Dynamics of age-structured populations

Objectives

The objectives of this laboratory exercise are to:

- Use the R programming environment for numerical analysis
- Develop and execute a computer program written in the R language
- Use functions
- Solve a system of linear difference equations to get the exact future population size
- Numerically obtain the dominant eigenvalue and right eigenvector of the population projection matrix
- Show that growth of an age-structured density-independent discrete time population model converges to the dominant eigenvalue of the projection matrix
- Show that the age distribution of an age-structured density-independent discrete time population model converges to the right eigenvector

Introduction

Demographic theory

In class we noted that we can represent simple population growth in discrete time with a system of difference equations. The following system of equations could represent an age-structured population with maximum lifespan of six years and three year old age at first reproduction. Notice how many transitions are set to zero.

$$\mathbf{n_1} = (0)n_1 + (0)n_2 + F_3n_3 + F_4n_4 + F_5n_5 + F_6n_6 \quad (1)$$
$$\mathbf{n_2} = s_1n_1 + (0)n_2 + (0)n_3 + (0)n_4 + (0)n_5 + (0)n_6 \quad (2)$$
$$\mathbf{n_3} = (0)n_1 + s_2n_2 + (0)n_3 + (0)n_4 + (0)n_5 + (0)n_6 \quad (3)$$
$$\mathbf{n_4} = (0)n_1 + (0)n_2 + s_3n_3 + (0)n_4 + (0)n_5 + (0)n_6 \quad (4)$$
$$\mathbf{n_5} = (0)n_1 + (0)n_2 + (0)n_3 + s_4n_4 + (0)n_5 + (0)n_6 \quad (5)$$
$$\mathbf{n_6} = (0)n_1 + (0)n_2 + (0)n_3 + (0)n_4 + s_5n_5 + (0)n_6 \quad (6)$$

Further, this model can be more compactly represented (and solved) using matrix algebra, in which case,

$$\mathbf{n_1} = \mathbf{Ln_0} \quad (7)$$

with solution

$$\mathbf{n_t} = \mathbf{L^t n_0} \quad (8)$$

from which we deduce that the asymptotic growth rate is given by the dominant eigenvalue and that the stable age distribution is given by the dominant eigenvector.

Calculating eigenvalues numerically

For very small models (*e.g.*, for species that live up to two or three years) or for species with life cycles that can be approximated by a two-stage Lefkovich matrix, we can calculate the dominant eigenvalue and dominant eigenvector by hand. Generally, however, we will want to evaluate these quantities numerically. The theory of numerically computing eigenvalues and eigenvectors is actually quite advanced. Fortunately, computation is usually straightforward in practice. This is because the non-profit software company NAG (for "Numerical Algorithms Group") has developed a very solid, portable library of Fortran routines for solving common problems from linear algebra – including eigenvalue/eigenvector problems. This software is called LAPACK. The relevant routines from LAPACK can be accessed in `R` using the function `eigen`, which takes as its argument a matrix in the usual form, *i.e.*, in our case the Leslie matrix, and returns the numerically evaluated eigenvalues and eigenvectors.

An age-structured population

In this exercise we will study a hypothetical species which lives for six years. The life history graph is illustrated below, showing the age-specific fecundities F_3 through F_6. Notice that two year olds don't reproduce in this species and that fecundity increases with age. First, we create a *function*, called `LeslieSolve` that takes as its `argument` the age-structured vector of individual abundances and returns the age-structured abundance vector in the next time step. The guts of this function are a system of linear difference equations that represent this life history.

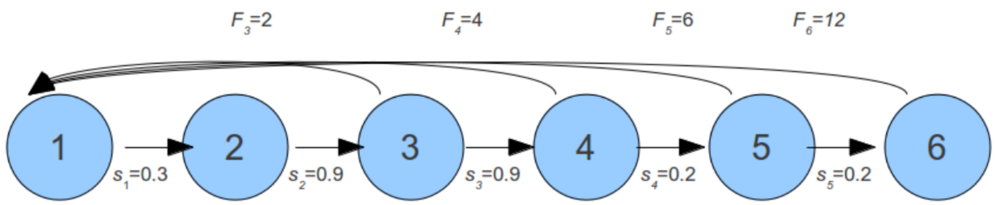


Figure 1: Life cycle graph.

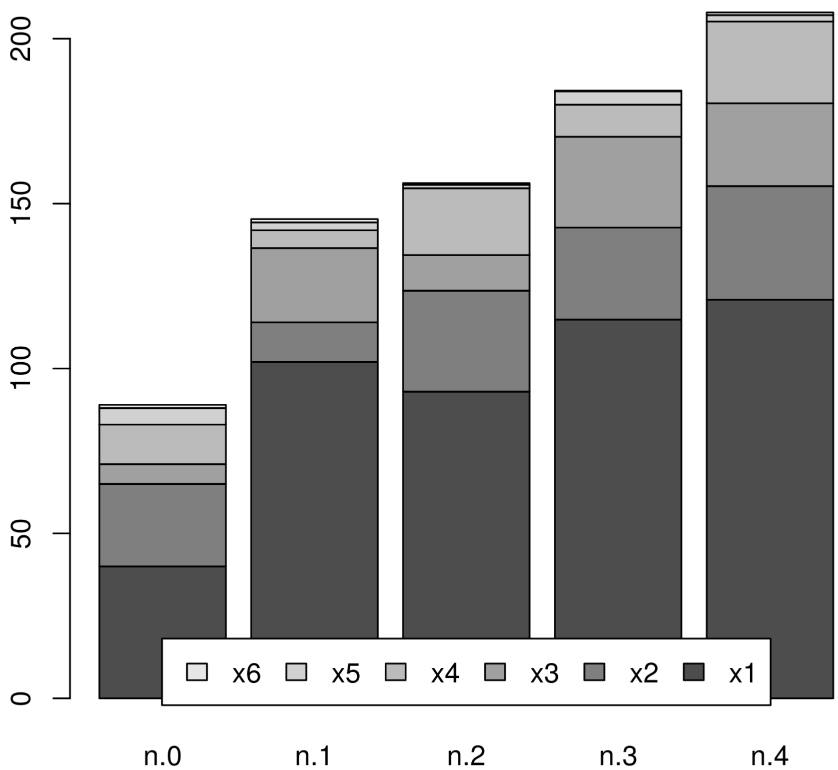
```
> LeslieSolve <- function(n) {
+     x1 <- ((2 * n[3]) + (4 * n[4]) + (6 * n[5]) + (12 * n[6]))
+     x2 <- (0.3 * n[1])
+     x3 <- (0.9 * n[2])
+     x4 <- (0.9 * n[3])
+     x5 <- (0.2 * n[4])
+     x6 <- (0.2 * n[5])
+     n.out <- c(x1, x2, x3, x4, x5, x6)
+ }
```

By declaring a vector of abundances `n` and using `LeslieSolve` we can compute the vector of population sizes at the next time step.

```
> n.0 <- c(40, 25, 6, 12, 5, 1)
> n.1 <- LeslieSolve(n.0)
```

Using the "column bind" operation (R function `cbind`), we can store the results of repeated iteration of this function and then use `barplot` to generate a graph.

```
> n <- cbind(n.0, n.1)
> n.2 <- LeslieSolve(n.1)
> n <- cbind(n, n.2)
> n.3 <- LeslieSolve(n.2)
> n <- cbind(n, n.3)
> n.4 <- LeslieSolve(n.3)
> n <- cbind(n, n.4)
> barplot(n, legend.text = c("x1", "x2", "x3", "x4", "x5", "x6"),
+     args.legend = c(x = "bottom", horiz = TRUE))
```



Open the file `age-structured-code.R` in RStudio. Run this code to confirm that the function works and reproduce the results obtained above. Then modify or add to this program to solve the following problems.

Exercises

1. Start with initial population where the abundance of x_1 is 100 and x_2 through x_6 are 0. Use `Leslie-Solve` to determine if this population increases or decreases. Modify the function so that s_1 is 0.15 and solve again. Now is the population increasing or decreasing. At (approximately) what value of s_1 is the population stable? Switch back to $s_1 = 0.3$ for the remainder of the exercise.

2. Perhaps one of the more unexpected properties of this model is that over time the relative abundance of the difference age classes settles down to the stable age distribution. As we know, and will show later

in this exercise, this stable age distribution is easy to calculate as the right eigenvector of the Leslie matrix. Importantly, however, the population doesn't immediately reach the stable age distribution. Rather, the stable age distribution is approached asymptotically (as is the growth rate, for that matter). This raises two questions. First, how fast does the actual age distribution approach the stable age distribution? How long does it take before transient dynamics are surpassed and the population exhibits stable growth? To answer these questions, initialize six populations with the following initial age distributions, iterate for 50 time steps, and plot the abundance of each age class over time. What behavior do you observe? In general, how quickly does population growth stabilize? What is the effect of concentrating the initial population in early versus late age classes?

- $\mathbf{n_0} = [10, 10, 10, 10, 10, 10]$
- $\mathbf{n_0} = [60, 0, 0, 0, 0, 0]$
- $\mathbf{n_0} = [0, 0, 0, 0, 0, 60]$
- $\mathbf{n_0} = [20, 0, 20, 0, 20, 0]$
- $\mathbf{n_0} = [20, 20, 20, 0, 0, 0]$
- $\mathbf{n_0} = [0, 0, 0, 20, 20, 20]$

What are the implications of these *transient dynamics* for interpreting data collected from natural populations?

3. Now, declare a two-dimensional "vector" `L` which contains the Leslie matrix for this species. (To clarify, one should think of a matrix in R as being a two-dimensional vector. If you declare a two-dimensional vector, R will automatically recognize it as a matrix, *i.e.*, run `m=rbind(c(1,2),c(3,4)); is.matrix(m)`.) *Hint:* use a combination of `rbind` and `c` or just the function `matrix` by itself.

4. Use the function eigen to return a list of the eigenvalues and eigenvectors of this population. Use the R function `Re` to extract the real part of the dominant eigenvalue – this is the λ of the matrix and is the long run population growth rate. (What is λ for this population?)

5. In this step we study the convergence of population growth to its asymptotic rate. For this purpose we will focus on the total population size (*i.e.*, the value obtained from summing the abundance of each age class). To do this, we will solve the model three different ways:

   - Using λ as a scalar multiplier for the scalar (unstructured) population size $N = n_1 + n_2 + n_3 + n_4 + n_5 + n_6$
   - Using λ as a scalar multiplier for the structured population vector $\mathbf{n_0} = [n_1, n_2, n_3, n_4, n_5, n_6]$
   - Using the function `LeslieSolve` to iterate the model exactly

   Using each of these approaches, solve the model for 50 time steps and plot the results for comparison, using $\mathbf{n_0} = [10, 10, 10, 10, 10, 10]$ as the initial population vector. Repeat using each of the other initial population vectors from the table above. What differences result from the three ways of representing population growth? When does it appear that these differences are most important?

6. For extra credit, show how the the stable age distribution approaches the normalized dominant eigenvector returned by `eigen`.