Analisi demografica di una popolazione reintrodotta di cervo

con effetto della densità dipendenza

# Caso studio

Questo esempio di analisi demografica si basa su un caso studio descritto in Jensen, A. L. (1995). *Simple density-dependent matrix model for population projection.* Ecological Modelling **77**, 43–48.

## La matrice di transizione

La matrice di transizione che descrive la dinamica di popolazione è la seguente:

B <- matrix(c(0.426, 1.290, 1.296, 1.120, 1.126, 1.554, 0,  
 0.713, -1, 0, 0, 0, 0, 0,  
 0, 0.645, -1, 0, 0, 0, 0,  
 0, 0, 0.648, -1, 0, 0, 0,  
 0, 0, 0, 0.560, -1, 0, 0,  
 0, 0, 0, 0, 0.563, -1, 0,  
 0, 0, 0, 0, 0, 0.777, -1),   
 nr = 7, byrow = TRUE)  
  
B

## [,1] [,2] [,3] [,4] [,5] [,6] [,7]  
## [1,] 0.426 1.290 1.296 1.12 1.126 1.554 0  
## [2,] 0.713 -1.000 0.000 0.00 0.000 0.000 0  
## [3,] 0.000 0.645 -1.000 0.00 0.000 0.000 0  
## [4,] 0.000 0.000 0.648 -1.00 0.000 0.000 0  
## [5,] 0.000 0.000 0.000 0.56 -1.000 0.000 0  
## [6,] 0.000 0.000 0.000 0.00 0.563 -1.000 0  
## [7,] 0.000 0.000 0.000 0.00 0.000 0.777 -1

Assegniamo dei nomi ai diversi stadi (righe e colonne) della matrice di transizione:

stages <- c("Newborns", "Yearlings", "Juveniles", "Adult1", "Adult2", "Adult3", "Adult3")

Uso la funzione *eigen* per calcolare il valore di lambda (tasso di incremento) della mia popolazione di cervo:

# Determinate lambda  
eigs.B <- eigen(B)  
lambda <- Re(eigs.B[["values"]][1])  
lambda

## [1] -1.585997

# Simulazione della dinamica di popolazione

Partendo da una popolazione iniziale con un certo numero di animali per ogni classe di età, vado a plottare la dinamica di popolazione per un periodo di 20 anni.  
Per prima cosa creo un vettore N0, che rappresenta la popolazione iniziale, in cui ho soltanto 4 femmine di 1 anno di età. La mia matrice di transizione rappresenta la dinamica della componente femminile della popolazione.

N0 <- matrix(c(0, 4, 0, 0, 0, 0, 0), ncol = 1)  
# N0  
years <- 20 # anni di simulazione  
# years  
N.projections <- matrix(0, nrow = nrow(B), ncol = years + 1) # matrice vuota per i risultati  
N.projections[, 1] <- N0  
N.projections

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]  
## [1,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [2,] 4 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [3,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [4,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [5,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [6,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [7,] 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [,15] [,16] [,17] [,18] [,19] [,20] [,21]  
## [1,] 0 0 0 0 0 0 0  
## [2,] 0 0 0 0 0 0 0  
## [3,] 0 0 0 0 0 0 0  
## [4,] 0 0 0 0 0 0 0  
## [5,] 0 0 0 0 0 0 0  
## [6,] 0 0 0 0 0 0 0  
## [7,] 0 0 0 0 0 0 0

Uso un ciclo *for* per proiettare nel futuro la popolazione. Il ciclo *for* mi permette di ripetere lo stesso calcolo per un determinato numero di volte, corrispondenti al numero di anni di simulazione:

for (i in 1:years) N.projections[, i + 1] <- N.projections[,+ i] +   
 matrix(((110-N.projections[,+ i])/110),nr=7) %\*% (-lambda) \* (N.projections[,+ i])  
N.projections

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]  
## [1,] 0 0.0000 0.00000 0.00000 0.00000 0.0000 0.00 0.0000 0.0000  
## [2,] 4 10.1133 24.67828 55.03705 98.65195 114.7932 106.86 111.6979 108.9635  
## [3,] 0 0.0000 0.00000 0.00000 0.00000 0.0000 0.00 0.0000 0.0000  
## [4,] 0 0.0000 0.00000 0.00000 0.00000 0.0000 0.00 0.0000 0.0000  
## [5,] 0 0.0000 0.00000 0.00000 0.00000 0.0000 0.00 0.0000 0.0000  
## [6,] 0 0.0000 0.00000 0.00000 0.00000 0.0000 0.00 0.0000 0.0000  
## [7,] 0 0.0000 0.00000 0.00000 0.00000 0.0000 0.00 0.0000 0.0000  
## [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17]  
## [1,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## [2,] 110.5919 109.6481 110.2044 109.8796 110.0703 109.9587 110.0242 109.9858  
## [3,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## [4,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## [5,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## [6,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## [7,] 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000  
## [,18] [,19] [,20] [,21]  
## [1,] 0.0000 0.0000 0.0000 0.0000  
## [2,] 110.0083 109.9951 110.0029 109.9983  
## [3,] 0.0000 0.0000 0.0000 0.0000  
## [4,] 0.0000 0.0000 0.0000 0.0000  
## [5,] 0.0000 0.0000 0.0000 0.0000  
## [6,] 0.0000 0.0000 0.0000 0.0000  
## [7,] 0.0000 0.0000 0.0000 0.0000

Rappresento graficamente il risultato:

matplot(0:years, rowSums(t(N.projections)), type = "l", ylab="Total no. of females",  
 xlab="Years")

