

# Assig4

## Demographic analysis

*Villaseñor-Derbez, J.C. & Faro, C*

```
suppressPackageStartupMessages({  
  library(readr)  
  library(tidyverse)  
  library(reshape)  
  library(popbio)  
})
```

## Description of the demographic information

- Mortality:  $Z = 1.39$  and  $M = 0.63$  (Garbin & Castello, 2014)
- Fecundity:  $a = -1.33354$ ,  $b = 3.238$  (from FISHBASE)
- von Bertalanffy growth parameters:  $L_{inf} = 102.0$ ,  $K = 0.55$ ,  $t_0 = -0.02$  (Uchiyama & Struhsaker, 1981)

## Define demographic parameters

```
# Define Von Bert parameters  
# Garbin & Castello, 2014  
# L_inf <- mean(c(80, 80, 87.12, 94, 97.9, 97.3, 112.34, 89.38, 92.5))  
# K <- mean(c(0.32, 0.6, 0.22, 0.38, 0.14, 0.25, 0.14, 0.38, 0.16))  
  
# Or from Us and Stiurskjpgfas, 1981  
L_inf <- 102  
K <- 0.55  
t_0 <- -0.02  
  
# Just to be clear, we use L-inf and t_0 from Ushisomething, and K from the review  
  
# Define fecundity parameters  
# From fishbase http://www.fishbase.se/Reproduction/FecundityList.php?ID=107&GenusName=Katsuwonus&Speci  
fec_a <- -1.33354  
fec_b <- 3.238  
# Define mortality  
m <- 0.63  
z <- 1.39
```

## Define the functions we will need

```
# Convert length to age using von bertalanffy model, solving for t  
length2age <- function(length, l_inf, K, t_o){  
  age <- (1/-K)*(log(1-(length/L_inf))) + t_o
```

```

    return(age)
}

# Convert age to length using von bertalanffy model
age2length <- function(age, l_inf, K, t_o){
  length <- l_inf*(1-exp(-K*(age-t_o)))
  return(length)
}

#Convert length to fecundity (number of eggs)
fecundity <- function(length, a, b){
  f <- 10^(a+(b*log10(length*10)))
  return(f)
}

```

## Create the matrix

```

A <- matrix(0, 14, 14) #initial empty matrix with all 0

# Populate matrix with mortality
for (i in 2:14){
  A[i,i-1] <- exp(-z)
}

# Populate matrix with fecundity
ages <- seq(0:13)-1
lengths <- age2length(ages, L_inf, K, t_0)
A[1,] <- fecundity(lengths, fec_a, fec_b)

A[is.na(A)] <- 0
A[2,1] <- 0.0000001
A[1,1] <- 0

colnames(A) <- ages
rownames(A) <- ages

knitr::kable(A, col.names = paste0("$a_{",ages,"}$"), row.names = F, caption = "Table 2- Population matrix A")

```

Table 1: Table 2- Population matrix A. The represents survivals, while the first row represent

$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
0e+00	1.657256e+07	7.026737e+07	1.294406e+08	1.758241e+08	2.072322e+08	2.270120e+08	2.389980e+08
1e-07	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	2.490753e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	2.490753e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	2.490753e-01	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.490753e-01	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.490753e-01	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.490753e-01	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	2.490753e-01
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
0e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

## Load the data

## Build a population vector

```
load("size_dist_1980.RData") #This is the n-vector for 1980, loads it to a vector called n

n <- n %>% {
  .$N}%>%
  c(rep(0, times = 8))

n_0 <- sum(A[1, 2:14]*n, na.rm = T)

n <- c(n_0, n)

n

## [1] 2.562588e+19 8.674537e+06 3.136387e+11 2.711014e+10 4.391505e+08
## [6] 3.880114e+06 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
## [11] 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

## Project the population

```
project <- popbio::pop.projection(A, n, 30)

pop <- project$stage.vectors %>%
  as.data.frame() %>%
  mutate(Age = as.factor(seq(0:13)-1)) %>%
  gather(Year, N, -Age) %>%
  mutate(Year = as.numeric(as.character(Year))) %>%
  select(Year, Age, N)

ggplot(pop, aes(x = Year, y = log(N), color = Age)) +
  geom_line() +
  theme_bw() +
  geom_hline(yintercept = 40, color = "black", linetype = "dashed", size = 1) +
  geom_vline(xintercept = 22, color = "black", linetype = "dashed", size = 1)
```

## Minimum catch size 1

```
A_min <- A

A_min[3,2] <- exp(-m)
```

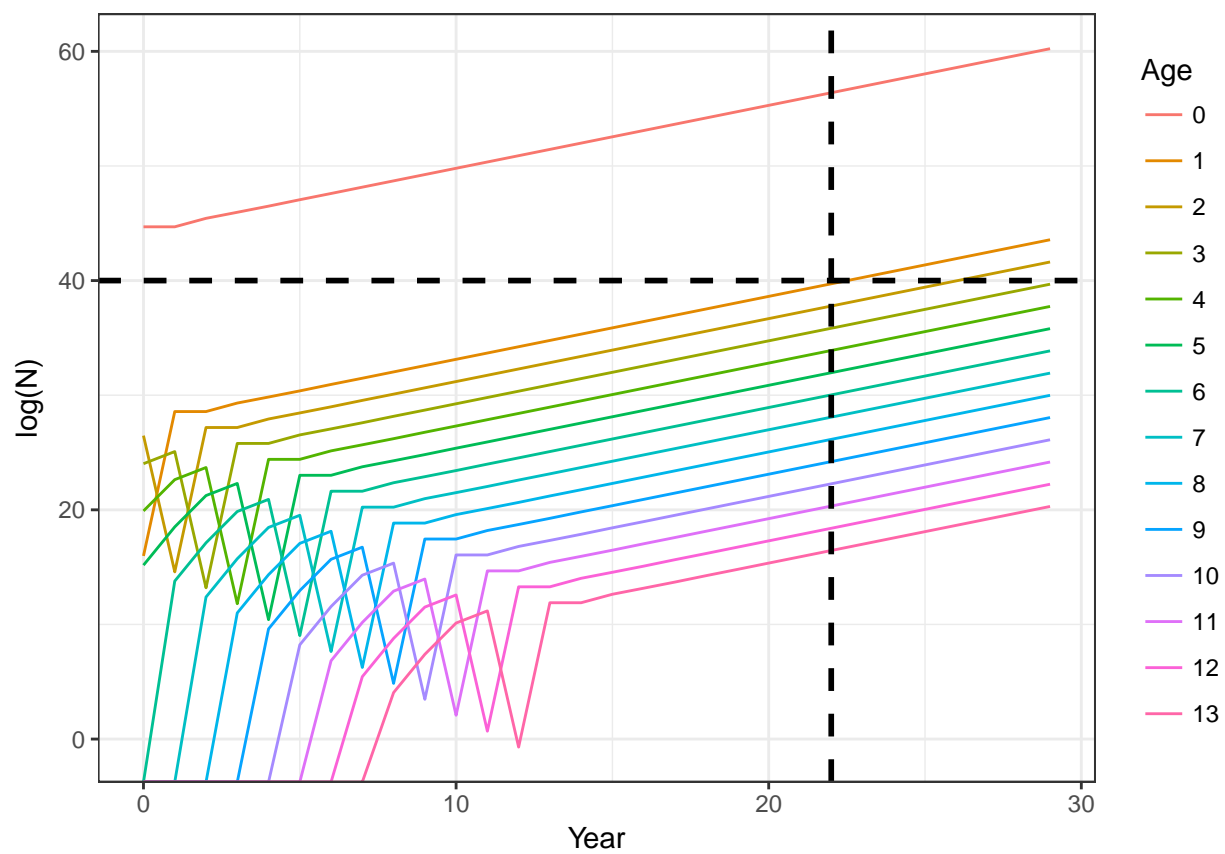


Figure 1: Figure 2 - Population size through time, represented by ages.

```
project_min <- popbio::pop.projection(A_min, n, 30)
```

```
pop <- project_min$stage.vectors %>%
  as.data.frame() %>%
  mutate(Age = as.factor(seq(0:13)-1)) %>%
  gather(Year, N, -Age) %>%
  mutate(Year = as.numeric(as.character(Year))) %>%
  select(Year, Age, N)

ggplot(pop, aes(x = Year, y = log(N), color = Age)) +
  geom_line() +
  theme_bw() +
  geom_hline(yintercept = 40, color = "black", linetype = "dashed", size = 1) +
  geom_vline(xintercept = 22, color = "black", linetype = "dashed", size = 1) +
  geom_vline(xintercept = 18, color = "red", linetype = "dashed", size = 1)
```

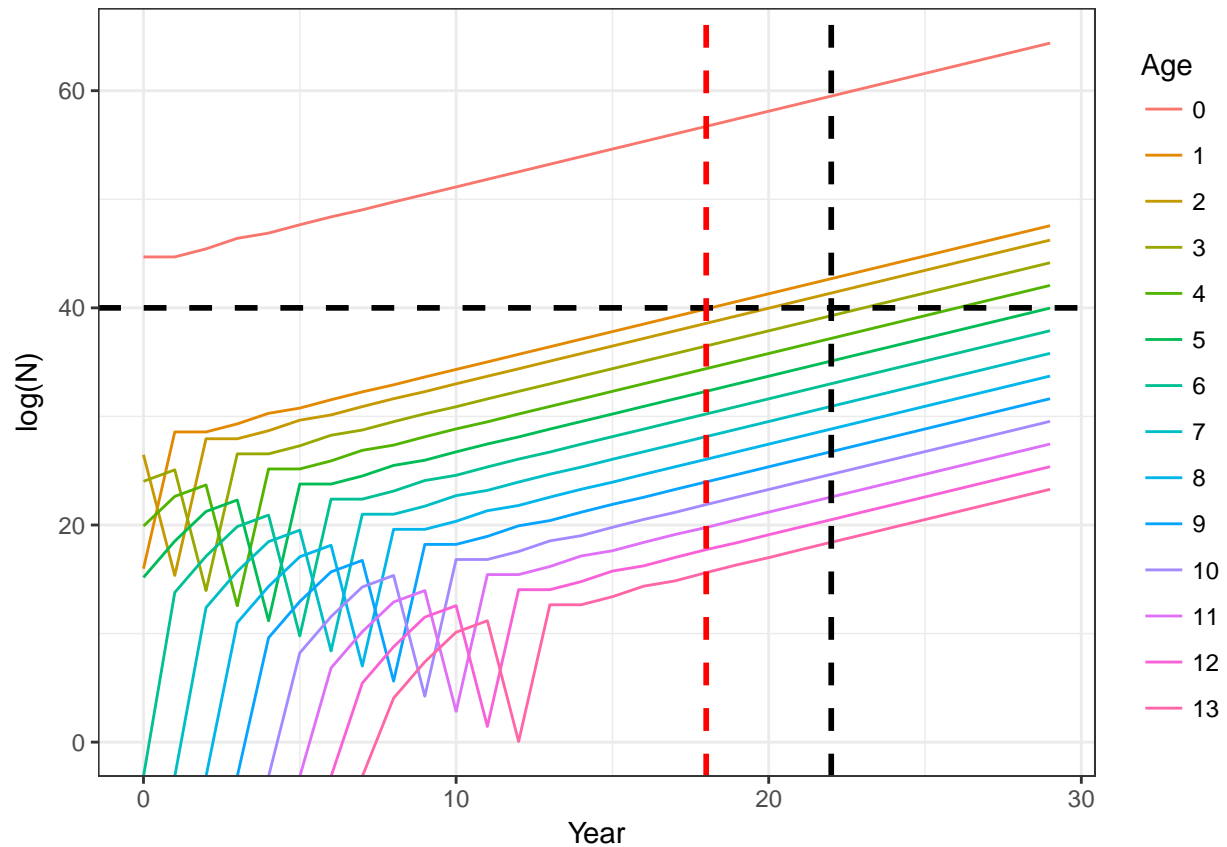


Figure 2: Figure 2 - Population size through time, represented by ages.

```
tot1 <- project$pop.sizes
tot2 <- project_min$pop.sizes
time <- seq(1:30)

data.frame(time, BAU = tot1, Int = tot2) %>%
  gather(scenario, popsize, -time) %>%
  ggplot(aes(x = time, y = log(popsize), color = scenario)) +
```

```
geom_point() +  
geom_line() +  
theme_bw()
```

