






Stage-Structured Population Modeling of Humpback Whales

Swati Tak (s220868)

Humpback whales: life stages and model parameters

Life stage	Calf 	Yearling 	Sub-adult 	Mature adult (reproductive) 	Older adult 
Age range (or time steps)	0 to 1 year old	1 to 2 years old	2 to 5 years old	5 to 12 years old	Above 12 years old
Time interval (Δt)*	1 year	1 year	3 years	8 years	37 years
Population size (N_0) at time=0	6000	6000	6000	6000	6000
Fecundity (f) (reproduction per couple per year)	0	0	0	0.25	0
Mortality (m) (fraction dying per year)	0.03	0.03	0.02	0.01	0.01
Survival rate (s)** (fraction surviving per year)	0.97	0.97	0.94	0.92	0.69

* Average number of years spent at each life stage. ** Calculated as $s = \exp(-m \cdot \Delta t)$ for each time step.

Transition matrix and temporal evolution of the population

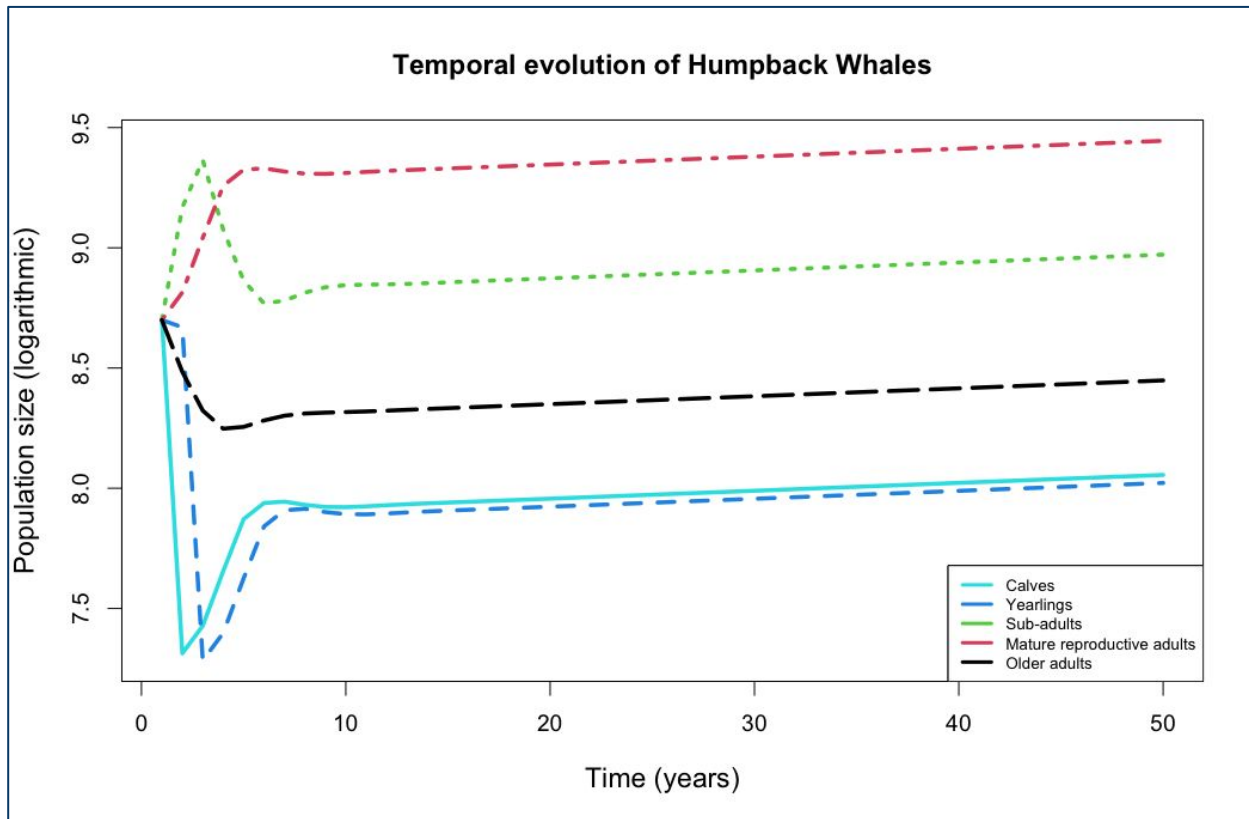
Transition matrix

$$P = \begin{bmatrix} 0 & 0 & 0 & 0.25 & 0 \\ 0.970 & 0 & 0 & 0 & 0 \\ 0 & 0.970 & 0.628 & 0 & 0 \\ 0 & 0 & 0.314 & 0.808 & 0 \\ 0 & 0 & 0 & 0.115 & 0.690 \end{bmatrix}$$

Stage-wise population sizes over time:

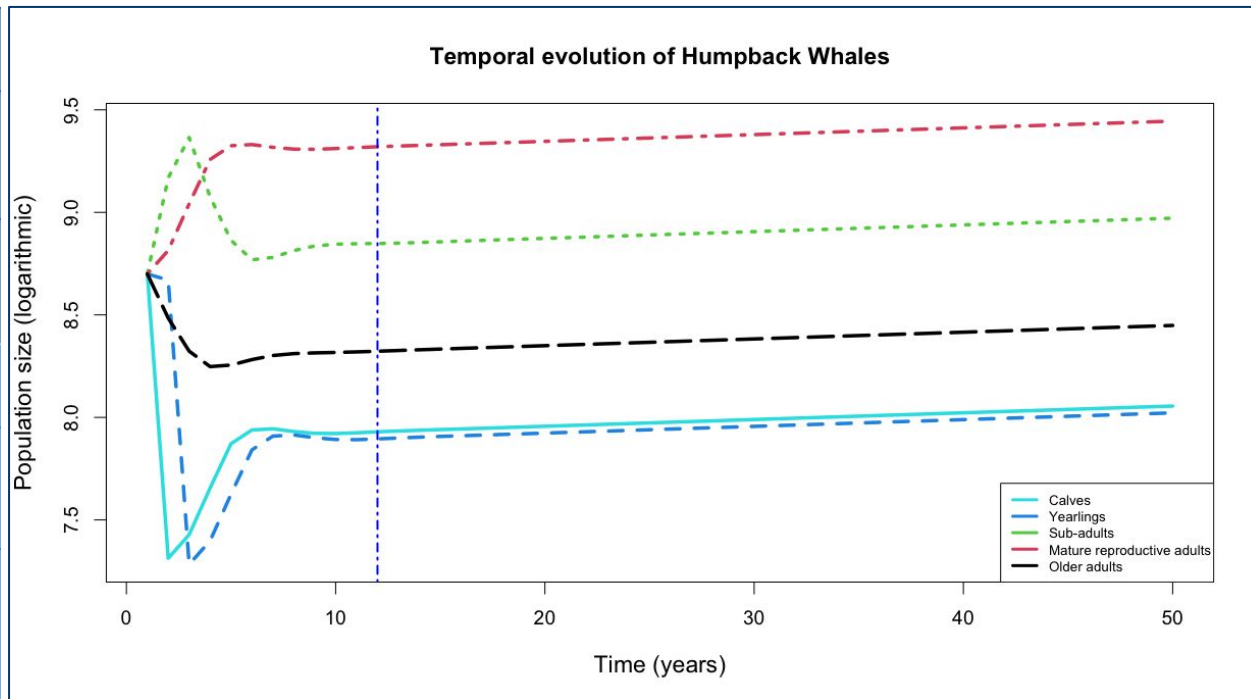
$$N_t = P^t \% N_0$$

	Calves	Yearlings	Sub-adults	Reproductive	Older adults
[1,]	6000.000	6000.000	6000.000	6000.000	6000.000
[2,]	1500.000	5822.673	9589.731	6729.890	4836.743
[3,]	1682.472	1455.668	11671.433	8446.335	4117.464
[4,]	2111.584	1632.748	8740.475	10486.245	3818.692
[5,]	2621.561	2049.177	7072.139	11213.844	3847.705
[6,]	2803.461	2544.083	6428.808	11277.819	3951.702
[7,]	2819.455	2720.606	6505.176	11127.538	4030.919
[8,]	2781.884	2736.127	6724.429	11030.125	4068.296
[9,]	2757.531	2699.667	6877.149	11020.271	4082.873
[10,]	2755.068	2676.034	6937.650	11060.253	4091.804
[11,]	2765.063	2673.643	6952.700	11111.540	4102.587
[12,]	2777.885	2683.343	6959.829	11157.691	4115.953
[13,]	2789.423	2695.786	6973.719	11197.206	4130.511
[14,]	2799.302	2706.983	6994.514	11233.484	4145.126
[15,]	2808.371	2716.570	7018.437	11269.315	4159.408
[16,]	2817.329	2725.371	7042.759	11305.766	4173.407
[17,]	2826.441	2734.064	7066.572	11342.844	4187.282
[18,]	2835.711	2742.907	7089.958	11380.268	4201.145
[19,]	2845.067	2751.903	7113.223	11417.838	4215.039
[20,]	2854.459	2760.982	7136.559	11455.487	4228.971



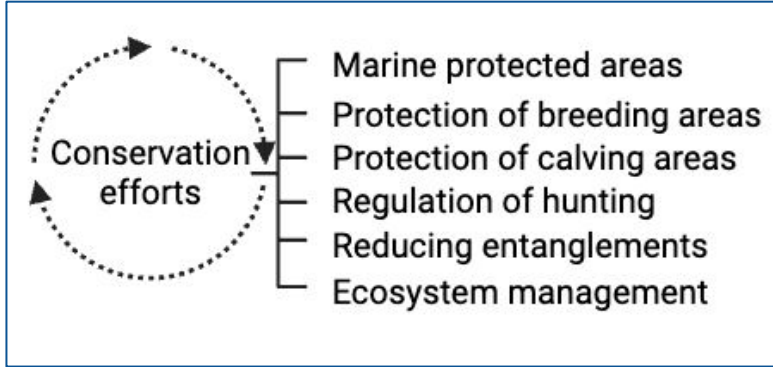
Eigen analysis and comparison with simulation results

lambda	1.004319
r_max $\Rightarrow \log(\text{lambda})$ (per year)	0.0043096
Curve slopes (mean)	0.0043515
Slope calculation	
Slope = Change in x coordinates / Change in y coordinates	
Coordinates for each curves were selected interactively – after the blue vertical line for the first pair of x and y coordinates and at the end of the curve for the second pair of x and y coordinates	



Both the simulation and the eigen analysis have resulted in somewhat similar values for long-term growth rates for the whale population.

Conservation measures



How to apply conversation measures in the model:

1. By reducing mortality rates
2. By increasing survival rates

When conservation efforts are targeted only on calves, the temporal evolution of all stages goes up in numbers.

This is because if more and more calves survive to transition to other stages, there will be more reproductive adults to add to the population size overall.

As can be seen from the population dynamics in the temporal evolution plot, calves and yearlings might show the most need for conservation efforts based on the way their numbers go down in the beginning. However, it is equally important to focus conservation efforts on all stages of humpback whales because calves, yearlings and sub-adults need mature adults and older adults for several reasons such as:

1. Protection from predators
2. Nourishment through mother's milk
3. Critical life skills to survive, feed and migrate

Image: P-shaped rings made by a humpback whale to catch krill.

Mathematical Biology

Weekly Exercise 6

Swati Tak

2023-10-09

Work environment

```
setwd("/Users/swati/Desktop/Mathematical Biology/")
library(demogR)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.3      v readr      2.1.4
## v forcats    1.0.0      v stringr   1.5.0
## v ggplot2    3.4.3      v tibble    3.2.1
## v lubridate  1.9.2      v tidyr     1.3.0
## v purrr      1.0.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(knitr)
```

Species information

```
# Stage-structured population based on age groups and sexual maturity
species <- "Humpback whale"

# Stage structure of humpback whales population
# Stage 1 Age group 1: Calf (<=1 year)
# Stage 2 Age group 2: Yearling (>1 and <2 years)
# Stage 3 Age group 3: Sub-adult(>=2 and <5 years)
# Stage 4 Age group 4: Mature adult (sexually mature 5 to 12 years)
# Stage 5 Age group 5: Fully mature (above 12 years)
```

Parameters: life stages, birth rate, mortality, fecundity and survival

```
# Two humpback whales can give birth to a calf in every 2 to 3 years
# Taking 2 yrs as standard, the intrinsic growth rate per year will be
r <- 0.25

# Population matrix
# 114000 total population in the world based on Google search
# N_0 <- matrix(c(6500, 6000, 9000, 84000, 8500), ncol=1)
N_0 <- matrix(c(6000, 6000, 6000, 6000, 6000), ncol=1)

Stages <- c("Calf (0-1 years)",
            "Yearling (1 to 2 years)",
            "Sub-adult (2 to 5 years)",
            "Mature adult (5 to 12 years)",
            "Fully mature (above 12 years, no reproduction)")

# Whale years lived in respective stage interval
Time.interval <- c(1, 1, 3, 8, 37)

# Stage-specific fertility rates
Fecundity <- c(0,0,0,r,0)

# Mortality rate at each life stage
# Arbitrary but higher at calf stage because deaths of calves and yearlings
# is more common than of adults, due to killings and predation
Mortality <- c(0.02, 0.03, 0.02, 0.01, 0.01)

# Set a for loop to calculate survival rates for each time interval
n <- length(Time.interval)
Survival_rates <- rep(0, n)

for (i in 1:(n)) {
  Survival_rates[i] <- exp(-Mortality[i] * Time.interval[i])
}

# Display the calculated Survival_rates and delta_t vectors
print(Survival_rates)
```

```
## [1] 0.9801987 0.9704455 0.9417645 0.9231163 0.6907343
```

Transition matrix (Leslie projection matrix)

```
# Create the Leslie matrix (projection or transition matrix)
f <- Fecundity
t <- Time.interval
s <- Survival_rates

# Create the transition matrix
```



```
P <- matrix(c(
  s[1]*(1-(1/t[1])), 0, 0, f[4], f[5],
  s[1]/t[1], s[2]*(1-(1/t[2])), 0, 0, 0,
  0, s[2]/t[2], s[3]*(1-(1/t[3])), 0, 0,
  0, 0, s[3]/t[3], s[4]*(1-(1/t[4])), 0,
  0, 0, 0, s[4]/t[4], s[5]),
  nr=5, byrow=TRUE)

print(P)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.0000000 0.0000000 0.0000000 0.2500000 0.0000000
## [2,] 0.9801987 0.0000000 0.0000000 0.0000000 0.0000000
## [3,] 0.0000000 0.9704455 0.6278430 0.0000000 0.0000000
## [4,] 0.0000000 0.0000000 0.3139215 0.8077268 0.0000000
## [5,] 0.0000000 0.0000000 0.0000000 0.1153895 0.6907343
```

Simulation for temporal evolution

```
# Simulate population growth through a for loop
time <- 50
N_projections <- matrix(0, nrow=nrow(P), ncol= time + 1)
N_projections[,1] <- N_0

for (i in 1:time) {
  N_projections[,i+1] <- P %*% N_projections[,i]
}

# Wrangle the simulation results
df <- t(N_projections)
colnames(df) <- c("Calves", "Yearlings", "Sub-adults",
  "Mature adults (reproductive)", "Older adults")
head(df)
```

```
##           Calves Yearlings Sub-adults Mature adults (reproductive) Older adults
## [1,] 6000.000 6000.000 6000.000 6000.000 6000.000
## [2,] 1500.000 5881.192 9589.731 6729.890 4836.743
## [3,] 1682.472 1470.298 11728.222 8446.335 4117.464
## [4,] 2111.584 1649.157 8790.327 10504.073 3818.692
## [5,] 2626.018 2069.772 7119.363 11243.894 3849.762
## [6,] 2810.973 2574.020 6478.443 11316.915 3956.591
```

```
# Removing the 51st row
df <- head(df, -1)

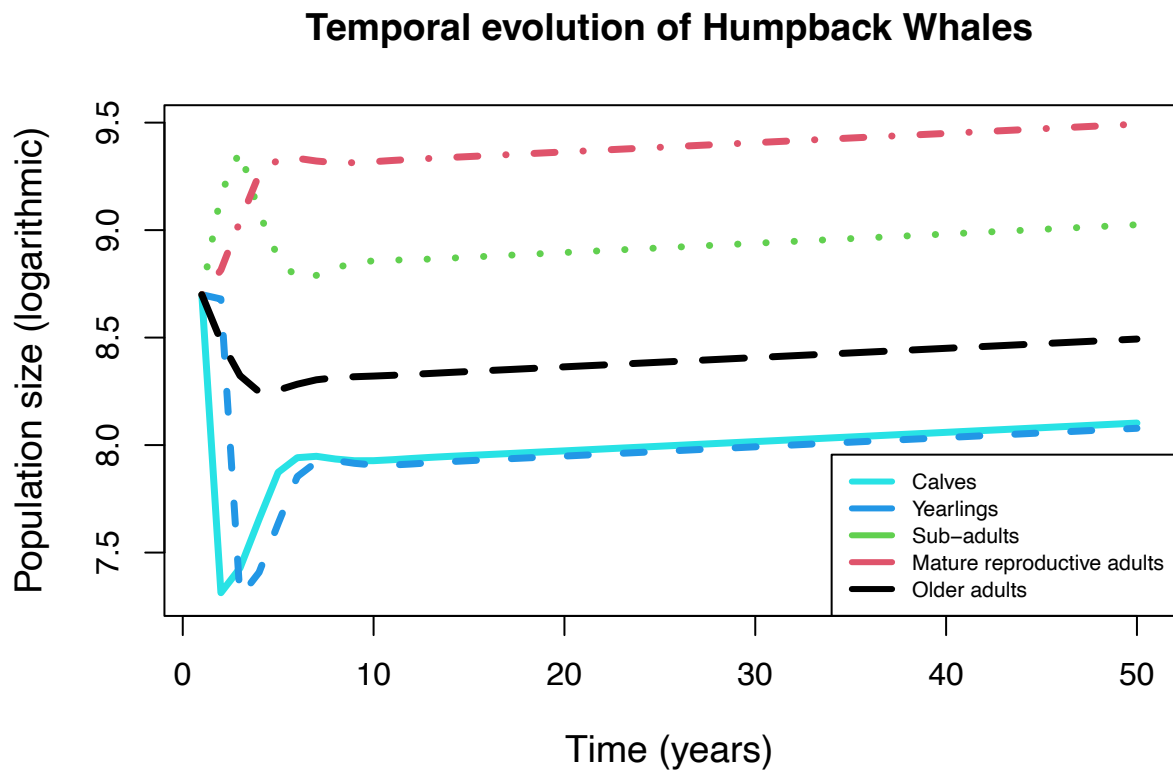
# Taking log of the population values
df <- log(df)
```


Results as a plot

```
# Increase font size for axes annotations
# par(cex=1.2)

# Plot the numerical solution as log(N) over time
matplot(1:time, df, type="l",
        lwd=c(3.5),
        col=5:1,
        cex.lab=1.2,
        xlab="Time (years)",
        ylab="Population size (logarithmic)",
        main="Temporal evolution of Humpback Whales")

# Add a legend
legend("bottomright",
      legend = c("Calves", "Yearlings", "Sub-adults",
                  "Mature reproductive adults", "Older adults"),
      col = 5:1,
      lwd = 3,
      cex = 0.7)
```



Growth rate from curve slopes

```
# Initialize a vector to store the slopes for each curve
slopes <- numeric(5)

# Loop through each curve
# for (i in 1:5) {
#   cat("Select starting point for Curve", i, "equilibrium phase:\n")
#   point1 <- locator(1)
#
#   cat("Select ending point for Curve", i, "equilibrium phase:\n")
#   point2 <- locator(1)

# Extract the x and y coordinates from the selected points
# x1 <- point1$x
# y1 <- point1$y
# x2 <- point2$x
# y2 <- point2$y

# Calculate the change in y and change in x
# delta_y <- y2 - y1
# delta_x <- x2 - x1

# Calculate the slope and store it in the 'slopes' vector
# slopes[i] <- delta_y / delta_x

# Print the calculated slope for the current curve
# cat("Slope for Curve", i, ":", slopes[i], "\n")
# }

# Manually inputting values for slope calculation
# as the above interactive feature doesn't work when knitting RMD to PDF
# These are from the curve for older mature adult whale
delta_y <- 8.105312 - 7.949865
delta_x <- 49.87822 - 12.60319

# Calculate slope for Older mature adult based on manually inputted values above
slope <- delta_y/delta_x

# Print the slope
cat("Slope for the curve:", slope, "\n")
```

```
## Slope for the curve: 0.004170272
```

```
# Print slopes for all curves based on interactive input for x and y coordinates
# for each stage curve as performed in R script
Slopes <- c(0.004308599, 0.004460539, 0.004385807, 0.004307011, 0.004295688)
cat("Slopes for all curves:", Slopes)
```

```
## Slopes for all curves: 0.004308599 0.004460539 0.004385807 0.004307011 0.004295688
```

```
cat("As evident from the slope values, the growth rate is in the range of 0.429 to 0.446.")
```

```
## As evident from the slope values, the growth rate is in the range of 0.429 to 0.446.
```

Eigen analysis

```
# Apply eigen to get the eigen analysis results
ea <- eigen(P)

# Calculate the dominant eigen value, that is, asymptomatic growth rate (lambda)
lambda <- max(Re(ea$values))
pop_structure <- ea$vectors

pop_structure_df <- as.data.frame(pop_structure)
colnames(pop_structure_df) <- colnames(df)
kable(pop_structure_df)
```

Calves	Yearlings	Sub-adults	Mature adults (reproductive)	Older adults
-0.1938717+0i	0+0i	0.2008087-0.0251248i	0.2008087+0.0251248i	0.1844992+0i
-0.1892156+0i	0+0i	0.2237879-0.3099278i	0.2237879+0.3099278i	-0.6551117+0i
-0.4877429+0i	0+0i	-0.7920541+0.0000000i	-0.7920541+0.0000000i	0.7033441+0i
-0.7788360+0i	0+0i	0.3222285+0.2694720i	0.3222285-0.2694720i	-0.2037266+0i
-0.2865879+0i	1+0i	-0.0028153-0.0954169i	-0.0028153+0.0954169i	0.0243155+0i

```
# Calculate maximum population growth rate
r_max <- log(lambda)
cat("The maximum intrinsic growth rate for the Humpback whale populations is:",
    r_max)
```

```
## The maximum intrinsic growth rate for the Humpback whale populations is: 0.004309572
```

```
# Expected population trend over time
if (lambda >= 1) {
  cat("\nThe", species, "population is expected to grow over time as the dominant
    eigen value is either equal to or greater than 1. Dominant eigen value:",
    lambda)
} else if (lambda < 1) {
  cat("\nThe", species, "population is expected to decline over time as the
    dominant eigen value is below 1. Dominant eigen value:", lambda)
}
```

```
##
```

```
## The Humpback whale population is expected to grow over time as the dominant
## eigen value is either equal to or greater than 1. Dominant eigen value: 1.004319
```

```
# Comparison of results from eigen analysis and temporal simulation  
mean(Slopes)
```

```
## [1] 0.004351529
```

```
r_max
```

```
## [1] 0.004309572
```