

Package ‘lefko3’

September 8, 2021

Type Package

Title Historical and Ahistorical Population Projection Matrix Analysis

Version 3.8.0

Date 2021-09-08

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](https://doi.org/10.1890/0012-9658(2000)081[1675:TDOPPD]2.0.CO;2)> and deVries and Caswell (2018) <[doi:10.1007/s12080-017-0353-0](https://doi.org/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 7.1.1

Suggests knitr, popbio, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

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

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

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2021-09-08 06:50:05 UTC

R topics documented:

add_IM	3
afleko2	8
cond_hmpm	15
create_IM	17
cypdata	23
cypvert	26
delete_IM	28
elasticity3	33
elasticity3.lefkoMat	35
elasticity3.list	38
elasticity3.matrix	41
fleko2	44
fleko3	50
historicalize3	57
image3	64
image3.lefkoElas	67
image3.lefkoMat	69
image3.lefkoSens	71
image3.list	74
image3.matrix	76
lambda3	78
lambda3.lefkoMat	81
lambda3.matrix	83
lathyrus	86
lefko3	90
lmean	90
ltre3	93
ltre3.lefkoMat	95
modelsearch	99
overwrite	106
projection3	108
repvalue3	111
repvalue3.lefkoMat	114
repvalue3.matrix	117
rleko2	120
rleko3	125
sensitivity3	130
sensitivity3.lefkoMat	132
sensitivity3.list	135
sensitivity3.matrix	139
sf_create	141
sf_distrib	145
slambda3	148
stablestage3	151
stablestage3.lefkoMat	153
stablestage3.matrix	156

subset_IM	159
summary.lefkoCondMat	163
summary.lefkoElas	166
summary.lefkoLTRE	167
summary.lefkoMat	169
summary.lefkoMod	170
summary.lefkoProj	172
supplemental	175
verticalize3	179

Index**190**

add_IM	<i>Add Matrices to lefkoMat Object</i>
--------	--

Description

Function add_IM() adds matrices to lefkoMat objects.

Usage

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

Arguments

IM	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 <i>*t*</i> 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("Sd1", "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, X03, X04, X05, 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_IM(mats_list, anthframe, hstages = NA, historical = FALSE,
  poporder = 1, patchorder = pch_ord, yearorder = yr_ord)

# POPN H (EXCLUDED 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 years given, including the associated component transition and fecundity matrices, data frame detailing the characteristics of ahistorical stages and the exact age-stage combinations corresponding to rows and columns in estimated matrices, and a data frame characterizing the patch and year combinations corresponding to these matrices. Unlike `rlefko2()` and `rlefko3()`, this function does not currently distinguish populations.

Usage

```

aflefko2(
  year = "all",

```

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

Arguments

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

patch	A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to <code>all</code> if matrices should be estimated for all patches. Defaults to <code>all</code> .
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 <code>lefkoSD</code> 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 <code>supplemental()</code> function. Can be used in place of or in addition to an overwrite table (see <code>overwrite</code> below) and a reproduction matrix (see <code>repmatrix</code> below).
repmatrix	A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in <code>supplement</code> . If left blank, then <code>afleko2()</code> will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just <code>0</code> , which will result in fecundity being estimated only for transitions noted in <code>supplement</code> above. Must be the dimensions of an ahistorical matrix.
overwrite	An optional data frame developed with the <code>overwrite()</code> function describing transitions to be overwritten either with given values or with other estimated transitions. Note that this function supplements overwrite data provided in <code>supplement</code> .
data	The original historical demographic data frame used to estimate vital rates (class <code>hfvdata</code>). The original data frame is required in order to initialize years and patches properly.
modelsuite	An optional <code>lefkoMod</code> object holding the vital rate models. If given, then <code>surv_model</code> , <code>obs_model</code> , <code>size_model</code> , <code>repst_model</code> , <code>fec_model</code> , <code>jsurv_model</code> , <code>jobs_model</code> , <code>jsize_model</code> , <code>jrepst_model</code> , <code>paramnames</code> , <code>yearcol</code> , and <code>patchcol</code> 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any survival probability model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any observation probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
size_model	A linear model predicting size. This can be a model of class <code>glm</code> or <code>glmer</code> , both of which require a predicted poisson variable under a log link, or a model of

	class <code>lm</code> or <code>lmer</code> , in which a Gaussian response is assumed. If given, then will overwrite any size model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>repst_model</code>	A linear model predicting reproduction probability. This can be a model of class <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>fec_model</code>	A linear model predicting fecundity. This can be a model of class <code>glm</code> or <code>glmer</code> , and requires a predicted poisson variable under a log link. If given, then will overwrite any fecundity model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>jsurv_model</code>	A linear model predicting juvenile survival probability. This can be a model of class <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile survival probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>jobs_model</code>	A linear model predicting juvenile sprouting or observation probability. This can be a model of class <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile observation probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>jsize_model</code>	A linear model predicting juvenile size. This can be a model of class <code>glm</code> or <code>glmer</code> , both of which require a predicted poisson variable under a log link, or a model of class <code>lm</code> or <code>lmer</code> , in which a Gaussian response is assumed. If given, then will overwrite any juvenile size model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>jrepst_model</code>	A linear model predicting reproduction probability of a mature individual that was immature in the previous year. This can be a model of class <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>paramnames</code>	A dataframe with two columns, the first showing the general model terms that will be used in matrix creation, and the second showing the equivalent terms used in modeling. Only required if <code>modelsuite</code> is not supplied.
<code>inda</code>	A numeric value to use for individual covariate a. Defaults to 0.
<code>indb</code>	A numeric value to use for individual covariate b. Defaults to 0.
<code>indc</code>	A numeric value to use for individual covariate c. Defaults to 0.
<code>surv_dev</code>	A numeric value to be added to the y-intercept in the linear model for survival probability.
<code>obs_dev</code>	A numeric value to be added to the y-intercept in the linear model for observation probability.
<code>size_dev</code>	A numeric value to be added to the y-intercept in the linear model for size.

repst_dev	A numeric value to be added to the y-intercept in the linear model for probability of reproduction.
fec_dev	A numeric value to be added to the y-intercept in the linear model for fecundity.
jsurv_dev	A numeric value to be added to the y-intercept in the linear model for juvenile survival probability.
jobs_dev	A numeric value to be added to the y-intercept in the linear model for juvenile observation probability.
jsize_dev	A numeric value to be added to the y-intercept in the linear model for juvenile size.
jrepst_dev	A numeric value to be added to the y-intercept in the linear model for juvenile reproduction probability.
repmo	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 <code>modelsuite</code> is supplied.
patchcol	The variable name or column number corresponding to patch in the dataset. Not needed if a <code>modelsuite</code> is supplied.
year.as.random	A logical term indicating whether coefficients for missing patches within vital rate models should be estimated as random intercepts. Defaults to FALSE, in which case missing 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.
final_age	The final age to model in the matrix, where the first age will be age 0.
continue	A logical value designating whether to allow continued survival of individuals going past the final age, using the demographic characteristics of the final age.
randomseed	A numeric value used as a seed to generate random estimates for missing occasion and patch coefficients, if either <code>year.as.random</code> or <code>patch.as.random</code> is set to TRUE. Defaults to <code>set.seed()</code> default.
negfec	A logical value denoting whether fecundity values estimated to be negative should be reset to 0. Defaults to FALSE.
reduce	A logical value denoting whether to remove ahistorical stages associated solely with 0 transitions. These are only removed in cases where the associated row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.
err_check	A logical value indicating whether to add matrices of vital rate probabilities associated with each matrix. Defaults to FALSE.

Value

If all inputs are properly formatted, then this function will return an object of class `lefkMat`, 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 years.

U	A list of survival transition matrices sorted as in A.
F	A list of fecundity matrices sorted as in A.
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 <code>afleko2()</code> , only one population may be analyzed at once, and so <code>pop = NA</code>
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 <code>err_check = TRUE</code> . This is a list of vital rate probability matrices, with 4 columns in the order of survival, observation probability, reproduction probability, and size transition probability.

Notes

This function will yield incorrect estimates if the models utilized incorporate state in 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 `afleko2()` will fail and produce an error.

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

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

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

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", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
" Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
" Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 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)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
minima <- c(0, 0, 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)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5)

lathframeln <- sf_create(sizes = sizevector, stagenames = stagevector,
repstatus = 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$fece2 <- round(lathvertln$fece2)
lathvertln$fece1 <- round(lathvertln$fece1)
lathvertln$fece3 <- round(lathvertln$fece3)

lathmodelsln2 <- modelsearch(lathvertln, historical = FALSE,
approach = "mixed", suite = "main",
vitalrates = c("surv", "obs", "size", "repst", "fec"), juvestimate = "Sd1",
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", "Sd1", "Sd", "Sd1"),
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 = lathframeIn, historical = FALSE)

lathmat2age <- aflefk2(year = "all", patch = "all",
  stageframe = lathframeIn, modelsuite = lathmodelsIn2, data = lathvertIn,
  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 `lefkMat` object, and forms one ahistorical matrix for each stage in time $t-1$.

Usage

```
cond_hmpm(hmpm, matchchoice = NULL, err_check = NULL)
```

Arguments

<code>hmpm</code>	A historical matrix projection model of class <code>lefkMat</code> .
<code>matchchoice</code>	A character denoting whether to use A, U, or F matrices. Defaults to A matrices.
<code>err_check</code>	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 `lefkCondMat` object, with the following elements:

<code>Acond</code>	A multi-level list holding the conditional A matrices derived from the input <code>lefkMat</code> 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.
<code>hstages</code>	A data frame matrix showing the pairing of ahistorical stages used to create historical stage pairs.
<code>ahstages</code>	A data frame detailing the characteristics of associated ahistorical stages.
<code>labels</code>	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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

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

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, NA, "D", "XSm", "Sm", "D",
  "XSm", "Sm", NA, NA),
  eststage2 = c(NA, 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, 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",
  individcol = "individ")

```

```
cypcondmats <- cond_hmpm(cypmatrix3r)
summary(cypcondmats)
```

create_IM

Create lefkoMat Object from Given Input Matrices

Description

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

Usage

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

Arguments

<code>mats</code>	A list of A matrices.
<code>stageframe</code>	A stageframe describing all stages utilized.
<code>hstages</code>	A data frame outlining the order of historical stages, if matrices provided in <code>mats</code> are historical. Defaults to NA.
<code>agestages</code>	A data frame outlining the order of ahistorical age-stages, if age-by-stage matrices are provided.
<code>historical</code>	A logical value indicating whether input matrices are historical or not. Defaults to FALSE.
<code>agebystage</code>	A logical value indicating whether input matrices are ahistorical age-by-stage matrices. If TRUE, then object <code>agestages</code> is required. Defaults to FALSE.
<code>UFdecomp</code>	A logical value indicating whether U and F matrices should be inferred. Defaults to TRUE.
<code>entrystage</code>	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.
<code>poporder</code>	The order of populations in the list supplied in object <code>mats</code> . 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_IM()`
`delete_IM()`
`subset_IM()`

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("Sd1", "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, X03, X04, X05, 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_IM(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("Sd1", "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	<i>Demographic Dataset of <i>Cypripedium candidum</i> Population, in Horizontal Format</i>
---------	--

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 referring to patch within the population.

sensor 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

```

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

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

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 <- rlfko2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsupp2r,
  yearcol = "year2", patchcol = "patchid", individcol = "individ")

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

lambda3(cypmatrix2r)

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

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

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 <- rlfko2(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)

```

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 referring to patch within the population.

sensor 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",
  fecca2col = "Pod.2", fecca3col = "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 <- rlefk2(data = cypraw_v2, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2"),
  size = c("size3added", "size2added"), supplement = cypsupp2r,
  yearcol = "year2", patchcol = "patchid", individcol = "individ")

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

lambda3(cypmatrix2r)

```

delete_IM

Delete Matrices from lefkoMat Object

Description

Function `delete_IM()` deletes matrices from `lefkoMat` objects.

Usage

```
delete_IM(IM, 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 <i>*t*</i> 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 delete_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("Sd1", "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, X03, X04, X05, 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_IM(mats_list, anthframe, hstages = NA, historical = FALSE,
  poporder = 1, patchorder = pch_ord, yearorder = yr_ord)

smaller_anth_lm <- delete_IM(anth_lefkoMat, patch = 3)
smaller_anth_lm

```

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

<code>mats</code>	A <code>lefkoMat</code> object, or population projection matrix, for which the stable stage distribution is desired.
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "indiv")

ehrlen3mean <- lmean(ehrlen3)
elasticity3(ehrlen3mean)

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

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

<code>mats</code>	An object of class <code>lefkoMat</code> .
<code>stochastic</code>	A logical value determining whether to conduct a deterministic (FALSE) or stochastic (TRUE) elasticity analysis. Defaults to FALSE.
<code>steps</code>	The number of occasions to project forward in stochastic simulation. Defaults to 10,000.
<code>time_weights</code>	Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among occasions.
<code>sparse</code>	A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
<code>append_mats</code>	A logical value indicating whether to include the original A, U, and F matrices in the output <code>lefkoElas</code> object.
<code>...</code>	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 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 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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

elasticity3(ehrlen3, stochastic = TRUE)

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

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

<code>mats</code>	A list of objects of class <code>matrix</code> .
<code>stochastic</code>	A logical value determining whether to conduct a deterministic (FALSE) or stochastic (TRUE) elasticity analysis. Defaults to FALSE.
<code>steps</code>	The number of occasions to project forward in stochastic simulation. Defaults to 10,000.
<code>time_weights</code>	Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among occasions.
<code>historical</code>	A logical value denoting whether the input matrices are historical. Defaults to FALSE.
<code>sparse</code>	A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
<code>append_mats</code>	A logical value indicating whether to include the original matrices input as object <code>mats</code> in the output <code>lefkoElas</code> object.
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

elasticity3(ehrlen3$A, stochastic = TRUE)

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

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

<code>mats</code>	An object of class <code>matrix</code> .
<code>sparse</code>	A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
<code>...</code>	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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
```

```

stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

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

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

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

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 = cypsuff2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

elasticity3(cypmatrix2r$A[[1]])

```

flefko2

Create Function-based Ahistorical Matrix Projection Model

Description

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

Usage

```

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

```

```

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

```

Arguments

year	A variable corresponding to observation occasion, or a set of such values, given in values associated with the year term used in linear model development. Can also equal <code>all</code> , in which case matrices will be estimated for all years. Defaults to <code>all</code> .
patch	A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to <code>all</code> if matrices should be estimated for all patches. Defaults to <code>all</code> .
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 <code>lefkoSD</code> 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 <code>supplemental()</code> function. Can be used in place of or in addition to an overwrite table (see <code>overwrite</code> below) and a reproduction matrix (see <code>repmat</code> below).
repmat	A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in <code>supplement</code> . If left blank, then <code>flefko2()</code> will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just <code>0</code> , which will result in fecundity being estimated only for transitions noted in <code>supplement</code> above. Must be the dimensions of an ahistorical matrix.
overwrite	An optional data frame developed with the <code>overwrite()</code> function describing transitions to be overwritten either with given values or with other estimated

transitions. Note that this function supplements overwrite data provided in supplement.

data	The original historical demographic data frame used to estimate vital rates (class <code>hfvdata</code>). The original data frame is required in order to initialize years and patches properly.
modelsuite	An optional <code>lefkMod</code> object holding the vital rate models. If given, then <code>surv_model</code> , <code>obs_model</code> , <code>size_model</code> , <code>repst_model</code> , <code>fec_model</code> , <code>jsurv_model</code> , <code>jobs_model</code> , <code>jsize_model</code> , <code>jrepst_model</code> , <code>paramnames</code> , <code>yearcol</code> , and <code>patchcol</code> 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any survival probability model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any observation probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
size_model	A linear model predicting size. This can be a model of class <code>glm</code> or <code>glmer</code> , both of which require a predicted poisson variable under a log link, or a model of class <code>lm</code> or <code>lmer</code> , in which a Gaussian response is assumed. If given, then will overwrite any size model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted poisson variable under a log link. If given, then will overwrite any fecundity model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile survival probability model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile observation probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
jsize_model	A linear model predicting juvenile size. This can be a model of class <code>glm</code> or <code>glmer</code> , both of which require a predicted poisson variable under a log link, or a model of class <code>lm</code> or <code>lmer</code> , in which a Gaussian response is assumed. If given,

	then will overwrite any juvenile size model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>jrepst_model</code>	A linear model predicting reproduction probability of a mature individual that was immature in the previous year. This can be a model of class <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing only the impacts of occasion t .
<code>paramnames</code>	A dataframe with two columns, the first showing the general model terms that will be used in matrix creation, and the second showing the equivalent terms used in modeling. Only required if <code>modelsuite</code> is not supplied.
<code>inda</code>	A numeric value to use for individual covariate a. Defaults to 0.
<code>indb</code>	A numeric value to use for individual covariate b. Defaults to 0.
<code>indc</code>	A numeric value to use for individual covariate c. Defaults to 0.
<code>surv_dev</code>	A numeric value to be added to the y-intercept in the linear model for survival probability.
<code>obs_dev</code>	A numeric value to be added to the y-intercept in the linear model for observation probability.
<code>size_dev</code>	A numeric value to be added to the y-intercept in the linear model for size.
<code>repst_dev</code>	A numeric value to be added to the y-intercept in the linear model for probability of reproduction.
<code>fec_dev</code>	A numeric value to be added to the y-intercept in the linear model for fecundity.
<code>jsurv_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile survival probability.
<code>jobs_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile observation probability.
<code>jsize_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile size.
<code>jrepst_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile reproduction probability.
<code>repmul</code>	A scalar multiplier of fecundity. Defaults to 1.
<code>yearcol</code>	The variable name or column number corresponding to year in occasion t in the dataset. Not needed if a <code>modelsuite</code> is supplied.
<code>patchcol</code>	The variable name or column number corresponding to patch in the dataset. Not needed if a <code>modelsuite</code> is supplied.
<code>year.as.random</code>	A logical term indicating whether coefficients for missing patches within vital rate models should be estimated as random intercepts. Defaults to <code>FALSE</code> , in which case missing monitoring occasion coefficients are set to 0.
<code>patch.as.random</code>	A logical term indicating whether coefficients for missing patches within vital rate models should be estimated as random intercepts. Defaults to <code>FALSE</code> , in which case missing patch coefficients are set to 0.

randomseed	A numeric value used as a seed to generate random estimates for missing occasion and patch coefficients, if either <code>year.as.random</code> or <code>patch.as.random</code> is set to TRUE. Defaults to <code>set.seed()</code> default.
negfec	A logical value denoting whether fecundity values estimated to be negative should be reset to 0. Defaults to FALSE.
reduce	A logical value denoting whether to remove ahistorical stages associated solely with 0 transitions. These are only removed in cases where the associated row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.
err_check	A logical value indicating whether to add matrices of vital rate probabilities associated with each matrix. Defaults to FALSE.

Value

If all inputs are properly formatted, then this function will return an object of class `flefkoMat`, 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 years. All matrices output in the <code>matrix</code> class.
U	A list of survival transition matrices sorted as in A. All matrices output in the <code>matrix</code> class.
F	A list of fecundity matrices sorted as in A. All matrices output in the <code>matrix</code> 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 <code>aflefko2()</code> .
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 <code>flefko2()</code> , only one population may be analyzed at once, and so <code>pop = NA</code>
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 <code>modelsuite</code> input.
prob_out	An optional element only added if <code>err_check = TRUE</code> . This is a list of vital rate probability matrices, with 4 columns in the order of survival, observation probability, reproduction probability, and size transition probability.

Notes

This function will yield incorrect estimates if the models utilized incorporate state in 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 occasions wish to estimate MPMs using a dataset incorporating multiple patches or subpopulations. Should the aim of analysis be a general MPM that does not distinguish these patches or subpopulations, the `patchcol` variable should be left to `NA`, which is the default.

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

Using the `err_check` option will produce a matrix of 4 columns, each characterizing a different vital rate. The product of each row yields an element in the associated $\$U$ 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", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
" Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
" Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 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)

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

lathvertIn <- 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 = lathframeIn,
```

```

stagesize = "size", 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 = "Sd1",
  bestfit = "AICc&k", sizedist = "gaussian", fecdist = "poisson",
  indiv = "indiv", 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", "Sd1", "Sd", "Sd1"),
  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 = lathframeIn, historical = FALSE)

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

summary(lathmat2ln)

```

flefko3

Create Function-based Historical Matrix Projection Model

Description

Function `flefko3()` returns function-based historical MPMs corresponding to the patches and years given, including the associated component transition and fecundity matrices, data frames detailing the characteristics of the ahistorical stages used and historical stage pairs created, and a data frame characterizing the patch and year combinations corresponding to these matrices. Unlike `rflefko3()`, this function currently does not distinguish populations within the same dataset.

Usage

```

flefko3(
  year = "all",
  patch = "all",
  stageframe,
  supplement = NA,

```

```

repmatrix = NA,
overwrite = NA,
data = NA,
modelsuite = NA,
surv_model = NA,
obs_model = NA,
size_model = NA,
repst_model = NA,
fec_model = NA,
jsurv_model = NA,
jobs_model = NA,
jsize_model = NA,
jrepst_model = NA,
paramnames = NA,
inda = 0,
indb = 0,
indc = 0,
surv_dev = 0,
obs_dev = 0,
size_dev = 0,
repst_dev = 0,
fec_dev = 0,
jsurv_dev = 0,
jobs_dev = 0,
jsize_dev = 0,
jrepst_dev = 0,
repmode = 1,
yearcol = NA,
patchcol = NA,
year.as.random = FALSE,
patch.as.random = FALSE,
randomseed = NA,
negfec = FALSE,
format = "ehrlen",
reduce = FALSE,
err_check = FALSE
)

```

Arguments

year	A variable corresponding to the observation occasion, or a set of such values, given in values associated with the year term used in linear model development. Can also equal <code>all</code> , in which case matrices will be estimated for all years. Defaults to <code>all</code> .
patch	A variable designating which patches or subpopulations will have matrices estimated. Should be set to specific patch names, or to <code>all</code> if matrices should be estimated for all patches. Defaults to <code>all</code> .
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 <code>flefkoSD</code> 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 <code>supplemental()</code> function. Can be used in place of or in addition to an <code>overwrite</code> table (see <code>overwrite</code> below) and a reproduction matrix (see <code>repmatrix</code> below).
repmatrix	A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in <code>supplement</code> . If left blank, then <code>flefko3()</code> will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in <code>supplement</code> 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 <code>overwrite()</code> function describing transitions to be overwritten either with given values or with other estimated transitions. Note that this function supplements <code>overwrite</code> data provided in <code>supplement</code> .
data	The historical vertical demographic data frame used to estimate vital rates (class <code>hfvdata</code>), which is required to initialize years and patches properly.
modelsuite	An optional <code>flefkoMod</code> object holding the vital rate models. If given, then <code>surv_model</code> , <code>obs_model</code> , <code>size_model</code> , <code>repst_model</code> , <code>fec_model</code> , <code>jsurv_model</code> , <code>jobs_model</code> , <code>jsize_model</code> , <code>jrepst_model</code> , <code>paramnames</code> , <code>yearcol</code> , and <code>patchcol</code> 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any survival probability model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any observation probability model given in <code>modelsuite</code> . 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 size. This can be a model of class <code>glm</code> or <code>glmer</code> , both of which require a predicted poisson variable under a log link, or a model of class <code>lm</code> or <code>lmer</code> , in which a Gaussian response is assumed. If given, then will overwrite any size model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted poisson variable under a log link. If given, then will overwrite any fecundity model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile survival probability model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any juvenile observation probability model given in <code>modelsuite</code> . 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 size. This can be a model of class <code>glm</code> or <code>glmer</code> , both of which require a predicted poisson variable under a log link, or a model of class <code>lm</code> or <code>lmer</code> , in which a Gaussian response is assumed. If given, then will overwrite any juvenile size model given in <code>modelsuite</code> . 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 <code>glm</code> or <code>glmer</code> , and requires a predicted binomial variable under a logit link. If given, then will overwrite any reproduction probability model given in <code>modelsuite</code> . This model must have been developed in a modeling exercise testing the impacts of occasions t and $t-1$.
paramnames	A dataframe with two columns, the first showing the general model terms that will be used in matrix creation, and the second showing the equivalent terms used in modeling. Only required if <code>modelsuite</code> is not supplied.
inda	A numeric value to use for individual covariate a. Defaults to 0.
indb	A numeric value to use for individual covariate b. Defaults to 0.
indc	A numeric value to use for individual covariate c. Defaults to 0.
surv_dev	A numeric value to be added to the y-intercept in the linear model for survival probability.
obs_dev	A numeric value to be added to the y-intercept in the linear model for observation probability.
size_dev	A numeric value to be added to the y-intercept in the linear model for size.
repst_dev	A numeric value to be added to the y-intercept in the linear model for probability of reproduction.

<code>fec_dev</code>	A numeric value to be added to the y-intercept in the linear model for fecundity.
<code>jsurv_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile survival probability.
<code>jobs_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile observation probability.
<code>jsize_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile size.
<code>jrepst_dev</code>	A numeric value to be added to the y-intercept in the linear model for juvenile reproduction probability.
<code>repm0d</code>	A scalar multiplier of fecundity. Defaults to 1.
<code>yearcol</code>	The variable name or column number corresponding to year in occasion t in the dataset. Not needed if <code>modelsuite</code> is supplied.
<code>patchcol</code>	The variable name or column number corresponding to patch in the dataset. Not needed if <code>modelsuite</code> is supplied.
<code>year.as.random</code>	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 coefficients corresponding to observation occasions are set to 0.
<code>patch.as.random</code>	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.
<code>randomseed</code>	A numeric value used as a seed to generate random estimates for missing occasion and patch coefficients, if either <code>year.as.random</code> or <code>patch.as.random</code> is set to TRUE. Defaults to <code>set.seed()</code> default.
<code>negfec</code>	A logical value denoting whether fecundity values estimated to be negative should be reset to 0. Defaults to FALSE.
<code>format</code>	A string indicating whether to estimate matrices in <code>ehr1en</code> format or <code>deVries</code> format. The latter adds one extra prior stage to account for the prior state of newborns. Defaults to <code>ehr1en</code> format.
<code>reduce</code>	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.
<code>err_check</code>	A logical value indicating whether to append matrices of vital rate probabilities associated with each matrix. These matrices are developed internally and can be used for error checking. Defaults to FALSE.

Value

If all inputs are properly formatted, then this function will return an object of class `lefk0Mat`, 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 years. All matrices output in the <code>matrix</code> class.
---	--

U	A list of survival transition matrices sorted as in A. All matrices output in the <code>matrix</code> class.
F	A list of fecundity matrices sorted as in A. All matrices output in the <code>matrix</code> 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 <code>aflefko2()</code> .
ahstages	A data frame detailing the characteristics of associated ahistorical stages, in the form of a modified stageframe that includes status as an entry stage through reproduction.
labels	A data frame showing the patch and year of each matrix in order. In <code>flefko3()</code> , only one population may be analyzed at once, and so <code>pop = NA</code> .
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 <code>modelsuite</code> input.
prob_out	An optional element only added if <code>err_check = TRUE</code> . This is a list of vital rate probability matrices, with 4 columns in the order of survival, observation probability, reproduction probability, and size transition probability.

Notes

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

The reproduction matrix (field `repmatrix`) may be supplied as either historical or ahistorical. If provided as ahistorical, then `flefko3()` will assume that all historical transitions involving stages noted for 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.

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 probably also lead to fatal errors.

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

Examples

```

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8,
9)
stagevector <- c("Sd", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
" Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
" Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 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)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 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 = "Sd1",
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", "Sd1", "Sd1", "mat", "Sd", "Sd1"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "Sd1", "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, "Sd1", NA, NA),
eststage1 = c(NA, NA, NA, NA, "Sd1", 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 = lathframe1n, historical = TRUE)

lathmat3ln <- flefko3(year = "all", patch = "all", stageframe = lathframe1n,
  modelsuite = lathmodels1n3, data = lathvert1n, supplement = lathsupp3,
  patchcol = "patchid", yearcol = "year2", year.as.random = FALSE,
  patch.as.random = FALSE, reduce = FALSE)

summary(lathmat3ln)

```

historicalize3	<i>Create Historical Vertical Data Frame from Ahistorical Vertical Data Frame</i>
----------------	---

Description

historicalize3() returns a vertically formatted demographic data frame organized to create historical projection matrices, given a vertically but ahistorically formatted data frame. This data frame is in standard lefko3 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,
  fec2col = 0,
  fec3col = 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

<code>data</code>	The horizontal data file.
<code>popidcol</code>	A variable name or column number corresponding to the identity of the population for each individual.
<code>patchidcol</code>	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.
<code>individcol</code>	A variable name or column number corresponding to the unique identity of each individual.
<code>year2col</code>	A variable name or column number corresponding to occasion t (year or time).
<code>year3col</code>	A variable name or column number corresponding to occasion $t+1$ (year or time).
<code>xcol</code>	A variable name or column number corresponding to the X coordinate of each individual in Cartesian space.
<code>ycol</code>	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 second variable name or column number corresponding to an individual covariate to be used in analysis, in occasion t .
indcovb3col	A second variable name or column number corresponding to an individual covariate to be used in analysis, in occasion $t+1$.
indcovc2col	A third variable name or column number corresponding to an individual covariate to be used in analysis, in occasion t .

indcovc3col	A third variable name or column number corresponding to an 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 (dead = 1, living = 0).
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 (dead = 1, living = 0).
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 (not observed = 1, observed = 0).
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 (not observed = 1, observed = 0).
repstrrel	This is a scalar multiplier to make the variable represented by repstrb2col equivalent to the variable represented by repstra2col. This can be useful if two reproductive status variables have related but unequal units, for example if repstrb2col refers to one-flowered stems while repstra2col refers to two-flowered stems.
fecrel	This is a scalar multiplier that makes the variable represented by fecb2col equivalent to the variable represented by feca2col. This can be useful if two fecundity variables have related but unequal units.
stage2col	Optional variable name or column number corresponding to life history stage in 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 . The <code>historicalize3()</code> function assumes that immature individuals are identified in this variable marked with a number equal to or greater than 1, and that mature individuals are marked as 0 or NA.
juv3col	A variable name or column number that marks individuals in immature stages in occasion $t+1$. The <code>historicalize3()</code> function assumes that immature individuals are identified in this variable marked with a number equal to or greater than 1, and that mature individuals are marked as 0 or NA.

stageassign	The stageframe object identifying the life history model being operationalized. Note that if stage2col is provided, then this stageframe is not utilized in stage designation.
stagesize	A variable name or column number describing which size variable to use in stage estimation. Defaults to NA, and can also take sizea, sizeb, sizec, or sizeadded, depending on which size variable is chosen.
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 <code>modelsearch()</code> , 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 zssignment. This can be useful when a matrix is desired without separation of reproductive and non-reproductive but mature stages of the same size. Only used if stageassign is set to a stageframe. Defaults to FALSE.
reduce	A logical variable determining whether unused variables and some invariant state variables should be removed from the output dataset. Defaults to TRUE.

Value

If all inputs are properly formatted, then this function will output a historical vertical data frame (class `hfvdata`), meaning that the output data frame will have three consecutive years of size and reproductive data per individual per row. This data frame is in standard format for all functions used in `lefko3`, and so can be used without further modification. Note that determination of state in 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 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.
sensor1, sensor2, sensor3	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	Density of individuals per unit designated in spacing. Only given if spacing is not NA.

Notes

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

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 `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",
  fecaa2col = "Pod.2", fecaa3col = "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", individcol = "individ")

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

lambda3(cypmatrix2r)

```

 image3

Create Matrix Image

Description

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

Usage

```
image3(mats, ...)
```

Arguments

<code>mats</code>	A <code>lefkoMat</code> object, or a single projection matrix, for which the dominant eigenvalue is desired.
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
```

```

eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

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

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

data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
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", indivcol = "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 <- rlfeko2(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

<code>mats</code>	A <code>lefkoElas</code> object.
<code>used</code>	A numeric value or vector designating the matrices to plot. Can also take the value "all", which plots all matrices. Defaults to "all".
<code>type</code>	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.
<code>...</code>	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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rleko3(data = lathvert, stageframe = lathframe,
  year = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
  supplement = lathsupp3, yearcol = "year2", indivcol = "indiv")

ehrlen_elas <- elasticity3(ehrlen3)

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

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

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

```

```

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
cypsups2r <- 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 = cypsups2r,
  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

<code>mats</code>	A <code>lefkoMat</code> object.
<code>used</code>	A numeric value or vector designating the matrices to plot. Can also take the value "all", which plots all matrices. Defaults to "all".
<code>type</code>	Character value indicating whether to plot A, U, or F matrices. Defaults to "A".
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "indiv")

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

# Cyripedium 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")

```

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

<code>mats</code>	A <code>lefkoSens</code> object.
<code>used</code>	A numeric value or vector designating the matrices to plot. Can also take the value "all", which plots all matrices. Defaults to "all".
<code>type</code>	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.
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
```

```

eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(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", indivcol = "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
cypsups2r <- 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 <- r1efko2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsups2r,
  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

<code>mats</code>	A list class object.
<code>used</code>	A numeric vector of projection matrices within <code>mats</code> to represent as matrix images. Can also take the text value "all", which will produce images of all matrices. Defaults to "all".
<code>...</code>	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", "Sd1", "VSm", "Sm", "VL", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)

```

```

indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

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

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

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

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

image3(ehrlen3$A, used = 1)

# Cyripedium 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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 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 <- rlefk2(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

<code>mats</code>	A matrix class object.
<code>...</code>	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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

image3(ehrlen3$U[[1]])

# Cypripedium example
data(cypdata)

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

```

```

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

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

cypraw_v1 <- verticalize3(data = cypdata, noyears = 6, firstyear = 2004,
  patchidcol = "patch", individcol = "plantid", blocksize = 4,
  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
cypsups2r <- 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 <- rlefk2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsups2r,
  yearcol = "year2", patchcol = "patchid", individcol = "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", "Sd1", "VSm", "Sm", "VL", "Flo", "Dorm")
repvector <- c(0, 0, 0, 0, 1, 0)
obsvector <- c(0, 1, 1, 1, 1, 1, 0)
matvector <- c(0, 0, 1, 1, 1, 1, 1)
immvector <- c(1, 1, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 100, 11, 103, 3500, 3800, 0.5)

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

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

lathsupp3 <- supplemental(stage3 = c("Sd", "Sd", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
```

```

type = c(1, 1, 1, 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)

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

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

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, 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	<i>Estimate Deterministic Population Growth Rates of lefkoMat Matrices</i>
------------------	--

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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
lambda3(ehrlen3mean)

# Cypridium 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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 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

<code>mats</code>	A population projection matrix of class <code>matrix</code> .
<code>sparse</code>	A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
<code>...</code>	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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(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 <- rlfko2(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 referring to patch within the population.

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

Volume88 Aboveground volume in cubic mm in 1988.

lnVol88 Natural logarithm of Volume88.

FCODE88 Equals 1 if flowering and 0 if not flowering in 1988.

Flow88 Number of flowers in 1988.

Intactseed88 Number of intact mature seeds produced in 1988. Not always an integer, as in some cases seed number was estimated via linear modeling.

Dead1988 Marked as 1 if known to be dead in 1988.

Dormant1988 Marked as 1 if known to be alive but vegetatively dormant in 1988.

- Missing1988** Marked as 1 if not found in 1988.
- Seedling1988** Marked as 1, 2, or 3 if observed as a seedling in year *t*. Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1988, 2 = likely that plant is a seedling in 1988, 3 = probable that plant is a seedling in 1988.
- Volume89** Aboveground volume in cubic mm in 1989.
- InVol89** Natural logarithm of Volume89.
- FCODE89** Equals 1 if flowering and 0 if not flowering in 1989.
- Flow89** Number of flowers in 1989.
- Intactseed89** Number of intact mature seeds produced in 1989. Not always an integer, as in some cases seed number was estimated via linear modeling.
- Dead1989** Marked as 1 if known to be dead in 1989.
- Dormant1989** Marked as 1 if known to be alive but vegetatively dormant in 1989.
- Missing1989** Marked as 1 if not found in 1989.
- Seedling1989** Marked as 1, 2, or 3 if observed as a seedling in year *t*. Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1989, 2 = likely that plant is a seedling in 1989, 3 = probable that plant is a seedling in 1989.
- Volume90** Aboveground volume in mm³ in 1990.
- InVol90** Natural logarithm of Volume90.
- FCODE90** Equals 1 if flowering and 0 if not flowering in 1990.
- Flow90** Number of flowers in 1990.
- Intactseed90** Number of intact mature seeds produced in 1990. Not always an integer, as in some cases seed number was estimated via linear modeling.
- Dead1990** Marked as 1 if known to be dead in 1990.
- Dormant1990** Marked as 1 if known to be alive but vegetatively dormant in 1990.
- Missing1990** Marked as 1 if not found in 1990.
- Seedling1990** Marked as 1, 2, or 3 if observed as a seedling in year *t*. Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1990, 2 = likely that plant is a seedling in 1990, 3 = probable that plant is a seedling in 1990.
- Volume91** Aboveground volume in mm³ in 1991.
- InVol91** Natural logarithm of Volume91.
- FCODE91** Equals 1 if flowering and 0 if not flowering in 1991.
- Flow91** Number of flowers in 1991.
- Intactseed91** Number of intact mature seeds produced in 1991. Not always an integer, as in some cases seed number was estimated via linear modeling.
- Dead1991** Marked as 1 if known to be dead in 1991.
- Dormant1991** Marked as 1 if known to be alive but vegetatively dormant in 1991.
- Missing1991** Marked as 1 if not found in 1991.
- Seedling1991** Marked as 1, 2, or 3 if observed as a seedling in year *t*. Numbers refer to certainty of assignment: 1 = certain that plant is a seedling in 1991, 2 = likely that plant is a seedling in 1991, 3 = probable that plant is a seedling in 1991.

Source

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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "indiv")

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", "Sd1", "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 = "size",
  censorkeep = NA, censor = TRUE)

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

ehrlen3 <- rlfko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

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

lambda3(ehrlen3mean)

```

`lefk3`*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 `lefk3` 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

`lefk3` 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. Ehrlén, and S. Kurokawa. 2021. *lefk3*: 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

<code>mats</code>	A <code>lefkoMat</code> object.
<code>matsout</code>	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:

<code>A</code>	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 <code>U</code> and <code>F</code> matrices. All matrices output in the <code>matrix</code> class.
<code>U</code>	A list of mean survival-transition matrices sorted as in <code>A</code> . All matrices output in the <code>matrix</code> class.
<code>F</code>	A list of mean fecundity matrices sorted as in <code>A</code> . All matrices output in the <code>matrix</code> class.
<code>hstages</code>	A data frame showing the pairing of ahistorical stages used to create historical stage pairs. Given if the MPM is historical.
<code>ahstages</code>	A data frame detailing the characteristics of associated ahistorical stages.
<code>labels</code>	A data frame detailing the order of population, patch, and year of each mean matrix. If <code>pop</code> , <code>patch</code> , or <code>year2</code> are <code>NA</code> in the original <code>labels</code> set, then these will be re-labeled as <code>A</code> , <code>1</code> , or <code>1</code> , respectively.
<code>matrixqc</code>	A short vector describing the number of non-zero elements in <code>U</code> and <code>F</code> 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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

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

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

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
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 <- rlefk2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsups2r,
  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

<code>mats</code>	A <code>lefkMat</code> object, population projection matrix, or list of population projection matrices.
<code>refmats</code>	A reference <code>lefkMat</code> object, or matrix, for use as the control. If missing, then is set to the same object as <code>mats</code> .
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "indiv")

ltre3(ehrlen3)

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

```

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(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 <- rlefk2(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)

```

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

<code>mats</code>	An object of class <code>lefkoMat</code> .
<code>refmats</code>	A reference <code>lefkoMat</code> object, or matrix, for use as the control. If missing, then is set to the same object as <code>mats</code> .
<code>ref</code>	A numeric value indicating which matrix or matrices in <code>refmats</code> to use as the control. The numbers used must correspond to the number of the matrices in the <code>labels</code> element of the associated <code>lefkoMat</code> object. The default setting, <code>NA</code> , uses all entries in <code>refmats</code> .
<code>stochastic</code>	A logical value determining whether to conduct a deterministic (<code>FALSE</code>) or stochastic (<code>TRUE</code>) elasticity analysis. Defaults to <code>FALSE</code> .
<code>steps</code>	The number of occasions to project forward in stochastic simulation. Defaults to 10,000.
<code>burnin</code>	The number of initial steps to ignore in stochastic projection when calculating stochastic elasticities. Must be smaller than <code>steps</code> . Defaults to 3000.
<code>time_weights</code>	Numeric vector denoting the probabilistic weightings of all matrices. Defaults to equal weighting among matrices.
<code>sparse</code>	A string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
<code>rseed</code>	Optional numeric value corresponding to the random seed for stochastic simulation.
<code>append_mats</code>	A logical value denoting whether to include the original <code>A</code> , <code>U</code> , and <code>F</code> matrices in the returned <code>lefkoLTRE</code> object. Defaults to <code>FALSE</code> .
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
```

```

stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

ltre3(ehrlen3, stochastic = TRUE)

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

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

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 <- rlfko2(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()` returns both a best-fit model for each vital rate, and a model table showing all models tested. The final output can be used as input in other functions within this package.

Usage

```

modelsearch(
  data,
  historical = TRUE,
  approach = "mixed",
  suite = "size",
  bestfit = "AICc&k",
  vitalrates = c("surv", "size", "fec"),
  surv = c("alive3", "alive2", "alive1"),
  obs = c("obsstatus3", "obsstatus2", "obsstatus1"),
  size = c("sizea3", "sizea2", "sizea1"),
  repst = c("repstatus3", "repstatus2", "repstatus1"),
  fec = c("feca3", "feca2", "feca1"),
  stage = c("stage3", "stage2", "stage1"),
  indiv = "individ",
  patch = NA,
  year = "year2",
  sizedist = "gaussian",
  fecdist = "gaussian",
  size.zero = FALSE,
  size.trunc = FALSE,
  fec.zero = FALSE,
  fec.trunc = FALSE,
  patch.as.random = TRUE,
  year.as.random = TRUE,
  juvestimate = NA,
  juvsize = FALSE,
  jsize.zero = FALSE,
  jsize.trunc = FALSE,
  fectime = 2,

```

```

  censor = NA,
  age = NA,
  indcovb = NA,
  indcovc = NA,
  show.model.tables = TRUE,
  global.only = FALSE,
  quiet = FALSE
)

```

Arguments

<code>data</code>	The vertical dataset to be used for analysis. This dataset should be of class <code>hfvddata</code> , but can also be a data frame formatted similarly to the output format provided by functions <code>verticalize3()</code> or <code>historicalize3()</code> , as long as all needed variables are properly designated.
<code>historical</code>	A logical variable denoting whether to assess the effects of state in occasion $t-1$ in addition to state in occasion t . Defaults to <code>TRUE</code> .
<code>approach</code>	The statistical approach to be taken for model building. The default is <code>mixed</code> , which uses the mixed model approach utilized in packages <code>lme4</code> and <code>glmmTMB</code> . Other options include <code>glm</code> , which uses <code>lm</code> , <code>glm</code> , <code>glm.nb</code> , and related functions in packages <code>MASS</code> , <code>stats</code> , and <code>pscl</code> .
<code>suite</code>	This describes the global model for each vital rate estimation and has the following possible values: <code>full</code> , includes main effects and all two-way interactions of size and reproductive status; <code>main</code> , includes main effects only of size and reproductive status; <code>size</code> , includes only size (also interactions between size in historical model); <code>rep</code> , includes only reproductive status (also interactions between status in historical model); <code>cons</code> , all vital rates estimated only as y-intercepts. If <code>approach = "glm"</code> and <code>year.as.random = FALSE</code> , then <code>year</code> is also included as a fixed effect, and, in the case of <code>full</code> , included in two-way interactions. Defaults to <code>size</code> .
<code>bestfit</code>	A variable indicating the model selection criterion for the choice of best-fit model. The default is <code>AICc&k</code> , which chooses the best-fit model as the model with the lowest <code>AICc</code> or, if not the same model, then the model that has the lowest degrees of freedom among models with $\Delta AICc \leq 2.0$. Alternatively, <code>AICc</code> may be chosen, in which case the best-fit model is simply the model with the lowest <code>AICc</code> value.
<code>vitalrates</code>	A vector describing which vital rates will be estimated via linear modeling, with the following options: <code>surv</code> , survival probability; <code>obs</code> , observation probability; <code>size</code> , overall size; <code>repst</code> , probability of reproducing; and <code>fec</code> , amount of reproduction (overall fecundity). Defaults to <code>c("surv", "size", "fec")</code> .
<code>surv</code>	A vector indicating the variable names coding for status as alive or dead in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("alive3", "alive2", "alive1")</code> .
<code>obs</code>	A vector indicating the variable names coding for observation status in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("obsstatus3", "obsstatus2", "obsstatus1")</code> .
<code>size</code>	A vector indicating the variable names coding for size in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("sizea3", "sizea2", "sizea1")</code> .

repst	A vector indicating the variable names coding for reproductive status in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("repstatus3", "repstatus2", "repstatus1")</code> .
fec	A vector indicating the variable names coding for fecundity in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("feca3", "feca2", "feca1")</code> .
stage	A vector indicating the variables coding for stage in occasions $t+1$, t , and $t-1$. Defaults to <code>c("stage3", "stage2", "stage1")</code> .
indiv	A variable indicating the variable name coding individual identity. Defaults to <code>indiv</code> .
patch	A variable indicating the variable name coding for patch, where patches are defined as permanent subgroups within the study population. Defaults to <code>NA</code> .
year	A variable indicating the variable coding for observation occasion t . Defaults to <code>year2</code> .
sizedist	The probability distribution used to model size. Options include <code>gaussian</code> for the Normal distribution (default), <code>poisson</code> for the Poisson distribution, and <code>negbin</code> for the negative binomial distribution (quadratic parameterization).
fecdist	The probability distribution used to model fecundity. Options include <code>gaussian</code> for the Normal distribution (default), <code>poisson</code> for the Poisson distribution, and <code>negbin</code> for the negative binomial distribution (quadratic parameterization).
size.zero	A logical variable indicating whether size distribution should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to <code>FALSE</code> .
size.trunc	A logical variable indicating whether size distribution is zero-truncated. Defaults to <code>FALSE</code> . Cannot be <code>TRUE</code> if <code>size.zero = TRUE</code> .
fec.zero	A logical variable indicating whether fecundity distribution should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to <code>FALSE</code> .
fec.trunc	A logical variable indicating whether fecundity distribution is zero-truncated. Defaults to <code>FALSE</code> . Cannot be <code>TRUE</code> if <code>fec.zero = TRUE</code> .
patch.as.random	If set to <code>TRUE</code> and <code>approach = "lme4"</code> , then <code>patch</code> is included as a random factor. If set to <code>FALSE</code> and <code>approach = "glm"</code> , then <code>patch</code> is included as a fixed factor. All other combinations of logical value and <code>approach</code> lead to <code>patch</code> not being included in modeling. Defaults to <code>TRUE</code> .
year.as.random	If set to <code>TRUE</code> and <code>approach = "lme4"</code> , then <code>year</code> is included as a random factor. If set to <code>FALSE</code> , then <code>year</code> is included as a fixed factor. All other combinations of logical value and <code>approach</code> lead to <code>year</code> not being included in modeling. Defaults to <code>TRUE</code> .
juvestimate	An optional variable denoting the stage name of the juvenile stage in the vertical dataset. If not <code>NA</code> , and <code>stage</code> is also given (see below), then vital rates listed in <code>vitalrates</code> other than <code>fec</code> will also be estimated from the juvenile stage to all adult stages. Defaults to <code>NA</code> , 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 <code>FALSE</code> , and is only used if <code>juvestimate</code> does not equal <code>NA</code> .

<code>jsize.zero</code>	A logical variable indicating whether size distribution of juveniles should be zero-inflated. Only applies to Poisson and negative binomial distributions. Defaults to FALSE.
<code>jsize.trunc</code>	A logical variable indicating whether size distribution in juveniles is zero-truncated. Defaults to FALSE. Cannot be TRUE if <code>jsize.zero = TRUE</code> .
<code>fectime</code>	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.
<code>censor</code>	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.
<code>age</code>	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 age x stage matrices.
<code>indcov_a</code>	Vector designating the names in occasions $t+1$, t , and $t-1$ of an individual covariate. Defaults to NA.
<code>indcov_b</code>	Vector designating the names in occasions $t+1$, t , and $t-1$ of an individual covariate. Defaults to NA.
<code>indcov_c</code>	Vector designating the names in occasions $t+1$, t , and $t-1$ of an individual covariate. Defaults to NA.
<code>show.model.tables</code>	If set to TRUE, then includes full modeling tables in the output. Defaults to TRUE.
<code>global.only</code>	If set to TRUE, then only global models will be built and evaluated. Defaults to FALSE.
<code>quiet</code>	If set to TRUE, then model building and selection will proceed without warnings and diagnostic messages being issued. Note that this will not affect warnings and messages generated as models themselves are tested. Defaults to FALSE.

Value

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

<code>survival_model</code>	Best-fit model of the binomial probability of survival from occasion t to occasion $t+1$. Defaults to 1.
<code>observation_model</code>	Best-fit model of the binomial probability of observation in occasion $t+1$ given survival to that occasion. Defaults to 1.
<code>size_model</code>	Best-fit model of size in 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 size in 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 size.
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 immature size.
juv_reproduction_table	Full dredge model table of immature reproduction probability.
criterion	Character variable denoting the criterion used to determine the best-fit model.
qc	Data frame with three variables: 1) Name of vital rate, 2) number of individuals used to model that vital rate, and 3) number of individual transitions used to model that vital rate.

Notes

The mechanics governing model building are fairly robust to errors and exceptions. The function attempts to build global models, and simplifies models automatically should model building

fail. Model building proceeds through the functions `lm()` (GLM with Gaussian response), `glm()` (GLM with Poisson or binomial response), `glm.nb()` (GLM with negative binomial response), `zeroinfl()` (zero-inflated Poisson or negative binomial response), `lmer()` (mixed model with Gaussian response), `glmer()` (mixed model with binomial or Poisson 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 in for the functions and packages mentioned. Package MuMIn is used for model dredging (see `dredge()`), and errors and warnings during dredging can be interpreted using the documentation for that package. Errors occurring during dredging lead to the adoption of the global model as the best-fit, and the user should view all logged errors and warnings to determine the best way to proceed. The `quiet = TRUE` option can be used to silence dredge warnings, but users should note that automated model selection can be viewed as a black box, and so care should be taken to ensure that the models run make biological sense, and that model quality is prioritized.

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

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

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", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
" Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
" Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 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)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

indataset <- c(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 4.6, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5,
0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 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 = "glm", suite = "main",
vitalrates = c("surv", "obs", "size", "repst", "fec"), juvestimate = "Sd1",
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", "Sd1", "Sd1", "mat", "Sd", "Sd1"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "Sd1", "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, "Sd1", NA, NA),
eststage1 = c(NA, NA, NA, NA, "Sd1", 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 = lathframe1n, historical = TRUE)

lathmat31n <- flefko3(year = "all", patch = "all", stageframe = lathframe1n,
  modelsuite = lathmodels1n3, data = lathvert1n, supplement = lathsupp3,
  patchcol = "patchid", yearcol = "year2", year.as.random = FALSE,
  patch.as.random = FALSE, reduce = FALSE)

summary(lathmat31n)

```

 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.
stage2	The name of the stage in occasion t in the transition to be replaced.
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. Use rep if all reproductive stages are to be used, and leave empty or use all if all stages in stageframe are to be used.
eststage3	The name of the stage to replace stage3. Only needed if a transition will be replaced by another estimated transition.
eststage2	The name of the stage to replace stage2. Only needed if a transition will be replaced by another estimated transition.

eststage1	The name of the stage to replace stage1. Only needed if a transition will be replaced by another estimated transition, and the matrix to be estimated is historical.
givenrate	A fixed rate or probability to replace for the transition described by stage3, stage2, and stage1.
type	A vector denoting the kind of transition between 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).

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, NA, "D", "XSm", "Sm", "D", "XSm",
  "Sm"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", "XSm",
  "XSm", "XSm"),
  eststage1 = c(NA, 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", "S"))

cypover3r

```

 projection3

Conduct Population Projection Simulations

Description

Function `projection3()` runs projection simulations. It projects the population forward in time by a user-defined number of occasions, and can perform these projections as replicates. Projections may be deterministic or stochastic. 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. There is no limit to the number of replicates.

Usage

```

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

```

Arguments

`mpm` A matrix projection model of class `lefkMat`, 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.
start_vec	An optional numeric vector denoting the starting stage distribution for the projection. Defaults to a single individual of each stage.
tweights	An optional numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among occasions.

Value

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

projection	A list of matrices showing the total number of individuals per stage per occasion, or showing the former with the projected stage distribution and reproductive value per stage per occasion followed by the total population size per occasion (all row-bound in order). Each matrix corresponds to a different patch or population. If more than 1 replicate is projected, then the results from these replicates will be stacked within the same data frame.
labels	A data frame showing the order of populations and patches in item <code>projection</code> .
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.

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.

Examples

```

# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "VSm", "Sm", "VL", "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 = "size",
  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", "Sd1", "Sd1", "Sd", "Sd1"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep"),
  stage1 = c("Sd", "rep", "Sd", "rep", "all", "all"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054),
  type = c(1, 1, 1, 1, 3, 3), type_t12 = c(1, 2, 1, 2, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe,
  year = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
  repmatrix = lathrepm, supplement = lathsupp3, yearcol = "year2",
  individcol = "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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

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

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, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
  NA, NA),
  eststage2 = c(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, "XSm", "XSm", "XSm", "XSm", "XSm",
  "XSm", NA, NA),
  givenrate = c(0.1, 0.1, 0.2, 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, NA, 0.5, 0.5),
  type = c(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),
  stageframe = cypframe_raw, historical = TRUE)

cypmatrix3r <- rlefk3(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 = "indiv")

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

```

Description

repvalue3() is a generic function that estimates returns the reproductive values of stages in a population projection matrix or a set of matrices. The specifics of estimation vary with the class of input object. This function is made to handle very large and sparse matrices supplied as lefkoMat objects or as individual matrices, and can be used with large historical matrices, IPMs, age x stage matrices, as well as 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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
repvalue3(ehrlen3mean)

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

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

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
cypsups2r <- 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, NA, 0.5, 0.5),
type =c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cyframe_raw, historical = FALSE)

cypmatrix2r <- rlefko2(data = cypraw_v1, stageframe = cyframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsups2r,
  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_1`, 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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
repvalue3(ehrlen3mean)

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

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

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 <- rlefk2(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

<code>mats</code>	A population projection matrix.
<code>sparse</code>	A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
<code>...</code>	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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
repvalue3(ehrlen3mean$A[[1]])

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

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

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 = cypsups2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

repvalue3(cypmatrix2r$A[[1]])

```

rlefko2

Create Raw Ahistorical Matrix Projection Model

Description

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 year associated with each matrix.

Usage

```

rlefko2(
  data,
  stageframe,
  year = "all",
  pop = NA,
  patch = NA,
  censor = FALSE,
  stages = NA,
  alive = c("alive3", "alive2"),
  size = c("sizea3", "sizea2"),
  repst = c("repstatus3", "repstatus2"),
  matst = c("matstatus3", "matstatus2"),
  fec = c("feca3", "feca2"),
  supplement = NA,
  repmatrix = NA,
  overwrite = NA,
  yearcol = NA,
  popcol = NA,
  patchcol = NA,
  indivcol = NA,
  censorcol = NA,

```

```

    censorkeep = 0,
    reduce = FALSE
  )

```

Arguments

data	A vertical demographic data frame, with variables corresponding to the naming conventions in <code>verticalize3()</code> .
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 linear model development. Can also equal <code>all</code> , in which case matrices will be estimated for all years. Defaults to <code>all</code> .
pop	A variable designating which populations will have matrices estimated. Should be set to specific population names, or to <code>all</code> 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 <code>all</code> if matrices should be estimated for all patches. Defaults to <code>all</code> .
censor	If TRUE, then data will be removed according to the variable set in <code>censorcol</code> , 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 <code>stageframe</code> exactly. If left blank, then <code>rlefk3()</code> will attempt to infer stages by matching values of <code>alive</code> , <code>size</code> , <code>repst</code> , and <code>matst</code> to characteristics noted in the associated <code>stageframe</code> .
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 size in occasions $t+1$ and t , respectively. Defaults to <code>c("sizea3", "sizea2")</code> .
repst	A vector of names of variables coding reproductive status in occasions $t+1$ and t , respectively. Defaults to <code>c("repstatus3", "repstatus2")</code> .
matst	A vector of names of variables coding maturity status in occasions $t+1$ and t , respectively. Defaults to <code>c("matstatus3", "matstatus2")</code> . Must be supplied if <code>stages</code> is not provided.
fec	A vector of names of variables coding fecundity in occasions $t+1$ and t , respectively. Defaults to <code>c("feca3", "feca2")</code> .
supplement	An optional data frame of class <code>lefkSD</code> that provides supplemental data that should be incorporated into the MPM. Three kinds of data may be integrated this way: transitions to be estimated via the use of proxy transitions, transition overwrites from the literature or supplemental studies, and transition multipliers for

fecundity. This data frame should be produced using the `supplemental()` function. Can be used in place of or in addition to an overwrite table (see `overwrite` below) and a reproduction matrix (see `repmatrix` below).

<code>repmatrix</code>	A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in <code>supplement</code> . If left blank, then <code>rlefko2()</code> will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in <code>supplement</code> above. Must be the dimensions of an ahistorical matrix.
<code>overwrite</code>	An optional data frame developed with the <code>overwrite()</code> function describing transitions to be overwritten either with given values or with other estimated transitions. Note that this function supplements overwrite data provided in <code>supplement</code> .
<code>yearcol</code>	The variable name or column number corresponding to occasion t in the dataset.
<code>popcol</code>	The variable name or column number corresponding to the identity of the population.
<code>patchcol</code>	The variable name or column number corresponding to patch in the dataset.
<code>indivcol</code>	The variable name or column number coding individual identity.
<code>sensorcol</code>	The variable name or column number denoting the censor status. Only needed if <code>sensor = TRUE</code> .
<code>sensorkeep</code>	The value of the censor variable denoting data elements to keep. Defaults to 0.
<code>reduce</code>	A logical value denoting whether to remove historical stages associated with only zero transitions. These are removed only if all row and column sums in ALL matrices estimated equal 0. Defaults to FALSE.

Value

If all inputs are properly formatted, then this function will return an object of class `lefkoMat`, 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 years. All matrices output in the <code>matrix</code> class.
U	A list of survival transition matrices sorted as in A. All matrices output in the <code>matrix</code> class.
F	A list of fecundity matrices sorted as in A. All matrices output in the <code>matrix</code> 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 <code>aflefko2()</code> .

ahstages	A data frame detailing the characteristics of associated ahistorical stages, in the form of a modified stageframe that includes status as an entry stage through reproduction.
labels	A data frame giving the population, patch, and year of each matrix in order.
matrixqc	A short vector describing the number of non-zero elements in U and F matrices, and the number of annual matrices.
dataqc	A vector showing the numbers of individuals and rows in the vertical dataset used as input.

Notes

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

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

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

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

Examples

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "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", "Sd1", "Sd", "Sd1"),
  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 <- rlefk2(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2"), supplement = lathsupp2, yearcol = "year2",
  individcol = "individ")

summary(ehrlen2)

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

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(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, NA, 0.5, 0.5),
type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefk2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsuff2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cypmatrix2r$A[[1]]

```

rlefk3

Create Raw Historical Matrix Projection Model

Description

rlefk3() 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 year associated with each matrix.

Usage

```

rlefk3(
  data,
  stageframe,
  year = "all",
  pop = NA,
  patch = NA,
  censor = FALSE,
  stages = NA,
  alive = c("alive3", "alive2", "alive1"),
  size = c("sizea3", "sizea2", "sizea1"),
  repst = c("repstatus3", "repstatus2", "repstatus1"),
  matst = c("matstatus3", "matstatus2", "matstatus1"),
  fec = c("feca3", "feca2", "feca1"),
  supplement = NA,
  repmatrix = NA,
  overwrite = NA,
  yearcol = NA,
  popcol = NA,
  patchcol = NA,
  indivcol = NA,
  censorcol = NA,
  censorkeep = 0,
  format = "ehrlen",
  reduce = FALSE,
  err_check = FALSE
)

```

Arguments

data	A vertical demographic data frame, with variables corresponding to the naming conventions in <code>verticalize3()</code> .
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 linear model development. Can also equal <code>all</code> , in which case matrices will be estimated for all years. Defaults to <code>all</code> .
pop	A variable designating which populations will have matrices estimated. Should be set to specific population names, or to <code>all</code> 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 <code>all</code> if matrices should be estimated for all patches. Defaults to <code>all</code> .
sensor	If TRUE, then data will be removed according to the variable set in <code>sensorcol</code> , such that only data with sensor 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 <code>stageframe</code> exactly. If left blank, then <code>rlefko3()</code> will attempt to infer stages by matching values of <code>alive</code> , <code>size</code> , <code>repst</code> , and <code>matst</code> to characteristics noted in the associated <code>stageframe</code> .
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 size in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("sizea3", "sizea2", "sizea1")</code> .
repst	A vector of names of variables coding reproductive status in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("repstatus3", "repstatus2", "repstatus1")</code> .
matst	A vector of names of variables coding maturity status in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("matstatus3", "matstatus2", "matstatus1")</code> . Must be supplied if <code>stages</code> is not provided.
fec	A vector of names of variables coding fecundity in occasions $t+1$, t , and $t-1$, respectively. Defaults to <code>c("feca3", "feca2", "feca1")</code> .
supplement	An optional data frame of class <code>lefkoSD</code> 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 <code>supplemental()</code> function. Can be used in place of or in addition to an <code>overwrite</code> table (see <code>overwrite</code> below) and a reproduction matrix (see <code>repmatrix</code> below).

repmatrix	A reproduction matrix, which is an optional matrix composed mostly of 0s, with non-zero values for each potentially new individual (row) born to each reproductive stage (column). Entries act as multipliers on fecundity, with 1 equaling full fecundity. Fecundity multipliers provided this way supplement rather than replace those provided in supplement. If left blank, then <code>rlefk3()</code> will assume that all stages marked as reproductive produce offspring at 1x that of fecundity estimated in provided linear models, and that fecundity will be into the first stage noted as propagule or immature. To prevent this behavior, input just 0, which will result in fecundity being estimated only for transitions noted in supplement above. May be the dimensions of either a historical or an ahistorical matrix. If the latter, then all stages will be used in occasion $t-1$ for each suggested ahistorical transition.
overwrite	An optional data frame developed with the <code>overwrite()</code> 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.
sensorcol	The variable name or column number denoting the censor status. Only needed if <code>sensor = TRUE</code> .
sensorkeep	The value of the censor variable denoting data elements to keep. Defaults to 0.
format	A string indicating whether to estimate matrices in <code>ehr1en</code> format or <code>deVries</code> format. The latter adds one extra prior stage to account for the prior state of newborns. Defaults to <code>ehr1en</code> format.
reduce	A logical value denoting whether to remove historical stages associated exclusively with zero transitions. These are removed only if all row and column sums in ALL matrices estimated equal 0. Defaults to <code>FALSE</code> .
err_check	A logical value indicating whether to append extra information used in matrix calculation within the output list. Used for development debugging purposes.

Value

If all inputs are properly formatted, then this function will return an object of class `lefkMat`, 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 years. All matrices output in the <code>matrix</code> class.
U	A list of survival transition matrices sorted as in A. All matrices output in the <code>matrix</code> class.
F	A list of fecundity matrices sorted as in A. All matrices output in the <code>matrix</code> 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 <code>aflefk2()</code> .
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 `flefk3()` 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 sub-populations. Should the aim of analysis be a general MPM that does not distinguish these patches or subpopulations, the `patchcol` variable should be left to NA, which is the default.

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

Examples

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk03(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

summary(ehrlen3)

# Cypridium 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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 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, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
  NA, NA),
eststage2 = c(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, "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, 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 <- rlefk3(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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 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)

cypsups2r <- 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 = cypsups2r,
  yearcol = "year2", patchcol = "patchid", individcol = "individ")

sensitivity3(cypmatrix2r)

```

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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
```

```

yearcol = "year2", individcol = "individ")

sensitivity3(ehrlen3, stochastic = TRUE)

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

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

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", individcol = "individ")

sensitivity3(cypmatrix2r)

```

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 lefkoSense, 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 `lefkMat` objects).

See Also

[sensitivity3\(\)](#)
[sensitivity3.lefkMat\(\)](#)
[sensitivity3.matrix\(\)](#)

Examples

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1))
```

```

stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

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

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 <- rlefk2(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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
sensitivity3(ehrlen3mean$A[[1]])

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

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(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, NA, 0.5, 0.5),
  type = c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
  stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefk2(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

sf_create() returns a data frame describing each ahistorical life history stage in the life history model. This data frame can be used as input into MPM creation functions such as [flefk3\(\)](#), where it determines how each stage is treated during matrix estimation.

Usage

```

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

```

```

    roundsize = 5
  )

```

Arguments

sizes	A numeric vector of the typical or representative size of each life history stage.
stagenames	An optional vector of stage names, in the same order as elements in sizes. If an IPM or function-based matrix with many stages is desired, then two stages that occur within the dataset and represent the lower and upper size limits of the IPM must be marked as <code>ipm</code> in this vector. These stages must be mature stages, and should have all characteristics other than size equal. If two or more groups of stages, each with its own characteristics, are to be developed for an IPM, then an even number of stages with two stages marking the minimum and maximum size of each group should be marked, with all other characteristics equal within each group.
repstatus	A vector denoting the binomial reproductive status of each life history stage. Defaults to 1.
obsstatus	A vector denoting the binomial observation status of each life history stage. Defaults to 1, but may be changed for unobservable stages.
propstatus	A vector denoting whether each life history stage is a propagule. Such stages are generally only used in fecundity estimation. Defaults to NA.
immstatus	A vector denoting whether each stage is immature. Must be composed of binomial values if given. Defaults to NA.
matstatus	A vector denoting whether each stage is mature. Must be composed of binomial values if given. Defaults to 1 for all stages defined in sizes.
minage	An optional vector denoting the minimum age at which a stage can occur. Only used in age x stage matrix development. Defaults to NA.
maxage	An optional vector denoting the maximum age at which a stage should occur. Only used in age x stage matrix development. Defaults to NA.
indataset	A vector designating which stages are found within the dataset. While <code>rlefk02()</code> and <code>rlefk03()</code> can use all stages in the input dataset, <code>flefk03()</code> and <code>flefk02()</code> can only handle size-classified stages with non-overlapping combinations of size and reproductive status, plus one immature stage. Stages that do not actually exist within the dataset should be marked as 0 in this vector.
binhalfwidth	A numeric vector giving the half-width of size bins. Required to classify individuals appropriately within size classes. Defaults to 0.5 for all sizes.
comments	An optional vector of text entries holding useful text descriptions of all stages.
ipmbins	If an IPM is desired, then this parameter sets the number of stages to create for that IPM. This number is in addition to any stages that are not size-classified. Defaults to 100, and numbers greater than this yield a warning about the loss of statistical power and increasing chance of matrix over-parameterization resulting from increasing numbers of stages.
roundsize	This parameter sets the precision of size classification, and equals the number of digits used in rounding sizes. Defaults to 5.

Value

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

Variables in this data frame include the following:

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

Examples

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

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

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

data(cypdata)

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(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 <- rlefk2(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,
  size3 = NA,
  size2 = NA,
  obs3 = NA,
  fec = NA,
  repst = NA,
  zisize = TRUE,
  zifec = TRUE
)

```

Arguments

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

size3	The name or column number of the variable corresponding to size in occasion *t+1*.
size2	The name or column number of the variable corresponding to size in occasion *t*. This term is required for both size and fecundity tests.
obs3	The name or column number of the variable corresponding to observation status in 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.
repst	The name or column number of the variable corresponding to reproductive status in occasion *t*. Required if fecundity distribution will be tested.
zysize	A logical value indicating whether to conduct a test of zero inflation in size. Defaults to TRUE.
zifec	A logical value indicating whether to conduct a test of zero inflation in fecundity. Defaults to TRUE.

Value

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

Notes

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

The specific test used for overdispersion is a chi-squared test of the dispersion parameter estimated using a generalized linear model predicting the response given size in 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

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8,
9)
stagevector <- c("Sd", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
" Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
" Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1)
obsvector <- c(0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 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)
```



```

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

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

```

slambda3

Estimate Stochastic Population Growth Rate

Description

Function `slambda3()` estimates the stochastic population growth rate, 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 `lefkMat` objects and lists of full A matrices.

Usage

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

Arguments

<code>mpm</code>	A matrix projection model of class <code>lefkMat</code> , or a list of full matrix projection matrices.
<code>times</code>	Number of occasions to iterate. Defaults to 10,000.
<code>tweights</code>	Numeric vector denoting the probabilistic weightings of annual matrices. Defaults to equal weighting among occasions.

Value

A data frame with the following variables:

<code>pop</code>	The identity of the population.
<code>patch</code>	The identity of the patch.
<code>a</code>	Estimate of stochastic growth rate, estimated as the arithmetic mean of the log population growth rate across simulated occasions.
<code>var</code>	The estimated variance of a .
<code>sd</code>	The standard deviation of a .
<code>se</code>	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 `weights` 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", "Sd1", "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

lathover3 <- overwrite(stage3 = c("Sd", "Sd", "Sd1"),
  stage2 = c("Sd", "Sd", "Sd"), stage1 = c("Sd", "rep", "rep"),
  givenrate = c(0.345, 0.345, 0.054))

ehrlen3 <- rlfko3(data = lathvert, stageframe = lathframe,
  year = c(1989, 1990), stages = c("stage3", "stage2", "stage1"),
  repmatrix = lathrepm, overwrite = lathover3, yearcol = "year2",
  individcol = "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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

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

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, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
  NA, NA),
  eststage2 = c(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, "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, NA, 0.5, 0.5),
  type = c(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),
  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", individcol = "individ")

cypstoch <- slambda3(cypmatrix3r)
cypstoch

```

 stablestage3

Estimate Stable Stage Distribution

Description

stablestage3() is a generic function that returns the stable stage distribution for a population projection matrix or set of matrices. This function is made to handle very large and sparse matrices supplied as lefkoMat objects or as individual matrices, and can be used with large historical matrices, IPMs, age x stage matrices, as well as 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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", individcol = "individ")

ehrlen3mean <- lmean(ehrlen3)
stablestage3(ehrlen3mean)

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

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
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 <- rlfko2(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_1`, 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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "indiv")

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

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

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", individcol = "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

<code>mats</code>	A population projection matrix of class <code>matrix</code> .
<code>sparse</code>	A text string indicating whether to use sparse matrix encoding ("yes") or dense matrix encoding ("no"). Defaults to "auto".
<code>...</code>	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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

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

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

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

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 <- rlefk2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsuff2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

stablestage3(cypmatrix2r$A[[1]])

```

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("Sd1", "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, X03, X04, X05, 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 summary of the object, 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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

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

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, NA, "D", "XSm", "Sm", "D", "XSm", "Sm",
  NA, NA),
  eststage2 = c(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, "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, NA, 0.5, 0.5),
  type = c(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),
  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",
  individcol = "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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

lathsupp2 <- supplemental(stage3 = c("Sd", "Sd1", "Sd", "Sd1"),
  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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
```

```

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

lathsupp2 <- supplemental(stage3 = c("Sd", "Sd1", "Sd", "Sd1"),
  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 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

ehrlen2 <- rlefk2(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 `flefk3()`, `flefk2()`, `rlefk3()`, `rlefk2()`, and `aflefk2()`.

Usage

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

Arguments

object	An object of class <code>lefkoMat</code> .
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
cypsups2r <- 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 = cypsups2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

summary(cypmatrix2r)

```

Description

A function to summarize objects of class `lefkoMod`. This function shows the best-fit models, summarizes the numbers of models in the model tables, shows the criterion used to determine the best-fit models, and provides some basic quality control information.

Usage

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

Arguments

`object` An R object of class `lefkoMod` resulting from `modelsearch()`.
`...` Other parameters.

Value

A summary of the object, showing the best-fit models for all vital rates, with constants of 0 or 1 used for unestimated models. This is followed by a summary of the number of models tested per vital rate, and a table showing the names of the parameters used to model vital rates and represent tested factors. At the end is a section describing the number of individuals and individual transitions used to estimate each vital rate best-fit model.

Examples

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 4.6, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8,
9)
stagevector <- c("Sd", "Sd1", "Dorm", "Sz1nr", "Sz2nr", "Sz3nr", "Sz4nr",
" Sz5nr", "Sz6nr", "Sz7nr", "Sz8nr", "Sz9nr", "Sz1r", "Sz2r", "Sz3r",
" Sz4r", "Sz5r", "Sz6r", "Sz7r", "Sz8r", "Sz9r")
repvector <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 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, 1)
matvector <- c(0, 0, 1, 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, 0)
propvector <- c(1, 0, 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, 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)

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

lathvertIn <- 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),
  sums_out = FALSE,
  ...
)

```

Arguments

<code>object</code>	A <code>lefkoProj</code> object.
<code>threshold</code>	A threshold population size to be searched for in projections. Defaults to 1.
<code>milepost</code>	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 <code>c(0, 0.25, 0.50, 0.75, 1.00)</code> .
<code>sums_out</code>	A logical value indicating whether to output population sums in matrix format, with columns corresponding to time and rows corresponding to replicate. Defaults to <code>FALSE</code>
<code>...</code>	Other parameters.

Value

If `sums_out = FALSE`, then there is no output beyond written statements describing the projection. If `sums_out = TRUE`, then the output is a list with two elements:

```
mat_sums
milepost_sums
```

Notes

If `sums_out = TRUE`, then the output from this function may be used to plot population size by replicate across time. This can enable analyses such as quasi-extinction analysis.

Examples

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep"),
  stage1 = c("Sd", "rep", "Sd", "rep", "all", "all"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054),
  type = c(1, 1, 1, 1, 3, 3), type_t12 = c(1, 2, 1, 2, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlefk3(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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
binvec <- c(0, 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", "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, 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, 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, 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 = cypsups3r, 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

supplemental() provides all necessary supplemental data for matrix estimation, particularly bringing together data on proxy rates, data to overwrite existing rates, identified reproductive transitions complete, and fecundity multipliers.

Usage

```

supplemental(
  stage3,
  stage2,
  stage1 = NA,
  eststage3 = NA,
  eststage2 = NA,
  eststage1 = NA,
  givenrate = NA,
  multiplier = NA,
  type = NA,
  type_t12 = NA,
  stageframe,
  historical = TRUE
)

```

Arguments

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

stage2	The name of the stage in occasion t in the transition to be replaced. Abbreviations for groups of stages are also useable (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 useable (see Notes).
eststage3	The name of the stage to replace stage3. Only needed if a transition will be replaced by another estimated transition.
eststage2	The name of the stage to replace stage2. Only needed if a transition will be replaced by another estimated transition.
eststage1	The name of the stage to replace stage1. Only needed if a transition will be replaced by another estimated transition, and the matrix to be estimated is historical. Stage 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, and NA entries for all other terms.
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 `lefkSD`. This object can be used as input in `flefk3()`, `flefk2()`, `rlefk3()`, `rlefk2()`, and `aflefk2()`.

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.
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

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.

Examples

```
# Lathyrus example
data(lathyrus)

sizevector <- c(0, 100, 13, 127, 3730, 3800, 0)
stagevector <- c("Sd", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
```

```

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

ehrlen3 <- rlefk3(data = lathvert, stageframe = lathframe, year = "all",
  stages = c("stage3", "stage2", "stage1"), supplement = lathsupp3,
  yearcol = "year2", indivcol = "individ")

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

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

sizevector <- c(0, 0, 0, 0, 0, 0, 1, 2.5, 4.5, 8, 17.5)
stagevector <- c("SD", "P1", "P2", "P3", "SL", "D", "XSm", "Sm", "Md", "Lg",
  "XLg")
repvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
obsvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
matvector <- c(0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(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, NA, 0.5, 0.5),
  type = c(1, 1, 1, 1, 1, 1, 1, 3, 3),
  stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefk2(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

`verticalize3()` returns a vertically formatted demographic data frame organized to create historical projection matrices, given a horizontally formatted input data frame.

Usage

```
verticalize3(  
  data,  
  noyears,  
  firstyear = 1,  
  popidcol = 0,  
  patchidcol = 0,  
  indivcol = 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 = TRUE,
  censor = FALSE,
  coordsRepeat = FALSE,
  spacing = NA,
  NAas0 = FALSE,
  NRasRep = FALSE,
  reduce = TRUE,
  a2check = FALSE
)

```

Arguments

<code>data</code>	The horizontal data file. A valid data frame is required as input.
<code>noyears</code>	The number of years or observation periods in the dataset. A valid integer is required as input.
<code>firstyear</code>	The first year or occasion of observation. Defaults to 1.
<code>popidcol</code>	A variable name or column number corresponding to the identity of the population for each individual.
<code>patchidcol</code>	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.
<code>individcol</code>	A variable name or column number corresponding to the identity of each individual.
<code>blocksize</code>	The number of variables corresponding to each occasion in the input dataset designated in <code>data</code> , 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.
<code>xcol</code>	A variable name(s) or column number(s) corresponding to the X coordinate of each individual, or each individual-occasion combination of the X coordinate, 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.
<code>ycol</code>	A variable name(s) or column number(s) corresponding to the Y coordinate of each individual, or each individual-occasion combination of the Y coordinate, 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.
<code>juvcol</code>	A variable name(s) or column number(s) that marks individuals in immature stages within the dataset. The <code>verticalize3()</code> 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.
<code>sizeacol</code>	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.

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.
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.
repstracol	A variable name(s) or column number(s) corresponding to the production of reproductive structures, such as flowers, associated with the first year or observation period in the input dataset. This can be binomial or count data, and is used to in analysis of the probability of reproduction. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.
repstrbcol	A second variable name(s) or column number(s) corresponding to the production of reproductive structures, such as flowers, associated with the first year or observation period in the input dataset. This can be binomial or count data, and is used to in analysis of the probability of reproduction. Can refer to the first instance, or all instances of these variables. In the latter case, the values should be entered as a vector.
fecacol	A variable name(s) or column number(s) denoting fecundity associated with the first year or observation 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 equal a full vector of all living status variables in temporal order.

deadacol	Variable name(s) or column number(s) providing information on whether an individual is alive at a given 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 equal a full vector of all dead status variables in temporal order.
obsacol	A variable name(s) or column number(s) providing information on whether an individual is in an observable stage at a given 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 equal a full vector of all observation status variables in temporal order.
nonobsacol	A variable name(s) or column number(s) providing information on whether an individual is in an unobservable stage at a given 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 equal a full vector of all non-observation status variables in temporal order.
sensorcol	A variable name(s) or column number(s) corresponding to the first entry of a censor variable, used to distinguish between entries to use and entries not to use, or to designate entries with special issues that require further attention. Can refer to the first instance of a censor status variable in the dataset, or equal a full vector of all censor status variables in temporal order. Can also refer to a single censor status variable used for the entire individual, if <code>singlecensor = TRUE</code> .
repstrrel	This is a scalar multiplier on variable <code>repstrbcol</code> to make it equivalent to <code>repstracol</code> . This can be useful if two reproductive status variables have related but unequal units, for example if <code>repstracol</code> refers to one-flowered stems while <code>repstrbcol</code> refers to two-flowered stems. Defaults to 1.
fecrel	This is a scalar multiplier on variable <code>fecbcol</code> to make it equivalent to <code>fecacol</code> . 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 equal a full vector of all stage identity variables in temporal order.
stageassign	The stageframe object identifying the life history model being operationalized. Note that if <code>stagecol</code> 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 <code>sizea</code> , <code>sizeb</code> , <code>sizec</code> , or <code>sizeadded</code> , depending on which size variable is chosen.
sensorkeep	The value of the censor variable identifying data to be included in analysis. Defaults to 0, but may take any value including NA. Note that if NA is the value to keep, then this function will alter all NAs to 0 values, and all other values to 1, treating 0 as the value to keep.
sensorRepeat	A logical value indicating whether the censor variable is a single column, or whether it repeats across occasion blocks. Defaults to TRUE.
sensor	A logical variable determining whether the output data should be censored using the variable defined in <code>sensorcol</code> . 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 <code>modelsearch()</code> , 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 <code>stageassign</code> 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 on FALSE except to inspect potential errors in the dataset.

Value

If all inputs are properly formatted, then this function will output a historical vertical data frame (class `hfvddata`), 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 $\text{lastseen} - \text{firstseen} + 1$.
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.
sensor1, sensor2, sensor3	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 unobservable stages, observation status (`obsstatus`) might not be inferred properly if a single size variable is used that does not yield sizes greater than 0 in all cases

in which individuals were observed. Such situations may arise, for example, in plants when leaf number is the dominant size variable used, but individuals occasionally occur with inflorescences but no leaves. In this instances, it helps to mark related variables as `sizeb` and `sizec`, because observation status will be interpreted in relation to all 3 size variables. Further analysis can then utilize only a single size variable, of the user's choosing. Similar issues can arise in reproductive status (`repstatus`).

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", "Sd1", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(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", "Sd1", "VSm", "Sm", "VL", "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", "Sd1", "Sd1", "Sd", "Sd1", "mat"),
  stage2 = c("Sd", "Sd", "Sd", "Sd", "rep", "rep", "Sd1"),
  stage1 = c("Sd", "rep", "Sd", "rep", "npr", "npr", "Sd"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "mat"),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "Sd1"),
  eststage1 = c(NA, NA, NA, NA, NA, NA, "NotAlive"),
  givenrate = c(0.345, 0.345, 0.054, 0.054, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, 0.345, 0.054, NA),
  type = c(1, 1, 1, 1, 3, 3, 1), type_t12 = c(1, 2, 1, 2, 1, 1, 1),
  stageframe = lathframe, historical = TRUE)

ehrlen3 <- rlfeko3(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, 0, 1, 1, 1, 1, 1)
immvector <- c(0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)
propvector <- c(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
indataset <- c(0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)
binvec <- c(0, 0, 0, 0, 0, 0.5, 0.5, 1, 1, 2.5, 7)

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

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, NA, 0.5, 0.5),
type =c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefk2(data = cypraw_v1, stageframe = cypframe_raw,
  year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
  size = c("size3added", "size2added"), supplement = cypsups2r,
  yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cyp2mean <- lmean(cypmatrix2r)
cyp2mean

# Cyripedium 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", indivcol = "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)

cypsups2r <- supplemental(stage3 = c("SD", "P1", "P2", "P3", "SL", "D",
  "XSm", "Sm", "SD", "P1"),
  stage2 = c("SD", "SD", "P1", "P2", "P3", "SL", "SL", "SL", "rep",
  "rep"),
  eststage3 = c(NA, NA, NA, NA, NA, NA, "D", "XSm", "Sm", NA, NA),
  eststage2 = c(NA, NA, NA, NA, NA, NA, "XSm", "XSm", "XSm", NA, NA),
  givenrate = c(0.10, 0.20, 0.20, 0.20, 0.25, NA, NA, NA, NA, NA),
  multiplier = c(NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.5, 0.5),
  type =c(1, 1, 1, 1, 1, 1, 1, 1, 3, 3),
  stageframe = cypframe_raw, historical = FALSE)

cypmatrix2r <- rlefk2(data = cypraw_v1, stageframe = cypframe_raw,

```

```
year = "all", patch = "all", stages = c("stage3", "stage2", "stage1"),
size = c("size3added", "size2added"), supplement = cypsups2r,
yearcol = "year2", patchcol = "patchid", indivcol = "individ")

cyp2mean <- lmean(cypmatrix2r)
cyp2mean
```

Index

* **datasets**
 cypdata, 23
 cypvert, 26
 lathyrus, 86

add_lm, 3, 18, 29, 160
aflefko2, 8, 107, 169, 176

cond_hmpm, 15
create_lm, 4, 17, 29, 160
cypdata, 23
cypvert, 26

delete_lm, 4, 18, 28, 160
dredge, 104

elasticity3, 33, 36, 39, 42
elasticity3.lefkoMat, 33, 35, 39, 42
elasticity3.list, 33, 36, 38, 42
elasticity3.matrix, 33, 36, 39, 41

flefko2, 44, 107, 142, 143, 169, 176
flefko3, 50, 107, 141–143, 169, 176

glm, 104
glm.nb, 104
glmer, 104
glmmTMB, 104

historicalize3, 57, 100

image, 64, 67, 69, 71, 74, 76
image3, 64
image3.lefkoElas, 67
image3.lefkoMat, 65, 69
image3.lefkoSens, 71
image3.list, 74
image3.matrix, 65, 76

lambda3, 78, 81, 84
lambda3.lefkoMat, 79, 81, 84

lambda3.matrix, 79, 81, 83
lathyrus, 86
lefko3, 90
lm, 104
lmean, 90
lmer, 104
ltre3, 93, 97
ltre3.lefkoMat, 94, 95

modelsearch, 61, 99, 146, 171, 183

overwrite, 10, 45, 52, 106, 122, 127

projection3, 108

repvalue3, 111, 115, 118
repvalue3.lefkoMat, 112, 114, 117, 118
repvalue3.matrix, 112, 115, 117
rlefko2, 8, 44, 107, 120, 142, 143, 169, 176
rlefko3, 8, 44, 50, 107, 125, 142, 143, 169, 176

sensitivity3, 130, 134, 137, 139
sensitivity3.lefkoMat, 131, 132, 137, 139
sensitivity3.list, 131, 134, 135, 139
sensitivity3.matrix, 131, 134, 137, 139
set.seed, 12, 48, 54
sf_create, 115, 141, 154
sf_distrib, 145
slambda3, 79, 81, 84, 148
stablestage3, 151, 155, 157
stablestage3.lefkoMat, 151, 153, 157
stablestage3.matrix, 151, 155, 156
subset_lm, 4, 18, 29, 159
summary.lefkoCondMat, 163
summary.lefkoElas, 36, 166
summary.lefkoLTRE, 94, 97, 167
summary.lefkoMat, 18, 169
summary.lefkoMod, 170
summary.lefkoProj, 172
supplemental, 10, 45, 52, 122, 126, 175

verticalize3, [100](#), [121](#), [126](#), [179](#)

vglm, [104](#)

zeroinfl, [104](#)