Population Projection

Caila Kucheravy

2024-01-04

We are gonna try making a population projection using the epigenetically estimated ages. This is VERY preliminary and I'm not sure I'll end up using it, since we don't have vital rates for each age category, and also I'm not sure if they ages are reliable enough to estimate growth rate from. Ideally I would like to do an integrated population analysis with the CMR data to estimate lambda, but still not sure if that will be possible.

Prep & Import Age Data:

```
rm(list=ls())

setwd("~/Documents/Master's/Analysis/Epigenetic Aging/Killer Whales")

library(tidyverse)
library(popbio)

sex.age_data <- read.csv("kw_ages_HA.csv") %>%
  select(sampleId, Year, Location, Sex, Age_Transformed, adj_age)
```

Add age classes to datasheet:

```
age_class_data <- sex.age_data %>%
  mutate(age_class = if_else(Sex == "M",
                             # For males, we have two observed age classes:
                             # Juvenile:  1.5-13 yrs
                             # Adult:    >13 yrs
                             if_else(adj_age < 13, "Juv. Male", "Repro. Male"),

                             # For females, we have three observed age classes:
                             # Juvenile:                 1.5-13 yrs
                             # Reproductive adult:       12-35 yrs
                             # Post-reproductive adult: >35 yrs
                             if_else(Sex == "F",
                                     if_else(adj_age < 13, "Juv. Female",
                                             if_else(adj_age < 35, "Repro. Female", "Post-Repro. Female
                                     ""))) %>%
  # Add three artificial female calves
  add_row(Sex = "F", adj_age = 1, age_class = "Calf") %>%
  add_row(Sex = "F", adj_age = 1, age_class = "Calf") %>%
  add_row(Sex = "F", adj_age = 1, age_class = "Calf")

summary <- age_class_data %>%
  group_by(age_class) %>%
  dplyr::summarise(n = n())
summary
```

```
## # A tibble: 5 x 2
##   age_class         n
##   <chr>         <int>
## 1 Calf              3
## 2 Juv. Female      12
## 3 Juv. Male        24
## 4 Repro. Female    17
## 5 Repro. Male      12
```

Use females only for population model:

```
age_class_female <- age_class_data %>%
  filter(Sex == 'F')
```

Matrix model based on Brault & Caswell (1993).

Construct a Lefkovitch (stage-structured) matrix to estimate population growth rate (λ):

```
                        # X = age class
initial_ages <- data.frame(X = c("Calf", "Juv. Female", "Repro. Female", "Post-Repro. Female"),
                         # Nx = the number counted in an age class
                         Nx = c(sum(age_class_female$age_class == "Calf"),
                                sum(age_class_female$age_class == "Juv. Female"),
                                sum(age_class_female$age_class == "Repro. Female"),
                                sum(age_class_female$age_class == "Post-Repro. Female"))
                         )

# Survival probabilities for (1) calves, (2) juveniles, (3) reproductive adults, (4) post-reproductive
# Since we don't have real survival probabilities for each age class, these are basically just made up
omega1 <- 0.9
omega2 <- 0.95
omega3 <- 0.95
omega4 <- 0.5

# Growth probabilities = reciprocals of mean stage durations
gamma2 <- 1/(13-1.5)
gamma3 <- 1/(35-13)

# ratio of the number of female offspring produced by the group to the number of female-years of exposu
m      <- 0.1

G1 = omega1^(1/2)
G2 = gamma2*omega2
P2 = (1 - gamma2)*omega2
G3 = gamma3*omega3
P3 = (1 - gamma3)*omega3
P4 = omega4
#F2 = (omega1^(1/2))*(G2*(m/2))
#F3 = (omega1^(1/2))*(1 + P3)*(m/2)
F2 = 0
F3 = m
```

Build a matrix:

```
                      #C     J     A     P
#kw_matrix <- matrix(c( 0,    0,  R_a,    0,
                   #  G_c,  P_j,    0,    0,
```

```r
                   #    0,   G_j,   P_a,    0,
                   #    0,    0,   G_a,   P_p),
                   # nrow = 4, byrow = T)

                      #C      J      A      P

kw_matrix <- matrix(c( 0, F2, F3,  0,
                      G1, P2,  0,  0,
                       0, G2, P3,  0,
                       0,  0, G3, P4),
                     nrow = 4, byrow = T)

eigen.analysis(kw_matrix)
```

```
## $lambda1
## [1] 0.9787336
##
## $stable.stage
## [1] 0.04952806 0.42199984 0.48474780 0.04372430
##
## $sensitivities
##            [,1]      [,2]      [,3]       [,4]
## [1,] 0.04273538 0.3641234 0.4182655 0.03772759
## [2,] 0.04408906 0.3756573 0.4315145 0.03892264
## [3,] 0.05942448 0.5063215 0.5816074 0.05246104
## [4,] 0.00000000 0.0000000 0.0000000 0.00000000
##
## $elasticities
##            [,1]       [,2]       [,3] [,4]
## [1,] 0.00000000 0.00000000 0.04273538    0
## [2,] 0.04273538 0.33292189 0.00000000    0
## [3,] 0.00000000 0.04273538 0.53887197    0
## [4,] 0.00000000 0.00000000 0.00000000    0
##
## $repro.value
## [1] 1.000000 1.031676 1.390522 0.000000
##
## $damping.ratio
## [1] 1.246345
```

Let's project the population 20 years:

```r
N = initial_ages$Nx
N_past = N
# Repeat matrix multiplication 20 times in a loop
for (step in 1:20) {              # code block inclosed in { }
  # Calculations
  N = kw_matrix%*%N                       # matrix multiplication A*N = N'
  pop = sum(N)                    # population size
  # Output using cat() function
  cat(round(N))                   # new population vector
  cat(" | pop= ")                 # population label
  cat(round(pop))                 # new population size
  cat(" | lambda= ")              # lambda label
  cat(sum(N)/sum(N_past))         # lambda estimate
```

```r
  cat("\n")                          # newline
  # Update N_past
  N_past = N
}                                    # close code block for loop
```

```
## 2 13 16 1 | pop= 32 | lambda= 1.003002
## 2 13 16 1 | pop= 32 | lambda= 0.9907571
## 2 13 16 1 | pop= 31 | lambda= 0.9849434
## 2 13 15 1 | pop= 31 | lambda= 0.9820012
## 2 13 15 1 | pop= 30 | lambda= 0.9804896
## 1 12 14 1 | pop= 30 | lambda= 0.9797029
## 1 12 14 1 | pop= 29 | lambda= 0.9792875
## 1 12 14 1 | pop= 28 | lambda= 0.9790639
## 1 12 14 1 | pop= 28 | lambda= 0.9789403
## 1 11 13 1 | pop= 27 | lambda= 0.9788697
## 1 11 13 1 | pop= 27 | lambda= 0.9788277
## 1 11 13 1 | pop= 26 | lambda= 0.9788013
## 1 11 12 1 | pop= 25 | lambda= 0.978784
## 1 10 12 1 | pop= 25 | lambda= 0.978772
## 1 10 12 1 | pop= 24 | lambda= 0.9787634
## 1 10 12 1 | pop= 24 | lambda= 0.978757
## 1 10 11 1 | pop= 23 | lambda= 0.9787521
## 1 10 11 1 | pop= 23 | lambda= 0.9787483
## 1 9 11 1 | pop= 22 | lambda= 0.9787453
## 1 9 11 1 | pop= 22 | lambda= 0.978743
```