Package 'luckieR'

May 9, 2026

Title Calculations of Luck in Structured Population Models

Version 0.1.0

Maintainer Christina Hernandez <`christinahernan@gmail.com`>

Description

User-friendly and generalized tools for the calculation of luck -- moments of variation in metrics like lifespan and lifetime reproductive output. We provide tools for calculating those moments and also performing decompositions into contributions from, for example, individual traits, environmental impacts, and luck (also called individual stochasticity). The functions included here are based on Snyder and Ellner (2024) <doi:10.1086/730557>, Cochran and Ellner (1992) <`https://www.jstor.org/stable/2937115`>, and Hernandez et al. (2024) <doi:10.1111/ele.14390>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports Matrix

URL `https://github.com/chrissy3815/luckieR`

BugReports `https://github.com/chrissy3815/luckieR/issues`

NeedsCompilation no

Author Robin Snyder [aut, cph],
Christina Hernandez [aut, cre],
Steve Ellner [aut]

Repository CRAN

Date/Publication 2026-04-13 11:20:02 UTC

Contents

`calcDistLRO` *Distribution of lifetime reproductive output*

Description

Calculates the distribution of lifetime reproductive output (LRO) the presence of environmental variation. Assumes a pre-breeding census (reproductiion happens before survival and growth).

Usage

```
calcDistLRO(Plist, Flist, Q, c0, maxClutchSize, maxLRO, Fdist = "Poisson")
```

Arguments

`Plist`	A list of survival/growth transition matrices. Plist[[q]][i,j] is the probability of transitioning from state j to state i in environment q.
`Flist`	A list of fecundity matrices. Flist[[q]][i,j] is the expected number of state i offspring from a state j parent in environment q
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

Called by probTraitCondLRO.

It is also possible to calculate the distribution of LRO by cross-classifying states by stage and number of offspring produced so far and calculating the state distribution at death. If s(z') is the survival probability for cross-classified state z', M(z', z) is the probability of transitioning from cross-classified state z to cross-classified state z', and N is the fundamental matrix for M: N = (I - M)^{-1}, then the state distribution at death, conditional on starting in state z, is (1 - s(z')) * N(z', z), where * denotes a Hadamard product (element-by-element multiplication, not matrix multiplication). (See, e.g., eq. 3.2.8 in Data-driven Modeling of Structured Populations: A Practical Guide to the Integral Projection Model, by Stephen P. Ellner, Dylan Z. Childs and Mark Rees, ed. 1, 2015.) However, this implicitly assumes that reproduction happens after survival and growth, i.e. a post-breeding census.

Value

The distribution of lifetime reproductive output

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.2, 0, 0, 0, 0.3, 0, 0, 0.4), 3, 3)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F1[1,] = 0.8*(0:2)
Plist = list (P1, P2)
Flist = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
maxClutchSize = 10
maxLRO=20
out = calcDistLRO (Plist, Flist, Q, c0, maxClutchSize, maxLRO)
```

`calcDistLRONoEnv` *Distribution of lifetime reproductive output*

Description

Calculates the distribution of lifetime reproductive output (LRO) the absence of environmental variation. Assumes a pre-breeding census (reproductiion happens before survival and growth).

Usage

```
calcDistLRONoEnv(Pmat, Fmat, c0, maxClutchSize, maxLRO, Fdist = "Poisson")
```

Arguments

`Pmat`	The survival/growth transition matrix. Pmat[i,j] is the probability of transitioning from state j to state i.
`Fmat`	The fecundity matrix. Fmat[i,j] is the expected number of state i offspring from a state j parent.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

It is also possible to calculate the distribution of LRO by cross-classifying states by stage and number of offspring produced so far and calculating the state distribution at death. If s(z') is the survival probability for cross-classified state z', M(z', z) is the probability of transitioning from cross-classified state z to cross-classified state z', and N is the fundamental matrix for M: N = (I - M)^{-1}, then the state distribution at death, conditional on starting in state z, is (1 - s(z')) * N(z', z), where * denotes a Hadamard product (element-by-element multiplication, not matrix multiplication). (See, e.g., eq. 3.2.8 in Data-driven Modeling of Structured Populations: A Practical Guide to the Integral Projection Model, by Stephen P. Ellner, Dylan Z. Childs and Mark Rees, ed. 1, 2015.) However, this implicitly assumes that reproduction happens after survival and growth, i.e. a post-breeding census.

Value

The distribution of lifetime reproductive output

Examples

```
Pmat = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
c0 = c(1,0,0)
maxClutchSize = 10
maxLRO=20
out = calcDistLRONoEnv (Pmat, Fmat, c0, maxClutchSize, maxLRO)
```

`calcDistLROPostBreeding`

Distribution of lifetime reproductive output

Description

Calculates the distribution of lifetime reproductive output (LRO) the presence of environmental variation. Assumes a post-breeding census (reproductiion happens after survival and growth).

Usage

```
calcDistLROPostBreeding(
  Plist,
  Flist,
  Q,
  c0,
  maxClutchSize,
  maxLRO,
  Fdist = "Poisson"
)
```

Arguments

`Plist`	A list of survival/growth transition matrices. Plist[[q]][i,j] is the probability of transitioning from state j to state i in environment q.
`Flist`	A list of fecundity matrices. Flist[[q]][i,j] is the expected number of state i offspring from a state j parent in environment q
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

Called by probTraitCondLROPostBreeding.

This function calculates the distribution of LRO by cross-classifying states by stage and number of offspring produced so far and calculating the state distribution at death. If s(z') is the survival probability for cross-classified state z', M(z', z) is the probability of transitioning from cross-classified state z to cross-classified state z', and N is the fundamental matrix for M: N = (I - M)^{-1}, then the state distribution at death, conditional on starting in state z, is (1 - s(z')) * N(z', z), where * denotes a Hadamard product (element-by-element multiplication, not matrix multiplication). (See, e.g., eq. 3.2.8 in Data-driven Modeling of Structured Populations: A Practical Guide to the Integral Projection Model, by Stephen P. Ellner, Dylan Z. Childs and Mark Rees, ed. 1, 2015.) This method implicitly assumes that reproduction happens after survival and growth, i.e. a post-breeding census.

Value

The distribution of lifetime reproductive output

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.2, 0, 0, 0, 0.3, 0, 0, 0.4), 3, 3)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F1[1,] = 0.8*(0:2)
Plist = list (P1, P2)
Flist = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
maxClutchSize = 10
maxLRO = 30
out = calcDistLROPostBreeding(Plist, Flist, Q, c0, maxClutchSize, maxLRO)
```

`calcDistLROPostBreedingNoEnv`
Distribution of lifetime reproductive output

Description

Calculates the distribution of lifetime reproductive output (LRO) the absence of environmental variation. Assumes a post-breeding census (reproductiion happens after survival and growth).

Usage

```
calcDistLROPostBreedingNoEnv(
  Pmat,
  Fmat,
  c0,
  maxClutchSize,
  maxLRO,
  Fdist = "Poisson"
)
```

Arguments

`Pmat`	The survival/growth transition matrix. Pmat[i,j] is the probability of transitioning from state j to state i.
`Fmat`	The fecundity matrix. Fmat[i,j] is the expected number of state i offspring from a state j parent.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

This function calculates the distribution of LRO by cross-classifying states by stage and number of offspring produced so far and calculating the state distribution at death. If s(z') is the survival probability for cross-classified state z', M(z', z) is the probability of transitioning from cross-classified state z to cross-classified state z', and N is the fundamental matrix for M: N = (I - M)^{-1}, then the state distribution at death, conditional on starting in state z, is (1 - s(z')) * N(z', z), where * denotes a Hadamard product (element-by-element multiplication, not matrix multiplication). (See, e.g., eq. 3.2.8 in Data-driven Modeling of Structured Populations: A Practical Guide to the Integral Projection Model, by Stephen P. Ellner, Dylan Z. Childs and Mark Rees, ed. 1, 2015.) This method implicitly assumes that reproduction happens after survival and growth, i.e. a post-breeding census.

Value

The distribution of lifetime reproductive output

Examples

```
Pmat = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
maxClutchSize = 10
maxLRO = 30
out = calcDistLROPostBreedingNoEnv (Pmat, Fmat, c0, maxClutchSize,
  maxLRO)
```

`calcDistOffspringCohort` *The distribution of offspring types in a cohort produced at the stable population distribution*

Description

If adults in a population can produce multiple types of offspring, then life can start in multiple states. Therefore, calculations of population-level mean, variance, and skewness in lifespan or lifetime reproductive output must take into account this variation in starting states. We often call this the "mixing distribution" and a reasonable standard mixing distribution is the distribution of offspring types in a cohort produced when the population is at its stable population distribution.

Usage

```
calcDistOffspringCohort(Amat, Fmat)
```

Arguments

`Amat`	An $n \times n$ matrix containing all of the transition and reproductive rates that project the population from time t to $t+1$. Generally given by `Umat+Fmat`.
`Fmat`	An $n \times n$ matrix of per capita rates of reproduction rates by any of the n states into offspring (new individuals of any of the n states)

Value

A vector of length n that contains the proportion of individuals in each of the n states in a cohort of offspring

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
Amat<- Umat+Fmat
mixdist<- calcDistOffspringCohort(Amat, Fmat)
```

`calcMoments` *Calculates the mean, the 2nd and 3rd non-central moments, and the skewness of a lifetime reward*

Description

Given a transition matrix and appropriate reward matrices, calculates the mean, the 2nd and 3rd non-central moments, and the skewness of some quantity (a "reward") over the stochastic trajectories generated by the transition matrix. If the reward matrices are for reproductive output, then this code calculates moments of lifetime reproductive output.

Usage

```
calcMoments(M, R1, R2, R3)
```

Arguments

`M`	The transition matrix
`R1`	A matrix. R1[i,j] is the expected reward (e.g. reproductive output) that occurs when making a transition from state j to state i. Reward matrices include the "dead" state: if there are n states, R1 should be size n+1 x n+1.
`R2`	A matrix. R2[i,j] is the expected squared reward (e.g. reproductive output) that occurs when making a transition from state j to state i. Reward matrices include the "dead" state: if there are n states, R2 should be size n+1 x n+1.
`R3`	A matrix. R3[i,j] is the expected cubed reward (e.g. reproductive output) that occurs when making a transition from state j to state i. Reward matrices include the "dead" state: if there are n states, R3 should be size n+1 x n+1.

Value

A list containing

- rho1Vec: A 1xn matrix whose jth entry gives the expected lifetime reward for a trajectory starting in state j
- mu2Vec: A 1xn matrix whose jth entry gives the central 2nd moment (variance) of lifetime reward for a trajectory starting in state j
- mu3Vec: A 1xn matrix whose jth entry gives the central 3rd moment of lifetime reward for a trajectory starting in state j
- skewnessVec: A 1xn matrix whose jth entry gives the skewness (3rd central moment / variance^3/2) of lifetime reward for a trajectory starting in state j

Examples

```
## Transition matrix: M\[i,j\] is the probability of transitioning
##  from state j to state i
M = matrix (0.1, 3, 3)
## Fecundity matrix: Fmat\[i,j\] is the expected number of size i offspring
## producted in one time step for a parent of size j
Fmat = matrix (0, 3, 3)
Fmat[1,] = c(0, 1, 2)
## Clutch size is Poisson distributed
R1 = matrix (0, 4, 4)
for (j in 1:3)
  R1[,j] = sum(Fmat[,j])
R2 = R1 + R1^2
R3 = R1 + R1^2 + R1^3
out = calcMoments (M, R1, R2, R3)
```

`distLifespanCondR2` *Distribution of lifespan conditional on LRO*

Description

Calculates the distribution of lifespan conditional on LRO in the presence of environmental variation.

Usage

```
distLifespanCondR2(
  Plist,
  Flist,
  Q,
  c0,
  maxClutchSize,
  maxLRO,
  maxAge,
  percentileCutoff = 0.99,
  Fdist = "Poisson"
)
```

Arguments

`Plist`	A list of survival/growth transition matrices. Plist[[q]][i,j] is the probability of transitioning from state j to state i in environment q.
`Flist`	A list of fecundity matrices. Flist[[q]][i,j] is the expected number of state i offspring from a state j parent in environment q
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.

`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`maxAge`	The maximum attainable age
`percentileCutoff`	A value between 0 and 1. Calculations are performed for values of LRO out to this percentile. Optional. The default value is 0.99.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

Assumes a pre-breeding census (reproduction happens before survival and growth). The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2), https://doi.org/10.1086/730557 and its online supplement.

Value

Returns a list containing the following:

- probLifespanCondR: A matrix whose [i,j]th entry is the probability that an individual with LRO = i-1 will have a lifespan of j years (i.e. age j-1).
- distKidsAtDeath: A vector whose jth entry is the probability of having an LRO of j-1.
- sdLifespanCondR: A vector whose jth entry is the standard deviation of lifespan conditional on having an LRO of j-1.
- CVLifespanCondR: A vector whose jth entry is the coefficient of variation of lifespan conditional on having an LRO of j-1.
- maxKidsIndex: calculations were performed out to values of LRO equal to maxKidsIndex-1.
- normalSurvProb: A vector whose jth entry is the unconditional probability of surviving j years (i.e. until age j-1).

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.2, 0, 0, 0, 0.3, 0, 0, 0.4), 3, 3)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F1[1,] = 0.8*(0:2)
Plist = list (P1, P2)
Flist = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
maxClutchSize = 12
maxLRO = 30
maxAge=20
out = distLifespanCondR2 (Plist, Flist, Q, c0, maxClutchSize,
  maxLRO, maxAge)
```

---

distLifespanCondR2NoEnv

Distribution of lifespan conditional on LRO

Description

Calculates the distribution of lifespan conditional on LRO in the absence of environmental variation.

Usage

```
distLifespanCondR2NoEnv(
  Pmat,
  Fmat,
  c0,
  maxClutchSize,
  maxLRO,
  maxAge,
  percentileCutoff = 0.99,
  Fdist = "Poisson"
)
```

Arguments

`Pmat`	The survival/growth transition matrix. Pmat[i,j] is the probability of transitioning from state j to state i.
`Fmat`	The fecundity matrix. Fmat[i,j] is the expected number of state i offspring from a state j parent.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`maxAge`	The maximum attainable age
`percentileCutoff`	A value between 0 and 1. Calculations are performed for values of LRO out to this percentile. Optional. The default value is 0.99.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

Assumes a pre-breeding census (reproduction happens before survival and growth). The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2), https://doi.org/10.1086/730557 and its online supplement. This is a wrapper function for distLifespanCondR2.

Value

Returns a list containing the following:

- probLifespanCondR: A matrix whose [i,j]th entry is the probability that an individual with LRO = i-1 will have a lifespan of j years (i.e. age j-1).
- distKidsAtDeath: A vector whose jth entry is the probability of having an LRO of j-1.
- sdLifespanCondR: A vector whose jth entry is the standard deviation of lifespan conditional on having an LRO of j-1.
- CVLifespanCondR: A vector whose jth entry is the coefficient of variation of lifespan conditional on having an LRO of j-1.
- maxKidsIndex: calculations were performed out to values of LRO equal to maxKidsIndex-1.
- normalSurvProb: A vector whose jth entry is the unconditional probability of surviving j years (i.e. until age j-1).

Examples

```
Pmat = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
c0 = c(1,0,0)
maxClutchSize = 12
maxLRO = 30
maxAge=20
out = distLifespanCondR2NoEnv (Pmat, Fmat, c0, maxClutchSize,
  maxLRO, maxAge)
```

`distLifespanCondR2PostBreeding` *Distribution of lifespan conditional on LRO*

Description

Calculates the distribution of lifespan conditional on LRO in the presence of environmental variation, assuming a post-breeding census.

Usage

```
distLifespanCondR2PostBreeding(
  Plist,
  Flist,
  Q,
  c0,
  maxClutchSize,
  maxLRO,
  maxAge,
  percentileCutoff = 0.99,
  Fdist = "Poisson"
)
```

Arguments

`Plist`	A list of survival/growth transition matrices. Plist[[q]][i,j] is the probability of transitioning from state j to state i in environment q.
`Flist`	A list of fecundity matrices. Flist[[q]][i,j] is the expected number of state i offspring from a state j parent in environment q
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`maxAge`	The maximum attainable age
`percentileCutoff`	A value between 0 and 1. Calculations are performed for values of LRO out to this percentile. Optional. The default value is 0.99.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

Assumes a post-breeding census (reproduction happens after survival and growth). The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2), https://doi.org/10.1086/730557 and its online supplement.

Value

Returns a list containing the following:

- probLifespanCondR: A matrix whose [i,j]th entry is the probability that an individual with LRO = i-1 will have a lifespan of j years (i.e. age j-1).
- distKidsAtDeath: A vector whose jth entry is the probability of having an LRO of j-1.
- sdLifespanCondR: A vector whose jth entry is the standard deviation of lifespan conditional on having an LRO of j-1.
- CVLifespanCondR: A vector whose jth entry is the coefficient of variation of lifespan conditional on having an LRO of j-1.
- maxKidsIndex: calculations were performed out to values of LRO equal to maxKidsIndex-1.
- normalSurvProb: A vector whose jth entry is the unconditional probability of surviving j years (i.e. until age j-1).

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.2, 0, 0, 0, 0.3, 0, 0, 0.4), 3, 3)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F1[1,] = 0.8*(0:2)
Plist = list (P1, P2)
Flist = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
maxClutchSize = 12
maxAge=20
maxLRO = 30
out = distLifespanCondR2PostBreeding (Plist, Flist, Q, c0,
  maxClutchSize, maxLRO, maxAge)
```

`distLifespanCondR2PostBreedingNoEnv`
Distribution of lifespan conditional on LRO

Description

Calculates the distribution of lifespan conditional on LRO in the presence of environmental variation, assuming a post-breeding census.

Usage

```
distLifespanCondR2PostBreedingNoEnv(
  Pmat,
  Fmat,
  c0,
  maxClutchSize,
  maxLRO,
  maxAge,
  percentileCutoff = 0.99,
  Fdist = "Poisson"
)
```

Arguments

`Pmat`	The survival/growth transition matrix. Pmat[i,j] is the probability of transitioning from state j to state i.
`Fmat`	The fecundity matrix. Fmat[i,j] is the expected number of state i offspring from a state j parent.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider

`maxLRO`	The maximum LRO to consider
`maxAge`	The maximum attainable age
`percentileCutoff`	A value between 0 and 1. Calculations are performed for values of LRO out to this percentile. Optional. The default value is 0.99.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

Assumes a post-breeding census (reproduction happens after survival and growth). The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2), https://doi.org/10.1086/730557 and its online supplement. This is a wrapper function for distLifespanCondR2PostBreeding.

Value

Returns a list containing the following:

- probLifespanCondR: A matrix whose [i,j]th entry is the probability that an individual with LRO = i-1 will have a lifespan of j years (i.e. age j-1).
- distKidsAtDeath: A vector whose jth entry is the probability of having an LRO of j-1.
- sdLifespanCondR: A vector whose jth entry is the standard deviation of lifespan conditional on having an LRO of j-1.
- CVLifespanCondR: A vector whose jth entry is the coefficient of variation of lifespan conditional on having an LRO of j-1.
- maxKidsIndex: calculations were performed out to values of LRO equal to maxKidsIndex-1.
- normalSurvProb: A vector whose jth entry is the unconditional probability of surviving j years (i.e. until age j-1).

Examples

```
Pmat = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
c0 = c(1,0,0)
maxClutchSize = 12
maxAge=20
maxLRO = 30
out = distLifespanCondR2PostBreedingNoEnv (Pmat, Fmat, c0, maxClutchSize,
  maxLRO, maxAge)
```

`firstBreed` *Probability of breeding and age at first breeding*

Description

Calculates the probability of breeding at least once, and the mean and variance of age at first breeding conditional on breeding, as a function of initial state. In order to use this function, the model specification must include state-dependent breeding probability, not just state-dependent mean fecundity. The probability of breeding at least once from now until death must be positive for all states. If not, remove those states from the model (can't start in any of them, going into them becomes death) and apply this function to the remaining states.

Usage

```
firstBreed(M0, pb)
```

Arguments

`M0`	An $s \times s$ matrix of transition rates among the s non-breeding states in the population
`pb`	A vector of length s containing the probability of breeding (in the next time step) for each state.

Details

In the `luckieR` package, we also provide the `probRepro()` function, which can be used in scenarios where the distribution of probability of breeding is not known. Instead, `probRepro` infers the probability of breeding by assuming that the mean fecundity (provided in `Fmat`) is the mean of a Poisson, Bernoulli, or fixed reproductive process.

Assumptions and coding follow Ellner et al. (2016) IPM monograph, chapter 3. The state transition matrix M is assumed to have the form M = p_b M_b + (1-p_b) M_0 to allow for costs of reproduction, but M_b and M_0 can be equal if there are no costs. Breeding does not necessarily imply producing any new recruits, for example clutch size conditional on breeding could be Poisson.

Value

A list of:

- `p_breed` A vector of length s containing the probability of breeding before death, given the starting state.
- `mean_age` A vector of length s containing the mean age of first reproduction, given the starting state.
- `var_age` A vector of length s containing the variance of age of first reproduction, given the starting state.

See Also

`probRepro()` if you do not have values for pb, `Rage::mature_age()` which computes the mean age at first entering a state with positive mean fecundity.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
pb<- c(0, 0.1, 0.4) # probability of breeding in the next time step, given current state
```

`flatten` *Convert 4-d array to matrix*

Description

Converts a 4-d transition array, such as the K array produced by make_AxT, to a cross-classified transition matrix ("megamatrix"), such as the A matrix produced by make_AxT

Usage

```
flatten(A4)
```

Arguments

`A4`	The 4-d transition array

Value

The cross-classified transition matrix

See Also

unfold, make_AxT

Examples

```
M = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0.1*(0:2)
B = mk_B (Fmat, Fdist="Bernoulli")
mT = 6
out = make_AxT(B, M, mT)
K = out$K
A = flatten(out$K) ## should equal out$A
```

`fundamental_matrix` *Calculate the fundamental matrix*

Description

The fundamental matrix, $\mathbf{N}$, of a Markov Chain contains the expected number of visits to each of the transient states before transitioning into an absorbing state. It is given by:

$$\mathbf{N} = (\mathbf{I} - \mathbf{P})^{-1},$$

where $\mathbf{P}$ is the matrix of transition probabilities among the transient states, and the power -1 indicates the matrix inverse.

Usage

```
fundamental_matrix(P)
```

Arguments

`P`	An $n \times n$ matrix of transition rates among the n transient states in the matrix population model. If the absorbing state is death, then this matrix is the `Umat` of transition rates among the living stages

Value

An $n \times n$ matrix where each ij entry is the expected number of time steps spent in state i given that an individual starts in state j

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
Nmat<- fundamental_matrix(Umat)
```

`makeCondFailureKernel` *Calculates the probability of not becoming successful before death and the transition kernel conditional on not becoming successful before death.*

Description

Calculates the probability of not hitting a set of absorbing states before dying (i.e. before a stochastic trajectory generated by the transition matrix ends). If the set of absorbing states is defined as "success," then this code calculates the probability of an individual dying before becoming successful. Also calculates the transition kernel conditional on not becoming successful before death.

Usage

```
makeCondFailureKernel(M, transientStates)
```

Arguments

`M`	The unconditional kernel (i.e. transition matrix)
`transientStates`	A vector listing the transient (non-absorbing) states

Value

A list containing

- q2Extended: the probability of not succeeding before death for all states. The probability of not succeeding before death for the successful states is 0 by definition.
- MCond: The kernel conditional on never becoming successful

Examples

```
M = matrix (0.1, 3, 3) ## States are small, medium, and large
## Transient states are small and medium.  "Success" is becoming
## large before you die.
transientStates = 1:2
out = makeCondFailureKernel (M, transientStates)
```

`makeCondKernel` *calculates the probability of becoming successful before death and the transition kernel conditional on becoming successful before death.*

Description

Calculates the probability of hitting a set of absorbing states before dying (i.e. before a stochastic trajectory generated by the transition matrix ends). If the set of absorbing states is defined as "success," then this code calculates the probability of an individual becoming successful before it dies. Also calculates the transition kernel conditional on becoming successful before death.

Usage

```
makeCondKernel(M, transientStates)
```

Arguments

`M`	The unconditional kernel (i.e. transition matrix)
`transientStates`	A vector listing the transient (non-absorbing) states

Value

A list containing

- q2Extended: the probability of success for all states. The probability of success for the successful states is 1 by definition.
- MCond: The kernel conditional on becoming successful

Examples

```
M = matrix (0.1, 3, 3) ## States are small, medium, and large
## Transient states are small and medium.  "Success" is becoming
## large before you die.
transientStates = 1:2
out = makeCondKernel (M, transientStates)
```

`makeM` *Make megamtrix*

Description

Calculates the transition matrix for cross-classified states from the transition matrices for each subclassification: i.e. calculates the megamatrix.

Usage

```
makeM(fastMatrixList, slowMatrix)
```

Arguments

`fastMatrixList`	A list of transition matrices for the state that cycles more rapidly in the cross-classified state. There should be one matrix for each state in the slow matrix. e.g., if the cross-classified state vector is (env. 1, stage 1; env. 1, stage 2; ... env. 1, stage n; env. 2, stage 1, ...), then stage is the fast state, and there should be a survival/growth matrix for each environment or a fecundity matrix for each environment.
`slowMatrix`	A transition matrix for the slow states in the cross-classified state. e.g., if the cross-classified state vector is (env. 1, stage 1; env. 1, stage 2; ... env. 1, stage n; env. 2, stage 1, ...), then environment is the slow state and slowMatrix is the environment transition matrix.

Value

M, the transition matrix for the cross-classified state

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.2, 0, 0, 0, 0.3, 0, 0, 0.4), 3, 3)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F1[1,] = 0.8*(0:2)
Plist = list (P1, P2)
Flist = list (F1, F2)
Q = matrix (1/2, 2, 2)
M = makeM (Plist, Q)
bigF = makeM(Flist, Q)
```

`makeMCondLROThreshold` *Utility function for calculating a kernel conditional on having LRO meeting or exceeding a threshold*

Description

Calculates the transition kernel conditional on producing at least threshold offspring over the course of a life, as well as related quantites. Assumes a post-breeding census (survival and growth happen before reproduction).

Usage

```
makeMCondLROThreshold(
  M,
  Fmat,
  threshold,
  m0,
  maxLRO = 12,
  maxClutchSize = 12,
  Fdist = "Poisson"
)
```

Arguments

`M`	the unconditional survival/growth transition matrix. M[i,j] is the probability of surviving and transitioning from state j to state i.
`Fmat`	the fecundity matrix. Fmat[i,j] is the expected number of size i offspring from a size j parent.
`threshold`	the number of offspring LRO should meet or exceed
`m0`	the state distribution of offspring
`maxLRO`	the maximum LRO is consider. Optional, with a default value of 12.
`maxClutchSize`	the maximum number of offspring born in one reproductive bout. Optional, with a default value of 12.
`Fdist`	the clutch size distribution. Currently supported values are "Poisson" and "Bernoulli." Optional, with a default value of "Poisson".

Details

if there are n sizes/stages, states 1–n are those sizes with LRO = 0 at the start of the time step (just after last year's survival and growth (and environment) updates. States n+1–2n are those sizes with LRO = 0 just after this year's fecundity update but before survival, growth (and environment) updates. States 2n+1–3n are those sizes with LRO = 1 at the start of the time step, and so on through LRO = maxLRO. To advance the state by one time step, it is necessary to multiply by ACondSucceed twice (once to update fecundity, then another time to update survival, growth, and, if necessary, the environment.

Value

A list containing the following. All matrices and vectors are defined over an extended size distribution: see Details.

- ACondSucceed: the size x #kids or size x env x #kids transition matrix conditional on reaching the LRO threshold before dying
- probSucceedCondZ: a vector whose jth entry is the probability of reaching the threshold before dying, conditional on beginning in size/stage j
- probSucceed: the probability of succeeding assuming that an individual's starting state is distributed according to m0
- a0CondSucceed: the offspring state distribution conditional on reaching the LRO threshold before dying.

Examples

```
M = matrix(0.1, 3, 3)
Fmat = matrix(0, 3, 3); Fmat[1,] = 0:2
threshold = 2
m0 = c(1, 0, 0)
out = makeMCondLROThreshold (M, Fmat, threshold, m0)
```

`makeMCondLROThresholdPostBreeding` *Utility function for calculating a kernel conditional on having LRO meeting or exceeding a threshold*

Description

Calculates the transition kernel conditional on producing at least threshold offspring over the course of a life, as well as related quantites. Assumes a post-breeding census (survival and growth happen before reproduction).

Usage

```
makeMCondLROThresholdPostBreeding(
  M,
  Fmat,
  threshold,
  m0,
  maxLRO = 12,
  maxClutchSize = 12,
  Fdist = "Poisson"
)
```

Arguments

`M`	the unconditional survival/growth transition matrix. M[i,j] is the probability of surviving and transitioning from state j to state i.
`Fmat`	the fecundity matrix. Fmat[i,j] is the expected number of size i offspring from a size j parent.
`threshold`	the number of offspring LRO should meet or exceed
`m0`	the state distribution of offspring
`maxLRO`	the maximum LRO is consider. Optional, with a default value of 12.
`maxClutchSize`	the maximum number of offspring born in one reproductive bout. Optional, with a default value of 12.
`Fdist`	the clutch size distribution. Currently supported values are "Poisson" and "Bernoulli." Optional, with a default value of "Poisson".

Value

A list containing the following. All matrices and vectors are defined over an extended size distribution: if there are n sizes/stages, states 1–n are those sizes with LRO = 0, states n+1–2n are those sizes with LRO = 1, and so on through LRO = maxLRO.

- ACondSucceed: the size x #kids or size x env x #kids transition matrix conditional on reaching the LRO threshold before dying
- probSucceedCondZ: a vector whose jth entry is the probability of reaching the threshold before dying, conditional on beginning in size/stage j
- probSucceed: the probability of succeeding assuming that an individual's starting state is distributed according to m0
- a0CondSucceed: the offspring state distribution conditional on reaching the LRO threshold before dying.

Examples

```
M = matrix(0.1, 3, 3)
Fmat = matrix(0, 3, 3); Fmat[1,] = 0:2
threshold = 2
m0 = c(1, 0, 0)
out = makeMCondLROThresholdPostBreeding(M, Fmat, threshold, m0)
```

`makePCondBreedDef3` *Utility function for calculating a kernel conditional on breeding.*

Description

Calculates the transition kernel conditional on producing at least one offspring, as well as related quantites. Assumes that the clutch sizes are Poisson-distributed.

Usage

```
makePCondBreedDef3(P, Fmat, c0)
```

Arguments

`P`	the survival/growth transition matrix
`Fmat`	the fecundity matrix. The clutch size of a parent of size j is Poisson-distributed with mean sum(Fmat[,j]).
`c0`	The size/stage distribution of offspring

Details

This function implements the calculations in Robin E. Snyder, Stephen P. Ellner, Giles Hooker. 2021. "Time and Chance: Using Age Partitioning to Understand How Luck Drives Variation in Reproductive Success." The American Naturalist 197(4). DOI: https://doi.org/10.1086/712874. It assumes that a breeder is an individual who has produced at least one offspring ("definition 3").

Value

A list containing the following. All matrices and vectors are defined over an extended size distribution: if there are n sizes/stages, states 1–n are those sizes with no offspring yet, states n+1–2n are those sizes with the first offspring produced in the current year, and states 2n+1–3n are those sizes having produced at least one offspring in some past year. See Details.

- PCondBreed: the transition matrix conditional on breeding before dying
- probEverBreedCondZ: a vector whose jth entry is the probability of breeding before dying, conditional on beginning in size/stage j
- PCondNeverBreedCondZ: the transition matrix conditional on not breeding before dying
- probEverBreedCondZ: a vector whose jth entry is the probability of not breeding before dying, conditional on beginning in size/stage j
- bigc0CondBreed: the offspring size/stage distribution conditional on the offspring breeding before dying.
- bigc0CondNeverBreed: the offspring size/stage distribution conditional on the offspring not breeding before dying.
- pM: the unconditional probability of breeding
- bigF: the fecundity matrix for the extended state space

See Also

makePDef3, which this function calls

Examples

```
P = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
c0 = c(1,0,0)
out = makePCondBreedDef3 (P, Fmat, c0)
```

`makePDef3` *Utility function for calculating a kernel conditional on breeding.*

Description

This code is called by makePCondBreedDef3. It assumes that clutch sizes are Poisson-distributed.

Usage

```
makePDef3(P, Fmat)
```

Arguments

`P`	the survival/growth transition matrix. P[i,j] is the probability of surviving and transitioning from size j to size i.
`Fmat`	the fecundity matrix. Fmat[i,j] is the expected number of size i offspring from a size j parent.

Details

This function calculates the extended state transition kernel defined in eq. S34 of the supplementary information in Robin E. Snyder, Stephen P. Ellner, Giles Hooker. 2021. "Time and Chance: Using Age Partitioning to Understand How Luck Drives Variation in Reproductive Success." The American Naturalist 197(4). DOI: https://doi.org/10.1086/712874. It assumes that a breeder is defined as an individual who has produced at least one offspring ("definition 3").

Value

bigP the extended state transition matrix (see Details)

See Also

makePCondBreedDef3, which calls this function

Examples

```
P = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
out = makePDef3 (P, Fmat)
```

`make_AxT` *Make the size x #kids transition matrix*

Description

Makes the 2-dim iteration matrix A for a size x #kids model, where #kids is the total number of offspring to date.

Usage

```
make_AxT(B, M, mT)
```

Arguments

`B`	The clutch size distribution matrix, where B[m,n] is the probability that a size n parent produces m-1 offspring in a single reproductive bout
`M`	The state transition matrix. States can be a single quantity, such as size or life history stage, or can be cross-classified, such as size x environment. M[i,j] is the probability of transitioning from state j to state i.
`mT`	The dimension of the #kids part of the 4-d array. I.e. the maximum #kids is mT-1.

Details

Called by distLifespanCondR2 and calcDistLRO.

Value

A list containing

- A: The 2-dim transition matrix for states cross-classified by size (or stage or...) and total number of offspring so far
- K: The 4-dim transition array, with dimensions (next size, next #kids, current size, current #kids). K is modified so individuals who get to the maximum values of #kids in the matrix stay there, but continue to grow/shrink/die.

Examples

```
M = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
B = mk_B (Fmat)
mT = 20
out = make_AxT(B, M, mT)
```

`meanConditionalLifespan` *Mean Conditional Lifespan*

Description

Calculates the mean lifespan for individuals starting in state j, conditional on them reaching state i before death. This calculation is based on Equation 6 in Cochran and Ellner (1992).

Usage

```
meanConditionalLifespan(M)
```

Arguments

`M`	An $n \times n$ matrix of transition rates among the n states in the matrix population model

Value

An $n \times n$ matrix where the (i, j) element represents the remaining expected lifespan of individuals starting in state j, conditional on them reaching state i before death, with NA if the transition from j to i is not possible.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
meanCondLifespan<- meanConditionalLifespan(Umat)
```

`meanConditionalTimes` *Compute conditional time to arriving in a state*

Description

Based on Equation 9 from Cochran and Ellner (1992), this function will calculate the expected time for an individual to transition from a starting state j to an arrival state i, conditional on arriving in state i.

Usage

```
meanConditionalTimes(M)
```

Arguments

`M`	An $n \times n$ matrix of transition rates among the n states in the matrix population model

Value

An $n \times n$ matrix where the (i, j) element represents the conditional mean time to reach i from j, with NA if the transition from j to i is not possible.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
meanCondTimes<- meanConditionalTimes(Umat)
```

`meanLifespan` *Calculate mean lifespan*

Description

Calculates the expected value of lifespan across individuals in a population whose dynamics can be described a matrix population model. The survival and transition rates among stages must be provided in the `Umat`. The default behavior is to return the expected remaining lifespan for individuals starting in all states of the population. This function also allows the user to provide a "mixing distribution" of starting states. Note that in the Markov Chain with Rewards framework, moments of lifespan are a special case of moments of LRO, where individuals accrue fixed rewards of 1 for each timestep that they survive.

Usage

```
meanLifespan(Umat, mixdist = NULL)
```

Arguments

`Umat`	An $n \times n$ matrix of transition rates among the n states in the matrix population model
`mixdist`	Optional, a vector of length n that contains the proportion of individuals starting in each state, to calculate a weighted average lifespan over the possible starting states

Value

Value(s) of expected remaining lifespan in terms of number of model timesteps. If `mixdist=NULL`, then `meanLifespan` returns a vector containing the expected remaining lifespan for individuals starting in each of the n states. If a mixing distribution is provided, then `meanLifespan` returns a single value, calculated as a weighted average over the offspring starting states.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
expectedVal<- meanLifespan(Umat)

# A standard choice for the mixing distribution is the distribution of
# offspring types in a cohort produced by the population at its stable
# distribution.
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
Amat<- Umat+Fmat
mixdist<- calcDistOffspringCohort(Amat, Fmat)
expectedVal<- meanLifespan(Umat, mixdist=mixdist)
```

`meanLRO` *Calculate mean lifetime reproductive output (LRO)*

Description

Calculates the expected value of lifetime reproductive output, or lifetime reproductive success, across individuals in a population whose dynamics can be described a matrix population model. The survival and transition rates among stages must be provided in the `Umat` and the mean per capita reproductive output (per time step of the population model) must be provided in the `Fmat`. The default behavior is to return the expected value of LRO for individuals starting in all states of the population. This function also allows the user to provide a "mixing distribution" of starting states, and/or to provide offspring weights such that offspring of different types contribute differently to reproductive success.

Usage

```
meanLRO(Umat, Fmat, mixdist = NULL, offspring_weight = NULL)
```

Arguments

`Umat`	An $n \times n$ matrix of transition rates among the n states in the matrix population model
`Fmat`	An $n \times n$ matrix of per capita rates of reproduction rates by any of the n states into offspring (new individuals of any of the n states)
`mixdist`	Optional, a vector of length n that contains the proportion of individuals starting in each state, to calculate a weighted average LRO over the entire population
`offspring_weight`	Optional, a vector of length n that contains the relative value of offspring of different types. If included, the `Fmat` is scaled column-wise by these weights, such that the rewards accrued differ across offspring types. For more details and an example of usage, see Hernandez et al. 2024. The natural history of luck: a synthesis study of structured population models. Ecology Letters, 27, e14390. Available from: https://doi.org/10.1111/ele.14390

Value

Value(s) of expected lifetime reproductive output. If `mixdist=NULL`, then `meanLRO` returns a vector containing the expected future LRO for individuals starting in each of the n states. If a mixing distribution is provided, then `meanLRO` returns a single value, calculated as a weighted average of the offspring starting states.

See Also

`calcDistOffspringCohort()`

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
expectedVal<- meanLRO(Umat, Fmat)

# A standard choice for the mixing distribution is the distribution of
# offspring types in a cohort produced by the population at its stable
# distribution.
Amat<- Umat+Fmat
mixdist<- calcDistOffspringCohort(Amat, Fmat)
expectedVal<- meanLRO(Umat, Fmat, mixdist)

# In Hernandez et al. 2024, we used the probability of surviving to reproduce
# to rescale reproductive rewards when individuals can produce multiple
# offspring types.
off_wts<- probRepro(Umat, Fmat, repro_var='poisson')
expectedVal<- meanLRO(Umat, Fmat, offspring_weight=off_wts)
```

`mk_B` *Make B matrix*

Description

Calculates B, the clutch size distribution matrix

Usage

```
mk_B(Fmat, maxKids = 20, Fdist = "Poisson")
```

Arguments

`Fmat`	the fecundity matrix. Fmat[i,j] is the expected number of size i offspring from a size j parent.
`maxKids`	The maximum clutch size. Optional, with a default value of 20.
`Fdist`	the clutch size distribution. Currently supported values are "Poisson" and "Bernoulli." Optional, with a default value of "Poisson".

Value

The clutch size distribution matrix B, where B[i,j] is the probability that a size j parent produces i-1 offspring in a single reproductive bout

Examples

```
Fmat = matrix (0, 3, 3); Fmat[1,] = 0.1*(0:2)
out = mk_B (Fmat, Fdist="Bernoulli")
```

`mk_BPostBreeding` *Make B matrix for a post-breeding census*

Description

Calculates B, the clutch size distribution matrix, for a post-breeding census

Usage

```
mk_BPostBreeding(M, Fmat, maxKids = 20, Fdist = "Poisson")
```

Arguments

`M`	The state transition matrix. States can be a single quantity, such as size or life history stage, or can be cross-classified, such as size x environment. M[i,j] is the probability of transitioning from state j to state i.
`Fmat`	the fecundity matrix. Fmat[i,j] is the expected number of size i offspring from a size j parent.
`maxKids`	The maximum clutch size. Optional, with a default value of 20.
`Fdist`	the clutch size distribution. Currently supported values are "Poisson" and "Bernoulli." Optional, with a default value of "Poisson".

Value

The clutch size distribution matrix B, where B[i,j] is the probability that a size j parent produces i-1 offspring in a single reproductive bout

Examples

```
Fmat = matrix (c(0,0,0, 0.5,0,0, 0,0,0), 3, 3)
M = matrix (c(0,0.5,0, 0,0,0.5, 0,0,0), 3, 3)
maxClutchSize=10
out = mk_BPostBreeding (M, Fmat, maxClutchSize, Fdist="Bernoulli")
```

---

partitionVarSkewnessEnvVar

Partitions variance and skewness of LRO

Description

Partitions variance and skewness of lifetime reproductive output (LRO) for models with environmental variation but without trait variation.

Usage

```
partitionVarSkewnessEnvVar(
  Plist,
  Flist,
  Q,
  c0,
  maxAge = 100,
  survThreshold = 0.05,
  Fdist = "Poisson",
  esR1 = NULL,
  esR2 = NULL,
  esR3 = NULL,
  bsR1 = NULL,
  bsR2 = NULL,
  bsR3 = NULL
)
```

Arguments

`Plist`	A list of survival/growth transition matrices. Plist[[q]][i,j] is the probability of transitioning from state j to state i in environment q.
`Flist`	A list of fecundity matrices. Flist[[q]][i,j] is the expected number of state i offspring from a state j parent in environment q
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxAge`	The maximum age an individual can attain. Optional. The default value is 100.
`survThreshold`	The threshold to use in determining lifespan (see return values). Optional. Default value is 0.05.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".
`esR1`	The 1st order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.

`esR2`	The 2nd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`esR3`	The 3rd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR1`	The 1st order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR2`	The 2nd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR3`	The 3rd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.

Details

A pre-breeding census is assumed: reproduction happens before survival and growth. The details of this calculation, including the definitions of the extended states, can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2) and its online supplemnt.

Value

A list containing the following: * birthStateVar: the contribution to Var(LRO) from birth state luck * birthEnvVar: the contribution to Var(LRO) from birth environment luck * survTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from survival trajectory luck at age j-1. * growthTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from growth trajectory luck at age j-1.

- envTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from environment trajectory luck at age j-1. * fecVar: a vector whose jth entry contains the contribution to Var(LRO) from fecundity luck at age j-1. * totVar: the total variance in LRO * birthStateSkewness: the contribution to LRO skewness from birth state luck
- birthEnvSkewness: the contribution to LRO skewness from birth environment luck * survTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from survival trajectory luck at age j-1. * growthTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from growth trajectory luck at age j-1. * envTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from environment trajectory luck at age j-1. * fecSkewness: a vector whose jth entry contains the contribution to LRO skewness from fecundity luck at age j-1. * totSkewness: the total skewness in LRO * lifespan: the age by which a proportion survThreshold of a cohort will be dead

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.2, 0, 0, 0, 0.3, 0, 0, 0.4), 3, 3)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
```

```
F2 = matrix (0, 3, 3); F1[1,] = 0.8*(0:2)
Plist = list (P1, P2)
Flist = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
out = partitionVarSkewnessEnvVar (Plist, Flist, Q, c0)
```

`partitionVarSkewnessEnvVarAndTraits` *Partitions variance and skewness of LRO*

Description

Partitions variance and skewness of lifetime reproductive output (LRO) for models with environmental variation and trait variation.

Usage

```
partitionVarSkewnessEnvVarAndTraits(
  PlistAllTraits,
  FlistAllTraits,
  Q,
  c0,
  traitDist,
  maxAge = 100,
  survThreshold = 0.05,
  Fdist = "Poisson",
  postBreedingCensus = FALSE
)
```

Arguments

`PlistAllTraits`	A list of lists of survival/growth transition matrices. Plist[[x]][[q]][i,j] is the probability of transitioning from state j to state i in environment q when an individual has trait x.
`FlistAllTraits`	A list of lists of fecundity matrices. Flist[[x]][[q]][i,j] is the expected number of state i offspring from a state j, trait x parent in environment q.
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`traitDist`	A vector whose jth entry is the probability of having trait j
`maxAge`	The maximum age an individual can attain. Optional. The default value is 100.
`survThreshold`	The threshold to use in determining lifespan (see return values). Optional. Default value is 0.05.

`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".
`postBreedingCensus`	Set this to TRUE to assume a post-breeding census (i.e. reproduction happens after survival and growth). Optional, with default value FALSE.

Details

A pre-breeding census is assumed: reproduction happens before survival and growth. The details of this calculation, including the definitions of the extended states, can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2) and its online supplement.

Value

A list containing the following:

- varFromTraits = the contribution to Var(LRO) from trait variation
- skewnessFromTraits = the contribution to LRO skewness from trait variation * totVar: the total variance in LRO
- totSkewness: the total skewness in LRO
- birthEnvVar: the contribution to Var(LRO) from birth environment luck
- birthEnvSkewness: the contribution to LRO skewness from birth environment luck * birthStateVar: the contribution to Var(LRO) from birth state luck
- birthStateSkewness: the contribution to LRO skewness from birth state luck
- survTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from survival trajectory luck at age j-1.
- survTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from survival trajectory luck at age j-1.
- growthTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from growth trajectory luck at age j-1.
- growthTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from growth trajectory luck at age j-1.
- envTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from environment trajectory luck at age j-1.
- envTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from environment trajectory luck at age j-1.
- fecVar: a vector whose jth entry contains the contribution to Var(LRO) from fecundity luck at age j-1.
- fecSkewness: a vector whose jth entry contains the contribution to LRO skewness from fecundity luck at age j-1.
- totVarCondX: a vector whose jth entry is Var(LRO | trait = j)
- totSkewnessCondX: a vector whose jth entry is the LRO skewness conditional on trait = j

- birthEnvVarCondX: a vector whose jth entry is the contribution of birth environment luck to Var(LRO | trait = j)
- birthEnvSkewnessCondX: a vector whose jth entry is the contribution of birth environment luck to LRO skewness conditional on trait = j
- birthStateVarCondX: a vector whose jth entry is the contribution of birth state luck to Var(LRO | trait = j)
- birthStateSkewnessCondX: a vector whose jth entry is the contribution of birth state luck to LRO skewness conditional on trait = j
- survTrajecVarCondX: a matrix whose i,jth entry contains the contribution to Var(LRO | trait = i) from survival trajectory luck at age j-1.
- survTrajecSkewnessCondX: a matrix whose i,jth entry contains the contribution to LRO skewness conditional on trait = i from survival trajectory luck at age j-1.
- growthTrajecVarCondX: a matrix whose i,jth entry contains the contribution to Var(LRO | trait = i) from growth trajectory luck at age j-1.
- growthTrajecSkewnessCondX: a matrix whose i,jth entry contains the contribution to LRO skewness conditional on trait = i from growth trajectory luck at age j-1.
- envTrajecVarCondX: a matrix whose i,jth entry contains the contribution to Var(LRO | trait = i) from environment trajectory luck at age j-1.
- envTrajecSkewnessCondX: a matrix whose i,jth entry contains the contribution to LRO skewness conditional on trait = i from environment trajectory luck at age j-1.
- fecVarCondX: a matrix whose i,jth entry contains the contribution to Var(LRO | trait = i) from fecundity luck at age j-1.
- fecSkewnessCondX: a matrix whose i,jth entry contains the contribution to LRO skewness conditional on trait = i from fecundity luck at age j-1.
- lifespanCondX: a vector whose jth entry is the age by which survThreshold proportion of a cohort with trait j would be dead

Examples

```
PlistAllTraits = FlistAllTraits = list ()
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = 0.9*P1
PlistAllTraits[[1]] = list (P1, P2)
P1 = matrix (c(0, 0.5, 0, 0, 0, 0.2, 0, 0, 0.7), 3, 3)
P2 = 0.9*P1
PlistAllTraits[[2]] = list (P1, P2)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = 0.8*F1
FlistAllTraits[[1]] = list (F1, F2)
F1 = matrix (0, 3, 3); F1[1,] = 0.9*(0:2)
F2 = 1.1*F1
FlistAllTraits[[2]] = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
traitDist = rep(0.5, 2)
out = partitionVarSkewnessEnvVarAndTraits (PlistAllTraits,
      FlistAllTraits, Q, c0, traitDist)
```

`partitionVarSkewnessEnvVarPostBreeding`
Partitions variance and skewness of LRO

Description

Partitions variance and skewness of lifetime reproductive output (LRO) for models with environmental variation but without trait variation, assuming a post-breeding census.

Usage

```
partitionVarSkewnessEnvVarPostBreeding(
  Plist,
  Flist,
  Q,
  c0,
  maxAge = 100,
  survThreshold = 0.05,
  Fdist = "Poisson",
  esR1 = NULL,
  esR2 = NULL,
  esR3 = NULL,
  bsR1 = NULL,
  bsR2 = NULL,
  bsR3 = NULL
)
```

Arguments

`Plist`	A list of survival/growth transition matrices. Plist[[q]][i,j] is the probability of transitioning from state j to state i in environment q.
`Flist`	A list of fecundity matrices. Flist[[q]][i,j] is the expected number of state i offspring from a state j parent in environment q
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxAge`	The maximum age an individual can attain. Optional. The default value is 100.
`survThreshold`	The threshold to use in determining lifespan (see return values). Optional. Default value is 0.05.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".
`esR1`	The 1st order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.

`esR2`	The 2nd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`esR3`	The 3rd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR1`	The 1st order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR2`	The 2nd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR3`	The 3rd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.

Details

A post-breeding census is assumed: reproduction happens after survival and growth. The details of this calculation, including the definitions of the extended states, can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2) and its online supplemnt.

Value

A list containing the following:

- birthStateVar: the contribution to Var(LRO) from birth state luck
- birthEnvVar: the contribution to Var(LRO) from birth environment luck
- survTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from survival trajectory luck at age j-1.
- growthTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from growth trajectory luck at age j-1.
- envTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from environment trajectory luck at age j-1.
- fecVar: a vector whose jth entry contains the contribution to Var(LRO) from fecundity luck at age j-1.
- totVar: the total variance in LRO
- birthStateSkewness: the contribution to LRO skewness from birth state luck
- birthEnvSkewness: the contribution to LRO skewness from birth environment luck
- survTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from survival trajectory luck at age j-1.
- growthTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from growth trajectory luck at age j-1.
- envTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from environment trajectory luck at age j-1.

- fecSkewness: a vector whose jth entry contains the contribution to LRO skewness from fecundity luck at age j-1.
- totSkewness: the total skewness in LRO
- lifespan: the age by which a proportion survThreshold of a cohort will be dead

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.2, 0, 0, 0, 0.3, 0, 0, 0.4), 3, 3)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F1[1,] = 0.8*(0:2)
Plist = list (P1, P2)
Flist = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
out = partitionVarSkewnessEnvVarPostBreeding (Plist, Flist, Q, c0)
```

`partitionVarSkewnessNoEnvVar`
Partitions variance and skewness of LRO

Description

Partitions variance and skewness of lifetime reproductive output (LRO) for models without trait or environmental variation.

Usage

```
partitionVarSkewnessNoEnvVar(
  Pmat,
  Fmat,
  c0,
  maxAge = 100,
  survThreshold = 0.05,
  Fdist = "Poisson",
  esR1 = NULL,
  esR2 = NULL,
  esR3 = NULL,
  bsR1 = NULL,
  bsR2 = NULL,
  bsR3 = NULL
)
```

Arguments

`Pmat`	The survival/growth transition matrix: Pmat[i,j] is the probability of transitioning from state j to state i.

`Fmat`	The fecundity matrix: Fmat[i,j] is the expected number of state i offspring from a state j parent
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxAge`	The maximum age an individual can attain. Optional. The default value is 100.
`survThreshold`	The threshold to use in determining lifespan (see return values). Optional. Default value is 0.05.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".
`esR1`	The 1st order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`esR2`	The 2nd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`esR3`	The 3rd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR1`	The 1st order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR2`	The 2nd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR3`	The 3rd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.

Details

A pre-breeding census is assumed: reproduction happens before survival and growth. The details of this calculation, including the definitions of the extended states, can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2) and its online supplemnt.

Value

A list containing the following: * birthStateVar: the contribution to Var(LRO) from birth state luck * survTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from survival trajectory luck at age j-1. * growthTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from growth trajectory luck at age j-1. * fecVar: a vector whose jth entry contains the contribution to Var(LRO) from fecundity luck at age j-1. * totVar: the total variance in LRO * birthStateSkewness: the contribution to LRO skewness from birth state luck * survTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from survival trajectory luck at age j-1. * growthTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from growth trajectory luck at age j-1. * fecSkewness: a vector whose jth entry contains the contribution to LRO skewness from fecundity luck at age j-1.

- totSkewness: the total skewness in LRO * lifespan: the age by which a proportion survThreshold of a cohort will be dead

Examples

```
Pmat = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
c0 = c(1,0,0)
out = partitionVarSkewnessNoEnvVar (Pmat, Fmat, c0)
```

---

partitionVarSkewnessNoEnvVarPostBreeding
Partitions variance and skewness of LRO for a post-breeding census

Description

Partitions variance and skewness of lifetime reproductive output (LRO) for models without trait or environmental variation, assuming a post-breeding census.

Usage

```
partitionVarSkewnessNoEnvVarPostBreeding(
  Pmat,
  Fmat,
  c0,
  maxAge = 100,
  survThreshold = 0.05,
  Fdist = "Poisson",
  esR1 = NULL,
  esR2 = NULL,
  esR3 = NULL,
  bsR1 = NULL,
  bsR2 = NULL,
  bsR3 = NULL
)
```

Arguments

`Pmat`	The survival/growth transition matrix: Pmat[i,j] is the probability of transitioning from state j to state i.
`Fmat`	The fecundity matrix: Fmat[i,j] is the expected number of state i offspring from a state j parent
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxAge`	The maximum age an individual can attain. Optional. The default value is 100.
`survThreshold`	The threshold to use in determining lifespan (see return values). Optional. Default value is 0.05.
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

`esR1`	The 1st order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`esR2`	The 2nd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`esR3`	The 3rd order reward matrix for the extended state space (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR1`	The 1st order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR2`	The 2nd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.
`bsR3`	The 3rd order reward matrix for the extended state space that includes the pre-birth "ur-state" (see Details). Optional. The default assumes Poisson-distributed clutch sizes.

Details

A post-breeding census is assumed: reproduction happens after survival and growth. The details of this calculation, including the definitions of the extended states, can be found in Robin E. Snyder and Stephen P. Ellner. 2024. "To prosper, live long: Understanding the sources of reproductive skew and extreme reproductive success in structured populations." The American Naturalist 204(2) and its online supplement.

Value

A list containing the following: * birthStateVar: the contribution to Var(LRO) from birth state luck * survTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from survival trajectory luck at age j-1. * growthTrajecVar: a vector whose jth entry contains the contribution to Var(LRO) from growth trajectory luck at age j-1. * fecVar: a vector whose jth entry contains the contribution to Var(LRO) from fecundity luck at age j-1. * totVar: the total variance in LRO * birthStateSkewness: the contribution to LRO skewness from birth state luck * survTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from survival trajectory luck at age j-1. * growthTrajecSkewness: a vector whose jth entry contains the contribution to LRO skewness from growth trajectory luck at age j-1. * fecSkewness: a vector whose jth entry contains the contribution to LRO skewness from fecundity luck at age j-1.

- totSkewness: the total skewness in LRO * lifespan: the age by which a proportion survThreshold of a cohort will be dead

Examples

```
Pmat = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
c0 = c(1,0,0)
out = partitionVarSkewnessNoEnvVarPostBreeding (Pmat, Fmat, c0)
```

`popMeanVar` *Population mean and variance of any attribute*

Description

Computes the population mean and variance of some attribute X, given the vectors of mean and variance conditional on individual 'type' Z. This substitutes for all of the individual formulas in Table 2 of Cochran and Ellner (1992). For example, if multiple offspring types are possible, this can be used to calculate mean and variance over the possible starting states.

Usage

```
popMeanVar(mean_by_type, var_by_type, mixdist)
```

Arguments

`mean_by_type`	A vector of length z containing the mean of attribute X for individuals of types 1 to z
`var_by_type`	A vector of length z containing the variance of attribute X for individuals of types 1 to z
`mixdist`	A vector of length z indicating the frequency distribution of individuals across the z types

Value

A list containing scalar values `pop_mean_X` and `pop_var_X` which give the mean and variance, respectively, of attribute X across the population

See Also

`meanLifespan()`, `meanLRO()`, `varLifespan()`, and `varLRO()`, which should give the same results when the user provides a mixing distribution.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
expectedVal<- meanLifespan(Umat)
varVal<- varLifespan(Umat)
mixdist<- c(0.3, 0.7, 0)
popVals<- popMeanVar(expectedVal, varVal, mixdist)
```

`probRepro` *Probability of reproducing at least once, based on starting state*

Description

Calculates the probability of reproducing at least once, for individuals starting in all possible starting states. In Hernandez et al. 2024, we used the probability of surviving to reproduce to rescale reproductive rewards when individuals can produce multiple offspring types.

Usage

```
probRepro(Umat, Fmat, repro_var = "poisson")
```

Arguments

`Umat`	An $n \times n$ matrix of transition rates among the n states in the matrix population model
`Fmat`	An $n \times n$ matrix of per capita rates of reproduction rates by any of the n states into offspring (new individuals of any of the n states)
`repro_var`	The form of variance in reproductive output, with possible values of `"poisson"`, `"bernoulli"` or `"fixed"`. This is required to calculate the probability of breeding this year from the values of mean per capita reproductive output.

Value

A vector of length n containing the probability of reproducing at least once, for individuals starting in each of the n states

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
probVals<- probRepro(Umat, Fmat, repro_var='poisson')
```

`probTraitCondLRO` *Distribution of a trait conditional on LRO*

Description

Calculates Prob(X | R), where X is a trait value and R is lifetime reproductive output (LRO) in the presence of environmental variation.

Usage

```
probTraitCondLRO(
  PlistAllTraits,
  FlistAllTraits,
  Q,
  c0,
  maxClutchSize,
  maxLRO,
  traitDist,
  Fdist = "Poisson"
)
```

Arguments

`PlistAllTraits`	A list of lists of survival/growth transition matrices. Plist[[x]][[q]][i,j] is the probability of transitioning from state j to state i in environment q when an individual has trait x.
`FlistAllTraits`	A list of lists of fecundity matrices. Flist[[x]][[q]][i,j] is the expected number of state i offspring from a state j, trait x parent in environment q.
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`traitDist`	A vector whose jth entry is the unconditional probability that the trait takes on the jth value
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2018. "Pluck or Luck: Does Trait Variation or Chance Drive Variation in Lifetime Reproductive Success?" The American Naturalist 191(4). DOI: 10.1086/696125

Value

A list containing the following:

- probXCondR: a matrix whose i,jth component is the probability that the trait has the jth value given that LRO = i-1
- probR: A vector whose jth entry is the unconditional probability that LRO = j-1
- probRCondX: A matrix whose i,jth entry is the probability that LRO = i-1 given that the trait takes the jth value.

Examples

```
PlistAllTraits = FlistAllTraits = list ()
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = 0.9*P1
PlistAllTraits[[1]] = list (P1, P2)
P1 = matrix (c(0, 0.5, 0, 0, 0, 0.2, 0, 0, 0.7), 3, 3)
P2 = 0.9*P1
PlistAllTraits[[2]] = list (P1, P2)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = 0.8*F1
FlistAllTraits[[1]] = list (F1, F2)
F1 = matrix (0, 3, 3); F1[1,] = 0.9*(0:2)
F2 = 1.1*F1
FlistAllTraits[[2]] = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
traitDist = rep(0.5, 2)
out = probTraitCondLRO (PlistAllTraits, FlistAllTraits, Q,
      c0, maxClutchSize=10, maxLRO=15, traitDist)
```

`probTraitCondLRONoEnv` *Distribution of a trait conditional on LRO*

Description

Calculates Prob(X | R), where X is a trait value and R is lifetime reproductive output (LRO) in the absence of environmental variation.

Usage

```
probTraitCondLRONoEnv(
  PlistAllTraits,
  FlistAllTraits,
  c0,
  maxClutchSize,
  maxLRO,
  traitDist,
  Fdist = "Poisson"
)
```

Arguments

`PlistAllTraits`	A list of lists of survival/growth transition matrices. PlistAllTraits[[x]][i,j] is the probability of transitioning from state j to state i when an individual has trait x.
`FlistAllTraits`	A list of lists of fecundity matrices. FlistAllTraits[[x]][i,j] is the expected number of state i offspring from a state j, trait x parent.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j

`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`traitDist`	A vector whose jth entry is the unconditional probability that the trait takes on the jth value
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2018. "Pluck or Luck: Does Trait Variation or Chance Drive Variation in Lifetime Reproductive Success?" The American Naturalist 191(4). DOI: 10.1086/696125

Value

A list containing the following:

- probXCondR: a matrix whose i,jth component is the probability that the trait has the jth value given that LRO = i-1
- probR: A vector whose jth entry is the unconditional probability that LRO = j-1
- probRCondX: A matrix whose i,jth entry is the probability that LRO = i-1 given that the trait takes the jth value.

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.5, 0, 0, 0, 0.2, 0, 0, 0.7), 3, 3)
PlistAllTraits = list(P1, P2)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F2[1,] = 0.9*(0:2)
FlistAllTraits = list (F1, F2)
c0 = c(1,0,0)
traitDist = rep(0.5, 2)
out = probTraitCondLRONoEnv (PlistAllTraits, FlistAllTraits,
      c0, maxClutchSize=10, maxLRO=15, traitDist)
```

`probTraitCondLROPostBreeding`

Distribution of a trait conditional on LRO

Description

Calculates Prob(X | R), where X is a trait value and R is lifetime reproductive output (LRO), for a post-breeding census model.

Usage

```
probTraitCondLROPostBreeding(
  PlistAllTraits,
  FlistAllTraits,
  Q,
  c0,
  maxClutchSize,
  maxLRO,
  traitDist,
  Fdist = "Poisson"
)
```

Arguments

`PlistAllTraits`	A list of lists of survival/growth transition matrices. Plist[[x]][[q]][i,j] is the probability of transitioning from state j to state i in environment q when an individual has trait x.
`FlistAllTraits`	A list of lists of fecundity matrices. Flist[[x]][[q]][i,j] is the expected number of state i offspring from a state j, trait x parent in environment q.
`Q`	The environment transition matrix. Q[i,j] is the probability of transitioning from environment j to environment i.
`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`traitDist`	A vector whose jth entry is the unconditional probability that the trait takes on the jth value
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2018. "Pluck or Luck: Does Trait Variation or Chance Drive Variation in Lifetime Reproductive Success?" The American Naturalist 191(4). DOI: 10.1086/696125

Value

A list containing the following:

- probXCondR: a matrix whose i,jth component is the probability that the trait has the jth value given that LRO = i-1
- probR: A vector whose jth entry is the unconditional probability that LRO = j-1
- probRCondX: A matrix whose i,jth entry is the probability that LRO = i-1 given that the trait takes the jth value.

Examples

```
PlistAllTraits = FlistAllTraits = list ()
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = 0.9*P1
PlistAllTraits[[1]] = list (P1, P2)
P1 = matrix (c(0, 0.5, 0, 0, 0, 0.2, 0, 0, 0.7), 3, 3)
P2 = 0.9*P1
PlistAllTraits[[2]] = list (P1, P2)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = 0.8*F1
FlistAllTraits[[1]] = list (F1, F2)
F1 = matrix (0, 3, 3); F1[1,] = 0.9*(0:2)
F2 = 1.1*F1
FlistAllTraits[[2]] = list (F1, F2)
Q = matrix (1/2, 2, 2)
c0 = c(1,0,0)
traitDist = rep(0.5, 2)
out = probTraitCondLROPostBreeding (PlistAllTraits, FlistAllTraits, Q,
      c0, maxClutchSize=10, maxLRO=15, traitDist)
```

`probTraitCondLROPostBreedingNoEnv`

Distribution of a trait conditional on LRO

Description

Calculates Prob(X | R), where X is a trait value and R is lifetime reproductive output (LRO), for a post-breeding census model in the absence of environmental variation

Usage

```
probTraitCondLROPostBreedingNoEnv(
  PlistAllTraits,
  FlistAllTraits,
  c0,
  maxClutchSize,
  maxLRO,
  traitDist,
  Fdist = "Poisson"
)
```

Arguments

`PlistAllTraits`	A list of lists of survival/growth transition matrices. PlistAllTraits[[x]][[q]][i,j] is the probability of transitioning from state j to state i in environment q when an individual has trait x.
`FlistAllTraits`	A list of lists of fecundity matrices. FlistAllTraits[[x]][[q]][i,j] is the expected number of state i offspring from a state j, trait x parent in environment q.

`c0`	A vector specifying the offspring state distribution: c0[j] is the probability that an individual is born in state j
`maxClutchSize`	The maximum clutch size to consider
`maxLRO`	The maximum LRO to consider
`traitDist`	A vector whose jth entry is the unconditional probability that the trait takes on the jth value
`Fdist`	The clutch size distribution. The recognized options are "Poisson" and "Bernoulli". Optional. The default value is "Poisson".

Details

The details of this calculation can be found in Robin E. Snyder and Stephen P. Ellner. 2018. "Pluck or Luck: Does Trait Variation or Chance Drive Variation in Lifetime Reproductive Success?" The American Naturalist 191(4). DOI: 10.1086/696125

Value

A list containing the following:

- probXCondR: a matrix whose i,jth component is the probability that the trait has the jth value given that LRO = i-1
- probR: A vector whose jth entry is the unconditional probability that LRO = j-1
- probRCondX: A matrix whose i,jth entry is the probability that LRO = i-1 given that the trait takes the jth value.

Examples

```
P1 = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
P2 = matrix (c(0, 0.5, 0, 0, 0, 0.2, 0, 0, 0.7), 3, 3)
PlistAllTraits = list (P1, P2)
F1 = matrix (0, 3, 3); F1[1,] = 0:2
F2 = matrix (0, 3, 3); F2[1,] = 0.9*(0:2)
FlistAllTraits = list (F1, F2)
c0 = c(1,0,0)
traitDist = rep(0.5, 2)
out = probTraitCondLROPostBreedingNoEnv (PlistAllTraits, FlistAllTraits,
      c0, maxClutchSize=10, maxLRO=15, traitDist)
```

`p_xT` *Helper function used for calculating size x #kids transition matrices*

Description

Function to take clutch size distribution matrix (B) and survival and growth or survival and growth and environment transition matrix (M), and compute the transition probabilities from size-class i and j total kids, to all size classes and l total kids.

Usage

```
p_xT(l, i, j, B, M)
```

Arguments

`l`	the value of total #kids in the next time step
`i`	the current state
`j`	the current total #kids
`B`	The clutch size distribution matrix, where B[m,n] is the probability that a size n parent produces m-1 offspring in a single reproductive bout
`M`	The state transition matrix. States can be a single quantity, such as size or life history stage, or can be cross-classified, such as size x environment. M[i,j] is the probability of transitioning from state j to state i.

Details

Called by make_AxT

Value

A vector whose kth entry is the probability that a size i individual with total number of offspring j will produce (l - j) offspring in the current reproductive bout and will transition to size k. If l-j is < 0, the return value is a vector of zeros.

See Also

make_AxT

Examples

```
M = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0:2
B = mk_B (Fmat)
p_xT (5, 2, 3, B, M)
```

`skewLifespan` *Calculate skewness of lifespan*

Description

Calculates the skewness of lifespan across individuals in a population whose dynamics can be described a matrix population model. The survival and transition rates among stages must be provided in the `Umat`. The default behavior is to return the skewness of remaining lifespan for individuals starting in all states of the population. This function also allows the user to provide a "mixing distribution" of starting states. Note that in the Markov Chain with Rewards framework, moments of lifespan are a special case of moments of LRO, where individuals accrue fixed rewards of 1 for each timestep that they survive.

Usage

```
skewLifespan(Umat, mixdist = NULL)
```

Arguments

`Umat`	An $n \times n$ matrix of transition rates among the n states in the matrix population model
`mixdist`	Optional, a vector of length n that contains the proportion of individuals starting in each state, to calculate a weighted average lifespan over the possible starting states

Value

Value(s) of skewness in remaining lifespan. If `mixdist=NULL`, then `skewLifespan` returns a vector containing the skewness in remaining lifespan for individuals starting in each of the n states. If a mixing distribution is provided, then `skewLifespan` returns a single value of skewness in remaining lifespan, calculated over the starting states according to the law of total cumulance.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
skewness<- skewLifespan(Umat)

# A standard choice for the mixing distribution is the distribution of
# offspring types in a cohort produced by the population at its stable
# distribution.
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
Amat<- Umat+Fmat
mixdist<- calcDistOffspringCohort(Amat, Fmat)
skewness<- skewLifespan(Umat, mixdist=mixdist)
```

`skewLRO` *Calculate skew in lifetime reproductive output*

Description

Calculates the skew of lifetime reproductive output, or lifetime reproductive success, across individuals in a population whose dynamics can be described a matrix population model. The survival and transition rates among stages must be provided in the `Umat` and the mean per capita reproductive output (per time step of the population model) must be provided in the `Fmat`. The default behavior is to assume Poisson-distributed annual reproduction among individuals in a given stage class. The default is to return the skewness of LRO for individuals starting in all states of the population. This function also allows the user to provide a "mixing distribution" of starting states, and/or to provide offspring weights such that offspring of different types contribute differently to reproductive success.

Usage

```
skewLRO(
  Umat,
  Fmat,
  repro_var = "poisson",
  mixdist = NULL,
  offspring_weight = NULL
)
```

Arguments

`Umat`	An $n \times n$ matrix of transition rates among the n states in the matrix population model
`Fmat`	An $n \times n$ matrix of per capita rates of reproduction rates by any of the n states into offspring (new individuals of any of the n states)
`repro_var`	The form of variance in per capita reproductive output, with possible values of `"poisson"`, `"bernoulli"` or `"fixed"`.
`mixdist`	Optional, a vector of length n that contains the proportion of individuals starting in each state, to calculate a weighted average LRO over the entire population
`offspring_weight`	Optional, a vector of length n that contains the relative value of offspring of different types. If included, the `Fmat` is scaled column-wise by these weights, such that the rewards accrued differ across offspring types. For more details and an example of usage, see Hernandez et al. 2024. The natural history of luck: a synthesis study of structured population models. Ecology Letters, 27, e14390. Available from: https://doi.org/10.1111/ele.14390

Value

Value(s) of skewness in lifetime reproductive output. If `mixdist=NULL`, then `skewLRO` returns a vector containing the skewness in future LRO for individuals starting in each of the n states. If a mixing distribution is provided, then `skewLRO` returns a single value of skewness, calculated over the offspring starting states using the law of total cumulance

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
skewness<- skewLRO(Umat, Fmat, repro_var='poisson')

# A standard choice for the mixing distribution is the distribution of
# offspring types in a cohort produced by the population at its stable
# distribution.
Amat<- Umat+Fmat
mixdist<- calcDistOffspringCohort(Amat, Fmat)
skew_overall<- skewLRO(Umat, Fmat, repro_var='poisson', mixdist=mixdist)

# In Hernandez et al. 2024, we used the probability of surviving to reproduce
# to rescale reproductive rewards when individuals can produce multiple
```

```
# offspring types.
off_wts<- probRepro(Umat, Fmat, repro_var='poisson')
skewness<- skewLRO(Umat, Fmat, repro_var='poisson', offspring_weight=off_wts)
```

`stable_dist` *Calculate the stable state distribution*

Description

The stable state distribution is the proportional distribution of the population across states (i.e., ages, stages, or sizes) when the population is growing at its asymptotically-constant equilibrium growth rate λ. The asymptotic population growth rate λ is given by the eigenvalue with the largest real part. The stable state distribution is the right eigenvector that corresponds to λ, normalized to sum to 1.

Usage

```
stable_dist(Amat)
```

Arguments

`Amat`	An $n{\times}n$ matrix containing all of the transition and reproductive rates that project the population from time t to $t+1$. Generally given by `Umat+Fmat`.

Value

A vector of length n that contains the proportion of individuals in each of the n states when the population is growing at its asymptotic population growth

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
Amat<- Umat+Fmat
ssd<- stable_dist(Amat)
```

`traitAve` *Calculates an average over a user-provided distribution*

Description

Calculates an average over a user-provided distribution

Usage

```
traitAve(x, traitDist)
```

Arguments

`x`	A vector
`traitDist`	A vector containing the probability of each value in x

Value

The average of x over traitDist

Examples

```
x = 1:3
traitDist = rep (1/3, 3)
traitAve (x, traitDist)
```

`traitMu3` *Calculates the third central moment over a user-provided distribution*

Description

Calculates the third central moment over a user-provided distribution

Usage

```
traitMu3(x, traitDist)
```

Arguments

`x`	A vector
`traitDist`	A vector containing the probability of each value in x

Value

The third central moment of x over traitDist

Examples

```
x = 1:3
traitDist = rep (1/3, 3)
traitMu3 (x, traitDist)
```

`traitVar` *Calculates a variance over a user-provided distribution*

Description

Calculates a variance over a user-provided distribution

Usage

```
traitVar(x, traitDist)
```

Arguments

`x`	A vector
`traitDist`	A vector containing the probability of each value in x

Value

The variance of x over traitDist

Examples

```
x = 1:3
traitDist = rep (1/3, 3)
traitVar (x, traitDist)
```

`unfold` *Convert a megamatrix to an array*

Description

Converts a cross-classified transition matrix ("megamatrix") to a 4-d transition array, such as the K array produced by make_AxT

Usage

```
unfold(A2, dim)
```

Arguments

`A2`	The megamatrix to be transformed
`dim`	A vector containing the dimensions of the array to be returned

Value

The transition array

See Also

flatten, make_AxT

Examples

```
M = matrix (c(0, 0.3, 0, 0, 0, 0.5, 0, 0, 0.5), 3, 3)
Fmat = matrix (0, 3, 3); Fmat[1,] = 0.1*(0:2)
B = mk_B (Fmat, Fdist="Bernoulli")
mT = 6
out = make_AxT (B, M, mT)
A = out$A
K = unfold (A, dim=c(3,6,3,6))  ## should equal out$K
```

`unvec` *Convert a vector to a matrix*

Description

Converts a vector to a matrix with user-defined dimensions.

Usage

```
unvec(nvec, nrow = NULL, ncol = NULL)
```

Arguments

`nvec`	The vector to be turned into a matrix
`nrow`	The number of rows in the matrix. Either nrow or ncol must be specified, but not both.
`ncol`	The number of columns in the matrix. Either nrow or ncol must be specified, but not both.

Details

If the vector has length N and the desired number of rows or columns is M, then N must be evenly divisible by M. The matrix is filled by columns.

Value

The matrix form of the vector

Examples

```
foo = 1:6
unvec (foo, nrow=2)
unvec (foo, ncol=2)
```

`varConditionalLifespan` *Variance of Conditional Lifespan*

Description

Calculates the variance in remaining lifespan for individuals starting in state j, conditional on them reaching state i before death. This calculation is based on Equation 7 in Cochran and Ellner (1992).

Usage

```
varConditionalLifespan(M)
```

Arguments

`M`	An $n \times n$ matrix of transition rates among the n states in the matrix population model

Value

An $n \times n$ matrix where the (i, j) element represents the variance of remaining lifespan of individuals starting in state j, conditional on them reaching state i before death, with NA if the transition from j to i is not possible.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
varCondLifespan<- varConditionalLifespan(Umat)
```

`varConditionalTimes` *Variance in conditional time to reach state i from state j*

Description

Calculates the variance in lifespan for individuals starting in state j, conditional on them reaching state i before death. This calculation is based on Equation 10 in Cochran and Ellner (1992).

Usage

```
varConditionalTimes(M)
```

Arguments

`M`	An $n \times n$ matrix of transition rates among the n states in the matrix population model

Value

An $n \times n$ matrix where the (i, j) element represents the variance in remaining expected lifespan of individuals starting in state j, conditional on them reaching state i before death, with NA if the transition from j to i is not possible.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
varCondTimes<- varConditionalTimes(Umat)
```

`varLifespan` *Calculate variance of lifespan*

Description

Calculates the variance of lifespan across individuals in a population whose dynamics can be described a matrix population model. The survival and transition rates among stages must be provided in the `Umat`. The default behavior is to return the variance of remaining lifespan for individuals starting in all states of the population. This function also allows the user to provide a "mixing distribution" of starting states. Note that in the Markov Chain with Rewards framework, moments of lifespan are a special case of moments of LRO, where individuals accrue fixed rewards of 1 for each timestep that they survive.

Usage

```
varLifespan(Umat, mixdist = NULL)
```

Arguments

`Umat`	An $n \times n$ matrix of transition rates among the n states in the matrix population model
`mixdist`	Optional, a vector of length n that contains the proportion of individuals starting in each state, to calculate a weighted average lifespan over the possible starting states

Value

Value(s) of variance in remaining lifespan. If `mixdist=NULL`, then `varLifespan` returns a vector containing the variance in remaining lifespan for individuals starting in each of the n states. If a mixing distribution is provided, then `varLifespan` returns a single value of variance in remaining lifespan, calculated over the starting states according to the law of total variance.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
variance<- varLifespan(Umat)

# A standard choice for the mixing distribution is the distribution of
# offspring types in a cohort produced by the population at its stable
# distribution.
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
Amat<- Umat+Fmat
mixdist<- calcDistOffspringCohort(Amat, Fmat)
variance<- varLifespan(Umat, mixdist=mixdist)
```

`varLRO` *Calculate variance in lifetime reproductive output*

Description

Calculates the variance of lifetime reproductive output, or lifetime reproductive success, across individuals in a population whose dynamics can be described a matrix population model. The survival and transition rates among stages must be provided in the `Umat` and the mean per capita reproductive output (per time step of the population model) must be provided in the `Fmat`. The default behavior is to assume Poisson-distributed annual reproduction among individuals in a given stage class. The default is to return the variance of LRO for individuals starting in all states of the population. This function also allows the user to provide a "mixing distribution" of starting states, and/or to provide offspring weights such that offspring of different types contribute differently to reproductive success.

Usage

```
varLRO(
  Umat,
  Fmat,
  repro_var = "poisson",
  mixdist = NULL,
  offspring_weight = NULL
)
```

Arguments

`Umat`	An $n \times n$ matrix of transition rates among the n states in the matrix population model
`Fmat`	An $n \times n$ matrix of per capita rates of reproduction rates by any of the n states into offspring (new individuals of any of the n states)
`repro_var`	The form of variance in per capita reproductive output, with possible values of `"poisson"`, `"bernoulli"` or `"fixed"`.
`mixdist`	Optional, a vector of length n that contains the proportion of individuals starting in each state, to calculate a weighted average LRO over the entire population

`offspring_weight`	Optional, a vector of length n that contains the relative value of offspring of different types. If included, the `Fmat` is scaled column-wise by these weights, such that the rewards accrued differ across offspring types. For more details and an example of usage, see Hernandez et al. 2024. The natural history of luck: a synthesis study of structured population models. Ecology Letters, 27, e14390. Available from: https://doi.org/10.1111/ele.14390

Value

Value(s) of variance in lifetime reproductive output. If `mixdist=NULL`, then `varLRO` returns a vector containing the variance in future LRO for individuals starting in each of the n states. If a mixing distribution is provided, then `varLRO` returns a single value of variance, calculated over the offspring starting states using the law of total variance.

Examples

```
Umat<- matrix(c(0.5, 0.1, 0.1, 0, 0.5, 0.3, 0, 0, 0.8), ncol=3)
Fmat<- matrix(c(0.2, 0, 0, 1, 0.5, 0, 5, 3, 0), ncol=3)
variance<- varLRO(Umat, Fmat, repro_var='poisson')

# A standard choice for the mixing distribution is the distribution of
# offspring types in a cohort produced by the population at its stable
# distribution.
Amat<- Umat+Fmat
mixdist<- calcDistOffspringCohort(Amat, Fmat)
variance_overall<- varLRO(Umat, Fmat, repro_var='poisson', mixdist=mixdist)

# In Hernandez et al. 2024, we used the probability of surviving to reproduce
# to rescale reproductive rewards when individuals can produce multiple
# offspring types.
off_wts<- probRepro(Umat, Fmat, repro_var='poisson')
variance<- varLRO(Umat, Fmat, repro_var='poisson', offspring_weight=off_wts)
```

`vec` *Convert a matrix to a vector*

Description

Converts a matrix to a vector by stacking columns

Usage

```
vec(nmat)
```

Arguments

`nmat`	The matrix to be converted

Value

The vectorized form of the matrix

Examples

```
foo = matrix (1:4, 2, 2)
vec (foo)
```

Index