage is a propagule.
- immstatus | A binomial variable showing whether each stage can occur as immature.
- matstatus | A binomial variable showing whether each stage occurs in maturity.
- indataset | A binomial variable describing whether each stage occurs in the input dataset.
- binhalfwidth_raw | The half-width of the size bin, as input.
- min_age | The minimum age at which the stage may occur.
- max_age | The maximum age at which the stage may occur.
- sizebin_min | The minimum size at which the stage may occur.
sizebin_max  The maximum size at which the stage may occur.
sizebin_center The centroid of the size bin at which the stage may occur.
sizebin_width The width of the size bin corresponding to the stage.
comments  A text field for stage descriptions.

Examples

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "Fla", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 113, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = reppvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
ehrlen3mean$A[1]

# Cypripedium example
rm(list=ls(all=TRUE))
```r
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0)
propvvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
                          repstatus = reppvector, obstatus = obsvvector, matstatus = matvvector,
                          propstatus = propvvector, immstatus = immvvector, indataset = indataset,
                          binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
                          patchidcol = "patch", individcol = "plantid", blocksize = 4,
                          sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
                          repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
                          stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
                          NRasRep = TRUE)

cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "D",
                              "XSm", "Sm", "SD", "P1"),
                          stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep",
                                      "rep"),
                          eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
                          eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
                          givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
                          multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
                          type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
                          stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
                          year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
                          size = c("size3added", "size2added"), supplement = cypsupp2r,
                          yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cyp2mean <- lmean(cypmatrix2r)
cyp2mean
```

**sf_distrib**

Test for overdispersion and zero inflation in size and fecundity
Description

Function sf_distrib takes a historically formatted vertical data as input and tests whether size and fecundity data are dispersed according to a Poisson distribution (where mean = variance), and whether the number of 0s exceeds expectations.

Usage

```r
sf_distrib(
  data,
  size3 = NA,
  size2 = NA,
  obs3 = NA,
  fec = NA,
  repst = NA,
  zisize = TRUE,
  zifec = TRUE
)
```

Arguments

data
- A historical vertical data file, which is a data frame of class hfvdata.

size3
- The name or column number of the variable corresponding to size in time \(t+1\).

size2
- The name or column number of the variable corresponding to size in time \(t\). This term is required for both size and fecundity tests.

obs3
- The name or column number of the variable corresponding to observation status in time \(t+1\). This should be used if observation status will be used as a vital rate to absorb states of size = 0.

fec
- The name or column number of the variable corresponding to fecundity. The name of the variable should correspond to the proper time, either time \(t\) or time \(t-1\).

repst
- The name or column number of the variable corresponding to reproductive status in time \(t\). Required if fecundity distribution will be tested.

zisize
- A logical value indicating whether to conduct a test of zero inflation in size. Defaults to TRUE.

zifec
- A logical value indicating whether to conduct a test of zero inflation in fecundity. Defaults to TRUE.

Value

Produces text describing the degree and significance of difference from expected dispersion, and the degree and significance of zero inflation. The tests are chi-squared score tests based on the expectations of mean = variance, and 0s as abundant as predicted by the value of lambda estimated from the dataset. See van der Broek (1995) for more details.
Notes

This function subsets the data in the same way as `modelsearch()` before testing underlying distributions, making the output much more appropriate than a simple analysis of size and fecundity variables in data.

The specific test used for overdispersion is a chi-squared test of the dispersion parameter estimated using a generalized linear model predicting the response given size in time *t*, under a quasi-Poisson distribution.

The specific test used for zero-inflation is the chi-squared test presented in van der Broek (1995).

Examples

```r
# Lathyrux example
data(lathyrus)
sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9)
stagevector <- c("Sd", "Sdl", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr", "Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r", "Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repsvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)
lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repsvector, obsstatus = obsvector, matstatus = matvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propvector)
lathvertln$feca2 <- round(lathvertln$feca2)
lathvertln$feca1 <- round(lathvertln$feca1)
lathvertln$feca3 <- round(lathvertln$feca3)

# The following will only test fecundity, since size is Gaussian.
# Zero-inflation will not be assessed in this example, since 0 values in
# fecundity have been excluded in the life history model.
sf_distrib(lathvertln, size2 = "sizea2", fec = "feca2", repst = "repstatus2", zifec = FALSE)
```
# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSml", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)

binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = reppvector, obsstatus = obsvector, matstatus = matvector, propstatus = propvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)

sf_distrib(cypraw_v1, size2 = "size2added", fec = "feca2", repst = "repstatus2", zisize = FALSE)

---

### slambda3

**Estimate Stochastic Population Growth Rate**

**Description**

Function slambda3() estimates the stochastic population growth rate, $\lambda$, defined as the long-term arithmetic mean of the log population growth rate estimated per simulated time (as given in equation 2 in Tuljapurkar, Horvitz, and Pascarella 2003). This term is estimated via projection of randomly sampled matrices, similarly to the procedure outlined in Box 7.4 of Morris and Doak (2002). Can handle both lefkoMat objects and lists of full A matrices.

**Usage**

slambda3(mpm, times = 10000L, tweights = NULL)
Arguments

- **npm**: A matrix projection model of class `lefskoMat`, or a list of full matrix projection matrices.
- **times**: Number of iterations to random samples. Defaults to 10,000.
- **tweights**: Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among times.

Value

A data frame with the following variables:

- **pop**: The identity of the population.
- **patch**: The identity of the patch.
- **a**: Estimate of stochastic growth rate, estimated as the arithmetic mean of the log population growth rate across simulated times.
- **var**: The estimated variance of a.
- **sd**: The standard deviation of a.
- **se**: The standard error of a.

Stochastic growth rate is estimated both at the patch level and at the population level. Population level estimates will be noted at the end of the data frame with 0 entries for patch designation.

Notes

Weightings given in `tweights` do not need to sum to 1. Final weightings used will be based on the proportion per element of the sum of elements in the user-supplied vector.

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
immsstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
...
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathrepm <- matrix(0, 7, 7)
lathrepm[1, 6] <- 0.345
lathrepm[2, 6] <- 0.054
lathover3 <- overwrite(stage3 = c("Sd", "Sd", "Sdl"),
stage2 = c("Sd", "Sd", "Sd"), stage1 = c("Sd", "rep", "rep"),
givenrate = c(0.345, 0.345, 0.054))
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe,
year = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
repmatrix = lathrepm, overwrite = lathover3, yearcol = "year2",
indivcol = "individ")
slambda3(ehrlen3)
# Cypripedium example
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
obsvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
matvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
immvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypraw_v1, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
rep_cyp_raw <- matrix(0, 11, 11)
rep_cyp_raw[1:2, 7:11] <- 0.5
cyprop2r <- overwrite(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3", "SL", "SL", "SL", "D", "XSm", "Sm", "D", "XSm", "Sm"),
stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL"),
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypraw_v1, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
cypmatrix3r <- rlefko3(data = cypraw_v1, stageframe = cypframe_raw, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), size = c("size3added", "size2added", "size1added"), repmatrix = rep_cyp_raw, overwrite = cypover3r, yearcol = "year2", patchcol = "patchid", indivcol = "individ")
cypstoch <- slambda3(cypmatrix3r)
cypstoch

stablestage3

Estimate Stable Stage Distribution

Description

stablestage3() is a generic function that returns the stable stage distribution for a population projection matrix or set of matrices. This function is made to handle very large and sparse matrices supplied as lefkoMat objects or as individual matrices, and can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

Usage

stablestage3(mats, ...)

Arguments

mats A lefkoMat object, or population projection matrix, for which the stable stage distribution is desired.

... Other parameters.

Value

The value returned depends on the class of the mats argument. See related functions for details.
See Also

stablestage3.lefkomat()
stablestage3.matrix()

Examples

# Lathyrus deterministic example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = reppvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)
lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054),
type = c(1, 1, 1, 1, 1, 1, 1),
type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")
ehrlen3mean <- lmean(ehrlen3)
stablestage3(ehrlen3mean)

# Cypripedium stochastic example
rm(list=ls(all=TRUE))
data(cypdata)
stablestage3.lefkoMat

Estimate Stable Stage Distribution of Matrices in lefkoMat Object

Description

stablestage3.lefkoMat() returns the deterministic stable stage distributions of all $A$ matrices in an object of class lefkoMat, as well as the long-run projected mean stage distribution in stochastic
analysis. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

Usage

```r
## S3 method for class 'lefkoMat'
stablestage3(
mats,
stochastic = FALSE,
times = 10000,
tweights = NA,
seed = NA,
sparse = "auto",
...)
```

Arguments

- `mats`: An object of class `lefkoMat`.
- `stochastic`: A logical value indicating whether to use deterministic (FALSE) or stochastic (TRUE) analysis. Defaults to FALSE.
- `times`: An integer variable indicating number of times to project if using stochastic analysis. Defaults to 10000.
- `tweights`: An optional vector indicating the probability weighting to use for each matrix in stochastic simulations. If not given, then defaults to equal weighting.
- `seed`: A number to use as a random number seed.
- `sparse`: A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
- `...`: Other parameters.

Value

This function returns the stable stage distributions (and long-run mean stage distributions in stochastic analysis) corresponding to the matrices in a `lefkoMat` object.

The output depends on whether the `lefkoMat` object used as input is ahistorical or historical, and whether the analysis is deterministic or stochastic. If ahistorical, then a single data frame is output, which includes the number of the matrix within the $A$ element of the input `lefkoMat` object, followed by the stage id (numeric and assigned through `sf_create()`), the stage name, and the estimated proportion of the stable stage distribution ($ss_prop$).

If a historical matrix is used as input, then two data frames are output into a list object. The `$hist` element contains a data frame in which the stable stage distribution is given in terms of across-year stage pairs. The structure includes the matrix number, the numeric stage designations for stages in times $t$ and $t-1$, respectively, followed by the respective stage names, and ending with the estimated proportion of the stable stage distribution for that stage within its matrix ($ss_prop$). The `$ahist` element contains the stable stage distribution in stages as given in the original stageframe. It includes a data frame with the matrix of origin, the numeric stage designation, stage name, and the
stable stage distribution estimated as the sum of distribution elements from \$hist corresponding to the equivalent stage in time \( t \), irrespective of stage in time \( t-1 \).

In addition to the data frames noted above, stochastic analysis will result in the additional output of a list of matrices containing the actual projected stage distributions across all projected times, in the order of population-patch combinations in the lefkoMat input.

Notes

In stochastic analysis, the projected mean distribution is the arithmetic mean across the final 1000 projected times if the simulation is at least 2000 projected times long. If between 500 and 2000 projected times long, then only the final 200 are used, and if fewer than 500 times are used, then all are used. Note that because stage distributions in stochastic simulations can change greatly in the initial portion of the run, we encourage a minimum of 2000 projected times per simulation, with 10000 preferred.

See Also

stablestage3()
stablestage3.matrix()

Examples

# Lathyrus deterministic example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 0, 0, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 0)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 0)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,

repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,

patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,

juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",

fecacol = "Intactseed88", deadacol = "Dead1988",

nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",

censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsup3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "mat"),

stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),

stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all", stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3, yearcol = "year2", individcol = "individ")
ehrlen3mean <- lmean(ehrlen3)
stablestage3(ehrlen3mean)

# Cypripedium stochastic example
rm(list=ls(all=TRUE))
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
revector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = revector, obsstatus = obsvector, matstatus = matvector, propstatus = propvector, immstatus = immvector, indataset = indataset, binnhalfwidth = binvec)
cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = cypframe_raw, historical = FALSE)
stablestage3.matrix

stablestage3(matrix2r, stochastic = TRUE)

stablestage3.matrix  
Estimate Stable Stage Distribution of a Single Population Projection Matrix

Description

stablestage3.matrix() returns the stable stage distribution for a population projection matrix. This function can handle large and sparse matrices, and so can be used with large historical matrices, IPMs, age x stage matrices, as well as smaller ahistorical matrices.

Usage

## S3 method for class 'matrix'
stablestage3(mats, sparse = "auto", ...)

Arguments

mats A population projection matrix of class matrix.
sparse A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
...
 Other parameters.

Value

This function returns the stable stage distribution corresponding to the input matrix.

See Also

stablestage3()

stablestage3.lefkoMat()

Examples

#Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
  immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
  propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
  patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
  juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
  fecacol = "Intactseed88", deadacol = "Dead1988",
  nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
  censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sd", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
  multiplier = c(0.345, 0.054, NA, NA, 0.345, 0.054),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
stablestage3(ehrlen3mean$A[[1]])

# Cypripedium stochastic example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004, patchidcol = "patch", individcol = "plantid", blocksize = 4, sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04", repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE, NRasRep = TRUE)

# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"), stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL", "rep", "rep"), eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA), eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA), multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA), type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3), stageframe = cypframe_raw, historical = FALSE)

cymatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw, year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"), size = c("size3added", "size2added"), supplement = cypsupp2r, yearcol = "year2", patchcol = "patchid", individ = "individ")

stablestage3(cymatrix2r$A[[1]])

---

**Summary of Class "lefkoCondMat"**

**Description**

This function provides basic information summarizing the characteristics of conditional matrices derived from a lefkoCondMat object.

**Usage**

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

**Arguments**

- `object` An object of class lefkoCondMat.
- `...` Other parameters.
Value

A summary of the object, showing the number of historical matrices, as well as the number of conditional matrices nested within each historical matrix.

Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3700, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 1, 1, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immsvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
immsstatus = immsvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censcorel = "Missing1988", censurekeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl", "Sd", "Sdl", "mat"),
stage2 = c("Sd", "Sdl", "Sd", "Sdl", "Sd"),
stage1 = c("Sd", "Sdl", "Sd", "Sdl", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA),
type = c(1, 1, 1, 3, 0, 3, 0),
type_t12 = c(1, 2, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", individcol = "individ")

lathcondmats <- cond_hmpm(ehrlen3)
summary(lathcondmats)
```

# Cypripedium example
data(cypdata)
```
sizevector <- c(0, 0, 0, 0, 0, 0, 0, 0, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
repsvector <- c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
dataset <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)

cypsupp3r <- supplemental(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3",
"SL", "SL", "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"),
stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL",
"SL", "SL", "SL", "SL", "rep", "rep"),
stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "SL", "SL",
"P3", "P3", "SL", "SL", "all", "all"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
"D", "XSm", "Sm", "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
"XSm", "XSm", "XSm", NA, NA),
eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
"XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.2, 0.25, 0.4, 0.4, NA, NA, NA, NA,
NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
type_t12 = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = cypframe_raw, historical = TRUE)

cypmatrix3r <- rleko3(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added", "size1added"),
supplement = cypsupp3r, yearcol = "year2", patchcol = "patchid",
indivcol = "individ")

cypcondmats <- cond_hmpm(cypmatrix3r)
summary(cypcondmats)
**summary.lefkoElas**  
*Summarize lefkoElas Objects*

**Description**

Function `summary.lefkoElas()` summarizes `lefkoElas` objects. Particularly, it breaks down elasticity values by the kind of ahistorical and, if applicable, historical transition.

**Usage**

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

**Arguments**

- `object`  
  A `lefkoElas` object.

- `...`  
  Other parameters.

**Value**

A list composed of 2 data frames. The first, `hist`, is a data frame showing the summed elasticities for all 16 kinds of historical transition per matrix, with each column corresponding to each elasticity matrix in order. The second, `ahist`, is a data frame showing the summed elasticities for all 4 kinds of ahistorical transition per matrix, with each column corresponding to each elasticity matrix in order.

**Examples**

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repsvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repsvector, obsstatus = obsvector, matstatus = matvector,
imstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",

```
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)
lathsupp3 <- supplemental(stage3 = c("Sd", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, 0.345, 0.054, NA, NA),
type = c(1, 1, 1, 1, 3, 3, 1),
type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)
lathsupp2 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl"),
stage2 = c("Sd", "Sdl", "rep", "rep"),
givenrate = c(0.345, 0.054, NA, NA),
multiplier = c(NA, NA, 0.345, 0.054),
type = c(1, 1, 1, 3),
stageframe = lathframe, historical = FALSE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"),
yearcol = "year", indivcol = "individ")
ehrlen2 <- rlefko2(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2"),
yearcol = "year2", indivcol = "individ")
ehrlen3elas <- elasticity3(ehrlen3)
ehrlen2elas <- elasticity3(ehrlen2)
summary(ehrlen3elas)
summary(ehrlen2elas)

---

**summary.lefkoMat**  
*Summary of Class "lefkoMat"*

**Description**

A function to simplify the viewing of basic information describing the matrices produced through functions `flefko3()`, `flefko2()`, `rlefko3()`, and `rlefko2()`.

**Usage**

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

Arguments

object An object of class lefkoMat.

... Other parameters.

Value

A summary of the object, showing the number of each type of matrix, the number of annual matrices, the number of estimated (non-zero) elements across all matrices and per matrix, the number of unique transitions in the dataset, the number of individuals, and summaries of the column sums of the survival-transition matrices.

Examples

data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NAasRep = TRUE)
# Here we use supplemental() to provide overwrite and reproductive info
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA, 0.5, 0.5),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1),
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
**summary.lefkoMod**

```r
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", individcol = "individ")

summary(cypmatrix2r)
```

---

### Description

A function to summarize objects of class `lefkoMod`. This function shows the best-fit models, summarizes the numbers of models in the model tables, shows the criterion used to determine the best-fit models, and provides some basic quality control information.

### Usage

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

### Arguments

- `object`: An R object of class `lefkoMod` resulting from `modelsearch()`.
- `...`: Other parameters.

### Value

A summary of the object, showing the best-fit models for all vital rates, with constants of 0 or 1 used for unestimated models. This is followed by a summary of the number of models tested per vital rate, and a table showing the names of the parameters used to model vital rates and represent tested factors. At the end is a section describing the number of individuals and individual transitions used to estimate each vital rate best-fit model.

### Examples

```r
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9)
stagevector <- c("Sd", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
                   "Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
                   "Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
```
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5)

lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immsstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvertln <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "lnVol88", repstracol = "Intactseed88",
fecacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframeln,
stagesize = "sizea", censorcol = "Missing1988", censorkeep = NA,
NAas0 = TRUE, censor = TRUE)

lathvertln$feca2 <- round(lathvertln$feca2)
lathvertln$feca1 <- round(lathvertln$feca1)
lathvertln$feca3 <- round(lathvertln$feca3)

lathmodelsln2 <- modelsearch(lathvertln, historical = FALSE,
approach = "mixed", suite = "main",
vitalrates = c("surv", "obs", "size", "repst", "fec"), juvestimate = "Sdl",
bestfit = "AICc&k", sizedist = "gaussian", fecdist = "poisson",
indiv = "individ", patch = "patchid", year = "year2",
year.as.random = TRUE, patch.as.random = TRUE, show.model.tables = TRUE,
quiet = TRUE)

summary(lathmodelsln2)

supplemental

Create an Data Frame of Supplemental Data for MPM Development

Description

supplemental() provides all necessary supplemental data for matrix estimation, particularly bringing together data on proxy rates, data to overwrite existing rates, identified reproductive transitions complete, and fecundity multipliers.

Usage

supplemental(
  stage3,
  stage2,
  stage1 = NA,
eststage3 = NA,
eststage2 = NA,
eststage1 = NA,
givenrate = NA,
multiplier = NA,
type = NA,
type_t12 = NA,
stageframe,
historical = TRUE
)

**Arguments**

- **stage3**: The name of the stage in time \(t+1\) in the transition to be replaced. Abbreviations for groups of stages are also useable (see Notes).
- **stage2**: The name of the stage in time \(t\) in the transition to be replaced. Abbreviations for groups of stages are also useable (see Notes).
- **stage1**: The name of the stage in time \(t-1\) in the transition to be replaced. Only needed if a historical matrix is to be produced. Abbreviations for groups of stages are also useable (see Notes).
- **eststage3**: The name of the stage to replace \(stage3\). Only needed if a transition will be replaced by another estimated transition.
- **eststage2**: The name of the stage to replace \(stage2\). Only needed if a transition will be replaced by another estimated transition.
- **eststage1**: The name of the stage to replace \(stage1\). Only needed if a transition will be replaced by another estimated transition, and the matrix to be estimated is historical. Stage \texttt{NotAlive} is also possible for raw hMPMs, as a means of handling the prior stage for individuals entering the population in time \(t\).
- **givenrate**: A fixed rate or probability to replace for the transition described by \(stage3\), \(stage2\), and \(stage1\).
- **multiplier**: A vector of numeric multipliers for fecundity, and NA entries for all other terms.
- **type**: A vector denoting the kind of transition between times \(t\) and \(t+1\) to be replaced. This should be entered as 1, S, or s for the replacement of a survival transition; 2, \(F\), or \(f\) for the replacement of a fecundity transition; or 3, \(R\), or \(r\) for a fecundity multiplier. If empty or not provided, then defaults to 1 for survival transition.
- **type_t12**: An optional vector denoting the kind of transition between times \(t-1\) and \(t\). Only necessary if a historical MPM in deVries format is desired. This should be entered as 1, S, or s for a survival transition; or 2, \(F\), or \(f\) for a fecundity transitions. Defaults to 1 for survival transition, with impacts only on the construction of deVries-format hMPMs.
- **stageframe**: The stageframe being used to produce the MPMs in the study.
- **historical**: A logical value indicating whether the MPMs intended will be historical or ahistorical. Defaults to TRUE.
Value

A data frame of class lefkoSD. This object can be used as input in `fleko3()`, `fleko2()`, `rleko3()`, `rleko2()`, and `afleko2()`.

Variables in this object include the following:

- **stage3**: Stage at time $t+1$ in the transition to be replaced.
- **stage2**: Stage at time $t$ in the transition to be replaced.
- **stage1**: Stage at time $t-1$ in the transition to be replaced.
- **eststage3**: Stage at time $t+1$ in the transition to replace the transition designated by stage3, stage2, and stage1.
- **eststage2**: Stage at time $t$ in the transition to replace the transition designated by stage3, stage2, and stage1.
- **eststage1**: Stage at time $t-1$ in the transition to replace the transition designated by stage3, stage2, and stage1.
- **givenrate**: A constant to be used as the value of the transition.
- **convtype**: Designates whether the transition from time $t$ to time $t+1$ is a survival transition probability (1), a fecundity rate (2), or a fecundity multiplier (3).
- **convtype_t12**: Designates whether the transition from time $t-1$ to time $t$ is a survival transition probability (1), a fecundity rate (2).

Notes

Fecundity multiplier data supplied via the `supplemental()` function acts in the same way as non-zero entries supplied via a reproductive matrix, but gets priority in all matrix creations. Thus, in cases where fecundity multipliers are provided for the same function via the reproductive matrix and function `supplemental()`, the latter is used.

Entries in stage3, stage2, and stage1 can include abbreviations for groups of stages. Use `rep` if all reproductive stages are to be used, `nrep` if all mature but non-reproductive stages are to be used, `mat` if all mature stages are to be used, `inmat` if all immature stages are to be used, `prop` if all propagule stages are to be used, `npr` if all non-propagule stages are to be used, and leave empty or use `all` if all stages in stageframe are to be used.

Examples

```r
# Lathyrus example
data(lathyrus)
sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
revector <- c(0, 0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
```
lathframe <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec,
propstatus = propvector)

lathvert <- verticalize3(lathyrus, noyears = 4, firstyear = 1988,
patchidcol = "SUBPLOT", individcol = "GENET", blocksize = 9,
juvcol = "Seedling1988", sizeacol = "Volume88", repstracol = "FCODE88",
feacol = "Intactseed88", deadacol = "Dead1988",
nonobsacol = "Dormant1988", stageassign = lathframe, stagesize = "sizea",
censorcol = "Missing1988", censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sdl", "Sdl", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)

# Cypripedium example
rm(list=ls(all=TRUE))
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
"XLg")
repvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizec = "Veg.04",
sizevector = c(...), stagenames = stagevector,
repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
immstatus = immvector, indataset = indataset, binhalfwidth = binvec)
repstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cypframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D",
"XSm", "Sm", "SD", "P1"),
"rep"),
eststage3 = c(NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3),
stageframe = cypframe_raw, historical = FALSE)
cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", individcol = "individ")
cyp2mean <- lmean(cypmatrix2r)
cyp2mean

verticalize3

Create Historical Vertical Data Frame From Horizontal Data Frame

Description

verticalize3() returns a vertically formatted demographic data frame organized to create historical projection matrices, given a horizontally formatted input data frame.

Usage

verticalize3(
data,
noyears,
firstyear = 1,
popidcol = 0,
patchidcol = 0,
individcol = 0,
blocksize = NA,
xcol = 0,
ycol = 0,
juvcol = 0,
sizeacol,
sizebcol = 0,
sizeccol = 0,
repstracol = 0,
repstrbcol = 0,
fecacol = 0,
fecbcol = 0,
indcovacol = 0,
indcovbcol = 0,
indcovccol = 0,
aliveacol = 0,
deadacol = 0,
obsacol = 0,
onobsacol = 0,
censoracol = 0,
repstrrel = 1,
fecrel = 1,
stagecol = 0,
stageassign = NA,
stagesize = NA,
censorkeep = 0,
censorRepeat = TRUE,
censor = FALSE,
coordsRepeat = FALSE,
spacing = NA,
NAas0 = FALSE,
NRasRep = FALSE,
reduce = TRUE,
a2check = FALSE
)

**Arguments**

**data**
The horizontal data file. A valid data frame is required as input.

**noyears**
The number of years or observation periods in the dataset. A valid integer is required as input.

**firstyear**
The first year or time of observation. Defaults to 1.

**popidcol**
A variable name or column number corresponding to the identity of the population for each individual.

**patchidcol**
A variable name or column number corresponding to the identity of the patch for each individual, if patches have been designated within populations.

**individcol**
A variable name or column number corresponding to the identity of each individual.

**blocksize**
The number of variables corresponding to each time in the input dataset designated in `data`, if a set pattern of variables is used for each observation time in the data frame used as input. If such a pattern is not used, and all variable names are properly noted as character vectors in the other input variables, then this may be set to NA. Defaults to NA.

**xcol**
A variable name(s) or column number(s) corresponding to the x coordinate of each individual, or each individual-time combination, in Cartesian space.
refer to the only instance, the first instance, or all instances of x variables. In the last case, the values should be entered as a vector.

**ycol** A variable name(s) or column number(s) corresponding to the y coordinate of each individual, or each individual-time combination, in Cartesian space. Can refer to the only instance, the first instance, or all instances of y variables. In the last case, the values should be entered as a vector.

**juvcol** A variable name(s) or column number(s) that marks individuals in immature stages within the dataset. The `verticalize3()` function assumes that immature individuals are identified in this variable marked with a number equal to or greater than 1, and that mature individuals are marked as 0 or NA. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.

**sizeacol** A variable name(s) or column number(s) corresponding to the size entry associated with the first year or observation time in the dataset. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.

**sizebcol** A second variable name(s) or column number(s) corresponding to the size entry associated with the first year or observation time in the dataset. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.

**sizeccol** A third variable name(s) or column number(s) corresponding to the size entry associated with the first year or observation time in the dataset. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.

**repstracol** A variable name(s) or column number(s) corresponding to the production of reproductive structures, such as flowers, associated with the first year or observation period in the input dataset. This can be binomial or count data, and is used to in analysis of the probability of reproduction. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.

**repstrbcol** A second variable name(s) or column number(s) corresponding to the production of reproductive structures, such as flowers, associated with the first year or observation period in the input dataset. This can be binomial or count data, and is used to in analysis of the probability of reproduction. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.

**fecacol** A variable name(s) or column number(s) denoting fecundity associated with the first year or observation time in the input dataset. This may represent egg counts, fruit counts, seed production, etc. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.

**fecbcol** A second variable name(s) or column number(s) denoting fecundity associated with the first year or observation time in the input dataset. This may represent egg counts, fruit counts, seed production, etc. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.
indcovacol A variable name(s) or column number(s) corresponding to an individual covariate to be used in analysis. Can refer to the only instance, the first instance, or all instances of these variables. In the last case, the values should be entered as a vector.

indcovbcol A variable name(s) or column number(s) corresponding to an individual covariate to be used in analysis. Can refer to the only instance, the first instance, or all instances of these variables. In the last case, the values should be entered as a vector.

indcovccol A second variable name(s) or column number(s) corresponding to an individual covariate to be used in analysis. Can refer to the only instance, the first instance, or all instances of these variables. In the last case, the values should be entered as a vector.

aliveacol Variable name(s) or column number(s) providing information on whether an individual is alive at a given time. If used, living status must be designated as binomial (living = 1, dead = 0). Can refer to the first instance of a living status variable in the dataset, or equal a full vector of all living status variables in temporal order.

deadacol Variable name(s) or column number(s) providing information on whether an individual is alive at a given time. If used, dead status must be designated as binomial (dead = 1, living = 0). Can refer to the first instance of a dead status variable in the dataset, or equal a full vector of all dead status variables in temporal order.

obsacol A variable name(s) or column number(s) providing information on whether an individual is in an observable stage at a given time. If used, observation status must be designated as binomial (observed = 1, not observed = 0). Can refer to the first instance of an observation status variable in the dataset, or equal a full vector of all observation status variables in temporal order.

nonobsacol A variable name(s) or column number(s) providing information on whether an individual is in an unobservable stage at a given time. If used, observation status must be designated as binomial (not observed = 1, observed = 0). Can refer to the first instance of a non-observation status variable in the dataset, or equal a full vector of all non-observation status variables in temporal order.

censorcol A variable name(s) or column number(s) corresponding to the first entry of a censor variable, used to distinguish between entries to use and entries not to use, or to designate entries with special issues that require further attention. Can refer to the first instance of a censor status variable in the dataset, or equal a full vector of all censor status variables in temporal order. Can also refer to a single censor status variable used for the entire individual, if singlecensor = TRUE.

repstrrel This is a scalar multiplier on variable repstrbcol to make it equivalent to repstracol. This can be useful if two reproductive status variables have related but unequal units, for example if repstracol refers to one-flowered stems while repstrbcol refers to two-flowered stems. Defaults to 1.

fecrel This is a scalar multiplier on variable fecbcol to make it equivalent to fecacol. This can be useful if two fecundity variables have related but unequal units. Defaults to 1.
stagecol

Optional variable name(s) or column number(s) corresponding to life history stage at a given time. Can refer to the first instance of a stage identity variable in the dataset, or equal a full vector of all stage identity variables in temporal order.

stageassign

The stageframe object identifying the life history model being operationalized. Note that if stagecol is provided, then this stageframe is not used for stage designation.

stagesize

A variable name or column number describing which size variable to use in stage estimation. Defaults to NA, and can also take sizea, sizeb, sizec, or sizeadded, depending on which size variable is chosen.

censorkeep

The value of the censor variable identifying data to be included in analysis. Defaults to 0, but may take any value including NA. Note that if NA is the value to keep, then this function will alter all NAs to 0 values, and all other values to 1, treating 0 as the value to keep.

censorRepeat

A logical value indicating whether the censor variable is a single column, or whether it repeats across time blocks. Defaults to TRUE.

censor

A logical variable determining whether the output data should be censored using the variable defined in censorcol. Defaults to FALSE.

coordsRepeat

A logical value indicating whether x and y coordinates correspond to a single x and a single y column. If TRUE, then each observation time has its own x and y variables. Defaults to FALSE.

spacing

The spacing at which density should be estimated, if density estimation is desired and x and y coordinates are supplied. Given in the same units as those used in the x and y coordinates given in xcol and ycol. Defaults to NA.

NAas0

If TRUE, then all NA entries for size and fecundity variables will be set to 0. This can help increase the sample size analyzed by modelsearch(), but should only be used when it is clear that this substitution is biologically realistic. Defaults to FALSE.

NRasRep

If TRUE, then will treat non-reproductive but mature individuals as reproductive during stage assignment. This can be useful when a matrix is desired without separation of reproductive and non-reproductive but mature stages of the same size. Only used if stageassign is set to a stageframe. Defaults to FALSE.

reduce

A logical variable determining whether unused variables and some invariant state variables should be removed from the output dataset. Defaults to TRUE.

a2check

A logical variable indicating whether to retain all data with living status at time \(t\) equal to 0. Defaults to FALSE, and should be kept on FALSE except to inspect potential errors in the dataset.

Value

If all inputs are properly formatted, then this function will output a historical vertical data frame (class hfvdata), meaning that the output data frame will have three consecutive times of size and reproductive data per individual per row. This data frame is in standard format for all functions used in lefko3, and so can be used without further modification.

Variables in this data frame include the following:
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rowid</td>
<td>Unique identifier for the row of the data frame.</td>
</tr>
<tr>
<td>popid</td>
<td>Unique identifier for the population, if given.</td>
</tr>
<tr>
<td>patchid</td>
<td>Unique identifier for patch within population, if given.</td>
</tr>
<tr>
<td>individ</td>
<td>Unique identifier for the individual.</td>
</tr>
<tr>
<td>year2</td>
<td>Year or time at time $t$.</td>
</tr>
<tr>
<td>firstseen</td>
<td>Year or time of first observation.</td>
</tr>
<tr>
<td>lastseen</td>
<td>Year or time of last observation.</td>
</tr>
<tr>
<td>obsage</td>
<td>Observed age in time $t$, assuming first observation corresponds to age $= 0$.</td>
</tr>
<tr>
<td>obslifespan</td>
<td>Observed lifespan, given as $\text{lastseen} - \text{firstseen} + 1$.</td>
</tr>
<tr>
<td>xpos1,xpos2,xpos3</td>
<td>X position in Cartesian space in times $t-1$, $t$, and $t+1$, respectively,</td>
</tr>
<tr>
<td>ypos1,ypos2,ypos3</td>
<td>Y position in Cartesian space in times $t-1$, $t$, and $t+1$, respectively,</td>
</tr>
<tr>
<td>size1,size2,size3</td>
<td>Main size measurement in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>sizeb1,sizeb2,sizeb3</td>
<td>Secondary size measurement in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>sizec1,sizec2,sizec3</td>
<td>Tertiary measurement in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>size1added,size2added,size3added</td>
<td>Sum of primary, secondary, and tertiary size measurements in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>repstra1,repstra2,repstra3</td>
<td>Main numbers of reproductive structures in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>repstrb1,repstrb2,repstrb3</td>
<td>Secondary numbers of reproductive structures in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>repstr1added,repstr2added,repstr3added</td>
<td>Sum of primary and secondary reproductive structures in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>fecal,feca2,feca3</td>
<td>Main numbers of offspring in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>fecb1,fecb2,fecb3</td>
<td>Secondary numbers of offspring in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>fec1added,fec2added,fec3added</td>
<td>Sum of primary and secondary fecundity in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>censor1,censor2,censor3</td>
<td>Censor state values in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
<tr>
<td>juvgiven1,juvgiven2,juvgiven3</td>
<td>Binomial variable indicating whether individual is juvenile in times $t-1$, $t$, and $t+1$. Only given if juvcol is provided.</td>
</tr>
<tr>
<td>obsstatus1,obsstatus2,obsstatus3</td>
<td>Binomial observation state in times $t-1$, $t$, and $t+1$, respectively.</td>
</tr>
</tbody>
</table>
Binomial reproductive state in times \( t-1 \), \( t \), and \( t+1 \), respectively.

Binomial offspring production state in times \( t-1 \), \( t \), and \( t+1 \), respectively.

Binomial maturity state in times \( t-1 \), \( t \), and \( t+1 \), respectively.

Binomial state as alive in times \( t-1 \), \( t \), and \( t+1 \), respectively.

Density of individuals per unit designated in spacing. Only given if spacing is not NA.

Notes

In some datasets on species with unobservable stages, observation status (\( \text{obsstatus} \)) might not be inferred properly if a single size variable is used that does not yield sizes greater than 0 in all cases in which individuals were observed. Such situations may arise, for example, in plants when leaf number is the dominant size variable used, but individuals occasionally occur with inflorescences but no leaves. In this instances, it helps to mark related variables as \( \text{sizeb} \) and \( \text{sizec} \), because observation status will be interpreted in relation to all 3 size variables. Further analysis can then utilize only a single size variable, of the user’s choosing. Similar issues can arise in reproductive status (\( \text{repstatus} \)).

Warnings that some individuals occur in state combinations that do not match any stages in the stageframe used to assign stages are common when first working with a dataset. Typically, these situations can be identified as \( \text{NoMatch} \) entries in \( \text{stage3} \), although such entries may crop up in \( \text{stage1} \) and \( \text{stage2} \), as well. In rare cases, these warnings will arise with no concurrent \( \text{NoMatch} \) entries, which indicates that the input dataset contained conflicting state data at once suggesting that the individual is in some stage but is also dead. The latter is removed if the conflict occurs in time \( t \) or time \( t-1 \), as only living entries are allowed in these times.

Care should be taken to avoid variables with negative values indicating size, fecundity, or reproductive or observation status. Negative values can be interpreted in different ways, typically reflecting estimation through other algorithms rather than actual measured data. Variables holding negative values can conflict with data management algorithms in ways that are difficult to predict.

Examples

```r
# Lathyrus example using blocksize - when repeated patterns exist in variable
# order
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sdl", "VSm", "Sm", "VLa", "Flo", "Dorm")
reppvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(1, 0, 0, 0, 0, 0, 0)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)
```
verticalize3

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propvector)


lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "SDl", "SDl", "SDl", "mat"), stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "SDl"), stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"), eststage3 = c("c(NA, NA, NA, NA, NA, NA, "mat"), eststage2 = c("c(NA, NA, NA, NA, NA, NA, "SDl"), eststage1 = c("c(NA, NA, NA, NA, NA, NA, "NotAlive"), givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA), multiplier = c("c(NA, NA, NA, NA, 0.345, 0.054, NA, NA), type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1), stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all", stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3, yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
ehrlen3mean$A[[1]]

# Lathyrus example without blocksize - when no repeated patterns exist in # variable order and all variables names are specified

data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "SDl", "VSm", "Sm", "VLa", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

lathframe <- sf_create(sizes = sizevector, stagenames = stagevector, repstatus = repvector, obsstatus = obsvector, matstatus = matvector, immstatus = immvector, indataset = indataset, binhalfwidth = binvec, propstatus = propvector)

repstracol = c("FCODE88", "FCODE89", "FCODE90", "FCODE91"),
fecacol = c("Intactseed88", "Intactseed89", "Intactseed90", "Intactseed91"),
deadacol = c("Dead1988", "Dead1989", "Dead1990", "Dead1991"),
stageassign = lathframe, stagesize = "sizea",
censorkeep = NA, censor = TRUE)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "mat"),
stage2 = c("Sd", "Sd", "Sd", "rep", "rep", "Sdl"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sdl"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
giventype = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)
ehrlen3 <- rlefko3(data = lathvert, stageframe = lathframe, year = "all",
stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
yearcol = "year2", indivcol = "individ")
ehrlen3mean <- lmean(ehrlen3)
ehrlen3mean$A[[1]]

# Cypripedium example using blocksize
rm(list=ls(all=TRUE))
data(cypdata)
sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg", "XLg")
reppvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0)
immvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)
cyframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = repvector, obststatus = obsvector, matstatus = matvector,
propstatus = propvector, immstatus = immvector, indataset = indataset,
binhalfwidth = binvec)
cyraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
patchidcol = "patch", individcol = "plantid", blocksize = 4,
sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
reppstracol = "Inf.04", repstrbcol = "Inf2.04", fecacol = "Pod.04",
stageassign = cyframe_raw, stagesize = "sizeadded", NAas0 = TRUE,
NRasRep = TRUE)
cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
  type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
  stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsupp2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cyp2mean <- lmean(cypmatrix2r)
cyp2mean

# Cypripedium example using partial repeat patterns with blocksize and part
# explicit variable name cast
data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
revector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

cypframe_raw <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = revector, obsstatus = obsvector, matstatus = matvector,
  propstatus = propvector, immstatus = immvector, indataset = indataset,
  binhalfwidth = binvec)

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
  patchidcol = "patch", individcol = "plantid", blocksize = 4,
  sizeacol = "Inf2.04", sizebcol = "Inf.04", sizeccol = "Veg.04",
  repstracol = c("Inf.04", "Inf.05", "Inf.06", "Inf.07", "Inf.08", "Inf.09"),
  repstrbcol = c("Inf2.04", "Inf2.05", "Inf2.06", "Inf2.07", "Inf2.08", "Inf2.09"),
  fecacol = "Pod.04", stageassign = cypframe_raw, stagesize = "sizeadded",
  NAas0 = TRUE, NRasRep = TRUE)

cypsupp2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "SL", "D", "XSm", "Sm", "SD", "P1"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
  type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
  stageframe = cypframe_raw, historical = FALSE)
eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 0.40, NA, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cypframe_raw,
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsupp2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")
cyp2mean <- lmean(cypmatrix2r)
cyp2mean
Index

* datasets
  cypdata, 11
  cypvert, 14
  lathyrus, 69
aflefko2, 3, 84, 144
cond_hmpm, 9
cypdata, 11
cypvert, 14
dredge, 81
elasticity3, 16, 19, 22, 25
elasticity3.lefkoMat, 16, 18, 22, 25
elasticity3.list, 16, 20, 21, 25
elasticity3.matrix, 16, 20, 22, 24
flefko2, 27, 84, 119, 139, 144
flefko3, 33, 84, 118, 119, 139, 144
glm, 80
glm.nb, 80
glmer, 80
glmmTMB, 80
historicalize3, 40, 77
image, 47, 49, 52, 54, 57, 59
image3, 47
image3.lefkoElas, 49
image3.lefkoMat, 48, 52
image3.lefkoSens, 54
image3.list, 57
image3.matrix, 48, 59
lambda3, 61, 64, 67
lambda3.lefkoMat, 62, 63, 67
lambda3.matrix, 62, 64, 66
lathyrus, 69
lefko3, 72
lm, 80
lmean, 73
lmer, 80
modelsearch, 44, 76, 123, 141, 150
overwrite, 4, 28, 35, 83, 98, 103
projection3, 85
repvalue3, 88, 92, 95
repvalue3.lefkoMat, 89, 90, 94, 95
repvalue3.matrix, 89, 92, 94
rlefko2, 3, 27, 84, 96, 119, 139, 144
rlefko3, 3, 27, 33, 84, 101, 119, 139, 144
sensitivity3, 107, 110, 113, 116
sensitivity3.lefkoMat, 107, 109, 113, 116
sensitivity3.list, 107, 110, 112, 116
sensitivity3.matrix, 107, 110, 113, 115
set.seed, 7, 31, 37
sf_create, 91, 117, 130
sf_distrib, 121
slambda3, 62, 64, 67, 124
stablestage3, 127, 131, 133
stablestage3.lefkoMat, 128, 129, 133
stablestage3.matrix, 128, 131, 133
summary.lefkoCondMat, 135
summary.lefkoElas, 26, 138
summary.lefkoMat, 139
summary.lefkoMod, 141
supplemental, 4, 28, 35, 98, 103, 142
verticalize3, 77, 97, 102, 146
vglm, 81
zeroinfl, 80, 81

157