ES 220 **Spring 2024**

Useful Project Functions – Details and Info

Code to install:

```
# Downloads latest version of project functions
download.file(url = "https://raw.githubusercontent.com/AldenGriffith/weplot/main/project-functions/project-functions.R",
              destfile= "project-functions.R",
              quiet = TRUE)

# Loads the functions
source("project-functions.R")
```

General functions for matrix models (deterministic or stochastic)

`get.lam` – Returns the population growth rate (dominant eigenvalue) of a population projection matrix.

Usage: `get.lam(A)`

`A` is a projection matrix.

`N.start` – Returns a population size vector with individuals in each stage/age class based on the proportions found after transient dynamics (“wiggles”) disappear when projecting the population (i.e. the “stable stage/age distribution”). This can be useful for providing a reasonable population size vector when starting a matrix model projection.

Usage: `N.start(A, Total.N)`

`A` is a projection matrix.

`Total.N` is the total population size to be distributed into each stage/age class.

`get.total.N` – Returns the total population size through time, summed across stage/age classes.

Usage: `get.total.N(N)`

`N` contains population size values from either a deterministic (2D array: rows = stage/age, columns = time) or stochastics (3D array: rows = stage/age, columns = time, layers = iterations) matrix model projection.

Functions for stochastic matrix models

`median.N` – Returns the median (of all iterations) total population size through time. In other words, this will collapse a stochastic matrix population projection into a single “line” through time such that half of the iterations resulted in population size greater than the line and half resulted in population sizes less than the line.

Usage: `median.N(N, time)`

`N` contains population size values from a stochastic model. It can either contain total population size values (2D array: rows = time, columns = iterations) or population size values for each stage/age (3D array: rows = stage/age, columns = time, layers = iterations).

`time` *(optional)* If the time argument is not included, the function will return values at all time points. If a value for time is provided (e.g. `time = 5`) then the function will return the median population size at the time point only. Note that the starting time is zero, and thus a value of a 5 corresponds to 5 time steps forward from the start.

`extinct.prob` – Returns the probability (expressed as a fraction) of the total population size decreasing below a specified minimum value.

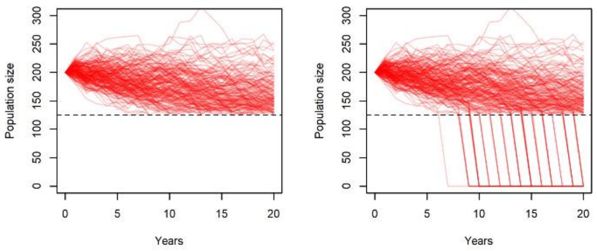
Usage: `extinct.prob(N, limit, time)`

`N` contains population size values from a stochastic model. It can either contain total population size values (2D array: rows = time, columns = iterations) or population size values for each stage/age (3D array: rows = stage/age, columns = time, layers = iterations).

`limit` specifies the minimum population size value (e.g. “quasi-extinction threshold”). Any iteration that goes below this value will be considered functionally “extinct”.

`time` (*optional*) specifies the time at which the population is evaluated. If the time argument is not included, the function will examine the entire period. If a value for time is provided (e.g. `time = 5`) then the function will calculate the probability up to that specific time. Note that the starting time is zero, and thus a value of a 5 corresponds to 5 time steps forward from the start.

`remove.extinct` – This function works in a similar way to `extinct.prob`, but rather than calculating a probability, it returns values of population sizes such that once an iteration has gone below a minimum size all future values are removed (e.g. `NA`, below left) or set to a specified value (e.g. zero, right):



Usage: `remove.extinct(N, limit, time, replace)`

`N` contains population size values from a stochastic model. It can either contain total population size values (2D array: rows = time, columns = iterations) or population size values for each stage/age (3D array: rows = stage/age, columns = time, layers = iterations).

`limit` specifies the minimum population size value (e.g. “quasi-extinction threshold”). Any iteration that goes below this value will be considered functionally “extinct”.

`time` (*optional*) specifies the time at which the population is evaluated. If the time argument is not included, the function will examine the entire period. If a value for time is provided (e.g. `time = 5`) then the function will calculate the probability up to that specific time. Note that the starting time is zero, and thus a value of a 5 corresponds to 5 time steps forward from the start.

`replace` (*optional*) specifies the replace value for populations that have gone below the minimum threshold. If not specified then values are simply removed (replaced with `NA`, above left).

`mat.dstoch` – Projects a structured population one year into the future with demographic stochasticity. This function can replace `A %*% N[,t]` in order to evaluate survival and reproduction *on an individual basis* instead of applying mean values to all individuals. It returns the population size vector at time t+1.

Usage: `mat.dstoch(A, Nt, repro.rows, repro.cols, post.breed)`

`A` is a projection matrix.

`Nt` is the population size vector at time t.

`repro.rows` (*optional*) specifies the row(s) that contain reproduction contributions. It assumes the 1st row if not specified.

`repro.cols` (*optional*) specifies the colums(s) that contain reproduction contributions. It assumes all columns except for the 1st if not specified.

`post.breed` (*optional*) indicates whether the life cycle census is post-breeding (`TRUE`; default) or pre-breeding (`FALSE`).