Package: exactLTRE (via r-universe)

October 8, 2026

Title An Exact Method for Life Table Response Experiment (LTRE) Analysis

Version 0.1.2.9000

Description Life Table Response Experiments (LTREs) are a method of comparative demographic analysis. The purpose is to quantify how the difference or variance in vital rates (stage-specific survival, growth, and fertility) among populations contributes to difference or variance in the population growth rate, ``lambda." We provide functions for one-way fixed design and random design LTRE, using either the classical methods that have been in use for several decades, or an fANOVA-based exact method that directly calculates the impact on lambda of changes in matrix elements, for matrix elements and their interactions. The equations and descriptions for the classical methods of LTRE analysis can be found in Caswell (2001, ISBN: 0878930965), and the fANOVA-based exact methods are described in Hernandez et al. (2023) <doi:10.1111/2041-210X.14065>. We also provide some demographic functions, including generation time from Bienvenu and Legendre (2015) <doi:10.1086/681104>. For implementation of exactLTRE where all possible interactions are calculated, we use an operator matrix presented in Poelwijk, Krishna, and Ranganathan (2016) <doi:10.1371/journal.pcbi.1004771>.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports matrixcalc, popdemo

Repository https://chrissy3815.r-universe.dev

Date/Publication 2025-06-29 20:50:37 UTC

RemoteUrl https://github.com/chrissy3815/exactltre

RemoteRef HEAD

RemoteSha 51490816f23a1aa145ac9545aee272cf21cb8c58

Contents

calc_matrix_effects . . . 2
calc_matrix_responses . . . 3
classicalLTRE . . . 4
classicalLTRE_fixed . . . 5
classicalLTRE_random . . . 6
collapse_mat_list . . . 7
cov_matrix . . . 8
exactLTRE . . . 9
exactLTRE_fixed . . . 11
exactLTRE_random . . . 13
fundamental_matrix . . . 14
generation_time . . . 15
lamDiff . . . 15
lamDiff_symmetric . . . 17
lamVar . . . 18
lifespan . . . 19
make.Gmatrix . . . 20
mean_matrix . . . 20
r_nought . . . 21
reMat . . . 22
run_matrix_checks . . . 22
variance_complete . . . 23

Index **25**

`calc_matrix_effects` *Effects of matrix parameters*

Description

Calculate the effects of differences or variance in vital rate entries in a set of matrix population models, for a given set of interaction terms.

Usage

```
calc_matrix_effects(responses, list_ind_vary)
```

Arguments

`responses`	A vector of the responses due to the parameters of a matrix population model and their interactions.
`list_ind_vary`	A list object, where each entry is a vector containing the indices (the combinations of the elements of `ind_vary`, an input parameter) that varied (were *not* held fixed) for the corresponding entry in the `responses` vector.

Value

A vector of effects, which are the responses due to a given interaction order, controlling for the lower-level interaction terms. These are the epsilon terms in an fANOVA decomposition.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
A_all<- collapse_mat_list(list(A1, A2, A3))
nu_var<- calc_matrix_responses(A_all, c(2,6,7,9), FUN=lamVar, maxint="all")
nu_diff<- calc_matrix_responses(list(A1,A2), c(2,6,7,9), FUN=lamDiff_symmetric, maxint="all")
epsilon_var<- calc_matrix_effects(nu_var$nus, nu_var$list_ind_vary)
epsilon_diff<- calc_matrix_effects(nu_diff$nus, nu_diff$list_ind_vary)
```

---

calc_matrix_responses *Responses to changing entries of a matrix*

Description

Calculate the responses of varying certain entries of a matrix. This function takes a set of matrix population models, the indices of parameters that vary in those matrices, and a response function. For example, difference or variance in lambda (the leading eigenvalue of a population projection matrix).

Usage

```
calc_matrix_responses(Aobj, ind_vary, FUN, maxint = "all")
```

Arguments

`Aobj`	An object containing all the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix.
`ind_vary`	A vector containing the column-wise (single-value) indices of the population projection matrices that vary.
`FUN`	The name of the function to be used for calculating responses. For example, `lamVar`, `lamDiff`, and `lamDiff_symmetric`
`maxint`	The maximum interaction order to be evaluated. The default input is "all" but this input can take any integer value. If maxint=3, then the output will include contributions terms up to 3-way interactions.

Value

This returns a list object, with 2 items: (1) a list of the indices varying for each of the responses terms; and (2) a vector of responses.

`list_ind_vary` is a list object, where each entry is a vector containing the indices (the combinations of the elements of `ind_vary`, an input parameter) that varied (were *not* held fixed) for the corresponding entry in the `nu` vector.

`nus` is a vector of responses, calculated using the function provided in `FUN`.

See Also

`lamVar`, `lamDiff`, and `lamDiff_symmetric`

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
A_all<- collapse_mat_list(list(A1, A2, A3))
nu_var<- calc_matrix_responses(A_all, c(2,6,7,9), FUN=lamVar, maxint="all")
nu_diff<- calc_matrix_responses(list(A1,A2), c(2,6,7,9), FUN=lamDiff_symmetric, maxint="all")
```

---

classicalLTRE *Classical LTRE analysis*

Description

Life Table Response Experiments (LTREs) are a method of comparative demographic analysis. The purpose is to quantify how the difference or variance in vital rates (stage-specific survival, growth, and fertility) among populations contributes to difference or variance in the population growth rate, "lambda." The equations and descriptions for the classical methods of LTRE analysis can be found in Caswell's 2001 textbook. The function we provide here can perform a one-way fixed design LTRE, or a random design LTRE.

Usage

```
classicalLTRE(Aobj, method = "random")
```

Arguments

`Aobj`	An object containing all the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix. For one-way fixed design, exactly 2 matrices must be provided, ordered as `[reference matrix, treatment matrix]`. For random design, any set of 2 or more matrices can be provided. The set of matrices passed in to this function must all have the same dimensions.
`method`	Either "random" or "fixed." The default behavior is "random." See details for more information.

Details

Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix. A one-way fixed design LTRE decomposes the difference in lambda due to differences at each position of the matrices. It should be used when the particular treatment levels are of interest. For a one-way fixed design LTRE, exactly 2 matrices must be provided, ordered as `[reference matrix, treatment matrix]`. The matrix of contributions returned from a classical method fixed design LTRE will have the same shape as the provided matrices.

A random design LTRE decomposes the variance in lambda due to variance and covariance in the entries at each position in the matrices. It assumes that the matrices being analyzed come from a population of similar matrices, without the particular treatment levels or population conditions being of interest in themselves. For a random design LTRE, at least 2 matrices must be provided. The matrix of contributions returned from a classical method random design LTRE will include both first-order terms (due to variance) and interaction terms (due to covariance). Therefore, if the provided matrix is 3x3, the matrix of contributions will be 9x9 (the size of the variance-covariance matrix is the square of the size of the original matrix). The contributions of variances are found on the diagonal of the contribution matrix, and the contributions of covariances are symmetric. So the contribution of covariance between two vital rate parameters is the sum of the two corresponding off-diagonal terms.

The equations and descriptions for the classical methods of LTRE analysis can be found in Caswell's 2001 textbook.

Value

A matrix of contributions to variance (random design) or difference (one-way fixed design) in lambda. Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
cont_diff<- classicalLTRE(list(A1,A2), method='fixed') # contributions to the difference in lambda
cont_var<- classicalLTRE(list(A1,A2,A3), method='random') # contributions to the variance of lambda
```

`classicalLTRE_fixed` *Classical LTRE analysis: one-way fixed design*

Description

Life Table Response Experiments (LTREs) are a method of comparative demographic analysis. The purpose is to quantify how the difference or variance in vital rates (stage-specific survival, growth, and fertility) among populations ' contributes to the difference or variance in the population growth rate, "lambda." ' The equations and descriptions for the classical methods of LTRE analysis ' can be found in Caswell's 2001 textbook.

Usage

```
classicalLTRE_fixed(Aref, Atreatment)
```

Arguments

`Aref`	The population projection matrix of the reference population. Depending on the experimental or observational dataset, this may be the control treatment, the first time period, the unharvested population, etc.
`Atreatment`	The population projection matrix of a treatment population.

Details

Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix. A fixed design LTRE decomposes the difference in lambda due to differences at each position of the matrices. For a fixed design LTRE, exactly 2 matrices must be provided, ordered as `[reference matrix, treatment matrix]`. The matrix of contributions returned from a classical method fixed design LTRE will have the same shape as the provided matrices.

In some cases, it may not be obvious how to identify the reference and the treatment matrix. The sum of contributions will be approximately equal to the observed difference in lambda between these two matrices, evaluated as lambda(Atreatment) - lambda(Aref). In cases where it doesn't 'matter' which way you, as a user, input these matrices, it is important to understand how to interpret positive and negative contributions.

The equations and descriptions for the classical methods of LTRE analysis can be found in Caswell's 2001 textbook.

Value

A matrix of contributions to the difference in lambda. Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
cont_diff<- classicalLTRE_fixed(A1, A2) # contributions to the difference in lambda
```

`classicalLTRE_random` *Classical LTRE analysis: random design*

Description

Life Table Response Experiments (LTREs) are a method of comparative demographic analysis. The purpose is to quantify how the difference or variance in vital rates (stage-specific survival, growth, and fertility) among populations contributes to the difference or variance in the population growth rate, "lambda." The equations and descriptions for the classical methods of LTRE analysis can be found in Caswell's 2001 textbook.

Usage

```
classicalLTRE_random(Aobj)
```

Arguments

Aobj	An object containing all the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix. Any set of 2 or more matrices can be provided. The set of matrices passed in must all have the same dimensions.

Details

Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix.

A random design LTRE decomposes the variance in lambda due to variance and covariance in the entries at each position in the matrices. For a random design LTRE, at least 2 matrices must be provided. The matrix of contributions returned from a classical method random design LTRE will include both first-order terms (due to variance) and interaction terms (due to covariance). Therefore, if the provided matrix is 3x3, the matrix of contributions will be 9x9 (the size of the variance-covariance matrix is the square of the size of the original matrix). The contributions of variances are found on the diagonal of the contribution matrix, and the contributions of covariances are symmetric. So the contribution of covariance between two vital rate parameters is the sum of the two corresponding off-diagonal terms.

The equations and descriptions for the classical methods of LTRE analysis can be found in Caswell's 2001 textbook.

Value

A matrix of contributions to variance in lambda.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
# contributions to the variance of lambda
cont_var<- classicalLTRE(list(A1,A2,A3), method='random')
```

---

collapse_mat_list *Collapse a list of matrices*

Description

Collapse a list of square matrices into a matrix where each row contains the column-wise vectorization of one of the original matrices.

Usage

```
collapse_mat_list(Alist)
```

Arguments

Alist	A list of matrices, of any length.

Value

A matrix where each row contains the column-wise vectorization of one of the original matrices.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
A_all<- collapse_mat_list(list(A1, A2, A3))
```

---

cov_matrix *The variance-covariance matrix for a set of population projection matrices*

Description

Calculate the variance-covariance matrix for the vital rates of a series of population projection matrices.

Usage

```
cov_matrix(Aobj)
```

Arguments

Aobj	A matrix where each row is the column-wise vectorization of a population projection matrix.

Details

The diagonal entries of the variance-covariance matrix give the variance of the entries in each index of the matrix. The off-diagonal entries of the variance-covariance matrix give the covariance of entries in each index of the matrix. The variance-covariance matrix is symmetrical, and the covariance of a given pair of vital rates is the sum of the two corresponding indices.

Value

The variance-covariance matrix for the vital rates of the population projection matrices. If the dimensions of each matrix in Aobj are n-by-n, then the variance-covariance matrix will have dimensions n^2-by-n^2.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
A_all<- collapse_mat_list(list(A1, A2, A3))
covmat<- cov_matrix(A_all)
```

`exactLTRE` *Exact LTRE analysis*

Description

Life Table Response Experiments (LTREs) are a method of comparative demographic analysis. The purpose is to quantify how the difference or variance in vital rates (stage-specific survival, growth, and fertility) among populations contributes to difference or variance in the population growth rate, "lambda." The exact method of LTRE is based on the principles of functional ANOVA. The equations and descriptions for the exact method will be published in a forthcoming paper, which we will link to.

Usage

```
exactLTRE(Aobj, method = "random", maxint = "all", fixed.directional = FALSE)
```

Arguments

`Aobj`	An object containing all the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix. For one-way fixed design, exactly 2 matrices must be provided. For a "directional" analyses, one of the provided matrices will serve as the baseline state (for example, a control or standard-of-reference). In this case, the matrices need to be ordered as `[baseline matrix, observed matrix]`. For random design, any set of 2 or more matrices can be provided. The set of matrices passed in must all have the same dimensions.
`method`	Either "random" or "fixed." The default behavior is "random." See details for more information.
`maxint`	The maximum interaction order to be evaluated. The default input is "all" but this input can take any integer value. If maxint=3, then the output will include contributions terms up to 3-way interactions.
`fixed.directional`	A true/false switch that allows the user to specify whether a directional LTRE should be used. The default behavior is to calculate a symmetric LTRE, where the mean matrix is used as the baseline matrix. See details for more guidance.

Details

Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix.

In a one-way fixed design LTRE, the particular treatment levels or conditions faced by a population are of interest, so one-way fixed design LTRE decomposes the difference in lambda due to differences at each position of the matrices. For a fixed design LTRE, exactly 2 matrices must be provided.

A random design LTRE treats the different matrices being compared as random samples from a set of population conditions, without specific focus on the treatments or conditions that each population experienced. This analysis decomposes the variance in lambda due to variance and covariance in the entries at each position in the matrices. For a random design LTRE, at least 2 matrices must be provided.

`fixed.directional=FALSE` is most appropriate for comparisons where none of the matrices are appropriate for use as a baseline or standard-of-reference. For example, if we want to ask which vital rates drive the difference in population growth rate for two populations of fish in separate but similar lakes, we want to use a *symmetric* analysis. In this case, the difference in lambda is decomposed using the mean matrix as the baseline. The decomposition is symmetric, meaning that if the treatment and reference matrix are swapped, the contributions from the vital rates will be equal in magnitude, but positive contributions will become negative and vice versa. In this case, it does not matter which order you provide the matrices in `Aobj`, but interpretation will require that you pay attention to the fact that the sum of contributions will be equal to the observed difference in lambda between the two matrices in `Aobj`, evaluated as lambda(A1) - lambda(A2).

`fixed.directional=TRUE` is most appropriate for comparisons between a control and treatment population in a controlled experiment, or for other cases where one of the populations can serve as a standard of reference (for example, the lowest elevation population, or one near the range center). In this case, the first matrix in `Aobj` is used as the baseline. This is a directional analysis, meaning that if the order in which the two matrices are provided were to be swapped (switching the baseline matrix and other observed matrix), the contributions of the vital rates would change. If you choose a directional analysis, be sure to provide the matrices in `Aobj` ordered as `[baseline matrix, observed matrix]`.

We set `fixed.directional=FALSE` as the default behavior because most population projection models are built with field-collected data rather than controlled experiment data.

Value

This returns a list object, with 3 items: (1) a vector of the matrix indices where the parameters vary between/among the matrices in Aobj; (2) a list of the indices varying for each of the contribution terms provided; (3) a vector of the contribution terms.If the method is "fixed" then these are contributions to the difference in lambda. If the method is "random" then these are the contributions to the variance in lambda.

`indices.varying` is a vector with the indices of parameters that vary. The numeric indices count down the columns of a given population projection matrix. For example, in a 3x3 matrix, the (2,2) position would be identified with a 5.

`varying.indices.list` is a list object, where each entry is a vector containing the indices (matching the `indices.varying` part of the output) that differed or varied for the corresponding entry in the `epsilon` vector.

epsilon is a vector of contributions to the variance or difference in lambda due to the observed values of the various life history parameters. For example, the contribution of adult survival to the Var(lambda) is determined by setting all parameters *except* adult survival to their mean values, and then calculating the variance of lambda across this manipulated set of matrices.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
cont_diff<- exactLTRE(list(A1,A2), method='fixed') # contributions to the difference in lambda
cont_var<- exactLTRE(list(A1,A2,A3), method='random') # contributions to the variance of lambda
```

---

exactLTRE_fixed *Exact LTRE analysis: fixed design*

Description

Life Table Response Experiments (LTREs) are a method of comparative demographic analysis. The purpose is to quantify how the difference or variance in vital rates (stage-specific survival, growth, and fertility) among populations contributes to difference or variance in the population growth rate, "lambda." The exact method of LTRE is based on the principles of functional ANOVA. The equations and descriptions for the exact method will be published in a forthcoming paper, which we will link to.

Usage

```
exactLTRE_fixed(Aobj, maxint = "all", fixed.directional = FALSE)
```

Arguments

Aobj	An object containing all the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix. For fixed design, exactly 2 matrices must be provided, ordered as [reference matrix, treatment matrix].
maxint	The maximum interaction order to be evaluated. The default input is "all" but this input can take any integer value. If maxint=3, then the output will include contributions terms up to 3-way interactions.
fixed.directional	A true/false switch that allows the user to specify whether a directional LTRE should be used. The default behavior is to calculate a symmetric LTRE, where the mean matrix is used as the baseline. See details for more guidance.

Details

Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix. A fixed design LTRE decomposes the difference in lambda due to differences at each position of the matrices. For a fixed design LTRE, exactly 2 matrices must be provided, ordered as [reference matrix, treatment matrix].

fixed.directional=FALSE is most appropriate for comparisons where it is not entirely obvious which population should be the reference and which should be the treatment (for example, when comparing a wet and a dry year). In this case, the difference in lambda is decomposed using the mean matrix as the baseline. The decomposition is symmetric, meaning that if the treatment and reference matrix are swapped, the contributions from the vital rates will be equal in magnitude, but positive contributions will become negative and vice versa.

fixed.directional=TRUE is most appropriate for comparisons between a control and treatment population in a controlled experiment. In this case, the reference matrix is treated as the baseline. This is a directional analysis, meaning that if the reference and treatment matrices were to be swapped, the contributions of the vital rates would change.

We set fixed.directional=FALSE as the default behavior because most population projection models are built with field-collected data rather than controlled experiment data.

Value

This returns a list object, with 3 items: (1) a vector of the matrix indices where the parameters vary between/among the matrices in Aobj; (2) a list of the indices varying for each of the contribution terms provided; (3) a vector of the contribution terms. For fixed design LTRE these are contributions to the difference in lambda.

indices.varying is a vector with the indices of parameters that vary. The numeric indices count down the columns of a given population projection matrix. For example, in a 3x3 matrix, the (2,2) position would be identified with a 5.

varying.indices.list is a list object, where each entry is a vector containing the indices (matching the indices.varying part of the output) that differed or varied for the corresponding entry in the epsilon vector.

epsilon is a vector of contributions to the difference in lambda due to the observed values of the various life history parameters. For example, the contribution to the difference in lambda of adult survival is determined by setting all parameters *except* adult survival to their mean values, and then calculating the difference in lambda in this manipulated set of matrices.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
cont_diff<- exactLTRE_fixed(list(A1,A2), maxint="all") # contributions to the difference in lambda
cont_diff<- exactLTRE_fixed(list(A1,A2), maxint=2) # only first- and second-order terms

# if A1 represents a control and A2 is a treatment:
cont_diff<- exactLTRE_fixed(list(A1,A2), maxint="all", fixed.directional=TRUE)
```

exactLTRE_random *Exact LTRE analysis: random design*

Description

Life Table Response Experiments (LTREs) are a method of comparative demographic analysis. The purpose is to quantify how the difference or variance in vital rates (stage-specific survival, growth, and fertility) among populations contributes to difference or variance in the population growth rate, "lambda." The exact method of LTRE is based on the principles of functional ANOVA. The equations and descriptions for the exact method will be published in a forthcoming paper, which we will link to.

Usage

```
exactLTRE_random(Aobj, maxint = "all")
```

Arguments

`Aobj`	An object containing all the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix. For random design, any set of 2 or more matrices can be provided. The set of matrices passed in must all have the same dimensions.
`maxint`	The maximum interaction order to be evaluated. The default input is "all" but this input can take any integer value. If maxint=3, then the output will include contributions terms up to 3-way interactions.

Details

Lambda is the asymptotic population growth rate, defined as the largest eigenvalue of the population projection matrix. A random design LTRE decomposes the variance in lambda due to variance and covariance in the entries at each position in the matrices. For a random design LTRE, at least 2 matrices must be provided.

Value

This returns a list object, with 3 items: (1) a vector of the matrix indices where the parameters vary between/among the matrices in Aobj; (2) a list of the indices varying for each of the contribution terms provided; (3) a vector of the contribution terms.If the method is "fixed" then these are contributions to the difference in lambda. If the method is "random" then these are the contributions to the variance in lambda.

`indices.varying` is a vector with the indices of parameters that vary. The numeric indices count down the columns of a given population projection matrix. For example, in a 3x3 matrix, the (2,2) position would be identified with a 5.

`varying.indices.list` is a list object, where each entry is a vector containing the indices (matching the `indices.varying` part of the output) that varied for the corresponding entry in the `epsilon` vector.

epsilon is a vector of contributions to the variance in lambda due to the observed values of the various life history parameters. For example, the contribution to the variance in lambda of adult survival is determined by setting all parameters *except* adult survival to their mean values, and then calculating the variance in lambda in this manipulated set of matrices.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
cont_var<- exactLTRE_random(list(A1,A2,A3), maxint='all') # contributions to the variance of lambda
```

---

fundamental_matrix *Fundamental matrix*

Description

Calculate the fundamental matrix. The fundamental matrix, generally referred to as **N**, contains the expected number of timesteps that an individual will spend in each age, stage, or size class of the matrix throughout their lifespan, given their current class. In the fundamental matrix, the current state is the column, and the future states are in the rows. For an age-classified matrix, this should be lower triangular.

Usage

```
fundamental_matrix(Umat)
```

Arguments

Umat	The survival components of the population projection matrix.

Value

The fundamental matrix, containing the expected number of timesteps that an individual will spend in each class of the population, given their current state. This matrix will have the same size as the input matrix.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
U1<- A1
U1[1,3]<- 0
# the upper right corner represents adult fertility in this model. U1, the
# survival matrix, contains all the transitions *except* for fertility.
N1<- fundamental_matrix(U1)
```

generation_time *Generation time*

Description

Generation time has been defined in multiple ways. This calculation comes from Bienvenu and Legendre (2015, The American Naturalist; doi:10.1086/681104). They define the generation time as the average time between two events in the genealogy of the population.

Usage

```
generation_time(Amat, Fmat)
```

Arguments

Amat	The full population projection matrix
Fmat	The fertility elements of the population projection matrix.

Value

The generation time, as a single number, given in the same units as the projection time step. If the projection interval is two weeks, the generation time will be the number of two-week intervals. You may wish to convert to a standard time step, like days or years.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
F1<- matrix(0, nrow=3, ncol=3)
F1[1,3]<- A1[1,3]
#F1 is all zeros, except the upper right corner which matches A1 for adult fertility
T<- generation_time(A1, F1)
```

lamDiff *Difference in lambda*

Description

In population projection matrices, the eigenvalue with the largest magnitude is the asymptotic population growth rate, referred to as lambda. This function calculates the difference in lambda between two population projection matrices, which must have the same dimensions. This function also has the option to hold some of the vital rates at the value in the baseline matrix. The resulting calculation is the difference in lambda when all the non-fixed vital rates are varying. For example, if all the vital rates are held fixed except for adult fertility, then the output is the difference in lambda due to difference in adult fertility. The difference is taken as $baselinematrix - observedmatrix$, and the function assumes that the provided matrices are ordered [baseline, observed].

Usage

```
lamDiff(Aobj, which.fixed = NULL)
```

Arguments

Aobj	An object containing the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix. Exactly 2 matrices should be provided. If more than 2 matrices are provided, the function will assume that the first is the baseline and the second is the observed matrix to be compared. Matrices beyond the first two will be ignored.
which.fixed	The column-wise indices (single-value index) of the vital rates to be held at their baseline values across the matrices in Aobj.

Details

This function differs from lamDiff_symmetric because it uses the first matrix in Aobj as the baseline matrix. So fixed parameters are set to the values in the baseline matrix. In lamDiff_symmetric, the fixed parameters would be set to their mean values.

lamDiff is most appropriate for comparisons between a control and treatment population in a controlled experiment or other settings where one of the populations can be considered as a standard-of-reference. lamDiff_symmetric is more appropriate for comparisons where none of the population matrices are obviously suitable as a baseline or standard-of-reference (for example, when comparing a wet and a dry year).

Value

A single value for the difference in lambda.

See Also

lamDiff_symmetric lamVar

Examples

```
Abaseline<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
Aobserved<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A_all<- list(Abaseline,Aobserved)
diff_all_vary<- lamDiff(A_all)
diff_fert_vary<- lamDiff(A_all, which.fixed=c(2,6,9))
```

`lamDiff_symmetric` *Difference in lambda, with the mean as baseline*

Description

In population projection matrices, the eigenvalue with the largest magnitude is the asymptotic population growth rate, referred to as lambda. This function calculates the difference in lambda between two population projection matrices, which must have the same dimensions. This function also has the option to hold some of the vital rates at their mean values across the provided matrices. The resulting calculation is the difference in lambda when all the non-fixed vital rates are varying. For example, if all the vital rates are held fixed except for adult fertility, then the output is the difference in lambda due to difference in adult fertility. The difference is taken as $observedmatrix1 - observedmatrix2$, where the provided matrices are ordered [observed matrix 1, observed matrix 2].

Usage

```
lamDiff_symmetric(Aobj, which.fixed = NULL)
```

Arguments

`Aobj`	An object containing the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix. Exactly 2 matrices should be provided. If more than 2 matrices are provided, the function will only use the first two.
`which.fixed`	The column-wise indices (single-value index) of the vital rates to be held at their mean values across the matrices in `Aobj`.

Details

This function differs from `lamDiff` because it uses the mean matrix as the baseline. So fixed parameters are set to their mean values. In `lamDiff`, the fixed parameters would be set to their respective values given by the baseline matrix.

`lamDiff` is most appropriate for comparisons between a control and treatment population in a controlled experiment or other settings where one of the populations can be considered as a standard-of-reference. `lamDiff_symmetric` is more appropriate for comparisons where none of the population matrices are obviously suitable as a baseline or standard-of-reference (for example, when comparing a wet and a dry year).

Value

A single value for the difference in lambda.

See Also

`lamDiff` `lamVar`

Examples

```
Aobs1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
Aobs2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A_all<- list(Aobs1,Aobs2)
diff_all_vary<- lamDiff_symmetric(A_all)
diff_fert_vary<- lamDiff_symmetric(A_all, which.fixed=c(2,6,9))
```

---

lamVar *Variance in lambda*

Description

In population projection matrices, the eigenvalue with the largest magnitude is the asymptotic population growth rate, referred to as lambda. This function calculates the variance in lambda among a group of population projection matrices, which must all be the same size. This function also has the option to hold some of the vital rates at their mean values across all the provided matrices. The resulting calculation is the variance of lambda with all the non-fixed vital rates are varying. For example, if all the vital rates are held fixed except for adult fertility, then the output is the variance in lambda due to variance in adult fertility.

Usage

```
lamVar(Aobj, which.fixed = NULL)
```

Arguments

`Aobj`	An object containing all the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix.
`which.fixed`	The column-wise indices (single-value index) of the vital rates to be held at their mean values across all matrices in `Aobj`.

Value

A single value of variance.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
A_all<- collapse_mat_list(list(A1, A2, A3))
var_all_vary<- lamVar(A_all)
var_fert_vary<- lamVar(A_all, which.fixed=c(2,6,9))
```

`lifespan` *Expected lifespan*

Description

The expected lifespan vector is calculated by multiplying a column of ones by the fundamental matrix. The fundamental matrix, generally referred to as **N**, contains the expected number of timesteps that an individual will spend in each age, stage, or size class of the matrix, given their current state.

Usage

```
lifespan(Umat, all_ages = TRUE)
```

Arguments

`Umat`	The survival components of the population projection matrix
`all_ages`	User specifies whether the function should return the expected lifespan remaining for all ages (all_ages="T") or only the expected lifespan at birth (all_ages="F").

Details

The expected lifespan vector contains the expected lifespan remaining for an individual in each age, stage, or size class of the population. If the user requests only the expected lifespan from birth, then only the first entry of the expected lifespan vector is returned.

Value

The expected lifespan is either a vector (if all_ages="T") or a single number for the expected lifespan of a newly born individual (if all_ages="F"). Expected lifespan is given in the same units as the projection time step. If the projection interval is two weeks, the lifespan will be the number of two-week intervals that an individual is expected to survive. You may wish to convert to a standard time step, like days or years.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
U1<- A1
U1[1,3]<- 0
# the upper right corner represents adult fertility in this model. U1, the
# survival matrix, contains all the transitions *except* for fertility.
eta<- lifespan(U1, all_ages=TRUE)
eta_1<- lifespan(U1, all_ages=FALSE) # eta_1 should match the first entry of eta
```

make.Gmatrix *G-matrix*

Description

The G-matrix is the operator used for calculating a vector of effects from a vector of responses, up to arbitrary interaction order. This recursive function for building up the G-matrix is presented in a paper about genetic epistasis from Poelwijk, Krishna, and Ranganathan (2016, PLOS Comp Bio doi:10.1371/journal.pcbi.1004771). When the G-matrix is multiplied on the right by a column vector of the observed responses, it will produce a column vector of the effects. In other words, the G-matrix adds and subtracts off the appropriate lower-order terms to arrive at the correct values of interaction effects.

Usage

```
make.Gmatrix(n)
```

Arguments

n	The number of observed parameters, mutation sites, etc.

Value

A matrix that is 2^n-by-2^n.

Examples

```
Gmat<- make.Gmatrix(3)
```

mean_matrix *Calculate the mean matrix*

Description

Calculate the mean matrix from a list of matrices. In the mean matrix, each element of the matrix is the mean value at that indexed position, across all the provided matrices.

Usage

```
mean_matrix(Aobj)
```

Arguments

Aobj	A list of matrix population models, which must all have the same dimensions.

Value

A single population projection matrix, with the same dimensions as the provided ones, where all vital rate entries are the mean across all provided matrices at the respective matrix index.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
Amean<- mean_matrix(list(A1,A2,A3))
```

---

r_nought *R0, the net reproductive output*

Description

The net reproductive output, R_0, is the expected number of offspring for one individual across their expected lifespan. It is calculated as the largest eigenvalue of the matrix product of the fertility matrix and the fundamental matrix. The fundamental matrix, generally referred to as **N**, contains the expected number of timesteps that an individual will spend in each age, stage, or size class of the matrix.

Usage

```
r_nought(Amat, Fmat)
```

Arguments

Amat	The full population projection matrix
Fmat	The fertility elements of the population projection matrix.

Value

The net reproductive output, a single value, is the number of offspring that an individual is expected to have over their lifespan.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
F1<- matrix(0, nrow=3, ncol=3)
F1[1,3]<- A1[1,3]
#F1 is all zeros, except the upper right corner which matches A1 for adult fertility
R0<- r_nought(A1, F1)
```

reMat *Reconstruct a matrix that was collapsed into a row vector*

Description

Reconstruct a matrix that was collapsed into a row vector. This function assumes that the matrix is square (as population projection matrices are). This function also assumes that the matrix was originally collapsed column-wise.

Usage

```
reMat(vecM, j = NULL)
```

Arguments

vecM	Either a single row containing a vectorized matrix, or a matrix where each row is a column-wise vectorized matrix.
j	Row-index of the target matrix to re-construct, if vecM is a matrix. If vecM is a single row vector, then this input can be neglected.

Value

A single square matrix

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
A1_remat<- reMat(as.vector(A1))
A_all<- rbind(as.vector(A1), as.vector(A2), as.vector(A3))
A3_remat<- reMat(A_all, j=3)
```

run_matrix_checks *Automated checks for a set of matrices*

Description

Run automated checks for a set of population projection matrices. This code will check if the matrices are square, strictly non-negative, ergodic, irreducible, and primitive. The last check is for whether the column sums are greater than 1 for presumed survival terms. For this portion, the code assumes that the first row represents only fertility, and that all other matrix entries represent only survival.

Usage

```
run_matrix_checks(Aobj)
```

Arguments

Aobj	An object containing the population projection matrices to be included in the analysis. It should either be a list, or a matrix where each row is the column-wise vectorization of a matrix.

Value

If all the checks pass, then nothing is returned. If one of the checks fails, then an error or warning message will be returned.

Examples

```
A1<- matrix(data=c(0,0.8,0, 0,0,0.7, 5,0,0.2), nrow=3, ncol=3)
A2<- matrix(data=c(0,0.9,0, 0,0,0.5, 4,0,0.3), nrow=3, ncol=3)
A3<- matrix(data=c(0,0.4,0, 0,0,0.6, 6,0,0.25), nrow=3, ncol=3)
run_matrix_checks(list(A1,A2,A3))
# A couple of examples that would throw errors or warnings:
# run_matrix_checks(c(0,0.8,0, -1,0,0.7, 5,0,0.2)) # has a negative value
# run_matrix_checks(c(0,0.8,0, 0,0.5,0.7, 5,0,0.2)) # has a column-sum greater than 1
```

---

variance_complete *A function for variance assuming a complete sample*

Description

The variance, assuming a complete sample, is the mean of the squared deviations. When you assume an incomplete sample (the standard assumption in statistics), the variance is calculated as the sum of squared deviations, divided by (N-1), where N is the number of observations in the sample. As such, the output of `variance_complete` will always be smaller than the output of `var`.

Usage

```
variance_complete(x)
```

Arguments

x	A numeric vector that represents a complete sample.

Value

The variance of the entries of x, calculated with the assumption that x represents a complete sample. Compare to the output of `var`.

See Also

`var`

Examples

```
test<- c(5, 6, 8, 10, 25)
Vc<- variance_complete(test)
# compare this output with that of var()
```

Index

calc_matrix_effects, 2
calc_matrix_responses, 3
classicalLTRE, 4
classicalLTRE_fixed, 5
classicalLTRE_random, 6
collapse_mat_list, 7
cov_matrix, 8

exactLTRE, 9
exactLTRE_fixed, 11
exactLTRE_random, 13

fundamental_matrix, 14

generation_time, 15

lamDiff, *3*, *4*, 15, *17*
lamDiff_symmetric, *3*, *4*, *16*, 17
lamVar, *3*, *4*, *16*, *17*, 18
lifespan, 19

make.Gmatrix, 20
mean_matrix, 20

r_nought, 21
reMat, 22
run_matrix_checks, 22

var, *23*
variance_complete, 23