Package ‘lefko3’

November 16, 2021

Type Package

Title Historical and Ahistorical Population Projection Matrix Analysis

Version 4.0.1

Date 2021-11-15

Description Creates matrix population models for use in population ecological analyses. Specializes on the construction of historical matrices, which are 2d matrices comprising 3 consecutive times of demographic information. Estimates both raw and function-based forms of historical and standard ahistorical matrices. It also estimates function-based age-by-stage matrices. Methodology based on Ehrlen (2000) <doi:10.1890/0012-9658(2000)081[1675:TDOPPD]2.0.CO;2> and deVries and Caswell (2018) <doi:10.1007/s12080-017-0353-0>.

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 7.1.2

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@g.ecc.u-tokyo.ac.jp>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2021-11-16 07:50:02 UTC

R topics documented:

add_lM *Add Matrices to lefkoMat Object*

Description

Function add_lM() adds matrices to lefkoMat objects.

Usage

```
add_lM(
  lM,
  Amats = NA,
  Umats = NA,
  Fmats = NA,
  UFdecomp = FALSE,
  entrystage = 1,
  pop = NA,
  patch = NA,
  year = NA
)
```

Arguments

lM	The lefkoMat object to add matrices to.
Amats	Either a single A matrix, or a list of A matrices. Not necessary if Umats and Fmats are both provided.
Umats	Either a single U matrix, or a list of U matrices. Not necessary if Amats and Fmats are both provided, or if UFdecomp = TRUE and entrystage is provided.
Fmats	Either a single F matrix, or a list of U matrices. Not necessary if Amats and Umats are both provided, or if UFdecomp = TRUE and entrystage is provided.
UFdecomp	A logical value indicating whether U and F matrices should be inferred from A matrices and the given entrystage. Defaults to TRUE.

`entrystage` The stage or stages produced by reproductive individuals. Used to determine which transitions are reproductive for U-F decomposition. Defaults to `1`, which corresponds to the first stage in the stageframe.

`pop` The population designation for each matrix. If object `lM` includes only a single population, then defaults to that designation. Otherwise requires a designation as input.

`patch` The patch designation for each matrix. If object `lM` includes only a single patch, then defaults to that designation. Otherwise requires a designation as input.

`year` The designation for occasion at time *t* corresponding to each matrix. Cannot be left empty.

Value

A `lefkoMat` object incorporating the new matrices within the object input in `lM`.

Notes

This function will not allow matrices of different dimension from those input in object `lM` to be added to that object.

Two of `Amats`, `Umats`, and `Fmats` must be provided for this function to proceed. Also, if `Amats`, `Umats`, and `Fmats` are all provided, then this function will default to replacing `Amats` with the sum of the respective `Umats` and `Fmats`.

See Also

`create_lM()`

`delete_lM()`

`subset_lM()`

Examples

```
# These matrices are of 9 populations of the plant species Anthyllis
# vulneraria, and were originally published in Davison et al. (2010) Journal
# of Ecology 98:255-267 (doi: 10.1111/j.1365-2745.2009.01611.x).

sizevector <- c(1, 1, 2, 3) # These sizes are not from the original paper
stagevector <- c("Sdl", "Veg", "SmFlo", "LFlo")
repvector <- c(0, 0, 1, 1)
obsvector <- c(1, 1, 1, 1)
matvector <- c(0, 1, 1, 1)
immvector <- c(1, 0, 0, 0)
propvector <- c(0, 0, 0, 0)
indataset <- c(1, 1, 1, 1)
binvec <- c(0.5, 0.5, 0.5, 0.5)

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

```
# POPN C 2003-2004
XC3 <- matrix(c(0, 0, 1.74, 1.74,
0.208333333, 0, 0, 0.057142857,
0.041666667, 0.076923077, 0, 0,
0.083333333, 0.076923077, 0.066666667, 0.028571429), 4, 4, byrow = TRUE)

# 2004-2005
XC4 <- matrix(c(0, 0, 0.3, 0.6,
0.32183908, 0.142857143, 0, 0,
0.16091954, 0.285714286, 0, 0,
0.252873563, 0.285714286, 0.5, 0.6), 4, 4, byrow = TRUE)

# 2005-2006
XC5 <- matrix(c(0, 0, 0.50625, 0.675,
0, 0, 0, 0.035714286,
0.1, 0.068965517, 0.0625, 0.107142857,
0.3, 0.137931034, 0, 0.071428571), 4, 4, byrow = TRUE)

# POPN E 2003-2004
XE3 <- matrix(c(0, 0, 2.44, 6.569230769,
0.196428571, 0, 0, 0,
0.125, 0.5, 0, 0,
0.160714286, 0.5, 0.133333333, 0.076923077), 4, 4, byrow = TRUE)

XE4 <- matrix(c(0, 0, 0.45, 0.646153846,
0.06557377, 0.090909091, 0.125, 0,
0.032786885, 0, 0.125, 0.076923077,
0.049180328, 0, 0.125, 0.230769231), 4, 4, byrow = TRUE)

XE5 <- matrix(c(0, 0, 2.85, 3.99,
0.083333333, 0, 0, 0,
0, 0, 0, 0,
0.416666667, 0.1, 0, 0.1), 4, 4, byrow = TRUE)

# POPN F 2003-2004
XF3 <- matrix(c(0, 0, 1.815, 7.058333333,
0.075949367, 0, 0.05, 0.083333333,
0.139240506, 0, 0, 0.25,
0.075949367, 0, 0, 0.083333333), 4, 4, byrow = TRUE)

XF4 <- matrix(c(0, 0, 1.233333333, 7.4,
0.223880597, 0, 0.111111111, 0.142857143,
0.134328358, 0.272727273, 0.166666667, 0.142857143,
0.119402985, 0.363636364, 0.055555556, 0.142857143), 4, 4, byrow = TRUE)

XF5 <- matrix(c(0, 0, 1.06, 3.372727273,
0.073170732, 0.025, 0.033333333, 0,
0.036585366, 0.15, 0.1, 0.136363636,
0.06097561, 0.225, 0.166666667, 0.272727273), 4, 4, byrow = TRUE)

# POPN G 2003-2004
XG3 <- matrix(c(0, 0, 0.245454545, 2.1,
0, 0, 0.045454545, 0,
0.125, 0, 0.090909091, 0,
0.125, 0, 0.090909091, 0.333333333), 4, 4, byrow = TRUE)

XG4 <- matrix(c(0, 0, 1.1, 1.54,
0.111111111, 0, 0, 0,
0, 0, 0, 0,
0.111111111, 0, 0, 0), 4, 4, byrow = TRUE)

XG5 <- matrix(c(0, 0, 0, 1.5,
0, 0, 0, 0,
0.090909091, 0, 0, 0,
0.545454545, 0.5, 0, 0.5), 4, 4, byrow = TRUE)

# POPN L 2003-2004
XL3 <- matrix(c(0, 0, 1.785365854, 1.856521739,
0.128571429, 0, 0, 0.010869565,
0.028571429, 0, 0, 0,
0.014285714, 0, 0, 0.02173913), 4, 4, byrow = TRUE)

XL4 <- matrix(c(0, 0, 14.25, 16.625,
0.131443299, 0.057142857, 0, 0.25,
0.144329897, 0, 0, 0,
0.092783505, 0.2, 0, 0.25), 4, 4, byrow = TRUE)

XL5 <- matrix(c(0, 0, 0.594642857, 1.765909091,
0, 0, 0.017857143, 0,
0.021052632, 0.018518519, 0.035714286, 0.045454545,
0.021052632, 0.018518519, 0.035714286, 0.068181818), 4, 4, byrow = TRUE)

# POPN O 2003-2004
XO3 <- matrix(c(0, 0, 11.5, 2.775862069,
0.6, 0.285714286, 0.333333333, 0.24137931,
0.04, 0.142857143, 0, 0,
0.16, 0.285714286, 0, 0.172413793), 4, 4, byrow = TRUE)

XO4 <- matrix(c(0, 0, 3.78, 1.225,
0.28358209, 0.171052632, 0, 0.166666667,
0.084577114, 0.026315789, 0, 0.055555556,
0.139303483, 0.447368421, 0, 0.305555556), 4, 4, byrow = TRUE)

XO5 <- matrix(c(0, 0, 1.542857143, 1.035616438,
0.126984127, 0.105263158, 0.047619048, 0.054794521,
0.095238095, 0.157894737, 0.19047619, 0.082191781,
0.111111111, 0.223684211, 0, 0.356164384), 4, 4, byrow = TRUE)

# POPN Q 2003-2004
XQ3 <- matrix(c(0, 0, 0.15, 0.175,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0, 0, 0), 4, 4, byrow = TRUE)

XQ4 <- matrix(c(0, 0, 0, 0.25,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0.666666667, 0, 1), 4, 4, byrow = TRUE)

XQ5 <- matrix(c(0, 0, 0, 1.428571429,
0, 0, 0, 0.142857143,
0.25, 0, 0, 0,
0.25, 0, 0, 0.571428571), 4, 4, byrow = TRUE)

# POPN R 2003-2004
XR3 <- matrix(c(0, 0, 0.7, 0.6125,
0.25, 0, 0, 0.125,
0, 0, 0, 0,
0.25, 0.166666667, 0, 0.25), 4, 4, byrow = TRUE)

XR4 <- matrix(c(0, 0, 0, 0.6,
0.285714286, 0, 0, 0,
0.285714286, 0.333333333, 0, 0,
0.285714286, 0.333333333, 0, 1), 4, 4, byrow = TRUE)

XR5 <- matrix(c(0, 0, 0.7, 0.6125,
0, 0, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0.333333333, 0.625), 4, 4, byrow = TRUE)

# POPN S 2003-2004
XS3 <- matrix(c(0, 0, 2.1, 0.816666667,
0.166666667, 0, 0, 0,
0, 0, 0, 0,
0, 0, 0, 0.166666667), 4, 4, byrow = TRUE)

XS4 <- matrix(c(0, 0, 0, 7,
0.333333333, 0.5, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0, 1), 4, 4, byrow = TRUE)

XS5 <- matrix(c(0, 0, 0, 1.4,
0, 0, 0, 0,
0, 0, 0, 0.2,
0.111111111, 0.75, 0, 0.2), 4, 4, byrow = TRUE)

mats_list <- list(XC3, XC4, XC5, XE3, XE4, XE5, XF3, XF4, XF5, XG3, XG4, XG5,
  XL3, XL4, XL5, XO3, XO4, XO5, XQ3, XQ4, XQ5, XR3, XR4, XR5, XS3, XS4, XS5)

yr_ord <- c(1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1,
  2, 3, 1, 2, 3)

pch_ord <- c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7,
  8, 8, 8, 9, 9, 9)

anth_lefkoMat <- create_lM(mats_list, anthframe, hstages = NA, historical = FALSE,
  poporder = 1, patchorder = pch_ord, yearorder = yr_ord)
```

```
# POPN H (EXCLDED FROM ANALYSIS B/C OF UNREALISTIC ELASTICITIES)
XH3 <- matrix(c(0, 0, 0.1125, 1.05,
0.2, 0, 0, 0,
0, 0.5, 0, 0,
0.2, 0.5, 0, 0), 4, 4, byrow = TRUE)

XH3u <- matrix(c(0, 0, 0, 0,
0.2, 0, 0, 0,
0, 0.5, 0, 0,
0.2, 0.5, 0, 0), 4, 4, byrow = TRUE)

XH4 <- matrix(c(0, 0, 0, 0,
0, 0, 0.5, 0,
0.8, 0.5, 0.25, 0.25,
0.2, 0, 0, 0.75), 4, 4, byrow = TRUE)

XH4u <- matrix(c(0, 0, 0, 0,
0, 0, 0.5, 0,
0.8, 0.5, 0.25, 0.25,
0.2, 0, 0, 0.75), 4, 4, byrow = TRUE)

XH5 <- matrix(c(0, 0, 0.2, 1.05,
0, 0, 0, 0,
0.001, 0.001, 0.333333333, 0, #ELEMENTS (3,1),(4,1),(3,2) REPLACED W NONZERO
0.001, 0, 0, 0), 4, 4, byrow = TRUE)

XH5u <- matrix(c(0, 0, 0, 0,
0, 0, 0, 0,
0.001, 0.001, 0.333333333, 0, #ELEMENTS (3,1),(4,1),(3,2) REPLACED W NONZERO
0.001, 0, 0, 0), 4, 4, byrow = TRUE)

anth_lefkoMat <- add_lM(anth_lefkoMat, Amats = list(XH3, XH4, XH5),
  Umats = list(XH3u, XH4u, XH5u), patch = c(10, 10, 10), year = c(1, 2, 3))

anth_lefkoMat
```

aflefko2 *Create Function-based Ahistorical Age x Stage Matrix Projection Model*

Description

Function aflefko2() returns ahistorical age x stage MPMs corresponding to the patches and occasion times given, including the associated component transition and fecundity matrices, data frames 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 occasion time combinations corresponding to these matrices.

Usage

```
aflefko2(
  year = "all",
  patch = "all",
  stageframe,
  supplement = NULL,
  repmatrix = NULL,
  overwrite = NULL,
  data = NA,
  modelsuite = NA,
  surv_model = NA,
  obs_model = NA,
  size_model = NA,
  sizeb_model = NA,
  sizec_model = NA,
  repst_model = NA,
  fec_model = NA,
  jsurv_model = NA,
  jobs_model = NA,
  jsize_model = NA,
  jsizeb_model = NA,
  jsizec_model = NA,
  jrepst_model = NA,
  paramnames = NA,
  inda = NULL,
  indb = NULL,
  indc = NULL,
  surv_dev = 0,
  obs_dev = 0,
  size_dev = 0,
  sizeb_dev = 0,
  sizec_dev = 0,
  repst_dev = 0,
  fec_dev = 0,
  jsurv_dev = 0,
  jobs_dev = 0,
  jsize_dev = 0,
  jsizeb_dev = 0,
  jsizec_dev = 0,
  jrepst_dev = 0,
  density = NA,
  repmod = 1,
  yearcol = NA,
  patchcol = NA,
  year.as.random = FALSE,
  patch.as.random = FALSE,
  random.inda = FALSE,
  random.indb = FALSE,
  random.indc = FALSE,
  final_age = 10,
  continue = TRUE,
  randomseed = NA,
  negfec = FALSE,
  reduce = FALSE,
  err_check = FALSE,
  exp_tol = 700,
  theta_tol = 1e+08
)
```

Arguments

year
: A variable corresponding to observation occasion, or a set of such values, given in values associated with the year term used in linear model development. Defaults to "all", in which case matrices will be estimated for all occasions.

patch
: A variable designating which patches or subpopulations will have matrices estimated. Defaults to "all", but can also be set to specific patch names.

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 survival and 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
: An optional reproduction matrix. This matrix is composed mostly of 0s, with non-zero entries acting as element identifiers and multipliers for fecundity (with 1 equaling full fecundity). If left blank, and no supplement is provided, then aflefko2() will assume that all stages marked as reproductive produce offspring at 1x that of estimated fecundity, and that offspring production will yield 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 stage-based 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 historical vertical demographic data frame used to estimate vital rates (class hfvdata). The original data frame is required in order to initialize occasions and patches properly.

modelsuite
: An optional lefkoMod object holding the vital rate models. If given, then surv_model, obs_model, size_model, sizeb_model, sizec_model, repst_model, fec_model,

`jsurv_model`, `jobs_model`, `jsize_model`, `jsizeb_model`, `jsizec_model`, `jrepst_model`, `paramnames`, `yearcol`, and `patchcol` are not required. No models should include size or reproductive status in occasion *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 occasion *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 occasion *t*.

`size_model` A linear model predicting primary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any primary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`sizeb_model` A linear model predicting secondary size. This can be a modelvof class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any secondary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`sizec_model` A linear model predicting tertiary size. This can be a modelvof class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any tertiary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *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 occasion *t*.

`fec_model` A linear model predicting fecundity. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. 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 occasion *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 occasion *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 occasion *t*.

`jsize_model` A linear model predicting juvenile primary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile primary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`jsizeb_model` A linear model predicting juvenile secondary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile secondary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`jsizec_model` A linear model predicting juvenile tertiary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile tertiary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *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 occasion *t*.

`paramnames` A dataframe with three columns, the first describing all terms used in linear modeling, the second (must be called `mainparams`), showing the general model terms that will be used in matrix creation (users should use `modelsearch()` at least once to see the proper names to be used in this column), and the third showing the equivalent terms used in modeling (must be named `modelparams`). Only required if `modelsuite` is not supplied.

`inda` Can be a single value to use for individual covariate `a` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`indb` Can be a single value to use for individual covariate `b` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`indc` Can be a single value to use for individual covariate `c` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`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 primary size.

`sizeb_dev` A numeric value to be added to the y-intercept in the linear model for secondary size.

`sizec_dev` A numeric value to be added to the y-intercept in the linear model for tertiary 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 primary size.

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

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

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

`density` A numeric value indicating density value to use to propagate matrices. Only needed if density is an explanatory term used in linear models. Defaults to `NA`.

`repmod` A scalar multiplier of fecundity. Defaults to `1`.

`yearcol` The variable name or column number corresponding to year in occasion *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 occasions within vital rate models should be estimated as random intercepts. Defaults to `FALSE`, in which case missing monitoring occasion 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`.

`random.inda` A logical value denoting whether to treat individual covariate `a` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`random.indb` A logical value denoting whether to treat individual covariate `b` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`random.indc` A logical value denoting whether to treat individual covariate `c` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`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 past the final age noted in the stageframe, using the demographic characteristics of the final age.

`randomseed` A numeric value used as a seed to generate random estimates for missing occasion 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 append matrices of vital rate probabilities associated with each matrix to the `lefkoMat` object generated. These matrices are developed internally and can be used for error checking. Defaults to `FALSE`.

`exp_tol` A numeric value used to indicate a maximum value to set exponents to in the core kernel to prevent numerical overflow. Defaults to `700`.

`theta_tol` A numeric value used to indicate a maximum value to theta as used in the negative binomial probability density kernel. Defaults to `100000000`, but can be reset to other values during error checking.

Value

If all inputs are properly formatted, then this function will return an object of class `lefkoMat`, which is a list that holds the matrix projection model and all of its metadata. Its structure is a list with the following elements:

`A` A list of full projection matrices in order of sorted patches and occasions. All matrices output in R's `matrix` class.

`U` A list of survival transition matrices sorted as in `A`. All matrices output in R's `matrix` class.

`F` A list of fecundity matrices sorted as in `A`. All matrices output in R's `matrix` class.

`hstages` A data frame matrix showing the pairing of ahistorical stages used to create historical stage pairs. Set to `NA` 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 6 columns in the order of survival, observation probability, reproduction probability, primary size transition probability, secondary size transition probability, and tertiary size transition probability.

Notes

Unlike `rlefko2()` and `rlefko3()`, this function does not currently distinguish populations.

This function will yield incorrect estimates if the models utilized incorporate state in occasion *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 `aflefko2()` will fail and produce an error.

Users may at times wish to estimate MPMs using a dataset incorporating multiple patches or subpopulations, but without discriminating between those 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 set to `NA`, which is the default.

Input options including multiple variable names must be entered in the order of variables in occasion *t*+1 and *t*. Rearranging the order will lead to erroneous calculations, and may lead to fatal errors.

Care should be taken to match the random status of year and patch to the states of those variables within the modelsuite. If they do not match, then they will be treated as zeroes in vital rate estimation.

Using the `err_check` option will produce a matrix of 6 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.

Users may produce age-based (Leslie) MPMs using this function. In that case, stages must be defined as occurring serially within single ages in the `stageframe`, with the possible exception of the final stage (which sometimes involves a perpetual stasis transition)..

Examples

```
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, 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)
minima <- c(0, 0, 1, 1, 1, 1, 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,
  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, 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, minage = minima, maxage = maxima)

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", age = "obsage",
  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", "Sd", "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)

lathmat2age <- aflefko2(year = "all", patch = "all",
  stageframe = lathframeln, modelsuite = lathmodelsln2, data = lathvertln,
  supplement = lathsupp2, patchcol = "patchid",
  yearcol = "year2", year.as.random = FALSE, patch.as.random = FALSE,
  final_age = 2, continue = TRUE, reduce = FALSE)

summary(lathmat2age)
```

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, 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,
  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", "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", "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, "D", "XSm", "Sm", "D",
    "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm",
    "XSm", "XSm", "XSm", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm",
    "XSm", "XSm", "XSm", 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, 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, 3),
  type_t12 = c(1, 2, 1, 2, 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)
```

create_lM *Create lefkoMat Object from Given Input Matrices*

Description

Function `create_lM()` creates lefkoMat objects from supplied matrices and extra information.

Usage

```
create_lM(
  mats,
  stageframe,
  hstages = NA,
  agestages = NA,
  historical = FALSE,
  agebystage = FALSE,
  UFdecomp = TRUE,
  entrystage = 1,
  poporder = 1,
  patchorder = 1,
  yearorder = NA
)
```

Arguments

`mats` A list of A matrices.

`stageframe` A stageframe describing all stages utilized.

`hstages` A data frame outlining the order of historical stages, if matrices provided in `mats` are historical. Defaults to NA.

`agestages` A data frame outlining the order of ahistorical age-stages, if age-by-stage matrices are provided.

`historical` A logical value indicating whether input matrices are historical or not. Defaults to FALSE.

`agebystage` A logical value indicating whether input matrices are ahistorical age-by-stage matrices. If TRUE, then object `agestages` is required. Defaults to FALSE.

`UFdecomp` A logical value indicating whether U and F matrices should be inferred. Defaults to TRUE.

`entrystage` The stage or stages produced by reproductive individuals. Used to determine which transitions are reproductive for U-F decomposition. Defaults to `1`, which corresponds to the first stage in the stageframe.

`poporder` The order of populations in the list supplied in object `mats`. Defaults to 1.

`patchorder` The order of patches in the list supplied in object `mats`. Defaults to 1.

`yearorder` The order of monitoring occasions in the list supplied in object `mats`. Defaults to NA, which leads to each matrix within each population-patch combination being a different monitoring occasion.

Value

A `lefkoMat` object incorporating the matrices input in object `mats` as object `A`, their U and F decompositions in objects `U` and `F` (if requested), the provided stageframe as object `ahstages`, the order of historical stages as object `hstages` (if `historical` = `TRUE`), the order of matrices as object `labels`, and a short quality control section used by the `summary.lefkoMat()` function.

Notes

U and F decomposition assumes that elements holding fecundity values are to be interpreted solely as fecundity rates. Users wishing to split these elements between fecundity and survival should do so manually after running this function.

Age-by-stage MPMs require an `agestages` data frame outlining the order of age-stages. This data frame has 3 variables: `stage_id`, which is the number of the stage as labelled by the equivalently named variable in the `stageframe`; `stage`, which is the official name of the stage as given in the equivalently named variable in the `stageframe`; and `age`, which of course gives the age associated with the stage at that time. The number of rows must be equal to the number of rows and columns of each entered matrix.

See Also

add_lM()

delete_lM()

subset_lM()

Examples

```
# These matrices are of 9 populations of the plant species Anthyllis
# vulneraria, and were originally published in Davison et al. (2010) Journal
# of Ecology 98:255-267 (doi: 10.1111/j.1365-2745.2009.01611.x).

sizevector <- c(1, 1, 2, 3) # These sizes are not from the original paper
stagevector <- c("Sdl", "Veg", "SmFlo", "LFlo")
repvector <- c(0, 0, 1, 1)
obsvector <- c(1, 1, 1, 1)
matvector <- c(0, 1, 1, 1)
immvector <- c(1, 0, 0, 0)
propvector <- c(0, 0, 0, 0)
indataset <- c(1, 1, 1, 1)
binvec <- c(0.5, 0.5, 0.5, 0.5)

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

# POPN C 2003-2004
XC3 <- matrix(c(0, 0, 1.74, 1.74,
0.208333333, 0, 0, 0.057142857,
0.041666667, 0.076923077, 0, 0,
0.083333333, 0.076923077, 0.066666667, 0.028571429), 4, 4, byrow = TRUE)

# 2004-2005
XC4 <- matrix(c(0, 0, 0.3, 0.6,
0.32183908, 0.142857143, 0, 0,
0.16091954, 0.285714286, 0, 0,
0.252873563, 0.285714286, 0.5, 0.6), 4, 4, byrow = TRUE)

# 2005-2006
XC5 <- matrix(c(0, 0, 0.50625, 0.675,
0, 0, 0, 0.035714286,
0.1, 0.068965517, 0.0625, 0.107142857,
0.3, 0.137931034, 0, 0.071428571), 4, 4, byrow = TRUE)

# POPN E 2003-2004
XE3 <- matrix(c(0, 0, 2.44, 6.569230769,
0.196428571, 0, 0, 0,
0.125, 0.5, 0, 0,
0.160714286, 0.5, 0.133333333, 0.076923077), 4, 4, byrow = TRUE)

XE4 <- matrix(c(0, 0, 0.45, 0.646153846,
0.06557377, 0.090909091, 0.125, 0,
0.032786885, 0, 0.125, 0.076923077,
0.049180328, 0, 0.125, 0.230769231), 4, 4, byrow = TRUE)

XE5 <- matrix(c(0, 0, 2.85, 3.99,
0.083333333, 0, 0, 0,
0, 0, 0, 0,
0.416666667, 0.1, 0, 0.1), 4, 4, byrow = TRUE)

# POPN F 2003-2004
XF3 <- matrix(c(0, 0, 1.815, 7.058333333,
0.075949367, 0, 0.05, 0.083333333,
0.139240506, 0, 0, 0.25,
0.075949367, 0, 0, 0.083333333), 4, 4, byrow = TRUE)

XF4 <- matrix(c(0, 0, 1.233333333, 7.4,
0.223880597, 0, 0.111111111, 0.142857143,
0.134328358, 0.272727273, 0.166666667, 0.142857143,
0.119402985, 0.363636364, 0.055555556, 0.142857143), 4, 4, byrow = TRUE)

XF5 <- matrix(c(0, 0, 1.06, 3.372727273,
0.073170732, 0.025, 0.033333333, 0,
0.036585366, 0.15, 0.1, 0.136363636,
0.06097561, 0.225, 0.166666667, 0.272727273), 4, 4, byrow = TRUE)

# POPN G 2003-2004
XG3 <- matrix(c(0, 0, 0.245454545, 2.1,
0, 0, 0.045454545, 0,
0.125, 0, 0.090909091, 0,
0.125, 0, 0.090909091, 0.333333333), 4, 4, byrow = TRUE)

XG4 <- matrix(c(0, 0, 1.1, 1.54,
0.111111111, 0, 0, 0,
0, 0, 0, 0,
0.111111111, 0, 0, 0), 4, 4, byrow = TRUE)

XG5 <- matrix(c(0, 0, 0, 1.5,
0, 0, 0, 0,
0.090909091, 0, 0, 0,
0.545454545, 0.5, 0, 0.5), 4, 4, byrow = TRUE)

# POPN L 2003-2004
XL3 <- matrix(c(0, 0, 1.785365854, 1.856521739,
0.128571429, 0, 0, 0.010869565,
0.028571429, 0, 0, 0,
0.014285714, 0, 0, 0.02173913), 4, 4, byrow = TRUE)

XL4 <- matrix(c(0, 0, 14.25, 16.625,
0.131443299, 0.057142857, 0, 0.25,
0.144329897, 0, 0, 0,
0.092783505, 0.2, 0, 0.25), 4, 4, byrow = TRUE)

XL5 <- matrix(c(0, 0, 0.594642857, 1.765909091,
0, 0, 0.017857143, 0,
0.021052632, 0.018518519, 0.035714286, 0.045454545,
0.021052632, 0.018518519, 0.035714286, 0.068181818), 4, 4, byrow = TRUE)

# POPN O 2003-2004
XO3 <- matrix(c(0, 0, 11.5, 2.775862069,
0.6, 0.285714286, 0.333333333, 0.24137931,
0.04, 0.142857143, 0, 0,
0.16, 0.285714286, 0, 0.172413793), 4, 4, byrow = TRUE)

XO4 <- matrix(c(0, 0, 3.78, 1.225,
0.28358209, 0.171052632, 0, 0.166666667,
0.084577114, 0.026315789, 0, 0.055555556,
0.139303483, 0.447368421, 0, 0.305555556), 4, 4, byrow = TRUE)

XO5 <- matrix(c(0, 0, 1.542857143, 1.035616438,
0.126984127, 0.105263158, 0.047619048, 0.054794521,
0.095238095, 0.157894737, 0.19047619, 0.082191781,
0.111111111, 0.223684211, 0, 0.356164384), 4, 4, byrow = TRUE)

# POPN Q 2003-2004
XQ3 <- matrix(c(0, 0, 0.15, 0.175,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0, 0, 0), 4, 4, byrow = TRUE)

XQ4 <- matrix(c(0, 0, 0, 0.25,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0.666666667, 0, 1), 4, 4, byrow = TRUE)

XQ5 <- matrix(c(0, 0, 0, 1.428571429,
0, 0, 0, 0.142857143,
0.25, 0, 0, 0,
0.25, 0, 0, 0.571428571), 4, 4, byrow = TRUE)

# POPN R 2003-2004
XR3 <- matrix(c(0, 0, 0.7, 0.6125,
0.25, 0, 0, 0.125,
0, 0, 0, 0,
0.25, 0.166666667, 0, 0.25), 4, 4, byrow = TRUE)

XR4 <- matrix(c(0, 0, 0, 0.6,
0.285714286, 0, 0, 0,
0.285714286, 0.333333333, 0, 0,
0.285714286, 0.333333333, 0, 1), 4, 4, byrow = TRUE)

XR5 <- matrix(c(0, 0, 0.7, 0.6125,
0, 0, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0.333333333, 0.625), 4, 4, byrow = TRUE)

# POPN S 2003-2004
XS3 <- matrix(c(0, 0, 2.1, 0.816666667,
0.166666667, 0, 0, 0,
0, 0, 0, 0,
0, 0, 0, 0.166666667), 4, 4, byrow = TRUE)

XS4 <- matrix(c(0, 0, 0, 7,
0.333333333, 0.5, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0, 1), 4, 4, byrow = TRUE)

XS5 <- matrix(c(0, 0, 0, 1.4,
0, 0, 0, 0,
0, 0, 0, 0.2,
0.111111111, 0.75, 0, 0.2), 4, 4, byrow = TRUE)

mats_list <- list(XC3, XC4, XC5, XE3, XE4, XE5, XF3, XF4, XF5, XG3, XG4, XG5,
  XL3, XL4, XL5, XO3, XO4, XO5, XQ3, XQ4, XQ5, XR3, XR4, XR5, XS3, XS4, XS5)

yr_ord <- c(1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1,
  2, 3, 1, 2, 3)

pch_ord <- c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7,
  8, 8, 8, 9, 9, 9)

anth_lefkoMat <- create_lM(mats_list, anthframe, hstages = NA, historical = FALSE,
  poporder = 1, patchorder = pch_ord, yearorder = yr_ord)

anth_lefkoMat

# A theoretical example showcasing historical matrices

sizevector <- c(1, 2, 3) # These sizes are not from the original paper
stagevector <- c("Sdl", "Veg", "Flo")
repvector <- c(0, 0, 1)
obsvector <- c(1, 1, 1)
matvector <- c(0, 1, 1)
immvector <- c(1, 0, 0)
propvector <- c(1, 0, 0)
indataset <- c(1, 1, 1)
binvec <- c(0.5, 0.5, 0.5)

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

A1 <- matrix(c(0.10, 0, 0, 0.12, 0, 0, 0.15, 0, 0,
  0.15, 0, 0, 0.17, 0, 0, 0.20, 0, 0,
  0.20, 0, 0, 0.22, 0, 0, 0.25, 0, 0,
  0, 0.20, 0, 0, 0.22, 0, 0, 0.25, 0,
  0, 0.25, 0, 0, 0.27, 0, 0, 0.30, 0,
  0, 0.30, 0, 0, 0.32, 0, 0, 0.35, 0,
  0, 0, 2.00, 0, 0, 3.00, 0, 0, 4.00,
  0, 0, 0.35, 0, 0, 0.37, 0, 0, 0.40,
  0, 0, 0.40, 0, 0, 0.42, 0, 0, 0.45), 9, 9, byrow = TRUE)

A2 <- matrix(c(0.10, 0, 0, 0.12, 0, 0, 0.15, 0, 0,
  0.15, 0, 0, 0.17, 0, 0, 0.20, 0, 0,
  0.20, 0, 0, 0.22, 0, 0, 0.25, 0, 0,
  0, 0.20, 0, 0, 0.22, 0, 0, 0.25, 0,
  0, 0.25, 0, 0, 0.27, 0, 0, 0.30, 0,
  0, 0.30, 0, 0, 0.32, 0, 0, 0.35, 0,
  0, 0, 5.00, 0, 0, 6.00, 0, 0, 7.00,
  0, 0, 0.35, 0, 0, 0.37, 0, 0, 0.40,
  0, 0, 0.40, 0, 0, 0.42, 0, 0, 0.45), 9, 9, byrow = TRUE)

A3 <- matrix(c(0.10, 0, 0, 0.12, 0, 0, 0.15, 0, 0,
  0.15, 0, 0, 0.17, 0, 0, 0.20, 0, 0,
  0.20, 0, 0, 0.22, 0, 0, 0.25, 0, 0,
  0, 0.20, 0, 0, 0.22, 0, 0, 0.25, 0,
  0, 0.25, 0, 0, 0.27, 0, 0, 0.30, 0,
  0, 0.30, 0, 0, 0.32, 0, 0, 0.35, 0,
  0, 0, 8.00, 0, 0, 9.00, 0, 0, 10.00,
  0, 0, 0.35, 0, 0, 0.37, 0, 0, 0.40,
  0, 0, 0.40, 0, 0, 0.42, 0, 0, 0.45), 9, 9, byrow = TRUE)

B1 <- matrix(c(0.10, 0, 0, 0.12, 0, 0, 0.15, 0, 0,
  0.15, 0, 0, 0.17, 0, 0, 0.20, 0, 0,
  0.20, 0, 0, 0.22, 0, 0, 0.25, 0, 0,
  0, 0.20, 0, 0, 0.22, 0, 0, 0.25, 0,
  0, 0.25, 0, 0, 0.27, 0, 0, 0.30, 0,
  0, 0.30, 0, 0, 0.32, 0, 0, 0.35, 0,
  0, 0, 11.00, 0, 0, 12.00, 0, 0, 13.00,
  0, 0, 0.35, 0, 0, 0.37, 0, 0, 0.40,
  0, 0, 0.40, 0, 0, 0.42, 0, 0, 0.45), 9, 9, byrow = TRUE)

B2 <- matrix(c(0.10, 0, 0, 0.12, 0, 0, 0.15, 0, 0,
  0.15, 0, 0, 0.17, 0, 0, 0.20, 0, 0,
  0.20, 0, 0, 0.22, 0, 0, 0.25, 0, 0,
  0, 0.20, 0, 0, 0.22, 0, 0, 0.25, 0,
  0, 0.25, 0, 0, 0.27, 0, 0, 0.30, 0,
  0, 0.30, 0, 0, 0.32, 0, 0, 0.35, 0,
  0, 0, 14.00, 0, 0, 15.00, 0, 0, 16.00,
  0, 0, 0.35, 0, 0, 0.37, 0, 0, 0.40,
  0, 0, 0.40, 0, 0, 0.42, 0, 0, 0.45), 9, 9, byrow = TRUE)

B3 <- matrix(c(0.10, 0, 0, 0.12, 0, 0, 0.15, 0, 0,
  0.15, 0, 0, 0.17, 0, 0, 0.20, 0, 0,
  0.20, 0, 0, 0.22, 0, 0, 0.25, 0, 0,
  0, 0.20, 0, 0, 0.22, 0, 0, 0.25, 0,
  0, 0.25, 0, 0, 0.27, 0, 0, 0.30, 0,
  0, 0.30, 0, 0, 0.32, 0, 0, 0.35, 0,
  0, 0, 17.00, 0, 0, 18.00, 0, 0, 19.00,
  0, 0, 0.35, 0, 0, 0.37, 0, 0, 0.40,
  0, 0, 0.40, 0, 0, 0.42, 0, 0, 0.45), 9, 9, byrow = TRUE)

histmats <- list(A1, A2, A3, B1, B2, B3)
stageframe <- exframe
pch_ord <- c("A", "A", "A", "B", "B", "B")
yr_ord <- c(1, 2, 3, 1, 2, 3)

hist_trial <- create_lM(histmats, exframe, historical = TRUE, UFdecomp = TRUE,
  entrystage = 1, patchorder = pch_ord, yearorder = yr_ord)

hist_trial
```

---

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

```
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 refering 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

Shefferson, R.P., R. Mizuta, and M.J. Hutchings. 2017. Predicting evolution in response to climate change: the example of sprouting probability in three dormancy-prone orchid species. *Royal Society Open Science* 4(1):160647.

Examples

```
# Cypripedium example using blocksize
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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

lambda3(cypmatrix2r)

# 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")
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,
  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", "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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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

lambda3(cypmatrix2r)
```

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 occasion *t* and `.3` to identify occasion *t*+1.

plantid A numeric variable giving a unique number to each individual.

patch A variable refering 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 occasion *t*.

Inf2.2 Number of double inflorescences in occasion *t*.

Inf.2 Number of inflorescences in occasion *t*.

Veg.2 Number of stems without inflorescences in occasion *t*.

Pod.2 Number of fruits in occasion *t*.

Inf2.3 Number of double inflorescences in occasion *t*+1.

Inf.3 Number of inflorescences in occasion *t*+1.

Veg.3 Number of stems without inflorescences in occasion *t*+1.

Pod.3 Number of fruits in occasion *t*+1.

Source

Shefferson, R.P., R. Mizuta, and M.J. Hutchings. 2017. Predicting evolution in response to climate change: the example of sprouting probability in three dormancy-prone orchid species. *Royal Society Open Science* 4(1):160647.

Examples

```
data(cypvert)

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)

cypframe_raw

cypraw_v2 <- 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", repstrrel = 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", "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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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_v2, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2"),
  size = c("size3added", "size2added"), supplement = cypsupp2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

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

lambda3(cypmatrix2r)
```

delete_lM *Delete Matrices from lefkoMat Object*

Description

Function delete_lM() deletes matrices from lefkoMat objects.

Usage

```
delete_lM(lM, mat_num = NA, pop = NA, patch = NA, year = NA)
```

Arguments

lM	The lefkoMat object to delete matrices from.
mat_num	Either a single integer corresponding to the matrix to remove within the labels element of lM, or a vector of such integers.
pop	The population designation for matrices to remove. Only used if mat_num is not given.
patch	The patch designation for matrices to remove. Only used if mat_num is not given.
year	The time *t* designation for matrices to remove. Only used if mat_num is not given.

Value

A lefkoMat object in which the matrices specified in lM have been removed.

Notes

If mat_num is not provided, then at least one of pop, patch, or year must be provided. If at least two of pop, patch, and year are provided, then function detele_lM() will identify matrices to remove as the intersection of provided inputs.

See Also

create_lM()

add_lM()

subset_lM()

Examples

```
# These matrices are of 9 populations of the plant species Anthyllis
# vulneraria, and were originally published in Davison et al. (2010) Journal
# of Ecology 98:255-267 (doi: 10.1111/j.1365-2745.2009.01611.x).

sizevector <- c(1, 1, 2, 3) # These sizes are not from the original paper
stagevector <- c("Sdl", "Veg", "SmFlo", "LFlo")
repvector <- c(0, 0, 1, 1)
obsvector <- c(1, 1, 1, 1)
matvector <- c(0, 1, 1, 1)
immvector <- c(1, 0, 0, 0)
propvector <- c(0, 0, 0, 0)
indataset <- c(1, 1, 1, 1)
binvec <- c(0.5, 0.5, 0.5, 0.5)

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

# POPN C 2003-2004
XC3 <- matrix(c(0, 0, 1.74, 1.74,
0.208333333, 0, 0, 0.057142857,
0.041666667, 0.076923077, 0, 0,
0.083333333, 0.076923077, 0.066666667, 0.028571429), 4, 4, byrow = TRUE)

# 2004-2005
XC4 <- matrix(c(0, 0, 0.3, 0.6,
0.32183908, 0.142857143, 0, 0,
0.16091954, 0.285714286, 0, 0,
0.252873563, 0.285714286, 0.5, 0.6), 4, 4, byrow = TRUE)

# 2005-2006
XC5 <- matrix(c(0, 0, 0.50625, 0.675,
0, 0, 0, 0.035714286,
0.1, 0.068965517, 0.0625, 0.107142857,
0.3, 0.137931034, 0, 0.071428571), 4, 4, byrow = TRUE)

# POPN E 2003-2004
XE3 <- matrix(c(0, 0, 2.44, 6.569230769,
0.196428571, 0, 0, 0,
0.125, 0.5, 0, 0,
0.160714286, 0.5, 0.133333333, 0.076923077), 4, 4, byrow = TRUE)

XE4 <- matrix(c(0, 0, 0.45, 0.646153846,
0.06557377, 0.090909091, 0.125, 0,
0.032786885, 0, 0.125, 0.076923077,
0.049180328, 0, 0.125, 0.230769231), 4, 4, byrow = TRUE)

XE5 <- matrix(c(0, 0, 2.85, 3.99,
0.083333333, 0, 0, 0,
0, 0, 0, 0,
0.416666667, 0.1, 0, 0.1), 4, 4, byrow = TRUE)

# POPN F 2003-2004
XF3 <- matrix(c(0, 0, 1.815, 7.058333333,
0.075949367, 0, 0.05, 0.083333333,
0.139240506, 0, 0, 0.25,
0.075949367, 0, 0, 0.083333333), 4, 4, byrow = TRUE)

XF4 <- matrix(c(0, 0, 1.233333333, 7.4,
0.223880597, 0, 0.111111111, 0.142857143,
0.134328358, 0.272727273, 0.166666667, 0.142857143,
0.119402985, 0.363636364, 0.055555556, 0.142857143), 4, 4, byrow = TRUE)

XF5 <- matrix(c(0, 0, 1.06, 3.372727273,
0.073170732, 0.025, 0.033333333, 0,
0.036585366, 0.15, 0.1, 0.136363636,
0.06097561, 0.225, 0.166666667, 0.272727273), 4, 4, byrow = TRUE)

# POPN G 2003-2004
XG3 <- matrix(c(0, 0, 0.245454545, 2.1,
0, 0, 0.045454545, 0,
0.125, 0, 0.090909091, 0,
0.125, 0, 0.090909091, 0.333333333), 4, 4, byrow = TRUE)

XG4 <- matrix(c(0, 0, 1.1, 1.54,
0.111111111, 0, 0, 0,
0, 0, 0, 0,
0.111111111, 0, 0, 0), 4, 4, byrow = TRUE)

XG5 <- matrix(c(0, 0, 0, 1.5,
0, 0, 0, 0,
0.090909091, 0, 0, 0,
0.545454545, 0.5, 0, 0.5), 4, 4, byrow = TRUE)

# POPN L 2003-2004
XL3 <- matrix(c(0, 0, 1.785365854, 1.856521739,
0.128571429, 0, 0, 0.010869565,
0.028571429, 0, 0, 0,
0.014285714, 0, 0, 0.02173913), 4, 4, byrow = TRUE)

XL4 <- matrix(c(0, 0, 14.25, 16.625,
0.131443299, 0.057142857, 0, 0.25,
0.144329897, 0, 0, 0,
0.092783505, 0.2, 0, 0.25), 4, 4, byrow = TRUE)

XL5 <- matrix(c(0, 0, 0.594642857, 1.765909091,
0, 0, 0.017857143, 0,
0.021052632, 0.018518519, 0.035714286, 0.045454545,
0.021052632, 0.018518519, 0.035714286, 0.068181818), 4, 4, byrow = TRUE)

# POPN O 2003-2004
XO3 <- matrix(c(0, 0, 11.5, 2.775862069,
0.6, 0.285714286, 0.333333333, 0.24137931,
0.04, 0.142857143, 0, 0,
0.16, 0.285714286, 0, 0.172413793), 4, 4, byrow = TRUE)

XO4 <- matrix(c(0, 0, 3.78, 1.225,
0.28358209, 0.171052632, 0, 0.166666667,
0.084577114, 0.026315789, 0, 0.055555556,
0.139303483, 0.447368421, 0, 0.305555556), 4, 4, byrow = TRUE)

XO5 <- matrix(c(0, 0, 1.542857143, 1.035616438,
0.126984127, 0.105263158, 0.047619048, 0.054794521,
0.095238095, 0.157894737, 0.19047619, 0.082191781,
0.111111111, 0.223684211, 0, 0.356164384), 4, 4, byrow = TRUE)

# POPN Q 2003-2004
XQ3 <- matrix(c(0, 0, 0.15, 0.175,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0, 0, 0), 4, 4, byrow = TRUE)

XQ4 <- matrix(c(0, 0, 0, 0.25,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0.666666667, 0, 1), 4, 4, byrow = TRUE)

XQ5 <- matrix(c(0, 0, 0, 1.428571429,
0, 0, 0, 0.142857143,
0.25, 0, 0, 0,
0.25, 0, 0, 0.571428571), 4, 4, byrow = TRUE)

# POPN R 2003-2004
XR3 <- matrix(c(0, 0, 0.7, 0.6125,
0.25, 0, 0, 0.125,
0, 0, 0, 0,
0.25, 0.166666667, 0, 0.25), 4, 4, byrow = TRUE)

XR4 <- matrix(c(0, 0, 0, 0.6,
0.285714286, 0, 0, 0,
0.285714286, 0.333333333, 0, 0,
0.285714286, 0.333333333, 0, 1), 4, 4, byrow = TRUE)

XR5 <- matrix(c(0, 0, 0.7, 0.6125,
0, 0, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0.333333333, 0.625), 4, 4, byrow = TRUE)

# POPN S 2003-2004
```

```
XS3 <- matrix(c(0, 0, 2.1, 0.816666667,
0.166666667, 0, 0, 0,
0, 0, 0, 0,
0, 0, 0, 0.166666667), 4, 4, byrow = TRUE)

XS4 <- matrix(c(0, 0, 0, 7,
0.333333333, 0.5, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0, 1), 4, 4, byrow = TRUE)

XS5 <- matrix(c(0, 0, 0, 1.4,
0, 0, 0, 0,
0, 0, 0, 0.2,
0.111111111, 0.75, 0, 0.2), 4, 4, byrow = TRUE)

mats_list <- list(XC3, XC4, XC5, XE3, XE4, XE5, XF3, XF4, XF5, XG3, XG4, XG5,
  XL3, XL4, XL5, XO3, XO4, XO5, XQ3, XQ4, XQ5, XR3, XR4, XR5, XS3, XS4, XS5)

yr_ord <- c(1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1,
  2, 3, 1, 2, 3)

pch_ord <- c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7,
  8, 8, 8, 9, 9, 9)

anth_lefkoMat <- create_lM(mats_list, anthframe, hstages = NA, historical = FALSE,
  poporder = 1, patchorder = pch_ord, yearorder = yr_ord)

smaller_anth_lM <- delete_lM(anth_lefkoMat, patch = 3)
smaller_anth_lM
```

---

density_input *Create a Data Frame of Elements Subject to Density Dependence*

Description

Function density_input() provides all necessary data to incorporate density dependence into a lefkoMat object, a list of matrices, or a single matrix. Three forms of density dependence are allowed, including the Ricker function, the Beverton-Holt function, the Usher function, and the logistic function. In each case, density must have an effect with at least a one time-step delay (see Notes).

Usage

```
density_input(
  mpm,
  stage3,
  stage2,
  stage1 = NA,
  age2 = NA,
  style = 1,
  time_delay = 1,
  alpha = NA,
  beta = NA,
  type = NA,
  type_t12 = NA
)
```

Arguments

mpm — The lefkoMat object that will be subject to density dependent projection.

stage3 — A vector showing the name or number of the stage in occasion *t*+1 in the transitions to be affected by density. Abbreviations for groups of stages are also usable (see Notes).

stage2 — A vector showing the name or number of the stage in occasion *t* in the transition to be affected by density. Abbreviations for groups of stages are also usable (see Notes).

stage1 — A vector showing the name or number of the stage in occasion *t*-1 in the transition to be affected by density. Only needed if a historical MPM is used. Abbreviations for groups of stages are also usable (see Notes).

age2 — A vector showing the age of the stage in occasion *t* in the transition to be affected by density. Only needed if an age-by-stage MPM is used.

style — A vector coding for the style of density dependence on each transition subject to density dependence. Options include 1, ricker, ric, or r for the Ricker function; 2, beverton, bev, and b for the Beverton-Holt function; 3, usher, ush, and u for the Usher function; and 4, logistic, log, and l for the logistic function. If only a single code is provided, then all noted transitions are assumed to be subject to this style of density dependence. Defaults to ricker.

time_delay — A vector indicating the number of occasions back on which density dependence operates. Defaults to 1, and may not equal any number less than 1. If a single number is input, then all noted transitions are assumed to be subject to this time delay.

alpha — A vector indicating the numeric values to use as the alpha term in the two parameter Ricker, Beverton-Holt, or Usher function, or the value of the carrying capacity *K* to use in the logistic equation (see Notes section for more on this term). If a single number is provided, then all noted transitions are assumed to be subject to this value of alpha.

beta — A vector indicating the numeric values to use as the beta term in the two parameter Ricker, Beverton-Holt, or Usher function. Not used in the logistic equation. If a single number is provided, then all noted transitions are assumed to be subject to this value of beta.

type — A vector denoting the kind of transition between occasions *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 occasions *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.

Value

A data frame of class `lefkoDens`. This object can be used as input in function `projection3()`.

Variables in this object include the following:

`stage3` Stage at occasion *t*+1 in the transition to be replaced.

`stage2` Stage at occasion *t* in the transition to be replaced.

`stage1` Stage at occasion *t*-1 in the transition to be replaced, if applicable.

`age2` Age at occasion *t* in the transition to be replaced, if applicable.

`style` Style of density dependence, coded as 1, 2, 3, or 4 for the Ricker, Beverton-Holt, Usher, or logistic function, respectively.

`time_delay` The time delay on density dependence, in time steps.

`alpha` The value of alpha in the Ricker, Beverton-Holt, or Usher function, or the value of carrying capacity, *K*, in the logistic function.

`beta` The value of beta in the Ricker, Beverton-Holt, or Usher function.

`type` Designates whether the transition from occasion *t* to occasion *t*+1 is a survival transition probability (1), or a fecundity rate (2).

`type_t12` Designates whether the transition from occasion *t*-1 to occasion *t* is a survival transition probability (1), a fecundity rate (2).

Notes

The parameters `alpha` and `beta` are applied according to the two-parameter Ricker function, the two-parameter Beverton-Holt function, the two-parameter Usher function, or the one-parameter logistic function. Although the default is that a 1 time step delay is assumed), greater time delays can be set through the `time_delay` option.

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, `immat` 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.

See Also

`start_input()`

`projection3()`

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, 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", "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)

e3d <- density_input(ehrlen3mean, stage3 = c("Sd", "Sdl"),
  stage2 = c("rep", "rep"), stage1 = c("all", "all"), style = 1,
  time_delay = 1, alpha = 1, beta = 0, type = c(2, 2), type_t12 = c(1, 1))
```

---

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 λ in the former case and the log of the stochastic λ 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()

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, 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", "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)
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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

elasticity3(cypmatrix2r)
```

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 λ is taken as the population growth rate. If stochastic, then stochastic λ, 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

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

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 occasions 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 occasions.

`sparse`	A text string indicating whether to use sparse matrix encoding (`"yes"`) or dense matrix encoding (`"no"`). Defaults to `"auto"`.
`append_mats`	A logical value indicating whether to include the original A, U, and F matrices in the output `lefkoElas` object.
`...`	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 occasions *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 occasion *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 suppled. 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 λ, while stochastic sensitivities are with regard to the log of the stochastic λ.

See Also

`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")
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", "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", indivcol = "individ")

elasticity3(ehrlen3, 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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

elasticity3(cypmatrix2r)
```

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

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

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 occasions 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 occasions.

`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"`.

`append_mats` A logical value indicating whether to include the original matrices input as object `mats` in the output `lefkoElas` object.

`...` 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 λ, while stochastic sensitivities are with regard to the log of the stochastic λ.

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")
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", "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", indivcol = "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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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

```
## 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

```
# 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)
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", "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)
elasticity3(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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

elasticity3(cypmatrix2r$A[[1]])
```

flefko2 *Create Function-based Ahistorical Matrix Projection Model*

Description

Function `flefko2()` returns ahistorical MPMs corresponding to the patches and occasion times 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 occasion time combinations corresponding to these matrices.

Usage

```
flefko2(
  year = "all",
  patch = "all",
  stageframe,
  supplement = NULL,
  repmatrix = NULL,
  overwrite = NULL,
  data = NA,
  modelsuite = NA,
  surv_model = NA,
  obs_model = NA,
  size_model = NA,
  sizeb_model = NA,
  sizec_model = NA,
  repst_model = NA,
  fec_model = NA,
  jsurv_model = NA,
  jobs_model = NA,
  jsize_model = NA,
  jsizeb_model = NA,
  jsizec_model = NA,
  jrepst_model = NA,
  paramnames = NA,
  inda = NULL,
  indb = NULL,
  indc = NULL,
  surv_dev = 0,
  obs_dev = 0,
  size_dev = 0,
  sizeb_dev = 0,
  sizec_dev = 0,
  repst_dev = 0,
  fec_dev = 0,
  jsurv_dev = 0,
  jobs_dev = 0,
  jsize_dev = 0,
  jsizeb_dev = 0,
  jsizec_dev = 0,
  jrepst_dev = 0,
  density = NA,
  repmod = 1,
  yearcol = NA,
  patchcol = NA,
  year.as.random = FALSE,
  patch.as.random = FALSE,
  random.inda = FALSE,
  random.indb = FALSE,
  random.indc = FALSE,
  randomseed = NA,
  negfec = FALSE,
  reduce = FALSE,
  err_check = FALSE,
  exp_tol = 700,
  theta_tol = 1e+08
)
```

Arguments

`year` A variable corresponding to observation occasion, or a set of such values, given in values associated with the year term used in linear model development. Defaults to `"all"`, in which case matrices will be estimated for all occasion times.

`patch` A variable designating which patches or subpopulations will have matrices estimated. Defaults to `"all"`, but can also be set to specific patch names.

`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 survival and 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` An optional reproduction matrix. This matrix is composed mostly of 0s, with non-zero entries acting as element identifiers and multipliers for fecundity (with 1 equaling full fecundity). If left blank, and no `supplement` is provided, then `flefko2()` will assume that all stages marked as reproductive produce offspring at 1x that of estimated fecundity, and that offspring production will yield 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 historical vertical demographic data frame used to estimate vital rates (class `hfvdata`). The original data frame is required in order to initialize times and patches properly.

`modelsuite` An optional `lefkoMod` object holding the vital rate models. If given, then `surv_model`, `obs_model`, `size_model`, `sizeb_model`, `sizec_model`, `repst_model`, `fec_model`, `jsurv_model`, `jobs_model`, `jsize_model`, `jsizeb_model`, `jsizec_model`, `jrepst_model`, `paramnames`, `yearcol`, and `patchcol` are not required. No models should include size or reproductive status in occasion *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 occasion *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 occasion *t*.

`size_model` A linear model predicting primary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any primary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`sizeb_model` A linear model predicting secondary size. This can be a modelvof class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any secondary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`sizec_model` A linear model predicting tertiary size. This can be a modelvof class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any tertiary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *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 occasion *t*.

`fec_model` A linear model predicting fecundity. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. 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 occasion *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 occasion *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 occasion *t*.

`jsize_model` A linear model predicting juvenile primary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile primary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`jsizeb_model` A linear model predicting juvenile secondary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile secondary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *t*.

`jsizec_model` A linear model predicting juvenile tertiary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile tertiary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing only the impacts of occasion *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 occasion *t*.

`paramnames` A dataframe with three columns, the first describing all terms used in linear modeling, the second (must be called `mainparams`), showing the general model terms that will be used in matrix creation (users should use `modelsearch()` at least once to see the proper names to be used in this column), and the third showing the equivalent terms used in modeling (must be named `modelparams`). Only required if `modelsuite` is not supplied.

`inda` Can be a single value to use for individual covariate `a` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`indb` Can be a single value to use for individual covariate `b` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`indc` Can be a single value to use for individual covariate `c` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`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 primary size.

`sizeb_dev` A numeric value to be added to the y-intercept in the linear model for secondary size.

`sizec_dev` A numeric value to be added to the y-intercept in the linear model for tertiary 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 primary size.

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

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

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

`density` A numeric value indicating density value to use to propagate matrices. Only needed if density is an explanatory term used in linear models. Defaults to `NA`.

`repmod` A scalar multiplier of fecundity. Defaults to `1`.

`yearcol` The variable name or column number corresponding to year in occasion *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 occasions within vital rate models should be estimated as random intercepts. Defaults to `FALSE`, in which case missing monitoring occasion 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`.

`random.inda` A logical value denoting whether to treat individual covariate `a` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`random.indb` A logical value denoting whether to treat individual covariate `b` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`random.indc` A logical value denoting whether to treat individual covariate `c` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`randomseed` A numeric value used as a seed to generate random estimates for missing occasion 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 append matrices of vital rate probabilities associated with each matrix to the `lefkoMat` object generated. These matrices are developed internally and can be used for error checking. Defaults to `FALSE`.

`exp_tol` A numeric value used to indicate a maximum value to set exponents to in the core kernel to prevent numerical overflow. Defaults to `700`.

`theta_tol` A numeric value used to indicate a maximum value to theta as used in the negative binomial probability density kernel. Defaults to `100000000`, but can be reset to other values during error checking.

Value

If all inputs are properly formatted, then this function will return an object of class `lefkoMat`, which is a list that holds the matrix projection model and all of its metadata. Its structure is a list with the following elements:

`A` A list of full projection matrices in order of sorted patches and occasion times. All matrices output in R's `matrix` class.

`U` A list of survival transition matrices sorted as in `A`. All matrices output in R's `matrix` class.

`F` A list of fecundity matrices sorted as in `A`. All matrices output in R's `matrix` class.

`hstages` A data frame matrix showing the pairing of ahistorical stages used to create historical stage pairs. Set to `NA` for ahistorical matrices.

`agestages` A data frame showing age-stage pairs. In this function, it is 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. In `flefko2()`, 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 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 6 columns in the order of survival, observation probability, reproduction probability, primary size transition probability, secondary size transition probability, and tertiary size transition probability.

Notes

Unlike `rlefko2()` and `rlefko3()`, this function does not currently distinguish populations.

This function will yield incorrect estimates if the models utilized incorporate state in occasion *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, but without discriminating between those 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 set to NA, which is the default.

Input options including multiple variable names must be entered in the order of variables in occasion *t*+1 and *t*. Rearranging the order will lead to erroneous calculations, and may lead to fatal errors.

Care should be taken to match the random status of year and patch to the states of those variables within the modelsuite. If they do not match, then they will be treated as zeroes in vital rate estimation.

Using the err_check option will produce a matrix of 6 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, 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)

lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
  immstatus = 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)

# Here we use supplemental to provide overwrite and reproductive info
lathsupp2 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl"),
  stage2 = c("Sd", "Sd", "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)

#Cypripedium example using three size metrics for classification
rm(list=ls(all=TRUE))

data(cypdata)
sizevector.f <- c(0, 0, 0, 0, 0, 0, seq(1, 12, by = 1), seq(0, 9, by = 1),
  seq(0, 8, by = 1), seq(0, 7, by = 1), seq(0, 6, by = 1), seq(0, 5, by = 1),
  seq(0, 4, by = 1), seq(0, 3, by = 1), 0, 1, 2, 0, 1, 0,
  0, 0, 1, 0)
sizebvector.f <- c(0, 0, 0, 0, 0, 0, rep(0, 12), rep(1, 10), rep(2, 9),
  rep(3, 8), rep(4, 7), rep(5, 6), rep(6, 5), rep(7, 4), rep(8, 3), 9, 9, 10,
  0, 1, 1, 2)
sizecvector.f <- c(0, 0, 0, 0, 0, 0, rep(0, 12), rep(0, 10), rep(0, 9),
  rep(0, 8), rep(0, 7), rep(0, 6), rep(0, 5), rep(0, 4), 0, 0, 0, 0, 0, 0,
  1, 1, 1, 1)
stagevector.f <- c("DS", "P1", "P2", "P3", "Sdl", "Dorm", "V1 I0 D0",
  "V2 I0 D0", "V3 I0 D0", "V4 I0 D0", "V5 I0 D0", "V6 I0 D0", "V7 I0 D0",
  "V8 I0 D0", "V9 I0 D0", "V10 I0 D0", "V11 I0 D0", "V12 I0 D0", "V0 I1 D0",
  "V1 I1 D0", "V2 I1 D0", "V3 I1 D0", "V4 I1 D0", "V5 I1 D0", "V6 I1 D0",
  "V7 I1 D0", "V8 I1 D0", "V9 I1 D0", "V0 I2 D0", "V1 I2 D0", "V2 I2 D0",
  "V3 I2 D0", "V4 I2 D0", "V5 I2 D0", "V6 I2 D0", "V7 I2 D0", "V8 I2 D0",
  "V0 I3 D0", "V1 I3 D0", "V2 I3 D0", "V3 I3 D0", "V4 I3 D0", "V5 I3 D0",
  "V6 I3 D0", "V7 I3 D0", "V0 I4 D0", "V1 I4 D0", "V2 I4 D0", "V3 I4 D0",
  "V4 I4 D0", "V5 I4 D0", "V6 I4 D0", "V0 I5 D0", "V1 I5 D0", "V2 I5 D0",
  "V3 I5 D0", "V4 I5 D0", "V5 I5 D0", "V0 I6 D0", "V1 I6 D0", "V2 I6 D0",
  "V3 I6 D0", "V4 I6 D0", "V0 I7 D0", "V1 I7 D0", "V2 I7 D0", "V3 I7 D0",
  "V0 I8 D0", "V1 I8 D0", "V2 I8 D0", "V0 I9 D0", "V1 I9 D0", "V0 I10 D0",
  "V0 I0 D1", "V0 I1 D1", "V1 I1 D1", "V0 I2 D1")
repvector.f <- c(0, 0, 0, 0, 0, rep(0, 13), rep(1, 59))
obsvector.f <- c(0, 0, 0, 0, 0, 0, rep(1, 71))
matvector.f <- c(0, 0, 0, 0, 0, rep(1, 72))
immvector.f <- c(0, 1, 1, 1, 1, rep(0, 72))
propvector.f <- c(1, rep(0, 76))
indataset.f <- c(0, 0, 0, 0, 0, rep(1, 72))
binvec.f <- c(0, 0, 0, 0, 0, rep(0.5, 72))
binbvec.f <- c(0, 0, 0, 0, 0, rep(0.5, 72))
bincvec.f <- c(0, 0, 0, 0, 0, rep(0.5, 72))

vertframe.f <- sf_create(sizes = sizevector.f, sizesb = sizebvector.f,
  sizesc = sizecvector.f, stagenames = stagevector.f, repstatus = repvector.f,
  obsstatus = obsvector.f, propstatus = propvector.f, immstatus = immvector.f,
  matstatus = matvector.f, indataset = indataset.f, binhalfwidth = binvec.f,
  binhalfwidthb = binbvec.f, binhalfwidthc = bincvec.f)

vert.data.f <- verticalize3(cypdata, noyears = 6, firstyear = 2004,
  individcol = "plantid", blocksize = 4, sizeacol = "Veg.04",
  sizebcol = "Inf.04", sizeccol = "Inf2.04", repstracol = "Inf.04",
  repstrbcol = "Inf2.04", fecacol = "Pod.04", censorcol = "censor",
  censorkeep = 1, censorRepeat = FALSE, stageassign = vertframe.f,
  stagesize = "sizeabc", NAas0 = TRUE, censor = FALSE)

vertmodels2f <- modelsearch(vert.data.f, historical = FALSE, suite = "main",
  sizeb = c("sizeb3", "sizeb2", "sizeb1"), sizec = c("sizec3", "sizec2", "sizec1"),
  approach = "glm", vitalrates = c("surv", "obs", "size", "repst", "fec"),
  sizedist = "negbin", sizebdist = "poisson", sizecdist = "poisson",
  fecdist = "poisson", patch.as.random = TRUE, year.as.random = TRUE)

vertsupp2f <- supplemental(stage3 = c("DS", "P1", "P2", "P3", "Sdl", "Sdl",
    "Dorm", "V1 I0 D0", "V2 I0 D0", "V3 I0 D0", "DS", "P1"),
  stage2 = c("DS", "DS", "P1", "P2", "P3", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl",
    "rep", "rep"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "Dorm", "V1 I0 D0", "V2 I0 D0",
    "V3 I0 D0", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "V1 I0 D0", "V1 I0 D0", "V1 I0 D0",
    "V1 I0 D0", 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, NA, 0.5 * 5000, 0.5 * 5000),
  type =c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3), stageframe = vertframe.f,
  historical = FALSE)

vert.mats.f2 <- flefko2(stageframe = vertframe.f, supplement = vertsupp2f,
  data = vert.data.f, modelsuite = vertmodels2f)
```

```
summary(vert.mats.f2)
```

flefko3 *Create Function-based Historical Matrix Projection Model*

Description

Function `flefko3()` returns function-based historical MPMs corresponding to the patches and occasion times 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 occasion time combinations corresponding to these matrices.

Usage

```
flefko3(
  year = "all",
  patch = "all",
  stageframe,
  supplement = NULL,
  repmatrix = NULL,
  overwrite = NULL,
  data = NA,
  modelsuite = NA,
  surv_model = NA,
  obs_model = NA,
  size_model = NA,
  sizeb_model = NA,
  sizec_model = NA,
  repst_model = NA,
  fec_model = NA,
  jsurv_model = NA,
  jobs_model = NA,
  jsize_model = NA,
  jsizeb_model = NA,
  jsizec_model = NA,
  jrepst_model = NA,
  paramnames = NA,
  inda = NULL,
  indb = NULL,
  indc = NULL,
  surv_dev = 0,
  obs_dev = 0,
  size_dev = 0,
  sizeb_dev = 0,
  sizec_dev = 0,
  repst_dev = 0,
  fec_dev = 0,
  jsurv_dev = 0,
  jobs_dev = 0,
  jsize_dev = 0,
  jsizeb_dev = 0,
  jsizec_dev = 0,
  jrepst_dev = 0,
  density = NA,
  repmod = 1,
  yearcol = NA,
  patchcol = NA,
  year.as.random = FALSE,
  patch.as.random = FALSE,
  random.inda = FALSE,
  random.indb = FALSE,
  random.indc = FALSE,
  randomseed = NA,
  negfec = FALSE,
  format = "ehrlen",
  reduce = FALSE,
  err_check = FALSE,
  exp_tol = 700,
  theta_tol = 1e+08
)
```

Arguments

`year` A variable corresponding to the observation occasion, or a set of such values, given in values associated with the year term used in linear model development. Defaults to `"all"`, in which case matrices will be estimated for all occasion times.

`patch` A variable designating which patches or subpopulations will have matrices estimated. Defaults to `"all"`, but can also be set to specific patch names.

`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 survival and 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` An optional reproduction matrix. This matrix is composed mostly of 0s, with non-zero entries acting as element identifiers and multipliers for fecundity (with

1 equaling full fecundity). If left blank, and no `supplement` is provided, then `flefko3()` will assume that all stages marked as reproductive produce offspring at 1x that of estimated fecundity, and that offspring production will yield 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 occasion *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 times and patches properly.

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

`size_model` A linear model predicting primary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any primary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing the impacts of occasions *t* and *t*-1.

`sizeb_model` A linear model predicting secondary size. This can be a modelvof class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any secondary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing the impacts of occasions *t* and *t*-1.

`sizec_model` A linear model predicting tertiary size. This can be a modelvof class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any tertiary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing the impacts of occasions *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. 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 occasions *t* and *t*-1.

`fec_model` A linear model predicting fecundity. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. 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 occasions *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 occasions *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 occasions *t* and *t*-1.

`jsize_model` A linear model predicting juvenile primary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile primary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing the impacts of occasions *t* and *t*-1.

`jsizeb_model` A linear model predicting juvenile secondary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile secondary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing the impacts of occasions *t* and *t*-1.

`jsizec_model` A linear model predicting juvenile tertiary size. This can be a model of class `glm`, `glmer`, `glmmTMB`, `zeroinfl`, `vglm`, `lm`, or `lmer`. If given, then will overwrite any juvenile tertiary size model given in `modelsuite`. This model must have been developed in a modeling exercise testing the impacts of occasions *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 occasions *t* and *t*-1.

`paramnames` A dataframe with three columns, the first describing all terms used in linear modeling, the second (must be called `mainparams`), showing the general model terms that will be used in matrix creation (users should use `modelsearch()` at least once to see the proper names to be used in this column), and the third showing the equivalent terms used in modeling (must be named `modelparams`). Only required if `modelsuite` is not supplied.

`inda` Can be a single value to use for individual covariate `a` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`indb` Can be a single value to use for individual covariate `b` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`indc` Can be a single value to use for individual covariate `c` in all matrices, a pair of values to use for times *t* and *t*-1 in historical matrices, or a vector of such values corresponding to each occasion in option `year`. Defaults to `NULL`.

`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 primary size.

`sizeb_dev` A numeric value to be added to the y-intercept in the linear model for secondary size.

`sizec_dev` A numeric value to be added to the y-intercept in the linear model for tertiary 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 primary size.

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

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

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

`density` A numeric value indicating density value to use to propagate matrices. Only needed if density is an explanatory term used in linear models. Defaults to `NA`.

`repmod` A scalar multiplier of fecundity. Defaults to `1`.

`yearcol` The variable name or column number corresponding to year in occasion *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 occasions within vital rate models should be estimated as random intercepts. Defaults to `FALSE`, in which case missing monitoring occasion 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`.

`random.inda` A logical value denoting whether to treat individual covariate `a` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`random.indb` A logical value denoting whether to treat individual covariate `b` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`random.indc` A logical value denoting whether to treat individual covariate `c` as a random, categorical variable. Otherwise is treated as a fixed, numeric variable. Defaults to `FALSE`.

`randomseed` A numeric value used as a seed to generate random estimates for missing occasion 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 `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 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 to the `lefkoMat` object generated. These matrices are developed internally and can be used for error checking. Defaults to `FALSE`.

`exp_tol` A numeric value used to indicate a maximum value to set exponents to in the core kernel to prevent numerical overflow. Defaults to `700`.

`theta_tol` A numeric value used to indicate a maximum value to theta as used in the negative binomial probability density kernel. Defaults to `100000000`, but can be reset to other values during error checking.

Value

If all inputs are properly formatted, then this function will return an object of class `lefkoMat`, which is a list that holds the matrix projection model and all of its metadata. Its structure is a list with the following elements:

`A` A list of full projection matrices in order of sorted patches and occasion times. All matrices output in R's `matrix` class.

`U` A list of survival transition matrices sorted as in `A`. All matrices output in R's `matrix` class.

`F` A list of fecundity matrices sorted as in `A`. All matrices output in R's `matrix` class.

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

`agestages` A data frame showing age-stage pairs. In this function, it is 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. 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 6 columns in the order of survival, observation probability, reproduction probability, primary size transition probability, secondary size transition probability, and tertiary size transition probability.

Notes

Unlike `rlefko3()`, this function currently does not distinguish populations within the same dataset.

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 be supplied as either historical or ahistorical. If provided as ahistorical, then `flefko3()` will assume that all historical transitions involving stages noted for occasions *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, but without discriminating between those 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 set to `NA`, which is the default.

Input options including multiple variable names must be entered in the order of variables in occasion *t*+1, *t*, and *t*-1. Rearranging the order will lead to erroneous calculations, and will may lead to fatal errors.

Care should be taken to match the random status of year and patch to the states of those variables within the modelsuite. If they do not match, then they will be treated as zeroes in vital rate estimation.

Using the `err_check` option will produce a matrix of 6 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, 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)

lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
  immstatus = 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)

lathmodelsln3 <- modelsearch(lathvertln, historical = TRUE,
  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)

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "mat", "Sd", "Sdl"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "Sdl", "rep", "rep"),
  stage1 = c("Sd", "rep", "Sd", "rep", "Sd", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, "mat", NA, NA),
  eststage2 = c(NA, NA, NA, NA, "Sdl", NA, NA),
  eststage1 = c(NA, NA, NA, NA, "Sdl", NA, NA),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, 0.345, 0.054),
  type = c(1, 1, 1, 1, 1, 3, 3), type_t12 = c(1, 2, 1, 2, 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 = TRUE,
  patch.as.random = TRUE, reduce = FALSE)

summary(lathmat3ln)
```

```
#Cypripedium example using three size metrics for classification
rm(list=ls(all=TRUE))

data(cypdata)
sizevector.f <- c(0, 0, 0, 0, 0, 0, seq(1, 12, by = 1), seq(0, 9, by = 1),
  seq(0, 8, by = 1), seq(0, 7, by = 1), seq(0, 6, by = 1), seq(0, 5, by = 1),
  seq(0, 4, by = 1), seq(0, 3, by = 1), 0, 1, 2, 0, 1, 0,
  0, 0, 1, 0)
sizebvector.f <- c(0, 0, 0, 0, 0, 0, rep(0, 12), rep(1, 10), rep(2, 9),
  rep(3, 8), rep(4, 7), rep(5, 6), rep(6, 5), rep(7, 4), rep(8, 3), 9, 9, 10,
  0, 1, 1, 2)
sizecvector.f <- c(0, 0, 0, 0, 0, 0, rep(0, 12), rep(0, 10), rep(0, 9),
  rep(0, 8), rep(0, 7), rep(0, 6), rep(0, 5), rep(0, 4), 0, 0, 0, 0, 0, 0,
  1, 1, 1, 1)
stagevector.f <- c("DS", "P1", "P2", "P3", "Sdl", "Dorm", "V1 I0 D0",
  "V2 I0 D0", "V3 I0 D0", "V4 I0 D0", "V5 I0 D0", "V6 I0 D0", "V7 I0 D0",
  "V8 I0 D0", "V9 I0 D0", "V10 I0 D0", "V11 I0 D0", "V12 I0 D0", "V0 I1 D0",
  "V1 I1 D0", "V2 I1 D0", "V3 I1 D0", "V4 I1 D0", "V5 I1 D0", "V6 I1 D0",
  "V7 I1 D0", "V8 I1 D0", "V9 I1 D0", "V0 I2 D0", "V1 I2 D0", "V2 I2 D0",
  "V3 I2 D0", "V4 I2 D0", "V5 I2 D0", "V6 I2 D0", "V7 I2 D0", "V8 I2 D0",
  "V0 I3 D0", "V1 I3 D0", "V2 I3 D0", "V3 I3 D0", "V4 I3 D0", "V5 I3 D0",
  "V6 I3 D0", "V7 I3 D0", "V0 I4 D0", "V1 I4 D0", "V2 I4 D0", "V3 I4 D0",
  "V4 I4 D0", "V5 I4 D0", "V6 I4 D0", "V0 I5 D0", "V1 I5 D0", "V2 I5 D0",
  "V3 I5 D0", "V4 I5 D0", "V5 I5 D0", "V0 I6 D0", "V1 I6 D0", "V2 I6 D0",
  "V3 I6 D0", "V4 I6 D0", "V0 I7 D0", "V1 I7 D0", "V2 I7 D0", "V3 I7 D0",
  "V0 I8 D0", "V1 I8 D0", "V2 I8 D0", "V0 I9 D0", "V1 I9 D0", "V0 I10 D0",
  "V0 I0 D1", "V0 I1 D1", "V1 I1 D1", "V0 I2 D1")
repvector.f <- c(0, 0, 0, 0, 0, rep(0, 13), rep(1, 59))
obsvector.f <- c(0, 0, 0, 0, 0, 0, rep(1, 71))
matvector.f <- c(0, 0, 0, 0, 0, rep(1, 72))
immvector.f <- c(0, 1, 1, 1, 1, rep(0, 72))
propvector.f <- c(1, rep(0, 76))
indataset.f <- c(0, 0, 0, 0, 0, rep(1, 72))
binvec.f <- c(0, 0, 0, 0, 0, rep(0.5, 72))
binbvec.f <- c(0, 0, 0, 0, 0, rep(0.5, 72))
bincvec.f <- c(0, 0, 0, 0, 0, rep(0.5, 72))

vertframe.f <- sf_create(sizes = sizevector.f, sizesb = sizebvector.f,
  sizesc = sizecvector.f, stagenames = stagevector.f, repstatus = repvector.f,
  obsstatus = obsvector.f, propstatus = propvector.f, immstatus = immvector.f,
  matstatus = matvector.f, indataset = indataset.f, binhalfwidth = binvec.f,
  binhalfwidthb = binbvec.f, binhalfwidthc = bincvec.f)

vert.data.f <- verticalize3(cypdata, noyears = 6, firstyear = 2004,
  individcol = "plantid", blocksize = 4, sizeacol = "Veg.04",
  sizebcol = "Inf.04", sizeccol = "Inf2.04", repstracol = "Inf.04",
  repstrbcol = "Inf2.04", fecacol = "Pod.04", censorcol = "censor",
  censorkeep = 1, censorRepeat = FALSE, stageassign = vertframe.f,
  stagesize = "sizeabc", NAas0 = TRUE, censor = FALSE)

vertmodels3f <- modelsearch(vert.data.f, historical = TRUE, suite = "main",
  sizeb = c("sizeb3", "sizeb2", "sizeb1"), sizec = c("sizec3", "sizec2", "sizec1"),
  approach = "glm", vitalrates = c("surv", "obs", "size", "repst", "fec"),
  sizedist = "negbin", sizebdist = "poisson", sizecdist = "poisson",
  fecdist = "poisson", patch.as.random = TRUE, year.as.random = TRUE)

vertsupp3f <- supplemental(stage3 = c("DS", "P1", "DS", "P1", "P2", "P2", "P3",
    "Sdl", "Sdl", "Sdl", "Dorm", "V1 I0 D0", "V2 I0 D0", "V3 I0 D0", "Dorm",
    "V1 I0 D0", "V2 I0 D0", "V3 I0 D0", "DS", "P1"),
  stage2 = c("DS", "DS", "DS", "DS", "P1", "P1", "P2", "P3", "Sdl", "Sdl", "Sdl",
    "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "Sdl", "rep", "rep"),
  stage1 = c("DS", "DS", "rep", "rep", "DS", "rep", "P1", "P2", "P3", "Sdl",
    "Sdl", "Sdl", "Sdl", "Sdl", "P3", "P3", "P3", "P3", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "Dorm", "V1 I0 D0",
    "V2 I0 D0", "V3 I0 D0", "Dorm", "V1 I0 D0", "V2 I0 D0", "V3 I0 D0", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "V1 I0 D0", "V1 I0 D0",
    "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "V1 I0 D0", "V1 I0 D0",
    "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", "V1 I0 D0", NA, NA),
  givenrate = c(0.10, 0.20, 0.10, 0.20, 0.20, 0.20, 0.20, 0.25, 0.40, 0.40, 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, 0.5 * 5000, 0.5 * 5000),
  type =c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
  stageframe = vertframe.f, historical = TRUE)

vert.mats.f3 <- flefko3(stageframe = vertframe.f, supplement = vertsupp3f,
  data = vert.data.f, modelsuite = vertmodels3f)
summary(vert.mats.f3)
```

historicalize3 *Create Historical Vertical Data Frame from Ahistorical Vertical Data Frame*

Description

Function 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 hfvdata format and can be used in all functions in the package.

Usage

```
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,
  obs2col = 0,
  obs3col = 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 or subpopulation 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 occasion *t* (year or time).

`year3col` A variable name or column number corresponding to occasion *t*+1 (year or time).

`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 occasion *t*.

`sizea3col` A variable name or column number corresponding to the primary size entry in occasion *t*+1.

`sizeb2col` A variable name or column number corresponding to the secondary size entry in occasion *t*.

`sizeb3col` A variable name or column number corresponding to the secondary size entry in occasion *t*+1.

`sizec2col` A variable name or column number corresponding to the tertiary size entry in occasion *t*.

`sizec3col` A variable name or column number corresponding to the tertiary size entry in occasion *t*+1.

`repstra2col` A variable name or column number corresponding to the production of reproductive structures, such as flowers, in occasion *t*. This can be binomial or count data, and is used to in analysis of the probability of reproduction.

`repstra3col` A variable name or column number corresponding to the production of reproductive structures, such as flowers, in occasion *t*+1. This can be binomial or count data, and is used to in analysis of the probability of reproduction.

`repstrb2col` A second variable name or column number corresponding to the production of reproductive structures, such as flowers, in occasion *t*. This can be binomial or count data.

`repstrb3col` A second variable name or column number corresponding to the production of reproductive structures, such as flowers, in occasion *t*+1. This can be binomial or count data.

`feca2col` A variable name or column number corresponding to fecundity in occasion *t*. This may represent egg counts, fruit counts, seed production, etc.

`feca3col` A variable name or column number corresponding to fecundity in occasion *t*+1. This may represent egg counts, fruit counts, seed production, etc.

`fecb2col` A second variable name or column number corresponding to fecundity in occasion *t*. This may represent egg counts, fruit counts, seed production, etc.

`fecb3col` A second variable name or column number corresponding to fecundity in occasion *t*+1. This may represent egg counts, fruit counts, seed production, etc.

`indcova2col` A variable name or column number corresponding to an individual covariate to be used in analysis, in occasion *t*.

`indcova3col` A variable name or column number corresponding to an individual covariate to be used in analysis, in occasion *t*+1.

`indcovb2col` A variable name or column number corresponding to a second individual covariate to be used in analysis, in occasion *t*.

`indcovb3col` A variable name or column number corresponding to a second individual covariate to be used in analysis, in occasion *t*+1.

`indcovc2col` A variable name or column number corresponding to a third individual covariate to be used in analysis, in occasion *t*.

`indcovc3col` A variable name or column number corresponding to a third individual covariate to be used in analysis, in occasion *t*+1.

`alive2col` A variable name or column number that provides information on whether an individual is alive in occasion *t*. If used, living status must be designated as binomial (living = `1`, dead = `0`).

`alive3col` A variable name or column number that provides information on whether an individual is alive in occasion *t*+1. If used, living status must be designated as binomial (living = `1`, dead = `0`).

`dead2col` A variable name or column number that provides information on whether an individual is dead in occasion *t*. If used, dead status must be designated as binomial (living = `0`, dead = `1`).

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

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

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

`repstrrel` This is a scalar multiplier making 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 making 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 occasion *t*.

`stage3col` Optional variable name or column number corresponding to life history stage in occasion *t*+1.

`juv2col` A variable name or column number that marks individuals in immature stages in occasion *t*. Function `historicalize3()` 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 occasion *t*+1. Function `historicalize3()` 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`, `sizeab`, `sizebc`, `sizeac`, `sizeabc`, or `sizeadded`, depending on which size variable within the input dataset is chosen. Note that the variable(s) chosen should be presented in the order of the primary, secondary, and tertiary variables in the stageframe input with `stageassign`. For example, choosing `sizeb` assumes that this size variable in the dataset is the primary variable in the stageframe.

`censor` 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 valid 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 occasions *t*-1 and *t*+1 gives preference to condition in occasion *t* within the input dataset. Conflicts in condition in input datasets that have both occasions *t* and *t*+1 listed per row are resolved by using condition in occasion *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 in occasion *t*.

`firstseen` Occasion of first observation.

`lastseen` Occasion of last observation.

`obsage` Observed age in occasion *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 occasions *t*-1, *t*, and *t*+1, respectively, if provided.

`ypos1,ypos2,ypos3`
Y position in Cartesian space in occasions *t*-1, *t*, and *t*+1, respectively, if provided.

`sizea1,sizea2,sizea3`
Main size measurement in occasions *t*-1, *t*, and *t*+1, respectively.

`sizeb1,sizeb2,sizeb3`
Secondary size measurement in occasions *t*-1, *t*, and *t*+1, respectively.

`sizec1,sizec2,sizec3`
Tertiary size measurement in occasions *t*-1, *t*, and *t*+1, respectively.

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

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

`repstrb1,repstrb2,repstrb3`
: Secondary numbers of reproductive structures in occasions *t*-1, *t*, and *t*+1, respectively.

`repstr1added,repstr2added,repstr3added`
: Sum of primary and secondary reproductive structures in occasions *t*-1, *t*, and *t*+1, respectively.

`feca1,feca2,feca3`
: Main numbers of offspring in occasions *t*-1, *t*, and *t*+1, respectively.

`fecb1,fecb2, fecb3`
: Secondary numbers of offspring in occasions *t*-1, *t*, and *t*+1, respectively.

`fec1added,fec2added,fec3added`
: Sum of primary and secondary fecundity in occasions *t*-1, *t*, and *t*+1, respectively.

`censor1,censor2,censor3`
: Censor status values in occasions *t*-1, *t*, and *t*+1, respectively.

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

`obsstatus1,obsstatus2,obsstatus3`
: Binomial observation status in occasions *t*-1, *t*, and *t*+1, respectively.

`repstatus1,repstatus2,repstatus3`
: Binomial reproductive status in occasions *t*-1, *t*, and *t*+1, respectively.

`fecstatus1,fecstatus2,fecstatus3`
: Binomial offspring production status in occasions *t*-1, *t*, and *t*+1, respectively.

`matstatus1,matstatus2,matstatus3`
: Binomial maturity status in occasions *t*-1, *t*, and *t*+1, respectively.

`alive1,alive2,alive3`
: Binomial status as alive in occasions *t*-1, *t*, and *t*+1, respectively.

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

Notes

Warnings that some individuals occur in state combinations that do not match any stages in the stageframe used to assign stages, and that some individuals match characteristics of several stages in the stageframe, 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 some cases, these warnings will arise with no concurrent `NoMatch` entries. These are important warnings and suggest that there is likely a problem with the stageframe. The most common such problems are: 1) stages have significant overlap in characteristics, with the most common being overlapping size bins caused by erroneous definitions of size bin halfwidths; and 2) some individuals exist in states not defined within the stageframe.

In some datasets 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. Alternatively, observation status may be input via `obs2col` and `obs3col` to force computation with given values (although this requires all instances of observation and non-observation to be known and coded ahead of time). Further analysis can then utilize only a single size variable, of the user's choosing. Similar issues can arise in reproductive status (`repstatus`).

Juvenile designation should only be used when juveniles fall outside of the size classification scheme used in determining stages. If juveniles are to be size classified along the size spectrum that adults also fall on, then it is best to treat juveniles as mature but not reproductive.

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.

Unusual errors (e.g. `"Error in pjf..."`) may occur in cases where the variables are improperly passed, or where seemingly numeric variables include text and so get automatically converted to string variables.

Density estimation is performed as a count of individuals alive and within the radius specified in `spacing` of the respective individual at some point in time.

Examples

```
data(cypvert)

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)

cypframe_raw

cypraw_v2 <- 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", repstrrel = 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", "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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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_v2, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2"),
  size = c("size3added", "size2added"), supplement = cypsupp2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "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")
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", "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", indivcol = "individ")

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

# 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, 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

image3(cypmatrix2r, used = 1, type = "U")
```

image3.lefkoElas *Create Matrix Image(s) for lefkoElas Object*

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

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

Arguments

`mats` A `lefkoElas` 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

```
# 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)
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", "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")

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, 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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

mats	A lefkoMat 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, U, or F matrices. Defaults to "A".
...	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")
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", "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", indivcol = "individ")

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

# 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

image3(cypmatrix2r, 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

```
## 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"`istorical 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")
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", "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")

ehrlen_sens <- sensitivity3(ehrlen3)

image3(ehrlen_sens, 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, 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

cypsens <- sensitivity3(cypmatrix2r)
image3(cypsens, used = 1, type = "a")
```

image3.list *Create Matrix Images for Matrices in a List*

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

```
## 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

```
# 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)
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"),
  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, 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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

```
## 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

```
# 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)
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", "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")

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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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

```
# 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)
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", "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)
lambda3(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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

lambda3(cypmatrix2r)
```

lambda3.lefkoMat *Estimate Deterministic Population Growth Rates of lefkoMat Matrices*

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 ahistorical matrices.

Usage

```
## 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

```
# 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)
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", "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)
lambda3(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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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 ahistorical matrices.

Usage

```
## 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

```
# 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)
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", "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)
lambda3(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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

lambda3(cypmatrix2r$A[[1]])
```

`lathyrus` *Demographic Dataset of* Lathyrus vernus *Population*

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 refering 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 NZumber 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<sup>3</sup> in 1990.

lnVol90 Natural logarithm of `Volume90`.

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

Flow90 Number of flowers in 1990.

Intactseed90 NZumber 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<sup>3</sup> in 1991.

lnVol91 Natural logarithm of `Volume91`.

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

Flow91 Number of flowers in 1991.

Intactseed91 NZumber 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

Ehrlen, J. 2000. The dynamics of plant populations: does the history of individuals matter? *Ecology* 81(6):1675-1684.

Examples

```
# 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")
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", "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", indivcol = "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", "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",
  juvcol = c("Seedling1988", "Seedling1989", "Seedling1990", "Seedling1991"),
  sizeacol = c("Volume88", "Volume89", "Volume90", "Volume91"),
  repstracol = c("FCODE88", "FCODE89", "FCODE90", "FCODE91"),
  fecacol = c("Intactseed88", "Intactseed89", "Intactseed90", "Intactseed91"),
  deadacol = c("Dead1988", "Dead1989", "Dead1990", "Dead1991"),
  nonobsacol = c("Dormant1988", "Dormant1989", "Dormant1990", "Dormant1991"),
  censorcol = c("Missing1988", "Missing1989", "Missing1990", "Missing1991"),
  stageassign = lathframe, stagesize = "sizea",
  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, 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]]

lambda3(ehrlen3mean)
```

lefko3 *Historical and Ahistorical Population Projection Matrix Analysis*

Description

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 monitoring occasions (2 time steps or periods) of demographic information. The package constructs both function-based and raw MPMs for both standard ahistorical (i.e. 2 occasions, 1 period) and historical analyses, and can also produce age-by-stage MPMs and IPMs. It also includes powerful functions to standardize demographic datasets.

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 Ehrlén

References

Shefferson, R.P., J. Ehrlen, and S. Kurokawa. 2021. *lefko3*: analyzing individual history through size-classified matrix population models. *Methods in Ecology and Evolution* 12(2): 378-382.

`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

```
# 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)
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", "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)
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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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
```

ltre3 *Conduct a Life Table Response Experiment*

Description

`ltre3()` is a generic function that returns life table response experiment (LTRE) or stochastic LTRE matrices for the input projection matrices.

Usage

```
ltre3(mats, refmats, ...)
```

Arguments

`mats`	A lefkoMat object, population projection matrix, or list of population projection matrices.
`refmats`	A reference lefkoMat object, or matrix, for use as the control. If missing, then is set to the same object as `mats`.
`...`	Other parameters.

Value

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

Notes

Deterministic LTRE is one-way, fixed, and based on the sensitivities of the matrix midway between each input matrix and the reference matrix, per Caswell (2001, Matrix Population Models, Sinauer Associates, MA, USA). Stochastic LTRE is per Davison et al. (2010, doi: 10.1111/j.1365-2745.2009.01611.x).

See Also

`ltre3.lefkoMat()`

`summary.lefkoLTRE()`

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, 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", "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")

ltre3(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)
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

ltre3(cypmatrix2r)
```

ltre3.lefkoMat — *Conduct a Life Table Response Experiment of a lefkoMat Object*

Description

`ltre3.lefkoMat()` returns a set of matrices of one-way LTRE (life table response experiment) or stochastic LTRE matrices contributions.

Usage

```
## S3 method for class 'lefkoMat'
ltre3(
  mats,
  refmats = NA,
  ref = NA,
  stochastic = FALSE,
  steps = 10000,
  burnin = 3000,
  time_weights = NA,
  sparse = "auto",
  rseed = NA,
  append_mats = FALSE,
  ...
)
```

Arguments

`mats` An object of class `lefkoMat`.

`refmats` A reference lefkoMat object, or matrix, for use as the control. If missing, then is set to the same object as `mats`.

`ref` A numeric value indicating which matrix or matrices in `refmats` to use as the control. The numbers used must correspond to the number of the matrices in the `labels` element of the associated `lefkoMat` object. The default setting, NA, uses all entries in `refmats`.

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

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

`burnin` The number of initial steps to ignore in stochastic projection when calculating stochastic elasticities. Must be smaller than `steps`. Defaults to 3000.

`time_weights` Numeric vector denoting the probabilistic weightings of all matrices. Defaults to equal weighting among matrices.

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

`rseed` Optional numeric value corresponding to the random seed for stochastic simulation.

`append_mats` A logical value denoting whether to include the original A, U, and F matrices in the returned `lefkoLTRE` object. Defaults to FALSE.

`...` Other parameters.

Value

This function returns an object of class `lefkoLTRE`. This includes a list of LTRE matrices as object `ltre_det` if a deterministic LTRE is called for, or a list of mean-value LTRE matrices as object `ltre_mean` and a list of SD-value LTRE matrices as object `ltre_sd` if a stochastic LTRE is called for. This is followed by the stageframe as object `ahstages`, the order of historical stages as object `hstages`, the age-by-stage order as object `agestages`, the order of matrices as object `labels`, and, if requested, the original A, U, and F matrices.

Notes

Deterministic LTRE is one-way, fixed, and based on the sensitivities of the matrix midway between each input matrix and the reference matrix, per Caswell (2001, Matrix Population Models, Sinauer Associates, MA, USA). Stochastic LTRE is simulated per Davison et al. (2010) Journal of Ecology 98:255-267 (doi: 10.1111/j.1365-2745.2009.01611.x).

Default behavior for stochastic LTRE uses the full population provided in `mats` as the reference if no `refmats` and `ref` is provided. If no `refmats` is provided but `ref` is, then the matrices noted in `ref` are used as the reference matrix set. Year and patch order is utilized from object `mats`, but not from object `refmats`, in which each matrix is assumed to represent a different year from one population. This function cannot currently handle multiple populations within the same `mats` object (although such analysis is possible if these populations are designated as patches instead).

See Also

ltre3()

summary.lefkoLTRE()

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, 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", "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")

ltre3(ehrlen3, 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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

ltre3(cypmatrix2r)
```

modelsearch *Develop Best-fit Vital Rate Estimation Models for MPM Development*

Description

Function modelsearch() runs exhaustive model building and selection for each vital rate needed to estimate a function-based MPM or IPM. It returns best-fit models for each vital rate, model table showing all models tested, and model quality control data. The final output can be used as input in other functions within this package.

Usage

```
modelsearch(
  data,
  stageframe = NULL,
  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", "sizea1"),
  sizeb = c(NA, NA, NA),
  sizec = c(NA, NA, NA),
  repst = c("repstatus3", "repstatus2", "repstatus1"),
  fec = c("feca3", "feca2", "feca1"),
  stage = c("stage3", "stage2", "stage1"),
  indiv = "individ",
  patch = NA,
  year = "year2",
  density = NA,
  sizedist = "gaussian",
  sizebdist = NA,
  sizecdist = NA,
  fecdist = "gaussian",
  size.zero = FALSE,
  sizeb.zero = FALSE,
  sizec.zero = FALSE,
  size.trunc = FALSE,
  sizeb.trunc = FALSE,
  sizec.trunc = FALSE,
  fec.zero = FALSE,
  fec.trunc = FALSE,
  patch.as.random = TRUE,
  year.as.random = TRUE,
  juvestimate = NA,
  juvsize = FALSE,
  jsize.zero = FALSE,
  jsizeb.zero = FALSE,
  jsizec.zero = FALSE,
  jsize.trunc = FALSE,
  jsizeb.trunc = FALSE,
  jsizec.trunc = FALSE,
  fectime = 2,
  censor = NA,
  age = NA,
  indcova = NA,
  indcovb = NA,
  indcovc = NA,
  random.indcova = FALSE,
  random.indcovb = FALSE,
  random.indcovc = FALSE,
  test.group = FALSE,
  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.

`stageframe` The stageframe characterizing the life history model used. Optional unless `test.group = TRUE`, in which case it is required. Defaults to `NULL`.

`historical` A logical variable denoting whether to assess the effects of state in occasion *t*-1, in addition to state in occasion *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 generalized linear modeling assuming that all factors are fixed.

`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); and `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 $\Delta AICc <= 2.0$. 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 occasions *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 occasions *t*+1, *t*, and *t*-1, respectively. Defaults to `c("obsstatus3","obsstatus2","obsstatus1")`.

size A vector indicating the variable names coding for the primary size variable on occasions *t*+1, *t*, and *t*-1, respectively. Defaults to c("sizea3","sizea2","sizea1").

sizeb A vector indicating the variable names coding for the secondary size variable on occasions *t*+1, *t*, and *t*-1, respectively. Defaults to c(NA,NA,NA), in which case sizeb is not used.

sizec A vector indicating the variable names coding for the tertiary size variable on occasions *t*+1, *t*, and *t*-1, respectively. Defaults to c(NA,NA,NA), in which case sizec is not used.

repst A vector indicating the variable names coding for reproductive status in occasions *t*+1, *t*, and *t*-1, respectively. Defaults to c("repstatus3","repstatus2","repstatus1").

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

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

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

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

year A text value indicating the variable coding for observation occasion *t*. Defaults to year2.

density A text value indicating the name of the variable coding for spatial density, should the user wish to test spatial density as a fixed factor affecting vital rates. Defaults to NA.

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

sizebdist The probability distribution used to model secondary size. Options include "gaussian" for the Normal distribution, "poisson" for the Poisson distribution, "negbin" for the negative binomial distribution (quadratic parameterization), and "gamma" for the Gamma distribution. Defaults to NA.

sizecdist The probability distribution used to model tertiary size. Options include "gaussian" for the Normal distribution, "poisson" for the Poisson distribution, "negbin" for the negative binomial distribution (quadratic parameterization), and "gamma" for the Gamma distribution. Defaults to NA.

fecdist The probability distribution used to model fecundity. Options include "gaussian" for the Normal distribution (default), "poisson" for the Poisson distribution, "negbin" for the negative binomial distribution (quadratic parameterization), and "gamma" for the Gamma distribution.

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

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

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

`size.trunc` A logical variable indicating whether the primary size distribution should be zero-truncated. Only applies to Poisson and negative binomial distributions. Defaults to `FALSE`. Cannot be `TRUE` if `size.zero = TRUE`.

`sizeb.trunc` A logical variable indicating whether the secondary size distribution should be zero-truncated. Only applies to Poisson and negative binomial distributions. Defaults to `FALSE`. Cannot be `TRUE` if `sizeb.zero = TRUE`.

`sizec.trunc` A logical variable indicating whether the tertiary size distribution should be zero-truncated. Only applies to Poisson and negative binomial distributions. Defaults to `FALSE`. Cannot be `TRUE` if `sizec.zero = TRUE`.

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

`fec.trunc` A logical variable indicating whether the fecundity distribution should be zero-truncated. Only applies to the Poisson and negative binomial distributions. Defaults to `FALSE`. Cannot be `TRUE` if `fec.zero = TRUE`.

`patch.as.random`

If set to `TRUE` and `approach = "mixed"`, 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 = "mixed"`, 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`.

`jsize.zero` A logical variable indicating whether the primary size distribution of juveniles should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to `FALSE`.

`jsizeb.zero` A logical variable indicating whether the secondary size distribution of juveniles should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to `FALSE`.

`jsizec.zero` A logical variable indicating whether the tertiary size distribution of juveniles should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to `FALSE`.

`jsize.trunc` A logical variable indicating whether the primary size distribution in juveniles should be zero-truncated. Defaults to `FALSE`. Cannot be `TRUE` if `jsize.zero = TRUE`.

`jsizeb.trunc` A logical variable indicating whether the secondary size distribution in juveniles should be zero-truncated. Defaults to `FALSE`. Cannot be `TRUE` if `jsizeb.zero = TRUE`.

`jsizec.trunc` A logical variable indicating whether the tertiary size distribution in juveniles should be zero-truncated. Defaults to `FALSE`. Cannot be `TRUE` if `jsizec.zero = TRUE`.

`fectime` A variable indicating which year of fecundity to use as the response term in fecundity models. Options include `2`, which refers to occasion *t*, and `3`, which refers to occasion *t*+1. Defaults to `2`.

`censor` A vector denoting the names of censoring variables in the dataset, in order from occasion *t*+1, followed by occasion *t*, and lastly followed by occasion *t*-1. Defaults to `NA`.

`age` Designates the name of the variable corresponding to age in the vertical dataset. Defaults to `NA`, in which case age is not included in linear models. Should only be used if building Leslie or age x stage matrices.

`indcova` Vector designating the names in occasions *t*+1, *t*, and *t*-1 of an individual covariate. Defaults to `NA`.

`indcovb` Vector designating the names in occasions *t*+1, *t*, and *t*-1 of a second individual covariate. Defaults to `NA`.

`indcovc` Vector designating the names in occasions *t*+1, *t*, and *t*-1 of a third individual covariate. Defaults to `NA`.

`random.indcova` A logical value indicating whether `indcova` should be treated as a random categorical factor, rather than as a fixed factor. Defaults to `FALSE`.

`random.indcovb` A logical value indicating whether `indcovb` should be treated as a random categorical factor, rather than as a fixed factor. Defaults to `FALSE`.

`random.indcovc` A logical value indicating whether `indcovc` should be treated as a random categorical factor, rather than as a fixed factor. Defaults to `FALSE`.

`test.group` A logical value indicating whether to include the `group` variable from the input `stageframe` as a fixed categorical variable in linear models. Defaults to `FALSE`.

`show.model.tables`
If set to TRUE, then includes full modeling tables in the output. Defaults to `TRUE`.

`global.only` If set to TRUE, then only global models will be built and evaluated. Defaults to `FALSE`.

`quiet` If set to TRUE, then model building and selection will proceed with most warnings and diagnostic messages silenced. Defaults to `FALSE`.

Value

This function yields an object of class `lefkoMod`, which is a list in which the first 13 elements are the best-fit models for survival, observation status, primary size, secondary size, tertiary size, reproductive status, fecundity, juvenile survival, juvenile observation, juvenile primary size, juvenile secondary size, juvenile tertiary size, and juvenile transition to reproduction, respectively, followed by 13 elements corresponding to the model tables for each of these vital rates, in order, followed by a data frame showing the order and names of variables used in modeling, followed by a single character element denoting the criterion used for model selection, and ending on a data frame with quality control data:

survival_model Best-fit model of the binomial probability of survival from occasion *t* to occasion *t*+1. Defaults to 1.

observation_model
Best-fit model of the binomial probability of observation in occasion *t*+1 given survival to that occasion. Defaults to 1.

size_model Best-fit model of the primary size metric on occasion *t*+1 given survival to and observation in that occasion. Defaults to 1.

sizeb_model Best-fit model of the secondary size metric on occasion *t*+1 given survival to and observation in that occasion. Defaults to 1.

sizec_model Best-fit model of the tertiary size metric on occasion *t*+1 given survival to and observation in that occasion. Defaults to 1.

repstatus_model
Best-fit model of the binomial probability of reproduction in occasion *t*+1, given survival to and observation in that occasion. Defaults to 1.

fecundity_model
Best-fit model of fecundity in occasion *t*+1 given survival to, and observation and reproduction in that occasion. Defaults to 1.

juv_survival_model
Best-fit model of the binomial probability of survival from occasion *t* to occasion *t*+1 of an immature individual. Defaults to 1.

juv_observation_model
Best-fit model of the binomial probability of observation in occasion *t*+1 given survival to that occasion of an immature individual. Defaults to 1.

juv_size_model Best-fit model of the primary size metric on occasion *t*+1 given survival to and observation in that occasion of an immature individual. Defaults to 1.

juv_sizeb_model
Best-fit model of the secondary size metric on occasion *t*+1 given survival to and observation in that occasion of an immature individual. Defaults to 1.

juv_sizec_model
Best-fit model of the tertiary size metric on occasion *t*+1 given survival to and observation in that occasion of an immature individual. Defaults to 1.

juv_reproduction_model
Best-fit model of the binomial probability of reproduction in occasion *t*+1, given survival to and observation in that occasion of an individual that was immature in occasion *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 occasion *t*+1 but immature in occasion *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 the primary size variable.

`sizeb_table` Full dredge model table of the secondary size variable.

`sizec_table` Full dredge model table of the tertiary size variable.

`repstatus_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 primary size in immature individuals.

`juv_sizeb_table` Full dredge model table of secondary size in immature individuals.

`juv_sizec_table` Full dredge model table of tertiary size in immature individuals.

`juv_reproduction_table` Full dredge model table of immature reproduction probability.

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

`qc` Data frame with four variables: 1) Name of vital rate, 2) number of individuals used to model that vital rate, 3) number of individual transitions used to model that vital rate, and 4) accuracy of model expressed as percent of predicted responses equal to actual responses (only in binomial models).

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 `lm()` (GLM with Gaussian response), `glm()` (GLM with Poisson, Gamma, or binomial response), `glm.nb()` (GLM with negative binomial response), `zeroinfl()` (GLM with zero-inflated Poisson or negative binomial response), `vglm()` (GLM with zero-truncated Poisson or negative binomial response), `lmer()` (mixed model with Gaussian response), `glmer()` (mixed model with binomial, Poisson, or Gamma response), and `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 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.

Care must be taken to build models that test the impacts of state in occasion *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.

If density dependence is explored through function modelsearch(), then the interpretation of density is not the full population size but rather the spatial density term included in the dataset.

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, 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)
```

```
lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
  immstatus = 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)

lathmodelsln3 <- modelsearch(lathvertln, historical = TRUE,
  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)

# Here we use supplemental() to provide overwrite and reproductive info
lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sdl", "Sdl", "mat", "Sd", "Sdl"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "Sdl", "rep", "rep"),
  stage1 = c("Sd", "rep", "Sd", "rep", "Sd", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, "mat", NA, NA),
  eststage2 = c(NA, NA, NA, NA, "Sdl", NA, NA),
  eststage1 = c(NA, NA, NA, NA, "Sdl", NA, NA),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, 0.345, 0.054),
  type = c(1, 1, 1, 1, 1, 3, 3), type_t12 = c(1, 2, 1, 2, 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 = TRUE,
  patch.as.random = TRUE, reduce = FALSE)

summary(lathmat3ln)
```

overwrite *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

```
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 occasion *t*+1 in the transition to be replaced. Abbreviations for groups of stages are also allowed (see Notes).

`stage2` The name of the stage in occasion *t* in the transition to be replaced. Abbreviations for groups of stages are also allowed (see Notes).

`stage1` The name of the stage in occasion *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 allowed (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.

`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 occasions *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 occasions *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.

Value

A data frame that puts the above vectors together and can be used as input in flefko3(), flefko2(), rlefko3(),rlefko2(), and aflefko2().

Variables in this data frame include the following:

stage3 Stage at occasion *t*+1 in the transition to be replaced.

stage2 Stage at occasion *t* in the transition to be replaced.

stage1 Stage at occasion *t*-1 in the transition to be replaced.

eststage3 Stage at occasion *t*+1 in the transition to replace the transition designated by stage3, stage2, and stage1.

eststage2 Stage at occasion *t* in the transition to replace the transition designated by stage3, stage2, and stage1.

eststage1 Stage at occasion *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 occasion *t* to occasion *t*+1 is a survival-transition probability (1) or a fecundity rate (2).

convtype_t12 Designates whether the transition from occasion *t*-1 to occasion *t* is a survival transition probability (1), a fecundity rate (2).

Notes

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, immat 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

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

cypover2r

cypover3r <- overwrite(stage3 = c("SD", "SD", "P1", "P1", "P2", "P3", "SL",
    "D", "XSm", "Sm", "D", "XSm", "Sm"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL",
    "SL", "SL", "SL"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "P3", "P3",
    "SL", "SL", "SL"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm",
    "Sm"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm",
    "XSm", "XSm"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm",
    "XSm", "XSm"),
  givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.25, NA, NA, NA, NA, NA, NA),
  type = c("S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S", "S"))

cypover3r
```

projection3 *Conduct Population Projection Simulations*

Description

Function projection3() runs projection simulations. It projects the population an patches forward in time by a user-defined number of occasions. Projections may be deterministic or stochastic, and may be density dependent either way. If deterministic, then projections will be cyclical if matrices exist covering multiple occasions for each population or patch. If stochastic, then annual matrices will be shuffled within patches and populations. Replicates may also be requested.

Usage

```
projection3(
  mpm,
  nreps = 1L,
  times = 10000L,
  stochastic = FALSE,
  standardize = FALSE,
  growthonly = TRUE,
  integeronly = FALSE,
  substoch = 0L,
  start_vec = NULL,
  start_frame = NULL,
  tweights = NULL,
  density = NULL
)
```

Arguments

mpm A matrix projection model of class lefkoMat, or a list of full matrix projection matrices.

nreps The number of replicate projections.

times Number of occasions to iterate per replicate. 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 occasion. Defaults to FALSE.

`growthonly` A logical value indicating whether to produce only the projected population size at each occasion, or a vector showing the stage distribution followed by the reproductive value vector followed by the full population size at each occasion. Defaults to TRUE.

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

`substoch` An integer value indicating whether to force survival- transition matrices to be substochastic in density dependent simulations. Defaults to `0`, which does not force substochasticity. Alternatively, `1` forces all survival-transition elements to range from 0.0 to 1.0, and `2` forces all column rows to total no more than 1.0.

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

`start_frame` An optional data frame characterizing stages, age-stages, or stage-pairs that should be set to non-zero values in the starting vector, and what those values should be. Can only be used with `lefkoMat` objects.

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

`density` An optional data frame describing the matrix elements that will be subject to density dependence, and the exact kind of density dependence that they will be subject to. The data frame used should be an object of class `lefkoDens`, which is the output from function `density_input()`.

Value

A list of class `lefkoProj`, which always includes the first three elements of the following, and also includes the remaining elements below when a `lefkoMat` object is used as input:

`projection` A list of lists of matrices showing the total number of individuals per stage per occasion. The first list corresponds to each pop-patch followed by each population. The inner list corresponds to replicates within each pop-patch or population.

`stage_dist` A list of lists of the actual stage distribution in each occasion in each replicate in each pop-patch or population. The list order is the same as in `projection`.

`rep_value` A list of lists of the actual reproductive value in each occasion in each replicate in each pop-patch or population. The list order is the same as in `projection`.

`pop_size` A list of data frames showing the total population size in each occasion per replicate (row within data frame) per pop-patch or population (list element).

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

`control` A short vector indicating the number of replicates and the number of occasions projected per replicate.

`ahstages` The original stageframe used in the study.

`hstages` A data frame showing the order of historical stage pairs.

`agestages` A data frame showing the order of age-stage pairs.

Notes

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.

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.

Starting vectors can be input in one of two ways: 1) as `start_vec` input, which is a vector of numbers of the numbers of individuals in each stage, stage pair, or age-stage, with the length of the vector necessarily as long as there are rows in the matrices of the MPM; or 2) as `start_frame` input, which is a data frame showing only those stages, stage pairs, or age-stages that should begin with more than 0 individuals, and the numbers of individuals that those stages should start with (this object is created using the `start_input()` function). If both are provided, then `start_frame` takes precedence and `start_vec` is ignored. If neither is provided, then `projection3()` automatically assumes that each stage, stage pair, or age-stage begins with a single individual. Importantly, if a `lefkoMat` object is not used, and a list of matrices is provided instead, then `start_frame` cannot be utilized and a full `start_vec` must be provided to conduct a simulation with starting numbers of individuals other than 1 per stage.

The resulting data frames in element `projection` are separated by pop-patch according to the order provided in element `labels`, but the matrices for each element of `projection` have the result of each replicate stacked in order on top of one another without any break or indication. Results for each replicate must be separated using the information provided in elements `control` and the 3 stage descriptor elements.

Density dependent projections are automatically set up if object `density` is input. If this object is not included, then density independent projections will be set up. Note that currently, density dependent projections can only be performed with `lefkoMat` objects.

The stage distributions and reproductive values produced are not the asymptotic values as would be given by the standardized right and left eigenvectors associated with the dominant eigenvalue of a matrix, but are vectors describing these values at the specific points in time projected. See equations 14.86 and 14.88 and section 14.4 on Sensitivity and Elasticity Analysis under Environmental Stochasticity in Caswell (2001, Matrix Population Models, Sinauer Associates) for more details.

See Also

`start_input()`

`density_input()`

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, 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"),
  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 = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
  supplement = lathsupp3, yearcol = "year2", indivcol = "individ")

lathproj <- projection3(ehrlen3, nreps = 5, 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")
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,
  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",
    "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
    "SL", "SL", "rep", "rep"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "P3", "P3",
    "SL", "SL", "SL", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
    NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.25, NA, NA, NA, NA, NA, NA,
    NA, NA),
  multiplier = c(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, 3, 3),
  type_t12 = c(1, 2, 1, 2, 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")

cypstoch <- projection3(cypmatrix3r, nreps = 5, stochastic = TRUE)
```

repvalue3 *Estimate Reproductive Value*

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 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")
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", "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", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
repvalue3(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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

repvalue3(cypmatrix2r, stochastic = TRUE)
```

repvalue3.lefkoMat *Estimate Reproductive Value Vectors of Matrices in a lefkoMat Object*

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 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 occasions 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 is chosen, and long-run mean reproductive value vectors if stochastic analysis is 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 deterministic and 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 reproductive value vector (rep_value). If stochastic and ahistorical, then a single data frame is output starting with the number of the population-patch (matrix_set), a string concatenating the names of the population and the patch (poppatch), the assigned stage id number (stage_id), and the stage name (stage), and the long-run mean reproductive value vector (rep_value).

If a historical matrix is used as input, then two data frames are output into a list object. The `hist` element describes the historical stage-pair reproductive values, while the `ahist` element describes the stage reproductive values. If deterministic, then `hist` contains a data frame including the matrix number (`matrix`), the numeric stage designations for stages in occasions *t* and *t*-1, (`stage_id_2` and `stage_id_1`, respectively), followed by the respective stage names (`stage_2` and `stage_1`), and ending with the estimated reproductive values (`rep_value`). The associated `ahist` element is as before. If stochastic, then the `hist` element contains a single data frame with the number of the population-patch (`matrix_set`), a string concatenating the names of the population and the patch (`poppatch`), the assigned stage id numbers in times *t* and *t*-1 (`stage_id_2` and `stage_id_2`, respectively), and the associated stage names (`stage_2` and `stage_1`, respectively), and the long-run mean reproductive values (`rep_value`). The associated `ahist` element is as before in the ahistorical, stochastic 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 occasions, 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 occasions if the simulation is at least 2000 projected occasions long. If between 500 and 2000 projected occasions long, then only the final 200 are used, and if fewer than 500 occasions 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 occasions per simulation, with 10000 preferred.

See Also

`repvalue3()`

`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")
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", "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", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
repvalue3(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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

repvalue3(cypmatrix2r, stochastic = TRUE)
```

repvalue3.matrix *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".
...	Other parameters.

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.

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")
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", "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", indivcol = "individ")
```

```
ehrlen3mean <- lmean(ehrlen3)
repvalue3(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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

repvalue3(cypmatrix2r$A[[1]])
```

rlefko2 *Create Raw Ahistorical Matrix Projection Model*

Description

Function rlefko2() returns raw ahistorical MPMs, including the associated component transition and fecundity matrices, a data frame describing the ahistorical stages used, and a data frame describing the population, patch, and occasion time 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"),
  sizeb = c(NA, NA),
  sizec = c(NA, NA),
  repst = c("repstatus3", "repstatus2"),
  matst = c("matstatus3", "matstatus2"),
  fec = c("feca3", "feca2"),
  supplement = NULL,
  repmatrix = NULL,
  overwrite = NULL,
  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 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 observation occasion, or a set of such values, given in values associated with the `year` term used in vital rate model development. Can also equal `all`, in which case matrices will be estimated for all occasion times. 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 occasions *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 occasions *t*+1 ans *t*, respectively.

`size` A vector of names of variables coding the primary size variable in occasions *t*+1 and *t*, respectively. Defaults to `c("sizea3","sizea2")`.

`sizeb` A vector of names of variables coding the secondary size variable in occasions *t*+1 and *t*, respectively. Defaults to `c(NA,NA)`.

`sizec` A vector of names of variables coding the tertiary size variable in occasions *t*+1 and *t*, respectively. Defaults to `c(NA,NA)`.

`repst` A vector of names of variables coding reproductive status in occasions *t*+1 and *t*, respectively. Defaults to `c("repstatus3","repstatus2")`. Must be supplied if `stages` is not provided.

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

`fec` A vector of names of variables coding fecundity in occasions *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. Should be used in place of or in addition to an overwrite table (see `overwrite` below) and a reproduction matrix (see `repmatrix` below).

`repmatrix` An optional reproduction matrix. This matrix is composed mostly of 0s, with non-zero entries acting as element identifiers and multipliers for fecundity (with 1 equaling full fecundity). If left blank, and no `supplement` is provided, then rlefko2() will assume that all stages marked as reproductive produce offspring at 1x that of estimated fecundity, and that offspring production will yield 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 occasion *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 the respective 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, which is a list that holds the matrix projection model and all of its metadata. Its structure is a list with the following elements:

A A list of full projection matrices in order of sorted populations, patches, and occasions. 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. Set to NA for ahistorical matrices.

agestages A data frame showing age-stage pairs. In this function, it is 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 also incorporate given or proxy survival transitions, input those given and proxy transitions through the `overwrite` options.

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 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. Otherwise the variable identifying patch needs to be named.

Input options including multiple variable names must be entered in the order of variables in occasion *t*+1 and *t*. Rearranging the order WILL lead to erroneous calculations, and may lead to fatal errors.

Although this function is capable of assigning stages given an input stageframe, it lacks the power of `verticalize3()` and `historicalize3()` in this regard. Users are strongly encouraged to use the latter two functions for stage assignment.

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, 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)

lathsupp2 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl"),
  stage2 = c("Sd", "Sd", "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 = lathframe, historical = FALSE)

ehrlen2 <- rlefko2(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2"), supplement = lathsupp2, yearcol = "year2",
  indivcol = "individ")

summary(ehrlen2)

# 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, 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

cypmatrix2r$A[[1]]
```

rlefko3 *Create Raw Historical Matrix Projection Model*

Description

Function `rlefko3()` returns raw historical MPMs, including the associated component transition and fecundity matrices, data frames describing the ahistorical stages used and the historical paired stages, and a data frame describing the population, patch, and occasion time 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"),
  sizeb = c(NA, NA, NA),
  sizec = c(NA, NA, NA),
  repst = c("repstatus3", "repstatus2", "repstatus1"),
  matst = c("matstatus3", "matstatus2", "matstatus1"),
  fec = c("feca3", "feca2", "feca1"),
  supplement = NULL,
  repmatrix = NULL,
  overwrite = NULL,
  yearcol = NA,
  popcol = NA,
  patchcol = NA,
  indivcol = NA,
  censorcol = NA,
  censorkeep = 0,
  format = "ehrlen",
  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 observation occasion, or a set of such values, given in values associated with the `year` term used in vital rate model development. Can also equal `all`, in which case matrices will be estimated for all occasions. 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 occasions *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 occasions *t*+1, *t*, and *t*-1, respectively.

`size` A vector of names of variables coding the primary size variable in occasions *t*+1, *t*, and *t*-1, respectively. Defaults to `c("sizea3","sizea2","sizea1")`.

`sizeb` A vector of names of variables coding the secondary size variable in occasions *t*+1, *t*, and *t*-1, respectively. Defaults to `c(NA,NA,NA)`.

`sizec` A vector of names of variables coding the tertiary size variable in occasions *t*+1, *t*, and *t*-1, respectively. Defaults to `c(NA,NA,NA)`.

`repst` A vector of names of variables coding reproductive status in occasions *t*+1, *t*, and *t*-1, respectively. Defaults to `c("repstatus3","repstatus2","repstatus1")`. Must be supplied if `stages` is not provided.

`matst` A vector of names of variables coding maturity status in occasions *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 occasions *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. Should be used in place of or in addition to an overwrite table (see `overwrite` below) and a reproduction matrix (see `repmatrix` below).

`repmatrix` An optional reproduction matrix. This matrix is composed mostly of 0s, with non-zero entries acting as element identifiers and multipliers for fecundity (with 1 equaling full fecundity). If left blank, and no `supplement` is provided, then `rlefko3()` will assume that all stages marked as reproductive produce offspring at 1x that of estimated fecundity, and that offspring production will yield 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 occasion *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 occasion *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`.

`format` A string indicating whether to estimate matrices in `ehrlen` format or `deVries` format. The latter adds one unborn 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 the respective 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. Defaults to `FALSE`.

Value

If all inputs are properly formatted, then this function will return an object of class `lefkoMat`, which is a list that holds the matrix projection model and all of its metadata. Its structure is a list with the following elements:

`A` A list of full projection matrices in order of sorted populations, patches, and occasions. 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
: A data frame showing age-stage pairs. In this function, it is 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 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 occasions *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. Otherwise the variable identifying patch needs to be named.

Input options including multiple variable names must be entered in the order of variables in occasion *t*+1, *t*, and *t*-1. Rearranging the order WILL lead to erroneous calculations, and may lead to fatal errors.

Although this function is capable of assigning stages given an input stageframe, it lacks the power of verticalize3() and historicalize3() in this regard. Users are strongly encouraged to use the latter two functions for stage assignment.

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, 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", "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")

summary(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)
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,
  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",
    "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
    "SL", "SL", "rep", "rep"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "P3", "P3",
    "SL", "SL", "SL", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
    NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.25, NA, NA, NA, NA, NA, NA,
    NA, NA),
  multiplier = c(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, 3, 3),
  type_t12 = c(1, 2, 1, 2, 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")

summary(cypmatrix3r)
```

sensitivity3 *Estimate Sensitivity of Population Growth Rate to Matrix Elements*

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 λ in the former case and the log of the stochastic λ 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

```
# 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)
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", "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, 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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

sensitivity3(cypmatrix2r)
```

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 λ is taken as the population growth rate. If stochastic, then the log of stochastic λ, 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",
  append_mats = FALSE,
  ...
)
```

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 occasions 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 occasions.
sparse	A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
append_mats	A logical value indicating whether to include the original A, U, and F matrices in the output lefkoSens object.
...	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, agestages, 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 λ.

See Also

sensitivity3()

sensitivity3.matrix()

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")
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", "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", indivcol = "individ")

sensitivity3(ehrlen3, 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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

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

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

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 occasions 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 occasions.
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".

`append_mats` A logical value indicating whether to include the original matrices input as object `mats` in the output `lefkoSense` object. Defaults to FALSE.

`...` 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, 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", "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")

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", "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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

sensitivity3(cypmatrix2r$A)
```

sensitivity3.matrix *Estimate Sensitivity of Population Growth Rate of a Single Matrix*

Description

sensitivity3.matrix() returns the sensitivities of λ 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

```
## 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.

Value

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")
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", "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", indivcol = "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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

sensitivity3(cypmatrix2r$A[[1]])
```

sf_create   *Create Stageframe for Population Matrix Projection Analysis*

Description

Function 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 including flefko3(), flefko2(), aflefko2(), rlefko3(), and rlefko2(), in which it determines how each stage is treated during matrix estimation.

Usage

```
sf_create(
  sizes,
  stagenames = NULL,
  sizesb = NULL,
  sizesc = NULL,
  repstatus = NULL,
  obsstatus = NULL,
  propstatus = NULL,
  matstatus = NULL,
  immstatus = NULL,
  minage = NULL,
  maxage = NULL,
  indataset = NULL,
  binhalfwidth = NULL,
  binhalfwidthb = NULL,
  binhalfwidthc = NULL,
  group = NULL,
  comments = NULL,
  roundsize = 5L,
  roundsizeb = 5L,
  roundsizec = 5L,
  ipmbins = 100L,
  ipmbinsb = NA_integer_,
  ipmbinsc = NA_integer_
)
```

Arguments

`sizes` A numeric vector of the typical or representative size of each life history stage. If making function-based MPMs, then this should be a vector composed of the midpoints of each size bin. If denoting the boundary of an automated size classification group, then should denote the absolute minimum size of that group, or the absolute size of that group (see Notes).

`stagenames` A vector of stage names, in the same order as elements in sizes. Can also be set to `ipm` for automated size classification (see Notes section).

`sizesb` An optional numeric vector for a second size metric for each life history stage. Only to be used if stages are defined by at least two size metrics in all cases. Same issues apply as in `sizes`.

`sizesc` An optional numeric vector for a third size metric for each life history stage. Only to be used if stages are defined by at least three size metrics in all cases. Same issues apply as in `sizes`.

`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 0.

`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`.

`immstatus` A vector denoting whether each stage is immature. Must be composed of binomial values if given. Defaults to the complement of vector `matstatus`.

`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 status variables. 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.

`binhalfwidthb` A numeric vector giving the half-width of size bins used for the optional second size metric. Required to classify individuals appropriately with two or three size classes. Defaults to 0.5 for all sizes.

`binhalfwidthc` A numeric vector giving the half-width of size bins used for the optional third size metric. Required to classify individuals appropriately with three size classes. Defaults to 0.5 for all sizes.

`group` An integer vector providing information on each respective stage's size classification group. If used, then function-based MPM creation functions `flefko2()`, `flefko3()`, and `aflefko2()` will estimate transitions only within these groups and for allowed cross-group transitions noted within the supplement table. Defaults to 0.

`comments` An optional vector of text entries holding useful text descriptions of all stages.

`roundsize` This parameter sets the precision of size classification, and equals the number of digits used in rounding sizes. Defaults to 5.

`roundsizeb` This parameter sets the precision of size classification in the optional second size metric, and equals the number of digits used in rounding sizes. Defaults to 5.

`roundsizec` This parameter sets the precision of size classification in the optional third size metric, and equals the number of digits used in rounding sizes. Defaults to 5.

`ipmbins` An integer giving the number of size bins to create using the primary size classification variable. 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.

`ipmbinsb` An optional integer giving the number of size bins to create using the secondary size classification variable. This number is in addition to any stages that are not size classified, as well as in addition to any automated size classification using the primary and tertiary size variables. Defaults to NA, and must be set to a positive integer for automated size classification to progress.

ipmbinsc An optional integer giving the number of size bins to create using the tertiary size classification variable. This number is in addition to any stages that are not size classified, as well as in addition to any automated size classification using the primary and secondary size variables. Defaults to NA, and must be set to a positive integer for automated size classification to progress.

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, stage group classification, 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.

size_b Size at which each stage occurs in terms of a second size variable, if one exists.

size_c Size at which each stage occurs in terms of a third size variable, if one exists.

min_age The minimum age at which the stage may occur.

max_age The maximum age at which the stage may occur.

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.

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 midpoint of the size bin at which the stage may occur.

sizebin_width The width of the size bin corresponding to the stage.

binhalfwidthb_raw
The half-width of the size bin of a second size variable, as input.

sizebinb_min The minimum size at which the stage may occur.

sizebinb_max The maximum size at which the stage may occur.

`sizebinb_center` The midpoint of the size bin at which the stage may occur, in terms of a second size variable.

`sizebinb_width` The width of the size bin corresponding to the stage, in terms of a second size variable.

`binhalfwidthc_raw` The half-width of the size bin of a third size variable, as input.

`sizebinc_min` The minimum size at which the stage may occur, in terms of a third size variable.

`sizebinc_max` The maximum size at which the stage may occur, in terms of a third size variable.

`sizebinc_center` The midpoint of the size bin at which the stage may occur, in terms of a third size variable.

`sizebinc_width` The width of the size bin corresponding to the stage, in terms of a third size variable.

`group` An integer denoting the size classification group that the stage falls within.

`comments` A text field for stage descriptions.

Notes

If an IPM or function-based matrix with automated size classification is desired, then two stages that occur within the dataset and represent the lower and upper size limits of the IPM must be marked with `ipm` in the stagenames vector. These stages should have all characteristics other than size equal, and the size input for whichever size will be classified automatically must include the minimum in one stage and the maximum in the other. The actual characteristics of the first stage encountered in the inputs will be used as the template for the creation of these sizes. Note that `ipm` refers to size classification with the primary size variable. To automate size classification with the secondary size variable, use `ipmb`, and to automate size classification with the tertiary size variable, use `ipmc`. To nest automated size classifications, use `ipmab` for the primary and secondary size variables, `ipmac` for the primary and tertiary size variables, `ipmbc` for the secondary and tertiary size variables, and `ipmabc` for all three size variables. The primary size variable can also be set with `ipma`.

If two or more groups of stages, each with its own characteristics, are to be developed for an IPM or function-based MPM, then an even number of stages with two stages marking the minimum and maximum size of each group should be marked with the same code as given above, with all other characteristics equal within each group.

Stage classification groups set with the `group` variable create zones within function-based matrices in which survival transitions are estimated. These groups should not be set if transitions are possible between all stages regardless of group. To denote specific transitions as estimable between stage groups, use the `supplemental()` function.

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, 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", "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)
ehrlen3mean$A[[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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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 Overdispersion and Zero Inflation in Size and Fecundity Distributions*

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

```
sf_distrib(
  data,
  sizea = NA,
  sizeb = NA,
  sizec = NA,
  obs3 = NA,
  fec = NA,
  repst = NA,
  zisizea = TRUE,
  zisizeb = TRUE,
  zisizec = TRUE,
  zifec = TRUE,
  fectime = 2
)
```

Arguments

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

`sizea` A vector holding the name or column number of the variables corresponding to primary size in occasions *t*+1 and *t*. Input only if `sizea` is to be tested.

`sizeb` A vector holding the name or column number of the variables corresponding to secondary size in occasions *t*+1 and *t*. Input only if `sizeb` is to be tested.

`sizec` A vector holding the name or column number of the variables corresponding to tertiary size in occasions *t*+1 and *t*. Input only if `sizec` is to be tested.

`obs3` The name or column number of the variable corresponding to observation status in occasion *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 occasion, either occasion *t* or occasion *t*-1. Input only if `fec` is to be tested.

`repst` The name or column number of the variable corresponding to reproductive status in occasion *t*. If not provided, then fecundity will be tested without subsetting to only reproductive individuals.

`zisizea` A logical value indicating whether to conduct a test of zero inflation in primary size. Defaults to `TRUE`.

`zisizeb` A logical value indicating whether to conduct a test of zero inflation in secondary size. Defaults to `TRUE`.

`zisizec` A logical value indicating whether to conduct a test of zero inflation in tertiary size. Defaults to `TRUE`.

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

`fectime` An integer indicating whether to treat fecundity as occurring in time *t* (`2`) or time *t*+1 (`3`). Defaults to `2`.

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 occasion *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

```
# 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")
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)

lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
  immstatus = 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)

# 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, sizea = c("sizea3", "sizea2"), fec = c("feca3", "feca2"),
  repst = c("repstatus3", "repstatus2"), zifec = FALSE)
```

```
# 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, 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,
  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, sizea = c("size3added", "size2added"),
  fec = c("feca3", "feca2"), repst = c("repstatus3", "repstatus2"),
  zisizea = TRUE)
```

slambda3 *Estimate Stochastic Population Growth Rate*

Description

Function slambda3() estimates the stochastic population growth rate, a, defined as the long-term arithmetic mean of the log population growth rate estimated per simulated occasion (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, dense_only = FALSE, tweights = NULL)
```

Arguments

mpm	A matrix projection model of class lefkoMat, or a list of full matrix projection matrices.
times	Number of occasions to iterate. Defaults to 10,000.
dense_only	A logical value indicating whether to force matrices to be run in dense format. Defaults to FALSE, and should only be used if errors occur when running under default conditions.
tweights	Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among occasions.

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 occasions.
var	The estimated variance of a.
sd	The standard deviation of a.
se	The standard error of a.

Notes

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.

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, 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"),
  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 = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
  supplement = lathsupp3, 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")
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,
  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",
    "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
    "SL", "SL", "rep", "rep"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "P3", "P3",
    "SL", "SL", "SL", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
    NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.25, NA, NA, NA, NA, NA, NA,
    NA, NA),
  multiplier = c(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, 3, 3),
  type_t12 = c(1, 2, 1, 2, 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")

cypstoch <- slambda3(cypmatrix3r, dense_only = TRUE)
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 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", "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", "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", indivcol = "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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

stablestage3(cypmatrix2r, stochastic = TRUE)
```

`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 ahistorical matrices.

Usage

```
## 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 occasions 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 deterministic and 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 stochastic and ahistorical, then a single data frame is output starting with the number of the population-patch (`matrix_set`), a string concatenating the names of the population and the patch (`poppatch`), the assigned stage id number (`stage_id`), and the stage name (`stage`), and the long-run average 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 describes the historical stage-pair distribution, while the `ahist` element describes the stage distribution. If deterministic, then `hist` contains a data frame including the matrix number (`matrix`), the numeric stage designations for stages in occasions *t* and *t*-1, (`stage_id_2` and stage_id_1, respectively), followed by the respective stage names (stage_2 and stage_1), and ending with the estimated stable stage-pair distribution. The associated ahist element is as before. If stochastic, then the hist element contains a single data frame with the number of the population-patch (matrix_set), a string concatenating the names of the population and the patch (poppatch), the assigned stage id numbers in times *t* and *t*-1 (stage_id_2 and stage_id_2, respectively), and the associated stage names (stage_2 and stage_1, respectively), and the long-run average stage distribution (ss_prop). The associated ahist element is as before in the ahistorical, stochastic 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 stage distributions across all projected occasions, 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 occasions if the simulation is at least 2000 projected occasions long. If between 500 and 2000 projected occasions long, then only the final 200 are used, and if fewer than 500 occasions 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 occasions 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")
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", "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", indivcol = "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")
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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

stablestage3(cypmatrix2r, 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, 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", "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", 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, 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

stablestage3(cypmatrix2r$A[[1]])
```

start_input *Create a Starting Vector for Population Projection*

Description

Function start_input() creates a data frame summarizing the non-zero elements of the start vector for use in population projection analysis via function projection3().

Usage

```
start_input(mpm, stage2, stage1 = NA, age2 = NA, value = 1)
```

Arguments

`mpm` The lefkoMat object to be used in projection analysis.

`stage2` A vector showing the name or number of a stage in occasion *t* that should be set to a positive number of individuals in the start vector. Abbreviations for groups of stages are also usable (see Notes). This input is required and has no default input.

`stage1` A vector showing the name or number of a stage in occasion *t*-1 that should be set to a positive number of individuals in the start vector. Abbreviations for groups of stages are also usable (see Notes). This is only used for historical MPMs, since the rows of hMPMs correspond to stage-pairs in times *t* and *t*-1 together. Only required for historical MPMs, and will result in errors if otherwise used.

`age2` A vector showing the age of each respective stage in occasion *t* that should be set to a positie number of individuals in the start vector. Only used for age-by-stage MPMs. Defaults to NA.

`value` A vector showing the values, in order, of the number of individuals set for the stage or stage-pair in question. Defaults to 1.

Value

A list of class `lefkoStart`, with 4 objects, which can be used as input in function `projection3()`. The last three include the `ahstages`, `hstages`, and `agestages` objects from the `lefkoMat` object supplied in `mpm`. The first element in the list is a data frame with the following variables:

`stage2` Stage at occasion *t*.

`stage_id_2` The stage number associated with `stage2`.

`stage1` Stage at occasion *t*-1, if historical. Otherwise NA.

`stage_id_1` The stage number associated with `stage1`.

`age2` The age of individuals in `stage2` and, if applicable, `stage1`. Only used in age-by-stage MPMs.

`row_num` A number indicating the respective starting vector element.

`value` Number of individuals in corresponding stage or stage-pair.

Notes

Entries in `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, `immat` 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.

See Also

`density_input()`

`projection3()`

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, 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", "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)

e3m_sv <- start_input(ehrlen3mean, stage2 = "Sd", stage1 = "Sd", value = 1000)
```

subset_lM *Create New lefkoMat Object as Subset of Another lefkoMat Object*

Description

Function subset_lM() creates a new lefkoMat object from a subset of matrices in another lefkoMat object.

Usage

```
subset_lM(lM, mat_num = NA, pop = NA, patch = NA, year = NA)
```

Arguments

lM	The lefkoMat object to select matrices from.
mat_num	Either a single integer corresponding to the matrix to select within the labels element of lM, or a vector of such integers.
pop	The population designation for matrices to select. Only used if mat_num is not given.
patch	The patch designation for matrices to select. Only used if mat_num is not given.
year	The time *t* designation for matrices to select. Only used if mat_num is not given.

Value

A lefkoMat object composed of the matrices specified in the options.

Notes

If mat_num is not provided, then at least one of pop, patch, or year must be provided. If at least two of pop, patch, and year are provided, then function subset_lM() will identify matrices as the intersection of provided inputs.

See Also

create_lM()

add_lM()

delete_lM()

Examples

```
# These matrices are of 9 populations of the plant species Anthyllis
# vulneraria, and were originally published in Davison et al. (2010) Journal
# of Ecology 98:255-267 (doi: 10.1111/j.1365-2745.2009.01611.x).

sizevector <- c(1, 1, 2, 3) # These sizes are not from the original paper
stagevector <- c("Sdl", "Veg", "SmFlo", "LFlo")
repvector <- c(0, 0, 1, 1)
obsvector <- c(1, 1, 1, 1)
matvector <- c(0, 1, 1, 1)
immvector <- c(1, 0, 0, 0)
propvector <- c(0, 0, 0, 0)
```

```
indataset <- c(1, 1, 1, 1)
binvec <- c(0.5, 0.5, 0.5, 0.5)

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

# POPN C 2003-2004
XC3 <- matrix(c(0, 0, 1.74, 1.74,
0.208333333, 0, 0, 0.057142857,
0.041666667, 0.076923077, 0, 0,
0.083333333, 0.076923077, 0.066666667, 0.028571429), 4, 4, byrow = TRUE)

# 2004-2005
XC4 <- matrix(c(0, 0, 0.3, 0.6,
0.32183908, 0.142857143, 0, 0,
0.16091954, 0.285714286, 0, 0,
0.252873563, 0.285714286, 0.5, 0.6), 4, 4, byrow = TRUE)

# 2005-2006
XC5 <- matrix(c(0, 0, 0.50625, 0.675,
0, 0, 0, 0.035714286,
0.1, 0.068965517, 0.0625, 0.107142857,
0.3, 0.137931034, 0, 0.071428571), 4, 4, byrow = TRUE)

# POPN E 2003-2004
XE3 <- matrix(c(0, 0, 2.44, 6.569230769,
0.196428571, 0, 0, 0,
0.125, 0.5, 0, 0,
0.160714286, 0.5, 0.133333333, 0.076923077), 4, 4, byrow = TRUE)

XE4 <- matrix(c(0, 0, 0.45, 0.646153846,
0.06557377, 0.090909091, 0.125, 0,
0.032786885, 0, 0.125, 0.076923077,
0.049180328, 0, 0.125, 0.230769231), 4, 4, byrow = TRUE)

XE5 <- matrix(c(0, 0, 2.85, 3.99,
0.083333333, 0, 0, 0,
0, 0, 0, 0,
0.416666667, 0.1, 0, 0.1), 4, 4, byrow = TRUE)

# POPN F 2003-2004
XF3 <- matrix(c(0, 0, 1.815, 7.058333333,
0.075949367, 0, 0.05, 0.083333333,
0.139240506, 0, 0, 0.25,
0.075949367, 0, 0, 0.083333333), 4, 4, byrow = TRUE)

XF4 <- matrix(c(0, 0, 1.233333333, 7.4,
0.223880597, 0, 0.111111111, 0.142857143,
0.134328358, 0.272727273, 0.166666667, 0.142857143,
0.119402985, 0.363636364, 0.055555556, 0.142857143), 4, 4, byrow = TRUE)
```

```
XF5 <- matrix(c(0, 0, 1.06, 3.372727273,
0.073170732, 0.025, 0.033333333, 0,
0.036585366, 0.15, 0.1, 0.136363636,
0.06097561, 0.225, 0.166666667, 0.272727273), 4, 4, byrow = TRUE)

# POPN G 2003-2004
XG3 <- matrix(c(0, 0, 0.245454545, 2.1,
0, 0, 0.045454545, 0,
0.125, 0, 0.090909091, 0,
0.125, 0, 0.090909091, 0.333333333), 4, 4, byrow = TRUE)

XG4 <- matrix(c(0, 0, 1.1, 1.54,
0.111111111, 0, 0, 0,
0, 0, 0, 0,
0.111111111, 0, 0, 0), 4, 4, byrow = TRUE)

XG5 <- matrix(c(0, 0, 0, 1.5,
0, 0, 0, 0,
0.090909091, 0, 0, 0,
0.545454545, 0.5, 0, 0.5), 4, 4, byrow = TRUE)

# POPN L 2003-2004
XL3 <- matrix(c(0, 0, 1.785365854, 1.856521739,
0.128571429, 0, 0, 0.010869565,
0.028571429, 0, 0, 0,
0.014285714, 0, 0, 0.02173913), 4, 4, byrow = TRUE)

XL4 <- matrix(c(0, 0, 14.25, 16.625,
0.131443299, 0.057142857, 0, 0.25,
0.144329897, 0, 0, 0,
0.092783505, 0.2, 0, 0.25), 4, 4, byrow = TRUE)

XL5 <- matrix(c(0, 0, 0.594642857, 1.765909091,
0, 0, 0.017857143, 0,
0.021052632, 0.018518519, 0.035714286, 0.045454545,
0.021052632, 0.018518519, 0.035714286, 0.068181818), 4, 4, byrow = TRUE)

# POPN O 2003-2004
XO3 <- matrix(c(0, 0, 11.5, 2.775862069,
0.6, 0.285714286, 0.333333333, 0.24137931,
0.04, 0.142857143, 0, 0,
0.16, 0.285714286, 0, 0.172413793), 4, 4, byrow = TRUE)

XO4 <- matrix(c(0, 0, 3.78, 1.225,
0.28358209, 0.171052632, 0, 0.166666667,
0.084577114, 0.026315789, 0, 0.055555556,
0.139303483, 0.447368421, 0, 0.305555556), 4, 4, byrow = TRUE)

XO5 <- matrix(c(0, 0, 1.542857143, 1.035616438,
0.126984127, 0.105263158, 0.047619048, 0.054794521,
0.095238095, 0.157894737, 0.19047619, 0.082191781,
0.111111111, 0.223684211, 0, 0.356164384), 4, 4, byrow = TRUE)
```

```
# POPN Q 2003-2004
XQ3 <- matrix(c(0, 0, 0.15, 0.175,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0, 0, 0), 4, 4, byrow = TRUE)

XQ4 <- matrix(c(0, 0, 0, 0.25,
0, 0, 0, 0,
0, 0, 0, 0,
1, 0.666666667, 0, 1), 4, 4, byrow = TRUE)

XQ5 <- matrix(c(0, 0, 0, 1.428571429,
0, 0, 0, 0.142857143,
0.25, 0, 0, 0,
0.25, 0, 0, 0.571428571), 4, 4, byrow = TRUE)

# POPN R 2003-2004
XR3 <- matrix(c(0, 0, 0.7, 0.6125,
0.25, 0, 0, 0.125,
0, 0, 0, 0,
0.25, 0.166666667, 0, 0.25), 4, 4, byrow = TRUE)

XR4 <- matrix(c(0, 0, 0, 0.6,
0.285714286, 0, 0, 0,
0.285714286, 0.333333333, 0, 0,
0.285714286, 0.333333333, 0, 1), 4, 4, byrow = TRUE)

XR5 <- matrix(c(0, 0, 0.7, 0.6125,
0, 0, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0.333333333, 0.625), 4, 4, byrow = TRUE)

# POPN S 2003-2004
XS3 <- matrix(c(0, 0, 2.1, 0.816666667,
0.166666667, 0, 0, 0,
0, 0, 0, 0,
0, 0, 0, 0.166666667), 4, 4, byrow = TRUE)

XS4 <- matrix(c(0, 0, 0, 7,
0.333333333, 0.5, 0, 0,
0, 0, 0, 0,
0.333333333, 0, 0, 1), 4, 4, byrow = TRUE)

XS5 <- matrix(c(0, 0, 0, 1.4,
0, 0, 0, 0,
0, 0, 0, 0.2,
0.111111111, 0.75, 0, 0.2), 4, 4, byrow = TRUE)

mats_list <- list(XC3, XC4, XC5, XE3, XE4, XE5, XF3, XF4, XF5, XG3, XG4, XG5,
  XL3, XL4, XL5, XO3, XO4, XO5, XQ3, XQ4, XQ5, XR3, XR4, XR5, XS3, XS4, XS5)

yr_ord <- c(1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1,
  2, 3, 1, 2, 3)
```

```
pch_ord <- c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7,
  8, 8, 8, 9, 9, 9)

anth_lefkoMat <- create_lM(mats_list, anthframe, hstages = NA, historical = FALSE,
  poporder = 1, patchorder = pch_ord, yearorder = yr_ord)

smaller_anth_lM <- subset_lM(anth_lefkoMat, patch = c(1, 2, 3),
  year = c(1, 2))
smaller_anth_lM
```

summary.lefkoCondMat *Summary of Class "lefkoCondMat"*

Description

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

Usage

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

Arguments

object	An object of class lefkoCondMat.
...	Other parameters.

Value

A text summary of the object shown on the console, showing the number of historical matrices, as well as the number of conditional matrices nested within each historical 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, 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", "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")

lathcondmats <- cond_hmpm(ehrlen3)
summary(lathcondmats)

# 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,
  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",
    "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
    "SL", "SL", "rep", "rep"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "P3", "P3",
    "SL", "SL", "SL", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
    NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.25, NA, NA, NA, NA, NA, NA,
    NA, NA),
  multiplier = c(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, 3, 3),
  type_t12 = c(1, 2, 1, 2, 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)
```

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

```
## 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

```
# 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)
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", "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)

lathsupp2 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl"),
  stage2 = c("Sd", "Sd", "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 = lathframe, historical = FALSE)

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

ehrlen2 <- rlefko2(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2"), supplement = lathsupp2,
  yearcol = "year2", indivcol = "individ")

ehrlen3elas <- elasticity3(ehrlen3)
ehrlen2elas <- elasticity3(ehrlen2)

summary(ehrlen3elas)
summary(ehrlen2elas)
```

summary.lefkoLTRE *Summarize lefkoLTRE Objects*

Description

Function summary.lefkoLTRE() summarizes lefkoLTRE objects. Particularly, it breaks down LTRE contributions by the kind of ahistorical and, if applicable, historical transition.

Usage

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

Arguments

object	A lefkoLTRE object.
...	Other parameters.

Value

A list composed of 2 (if deterministic) or 4 (if stochastic) data frames. If deterministic, then hist_det is a data frame showing the summed LTRE contributions for all 16 kinds of historical transition per matrix, with each column corresponding to each A matrix in order, followed by all summed positive and all summed negative contributions. Object ahist_det is a data frame showing the summed LTRE contributions for all 4 kinds of ahistorical transition per matrix, with order as before, followed by summed positive and summed negative contributions. If stochastic, then hist_mean and hist_sd are the summed LTRE contributions for the mean vital rates and variability in vital rates, respectively, according to all 16 historical transition types, followed by summed positive and negative contributions, and ahist_mean and ahist_sd are the equivalent ahistorical versions.

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, 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", "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)

lathsupp2 <- supplemental(stage3 = c("Sd", "Sdl", "Sd", "Sdl"),
  stage2 = c("Sd", "Sd", "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 = lathframe, historical = FALSE)

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

ehrlen2 <- rlefko2(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2"), supplement = lathsupp2,
  yearcol = "year2", indivcol = "individ")

ehrlen3ltre <- ltre3(ehrlen3)
```

```
summary(ehrlen3ltre)
```

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(), rlefko2(), and aflefko2().

Usage

```
## S3 method for class 'lefkoMat'
summary(object, colsums = TRUE, ...)
```

Arguments

object	An object of class lefkoMat.
colsums	A logical value indicating whether column sums should be shown for U matrices, allowing users to check stage survival probabilities. Defaults to TRUE.
...	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, 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,
  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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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")

summary(cypmatrix2r)
```

summary.lefkoMod *Summary of Class "lefkoMod"*

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

```
## 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 numbers of individuals and of individual transitions used to estimate each vital rate best-fit model, along with the accuracy of each binomial model.

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, 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)

lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
  repstatus = repvector, obsstatus = obsvector, matstatus = matvector,
  immstatus = 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)
```

summary.lefkoProj *Summarize lefkoProj Objects*

Description

Function summary.lefkoProj() summarizes lefkoProj objects. Particularly, it breaks down the data frames provided in the projection element in ways meaningful for those running simulations.

Usage

```
## S3 method for class 'lefkoProj'
summary(object, threshold = 1, milepost = c(0, 0.25, 0.5, 0.75, 1), ...)
```

Arguments

object	A lefkoProj object.
threshold	A threshold population size to be searched for in projections. Defaults to 1.
milepost	A numeric vector indicating at which points in the projection to assess detailed results. Can be input as integer values, in which case each number must be between 1 and the total number of occasions projected in each projection, or decimals between 0 and 1, which would then be translated into the corresponding projection steps of the total. Defaults to c(0,0.25,0.50,0.75,1.00).
...	Other parameters.

Value

Apart from a statement of the results, we have the following item in the output:

milepost_sums	A data frame showing the number of replicates at each of the milepost times that is above the threshold population/patch size.

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, 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, 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, nreps = 5, stochastic = TRUE)
summary(lathproj)

# 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, 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,
  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",
    "D", "XSm", "Sm", "D", "XSm", "Sm", "SD", "P1"),
  stage2 = c("SD", "SD", "SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "SL",
    "SL", "SL", "rep", "rep"),
  stage1 = c("SD", "rep", "SD", "rep", "SD", "P1", "P2", "P3", "P3", "P3",
    "SL", "SL", "SL", "mat", "mat"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
    NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  eststage1 = c(NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm", "XSm",
    "XSm", NA, NA),
  givenrate = c(0.1, 0.1, 0.2, 0.2, 0.2, 0.2, 0.25, NA, NA, NA, NA, NA, NA,
    NA, NA),
  multiplier = c(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, 3, 3),
  type_t12 = c(1, 2, 1, 2, 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")

cypstoch <- projection3(cypmatrix3r, nreps = 5, stochastic = TRUE)
summary(cypstoch)
```

supplemental *Create a Data Frame of Supplemental Data for MPM Development*

Description

Function 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 = 1,
  type = NA,
  type_t12 = NA,
  stageframe,
  historical = TRUE
)
```

Arguments

`stage3` The name of the stage in occasion *t*+1 in the transition to be replaced. Abbreviations for groups of stages are also usable (see Notes).

`stage2` The name of the stage in occasion *t* in the transition to be replaced. Abbreviations for groups of stages are also usable (see Notes).

`stage1` The name of the stage in occasion *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 usable (see Notes).

`eststage3` The name of the stage to replace `stage3` in a proxy transition. Only needed if a transition will be replaced by another estimated transition.

`eststage2` The name of the stage to replace `stage2` in a proxy transition. Only needed if a transition will be replaced by another estimated transition.

`eststage1` The name of the stage to replace `stage1` in a proxy historical transition. Only needed if a transition will be replaced by another estimated transition, and the matrix to be estimated is historical. Stage `NotAlive` is also possible for raw hMPMs as a means of handling the prior stage for individuals entering the population in occasion *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 or for proxy transitions. Defaults to `1`.

`type` A vector denoting the kind of transition between occasions *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 occasions *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 `flefko3()`, `flefko2()`, `rlefko3()`, `rlefko2()`, and `aflefko2()`.

Variables in this object include the following:

`stage3` Stage at occasion *t*+1 in the transition to be replaced.

`stage2` Stage at occasion *t* in the transition to be replaced.

`stage1` Stage at occasion *t*-1 in the transition to be replaced.

`eststage3` Stage at occasion *t*+1 in the transition to replace the transition designated by `stage3`, `stage2`, and `stage1`.

`eststage2` Stage at occasion *t* in the transition to replace the transition designated by `stage3`, `stage2`, and `stage1`.

`eststage1` Stage at occasion *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.

`multiplier` A multiplier for proxy transitions or for fecundity.

`convtype` Designates whether the transition from occasion *t* to occasion *t*+1 is a survival transition probability (1), a fecundity rate (2), or a fecundity multiplier (3).

`convtype_t12` Designates whether the transition from occasion *t*-1 to occasion *t* is a survival transition probability (1), a fecundity rate (2).

Notes

Negative values are not allowed in `givenrate` and `multiplier` input.

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, `immat` 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. Also use `groupX` to denote all stages in group X (e.g. `group1` will use all stages in the respective stageframe's group 1).

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, 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", "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)
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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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
```

verticalize3 *Create Historical Vertical Data Frame from Horizontal Data Frame*

Description

Function `verticalize3()` returns a vertically formatted demographic data frame organized to create historical projection matrices, given a horizontally formatted input data frame. It also handles stage assignments if given an appropriate stageframe.

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,
  nonobsacol = 0,
  censorcol = 0,
  repstrrel = 1,
  fecrel = 1,
  stagecol = 0,
  stageassign = NA,
  stagesize = NA,
  censorkeep = 0,
  censorRepeat = FALSE,
  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 occasions in the dataset. A valid integer is required as input.

`firstyear` The first year or occasion 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 or subpopulation 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 occasion in the input dataset designated in `data`, if a set pattern of variables is used for each observation occasion 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 of each individual at each occasion, in Cartesian space. Can 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 of each individual at each occasion, 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. This 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 occasion 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. This variable should refer to the first size variable in the stageframe, unless `stagesize = "sizeadded"`.

`sizebcol` A second variable name(s) or column number(s) corresponding to the size entry associated with the first year or observation occasion 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. This variable should refer to the second size variable in the stageframe, unless `stagesize = "sizeadded"`.

`sizeccol` A third variable name(s) or column number(s) corresponding to the size entry associated with the first year or observation occasion 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. This variable should refer to the third size variable in the stageframe, unless `stagesize = "sizeadded"`.

`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 analyze 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 analyze 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 occasion 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 occasion 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 occasion. 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 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 occasion. 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 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 occasion. 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 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 occasion. 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 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 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 occasion. Can refer to the first instance of a stage identity variable in the dataset, or 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, sizeab, sizebc, sizeac, sizeabc, or sizeadded, depending on which size variable within the input dataset is chosen. Note that the variable(s) chosen should be presented in the order of the primary, secondary, and tertiary variables in the stageframe input with stageassign. For example, choosing sizeb assumes that this size is the primary variable in the stageframe.

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 new value to keep.

censorRepeat — A logical value indicating whether the censor variable is a single column, or whether it repeats across occasion blocks. Defaults to FALSE.

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 single X and Y columns. If TRUE, then each observation occasion 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 occasion *t* equal to 0. Defaults to `FALSE`, and should be kept `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 occasions 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:

`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 occasion *t*.

`firstseen` Occasion of first observation.

`lastseen` Occasion of last observation.

`obsage` Observed age in occasion *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 occasions *t*-1, *t*, and *t*+1, respectively, if provided.

`ypos1,ypos2,ypos3`
Y position in Cartesian space in occasions *t*-1, *t*, and *t*+1, respectively, if provided.

`sizea1,sizea2,sizea3`
Main size measurement in occasions *t*-1, *t*, and *t*+1, respectively.

`sizeb1,sizeb2,sizeb3`
Secondary size measurement in occasions *t*-1, *t*, and *t*+1, respectively.

`sizec1,sizec2,sizec3`
Tertiary measurement in occasions *t*-1, *t*, and *t*+1, respectively.

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

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

`repstrb1,repstrb2,repstrb3`
: Secondary numbers of reproductive structures in occasions *t*-1, *t*, and *t*+1, respectively.

`repstr1added,repstr2added,repstr3added`
: Sum of primary and secondary reproductive structures in occasions *t*-1, *t*, and *t*+1, respectively.

`feca1,feca2,feca3`
: Main numbers of offspring in occasions *t*-1, *t*, and *t*+1, respectively.

`fecb1,fecb2, fecb3`
: Secondary numbers of offspring in occasions *t*-1, *t*, and *t*+1, respectively.

`fec1added,fec2added,fec3added`
: Sum of primary and secondary fecundity in occasions *t*-1, *t*, and *t*+1, respectively.

`censor1,censor2,censor3`
: Censor state values in occasions *t*-1, *t*, and *t*+1, respectively.

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

`obsstatus1,obsstatus2,obsstatus3`
: Binomial observation state in occasions *t*-1, *t*, and *t*+1, respectively.

`repstatus1,repstatus2,repstatus3`
: Binomial reproductive state in occasions *t*-1, *t*, and *t*+1, respectively.

`fecstatus1,fecstatus2,fecstatus3`
: Binomial offspring production state in occasions *t*-1, *t*, and *t*+1, respectively.

`matstatus1,matstatus2,matstatus3`
: Binomial maturity state in occasions *t*-1, *t*, and *t*+1, respectively.

`alive1,alive2,alive3`
: Binomial state as alive in occasions *t*-1, *t*, and *t*+1, respectively.

`density`
: Radial density of individuals per unit designated in `spacing`. Only given if `spacing` is not NA.

Notes

In some datasets on species with unobserveable 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`).

Juvenile designation should only be used when juveniles fall outside of the size classification scheme used in determining stages. If juveniles are to be size classified along the size spectrum that adults also fall on, then it is best to treat juveniles as mature but not reproductive.

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 occasion *t* or *t*-1, as only living entries are allowed in time *t* and time *t*-1 may involve living entries as well as unliving entries immediately prior to birth.

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.

Unusual errors (e.g. "Error in .pfj...") may occur in cases where the variables are improperly passed, where seemingly numeric variables include text, or where the blocksize is improperly set.

Density estimation is performed as a count of individuals alive and within the radius specified in spacing of the respective individual at some point in time.

Examples

```
# 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")
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", "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", 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, 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",
  juvcol = c("Seedling1988", "Seedling1989", "Seedling1990", "Seedling1991"),
  sizeacol = c("Volume88", "Volume89", "Volume90", "Volume91"),
  repstracol = c("FCODE88", "FCODE89", "FCODE90", "FCODE91"),
  fecacol = c("Intactseed88", "Intactseed89", "Intactseed90", "Intactseed91"),
  deadacol = c("Dead1988", "Dead1989", "Dead1990", "Dead1991"),
  nonobsacol = c("Dormant1988", "Dormant1989", "Dormant1990", "Dormant1991"),
  censorcol = c("Missing1988", "Missing1989", "Missing1990", "Missing1991"),
  stageassign = lathframe, stagesize = "sizea",
  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, 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")
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,
  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, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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
```

```
# 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")
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,
  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", "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, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, 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
```

Index