

An introduction to using population matrix models in R

After completing this assignment, you should know how to

- develop a population matrix model using R
- understand the components of a population matrix model
- use a population matrix model to project population size in the future, estimate λ , and determine the stable age distribution
- perform an elasticity analysis and interpret the values

To understand the dynamics of populations – for example, how fast they are growing (or declining) – we must be able to estimate demographic parameters like rates of survival and reproduction. These rates are often dependent on the age (or stage) of the organism (Gotelli 2001). The study of these vital statistics is known as demography. Ecologists have many tools to aid in describing and analyzing the demographic parameters (like birth and death rates) of populations. Life tables are one example of such a tool – they provide an age- or stage-specific schedule of survival and reproduction for a particular population. Here we will learn about other tools for representing and working with these demographic parameters to describe population dynamics: life cycle diagrams and matrix models.

Overview of matrix models

Life cycle diagrams

These simple diagrams provide a visualization of the age structure and vital rates of a population. Each circle represents the age class (or stage). Each arrow represents transitions between age classes or stages.

P_i = probability that an individual in age class i will survive to age class $i + 1$

F_i = fertility rate of each age class i

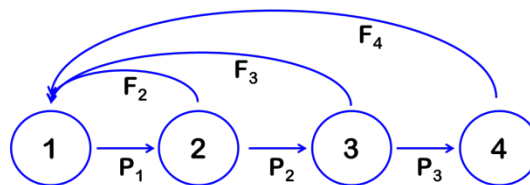


Figure 1. Example of a life cycle diagram for a population with four age classes. Adapted from Donovan and Welden (2002).

Matrix models

Information from a life cycle diagram can be translated into the form of a “matrix,” a rectangular array of values. The elements of the matrix (these values) represent the vital rates that characterize a population. Using matrix algebra, one can determine λ (the finite rate of increase), the stable age distribution, project future population sizes, and estimate other demographic

parameters based on a population matrix. There are two types of matrix models used to describe populations: a **Leslie matrix** can be used to summarize information about the *age-specific* survival and fertility rates of a population. A similar model that uses *stages* instead of *ages* is called a **Lefkovich matrix**. It may not always be possible (or useful) to determine the age of an organism; instead we may group an organism's life history into stages (like egg, larva, adult, for insects; seedling, small rosette, large rosette, reproductive, for a plant, etc.) (Gotelli 2001).

Translating information in a life cycle diagram to a Leslie (or Lefkovich) matrix

Let's say we have a population with four age classes like that in the life cycle diagram above (Fig. 1) We could set up a Leslie matrix from the vital rates shown in this diagram, using the conventions illustrated in Fig. 2.

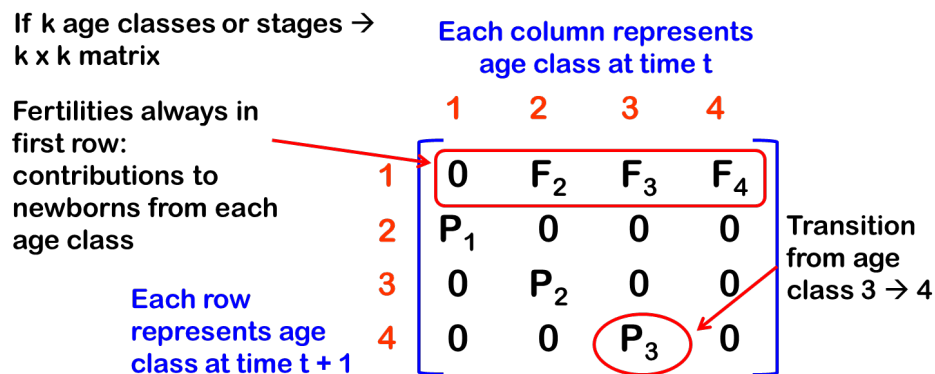


Figure 2. Example of assembling a Leslie matrix based on the vital rates shown in Fig. 1. (Adapted from Donovan and Welden 2002).

Population projection using a Leslie matrix

We can represent the current population by a single column vector with the number of individuals in each age class (N_0). Using matrix multiplication, we can find the number of individuals in each age class in the next time step, represented by another vector (N_1) (Fig. 3). But how do you multiply a matrix by a vector? There are special rules for this. You can do it by hand as illustrated in Fig. 4.

$$\begin{bmatrix} 0 & F_2 & F_3 & F_4 \\ P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \end{bmatrix} \times \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix}$$

Leslie matrix N_0 N_1

Figure 3. Using matrix multiplication to project population size. (Adapted from Donovan and Welden 2002).

Rules for multiplying a matrix by a vector

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} aw + bx + cy + dz \\ ew + fx + gy + hz \\ iw + jx + ky + lz \\ mw + nx + oy + pz \end{bmatrix}$$

Leslie matrix N_0 N_1

Example of projecting population size one step into the future

$$\begin{bmatrix} 0 & 0.9 & 1.4 & 1.1 \\ 0.7 & 0 & 0 & 0 \\ 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0.3 & 0 \end{bmatrix} \times \begin{bmatrix} 20 \\ 40 \\ 21 \\ 9 \end{bmatrix} = \begin{bmatrix} 0(20)+0.9(40)+1.4(21)+1.1(9) \\ 0.7(20)+0(40)+0(21)+0(9) \\ 0(20)+0.6(40)+0(21)+0(9) \\ 0(20)+0(40)+0.3(21)+0(9) \end{bmatrix} = \begin{bmatrix} 75.3 \\ 14 \\ 24 \\ 6.3 \end{bmatrix}$$

Leslie matrix N_0 N_1 N_1

Figure 4. Example of multiplying a matrix by a vector to project population size. (Adapted from Donovan and Welden 2002).

We can then use our two population vectors, N_0 and N_1 to determine λ (lambda), the finite rate of increase. Just add the elements of each vector to determine the population size at each time point.

$$N_0 = 20 + 40 + 21 + 9 = 90$$

$$N_1 = 75.3 + 14 + 24 + 6.3 = 119.6$$

Then we use the equation $\lambda = N_{t+1} / N_t$: $\lambda = N_1/N_0 = 119.6/90 = 1.329$

Of course, doing this by hand, especially to project population size farther into the future would be very tedious. Next we will learn how to use R to work with these matrix models.

Using R to project population size, determine, λ , and more

You will practice the process of constructing and manipulating a matrix model using R (in RStudio), with the help of the R package `popbio` (Stubben and Milligan 2007). You will get an introduction to these methods using a tool called “swirl.” Before proceeding, you should read and follow the steps for installing and starting swirl in this document: “Getting started with swirl.” You should install the course “Population_matrix_models” according to the instructions in that file. Start swirl and go to the first lesson of the course, “Setting up matrix.”

Lesson 1: Setting up matrix

Follow the prompts in the swirl lesson which will first introduce you to setting up matrices in general. In the *second half* of the swirl lesson, you will set up a Leslie matrix model (Fig. 5b) that corresponds to the life cycle diagram shown in Fig. 5a.

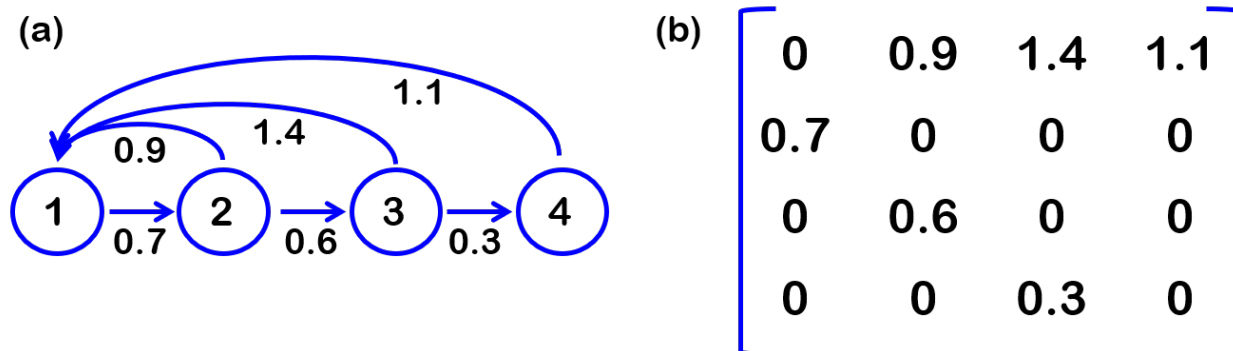


Figure 5. (a) Life cycle diagram for a hypothetical population with four age classes. (b) Leslie matrix corresponding to this life cycle diagram.

To enter the matrix from above, you will first need to create a vector to define the names of the age or stage classes:

```
ages = c("Age 1", "Age 2", "Age 3", "Age 4")
```

The next step is to enter the values of the matrix itself, using the `matrix()` function. We will call our matrix “A” and enter the values by rows. Here is the code you will need to use to match the data in Fig. 5b:

```
A = matrix(c(0,0.9,1.4,1.1,0.7,0,0,0,0,0.6,0,0,0,0,0.3,0),nrow=4,
byrow=T,dimnames=list(ages,ages))
```

#Note: we coded things slightly differently in FISH 458/558 to make the code a bit more readable and adaptable.

Here’s a breakdown of this code: After entering the 16 elements of our matrix (make sure you have 16 numbers!), `nrow=4` specifies 4 rows, `byrow=T` indicates filling by row (instead of column), `dimnames=list(...)` labels both rows and columns with the age classes that you defined above in the vector you created.

For our next step, we will need to define the initial population size of our population, broken down between our 4 age classes. If we have 20 1-year-olds, 40 2-year-olds, 21 3-year-olds, and 9 4-year-olds, we can define our initial population size vector N_0 as

```
N0=c(20,40,21,9)
```

You will use this vector to project population size into the future using matrix multiplication, in the last step of lesson 1 in this swirl course. When you are done, move on to Lesson 2: Analyzing matrix models.

Lesson 2: Analyzing matrix models.

This lesson makes use of the R package `popbio` which will allow for more sophisticated analyses and visualizations using the matrix model you developed in lesson 1. Note that this

lesson is pre-loaded with the matrix model A and the initial population vector N_0 that you created in lesson 1, so you do not need to repeat those steps.

Whenever we make use of an R package, we should first make sure we have it installed. If you do not have the `popbio` package already, swirl will install it for you. Once installed, you must load the package with the code `library(name of package)`. In this case, we would enter

```
library(popbio)
```

[Note, R already did this for you when you started this swirl lesson, but it's a good step to remember when you are working outside of this swirl course!]

In the package `popbio`, the function `pop.projection()` allows you to project the population size any number of time steps into the future. To determine our population size 10 steps into the future based on our matrix A and the initial population size vector N_0 , you would use this code:

```
projA=pop.projection(A,N0,10)
```

For your reference, the important output of the `pop.projection()` function includes:

<code>\$lambda</code>	estimate of λ based on the last 2 time steps of the projection
<code>\$stable.stage</code>	stable age (stage) distribution based on last population estimate
<code>\$stage.vectors</code>	projects number in each age class/stage at each time step
<code>\$pop.sizes</code>	total population at each time step
<code>\$pop.changes</code>	estimate of λ at each time step ($\lambda = N_{t+1}/N_t$)

(Stubben and Milligan 2007)

We can use this output to plot population growth over time. The swirl lesson will first have you make a quick graph to see the pattern of population growth. Then you will create a nicely formatted graph with proper x and y labels. When requested, write code to reproduce as best you can the plot shown in Fig. 6. (Some hints for modifying your plot: `pch=16` makes solid circles instead of the default open points, `type="b"` allows you to plot both points and lines connecting them.)

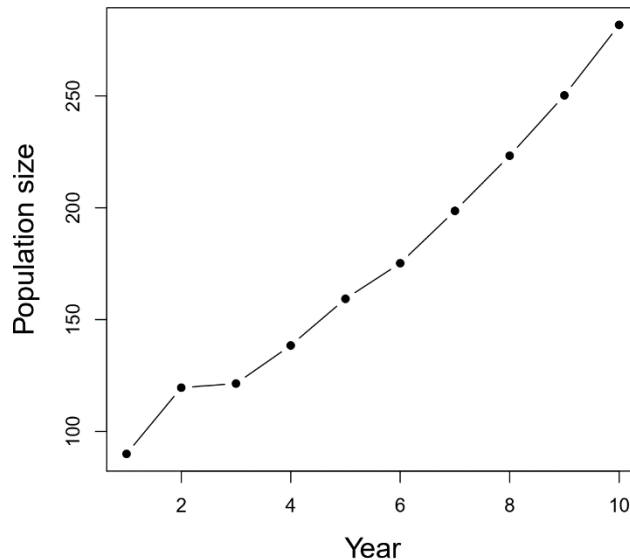


Figure 6. Population size projected over 10 years.

You will next be instructed on how to use the `eigen.analysis()` function to learn more about the structure and dynamics of the population represented by our Leslie matrix.

For your reference, the important output of the `eigen.analysis()` function includes:

<code>\$lambda1</code>	λ of population (when stabilized)
<code>\$stable.stage</code>	stable age (stage) distribution
<code>\$sensitivities</code>	sensitivity of each vital rate in matrix
<code>\$elasticities</code>	elasticity of each vital rate in matrix
<code>\$repro.value</code>	reproductive value: expected contribution of each individual to current and future reproduction
<code>\$damping.ratio</code>	rate of convergence to a stable stage distribution

(Stubben and Milligan 2007)

These parameters are all explained further in the swirl lesson. In particular, you will learn how we can interpret elasticity values to determine which vital rates (survival probabilities or fertility rates) contribute the most to the growth rate (λ) of the population, which can be useful information in the management of invasive species or the conservation of threatened species (Conard 2011, Stevens 2009).

References

- Conard, J.M. 2011. Demographic vital rates and population growth: an introduction to projection matrices and elasticity analysis. Teaching Issues and Experiments in Ecology, Vol. 7: Practice #3. Available from http://tiee.esa.org/vol/v7/issues/data_sets/conard/abstract.html
- Donovan, T.M., and C.W. Welden. 2002. Spreadsheet exercises in ecology and evolution. Sinauer Associates, Sunderland, MA.

Gotelli, N.J. 2001. A primer of ecology. Sinauer Associates, Sunderland, MA.

Stevens, M.H.H. 2009. A primer of ecology with R. Springer Science+Business Media, New York, NY.

Stubben, C.J., and B.G. Milligan. 2007. Estimating and analyzing demographic models using the popbio package in R. Journal of Statistical Software 22(11).