

Lab 2

Sawyer Balint

This document is available at <https://github.com/sjbalint/BI521/tree/main/scripts/labs>

```
#import packages
library(tidyverse) #for data wrangling
library(here) #for filepath management
library(ggsci) #for colors
library(scales) #for log axis breaks

#custom graphing theme
#including geoms to reduce repetition
theme <- list(
  theme_classic(),
  scale_color_jama(),
  scale_fill_jama(),
  theme(legend.position="right"),
  geom_line(),
  labs(x="Year",y="Population", color="Age")
)

logtheme <- list(
  scale_y_log10(labels = trans_format(log10, math_format(10^.x))),
  annotation_logticks(sides="l")
)
```

TASK 1: Calculate the age distribution in Year 2 if the population in Year 1 is composed of a single age-class, for each of the four Year-1 populations below.

```
#make the matrix
M <- matrix(c(0,0,0,2000,
              0.005,0,0,0,
              0,0.2,0,0,
              0,0,0.5,0),
            ncol=4, nrow=4, byrow=TRUE)

#check structure
M
```

```
##      [,1] [,2] [,3] [,4]
## [1,] 0.000 0.0  0.0 2000
## [2,] 0.005 0.0  0.0   0
## [3,] 0.000 0.2  0.0   0
## [4,] 0.000 0.0  0.5   0
```

```
#make vectors
v1_1 <- c(1000,0,0,0)
v2_1 <- c(0,1000,0,0)
v3_1 <- c(0,0,1000,0)
v4_1 <- c(0,0,0,1000)

#calculate age distributions and print
M %*% v1_1
```

```
##      [,1]
## [1,]    0
## [2,]    5
## [3,]    0
## [4,]    0
```

```
M %*% v2_1
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]  200
## [4,]    0
```

```
M %*% v3_1
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,]    0
## [4,]  500
```

```
M %*% v4_1
```

```
##      [,1]
## [1,] 2e+06
## [2,] 0e+00
## [3,] 0e+00
## [4,] 0e+00
```

TASK 2: Calculate the age distribution in Year 2 if the population in Year 1 is composed of equal age-classes.

```
#make vector
v5_1 <- c(1000,1000,1000,1000)

#calculate age distribution and print
M %*% v5_1
```

```
##      [,1]
## [1,] 2e+06
## [2,] 5e+00
## [3,] 2e+02
## [4,] 5e+02
```

What do you notice comparing $v_{ex5}(2)$ to the results for the task before?

$v_{ex5}(2)$ is the maximum population of each age distribution from the previous task.

TASK 3: Calculate the age distribution in Year 4 for the above initial age distributions $v_{ex1}(1), v_{ex2}(1), v_{ex3}(1), v_{ex4}(1), v_{ex5}(1)$.

```
#calculate with repetition as requested
```

```
v1_2 <- M %*% v1_1
```

```
v1_3 <- M %*% v1_2
```

```
v1_4 <- M %*% v1_3
```

```
#print results
```

```
v1_4
```

```
##      [,1]
```

```
## [1,]  0.0
```

```
## [2,]  0.0
```

```
## [3,]  0.0
```

```
## [4,]  0.5
```

```
#calculate efficiently
```

```
v1_4 <- M %*% M %*% M %*% v1_1
```

```
v1_4 #same answer as before
```

```
##      [,1]
```

```
## [1,]  0.0
```

```
## [2,]  0.0
```

```
## [3,]  0.0
```

```
## [4,]  0.5
```

```
#test all other scenarios
```

```
M %*% M %*% M %*% v2_1
```

```
##      [,1]
```

```
## [1,] 2e+05
```

```
## [2,] 0e+00
```

```
## [3,] 0e+00
```

```
## [4,] 0e+00
```

```
M %*% M %*% M %*% v3_1
```

```
##      [,1]
```

```
## [1,]    0
```

```
## [2,] 5000
```

```
## [3,]    0
```

```
## [4,]    0
```

```
M %*% M %*% M %*% v4_1
```

```
##      [,1]
## [1,]    0
## [2,]    0
## [3,] 2000
## [4,]    0
```

```
M %*% M %*% M %*% v5_1
```

```
##      [,1]
## [1,] 2e+05
## [2,] 5e+03
## [3,] 2e+03
## [4,] 5e-01
```

TASK 4: Using a for loop, simulate 100 years of the salmon population with initial population distribution $v_{ex6} = [1000, 500, 250, 125]$. The result should be a data.frame with the first column denoting year, and four additional columns (one for the abundance of each year-class).

```
#empty list to store results
result.list <- list()

#initial vector of age distribution
v <- c(1000, 500, 250, 125)

for (year in 0:100){

  #make data frame of results
  result <- data.frame(year=year,
                        N_1 = v[1],
                        N_2 = v[2],
                        N_3 = v[3],
                        N_4 = v[4])

  #record result
  result.list <- append(result.list, list(result))

  #perform calculation
  #perform calculation after logging because we're starting at 0
  #to include initial conditions
  v <- M %*% v
}

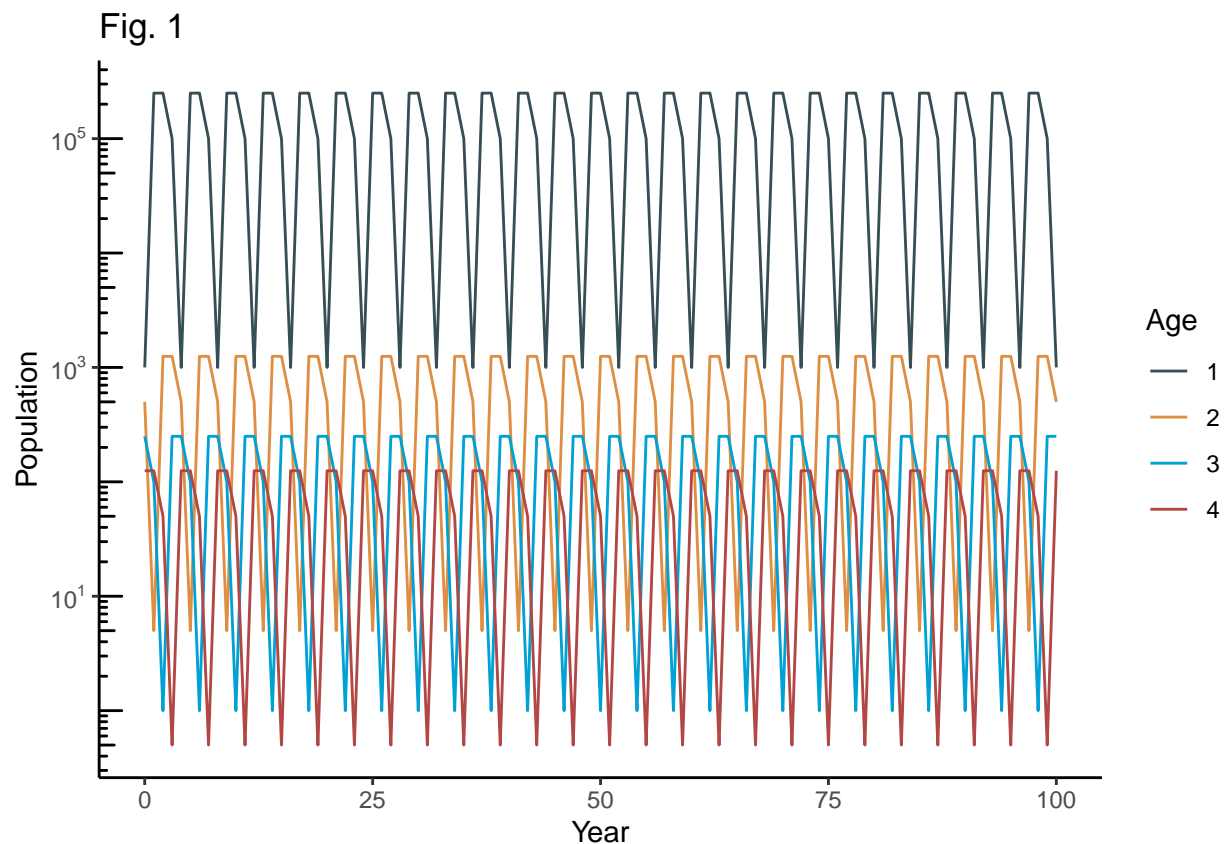
#final dataframe
#binding dataframes from a list is faster than rbinding rows within the for-loop
result.df <- bind_rows(result.list)

#view results
head(result.df)
```

```
##   year   N_1  N_2 N_3   N_4
## 1    0   1000  500 250 125.0
## 2    1 250000    5  100 125.0
## 3    2 250000 1250    1   50.0
## 4    3 100000 1250 250    0.5
## 5    4   1000  500 250 125.0
## 6    5 250000    5  100 125.0
```

```
#long dataframe for plotting
plot.df <- result.df %>%
  pivot_longer(!year) %>%
  mutate(name=factor(name, labels=c(1:4)))

#make a quick plot
#these are some impressive oscillations...
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  logtheme+
  labs(title="Fig. 1")
```



TASK 5: Using a for loop, simulate 100 years of the salmon population with initial population distribution $v_{ex7} = [1000, 500, 250, 125, 0]$. The result should be a data.frame with the first column denoting year, and five additional columns (one for the abundance of each year-class).

```

#make the new matrix
M <- matrix(c(0,0,0,2000*0.9,2000,
              0.005,0,0,0,0,
              0,0.2,0,0,0,
              0,0,0.5,0,0,
              0,0,0,0.1,0),
            ncol=5, nrow=5, byrow=TRUE)

#wrap this in a function to save both of us some time
fishyfunction <- function(M, v){

  #empty list to store results
  result.list <- list()

  for (year in 0:100){

    #make data frame of results
    result <- data.frame(year=year,
                        N_1 = v[1],
                        N_2 = v[2],
                        N_3 = v[3],
                        N_4 = v[4],
                        N_5 = v[5])

    #record result
    result.list <- append(result.list, list(result))

    #perform calculation
    #perform calculation after logging because we're starting at 0
    #to include initial conditions
    v <- M %*% v

  }

  #final dataframe
  result.df <- bind_rows(result.list)

  return(result.df)

}

#initial vector of age distribution
v <- c(1000, 500, 250, 125, 0)

#custom function
century.df <- fishyfunction(M, v)

head(century.df)

```

```

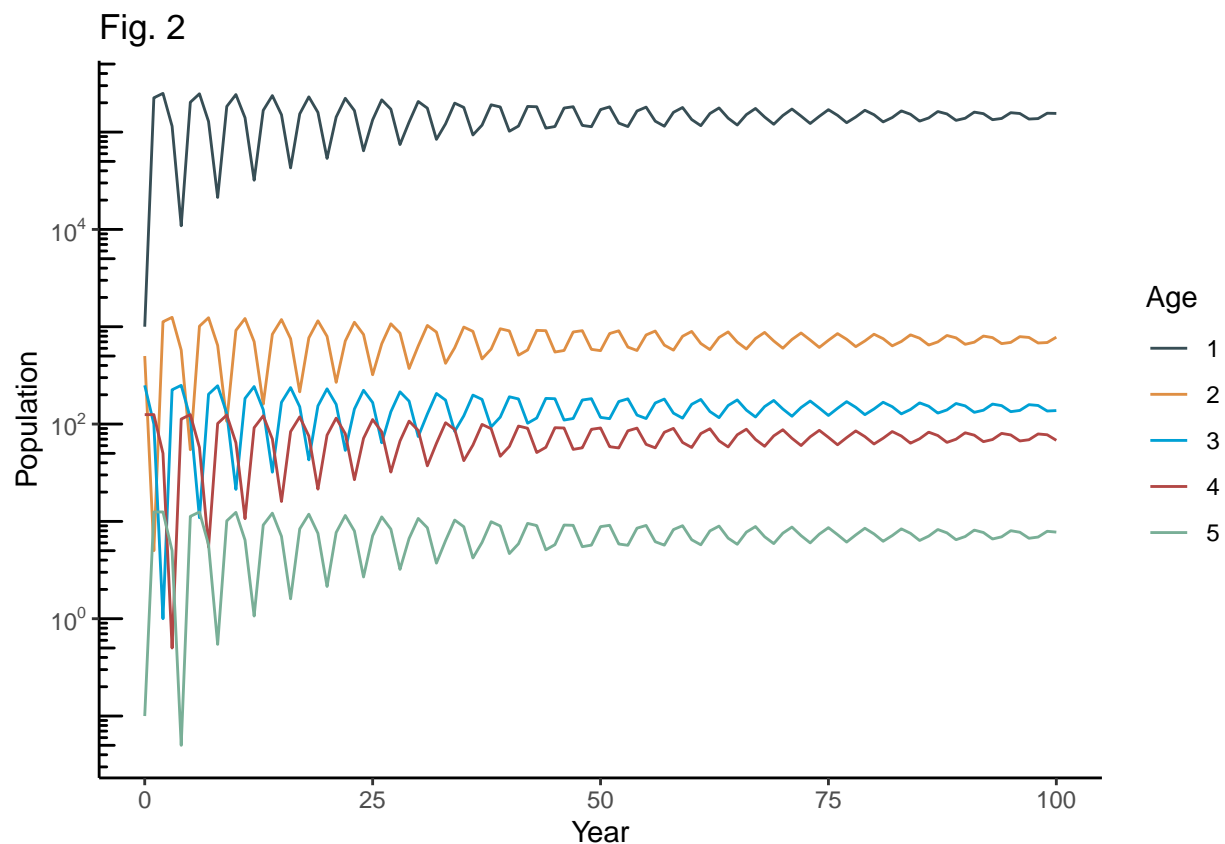
##   year   N_1   N_2 N_3   N_4   N_5
## 1    0   1000 500.0 250 125.0  0.00
## 2    1 225000    5.0 100 125.0 12.50
## 3    2 250000 1125.0    1  50.0 12.50

```

```
## 4    3 115000 1250.0 225    0.5  5.00
## 5    4  10900  575.0 250  112.5  0.05
## 6    5 202600   54.5 115  125.0 11.25
```

```
#long dataframe for plotting
plot.df <- century.df %>%
  pivot_longer(!year) %>%
  mutate(name=factor(name, labels=c(1:5)),
         #can't have zeros with a log scale, so approximate zero instead
         value=ifelse(value==0,0.1,value))

#make a plot
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  logtheme+
  labs(title="Fig. 2")
```



What is happening different in this simulation?

The oscillations in population within age classes are decreasing over time. The population is approaching a steady state.

TASK 6: Apply the inverse matrix to the final year of the simulations you ran before.

```
#final year of previous simulation
v <- as.numeric(century.df[nrow(century.df),2:6])
```

```
#apply inverse matrix
M
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.000 0.0 0.0 1800.0 2000
## [2,] 0.005 0.0 0.0 0.0 0
## [3,] 0.000 0.2 0.0 0.0 0
## [4,] 0.000 0.0 0.5 0.0 0
## [5,] 0.000 0.0 0.0 0.1 0
```

```
solve(M)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.000000e+00 200 0 0 0
## [2,] 0.000000e+00 0 5 0 0
## [3,] 0.000000e+00 0 0 2 0
## [4,] -1.156482e-20 0 0 0 10
## [5,] 5.000000e-04 0 0 0 -9
```

```
inverse_V <- signif(solve(M) %*% v, digits=3)
inverse_V
```

```
##      [,1]
## [1,] 1.56e+05
## [2,] 6.88e+02
## [3,] 1.36e+02
## [4,] 7.76e+01
## [5,] 7.91e+00
```

```
#inverse M should be the same as the 99th row
as.numeric(century.df[nrow(century.df)-1,2:6])
```

```
## [1] 1.562095e+05 6.882060e+02 1.361273e+02 7.759957e+01 7.911427e+00
```

What value did you get?

1.56×10^5 , 688, 136, 77.6, 7.91

Explain the result in terms of the basic concept of the matrix inverse.

The matrix inverse is undoing the effect of the matrix. So, applying the matrix inverse once allows us to “go back in time” one time step. Indeed, we find that the values match the second to last year of the simulation.

TASK 7: Iterate the population dynamics of `M_notsalmon_1` for 100 steps (years) using the same $v_{ex7} = [1000, 500, 250, 125, 0]$ as above.


```

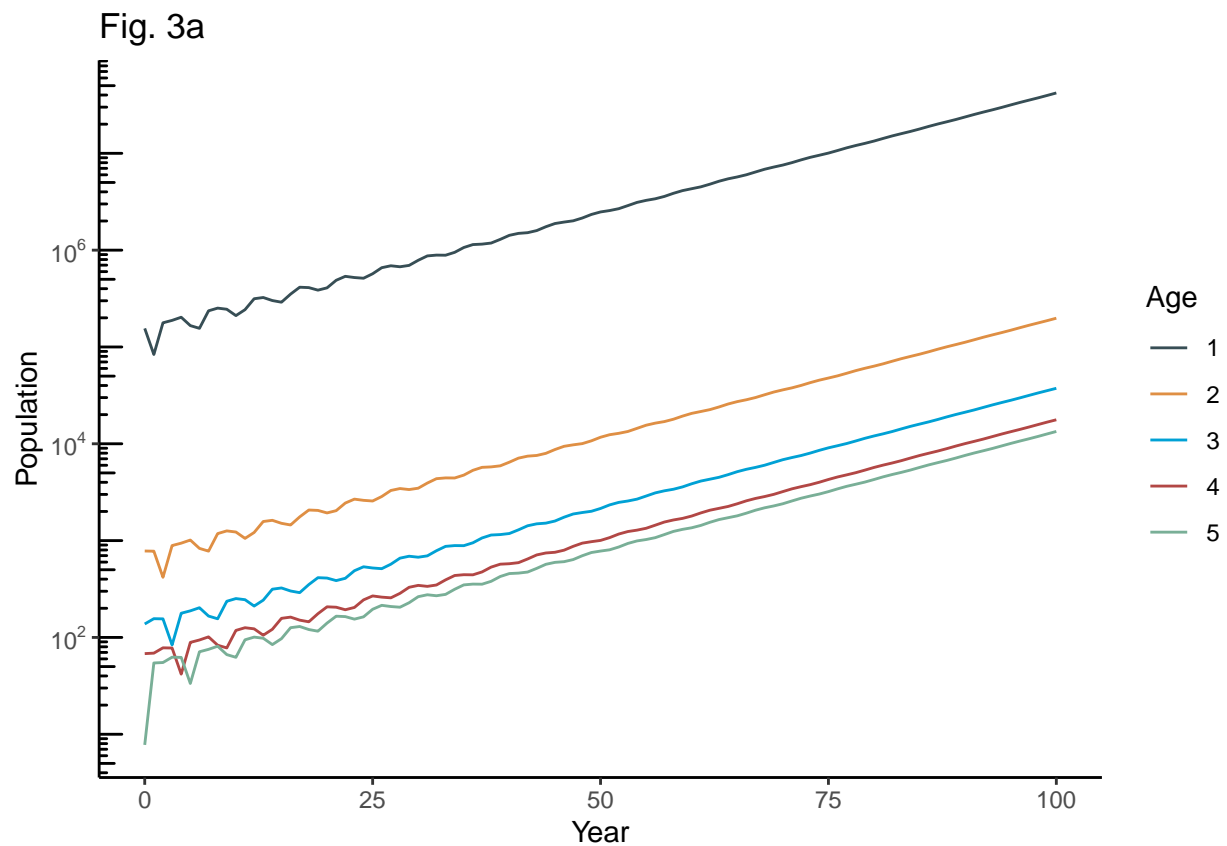
#make the new matrix
M <- matrix(c(0,0,0,2000*0.5,2000,
              0.005,0,0,0,0,
              0,0.2,0,0,0,
              0,0,0.5,0,0,
              0,0,0,0.8,0),
            ncol=5, nrow=5, byrow=TRUE)

#custom function
century.df <- fishyfunction(M, v)

#long dataframe for plotting
plot.df <- century.df %>%
  pivot_longer(!year) %>%
  mutate(name=factor(name, labels=c(1:5)),
         #can't have zeros with a log scale, so approximate zero instead
         value=ifelse(value==0,0.1,value))

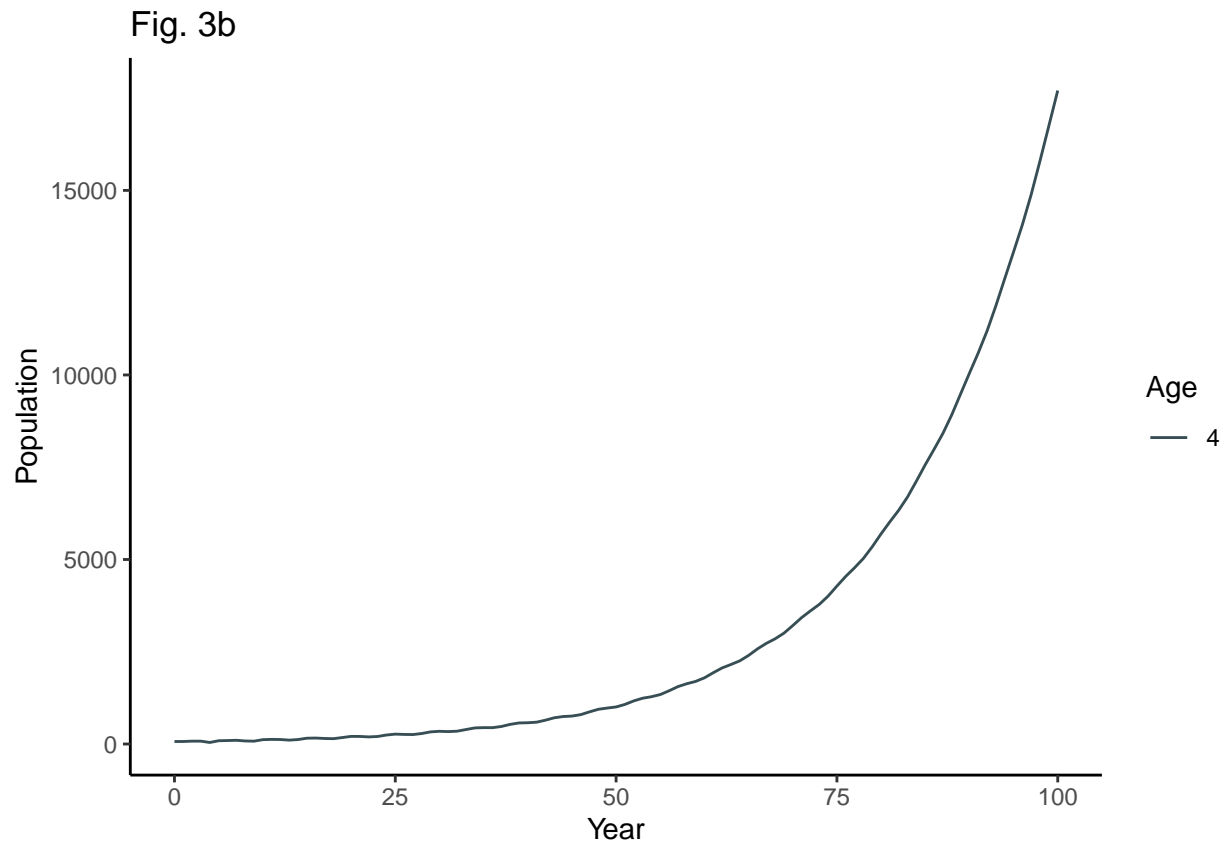
#make a plot
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  logtheme+
  labs(title="Fig. 3a")

```



```
#make a plot for age 4

plot.df %>%
  filter(name=="4") %>%
  ggplot(aes(year, value, color=name))+
  theme+
  labs(title="Fig. 3b")
```



Plot the abundance of Age-4 fish across the 100 year simulation. What happens?

The population of all age classes is increasing exponentially (which appears linear on log10 scale) and does not reach steady state.

TASK 8: Simulate the population for 100 years starting with the population distributed as the first eigenvector of `M_notsalmon_1`.

```
#initial vector of age distribution
v <- 1000*Re(eigen(M)$vectors[,1]) #multiply by 1000 to have numbers greater than 1

#custom function
century.df <- fishyfunction(M, v)

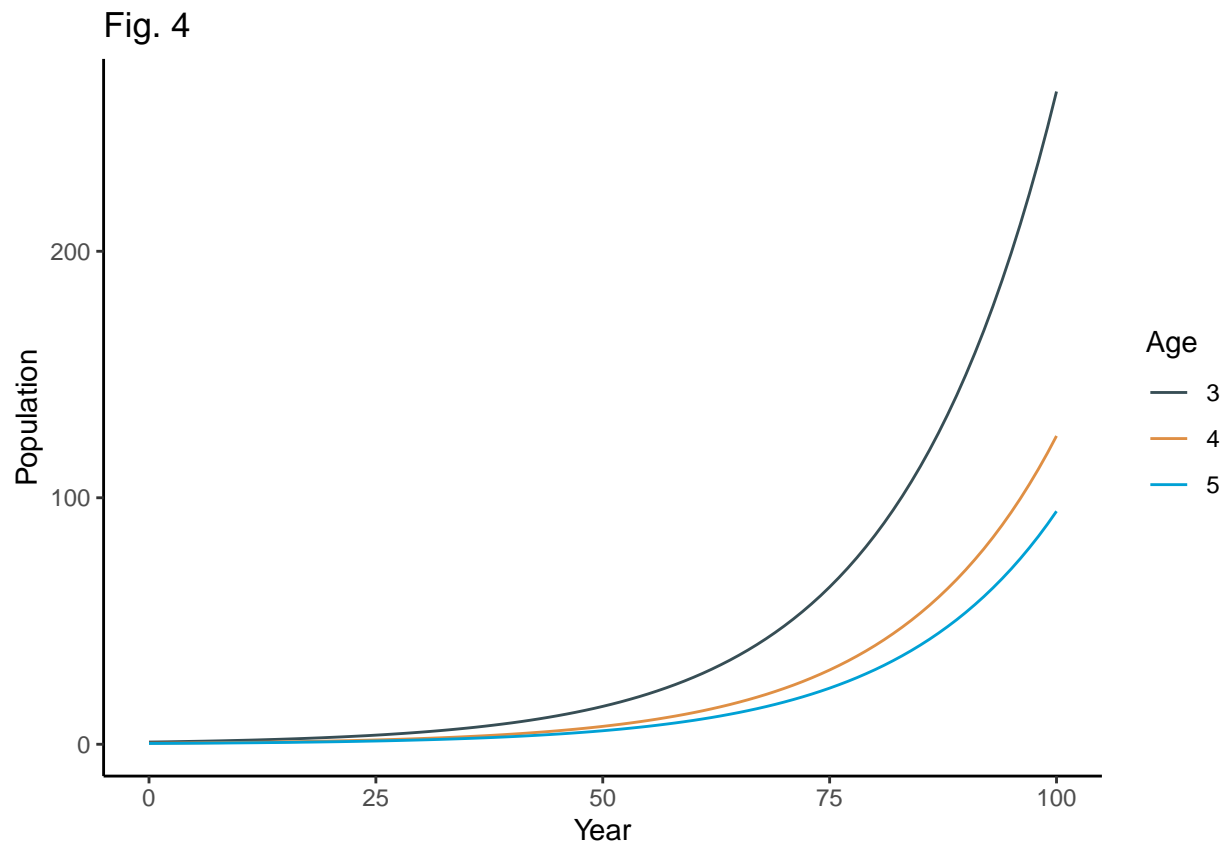
#long dataframe for plotting
plot.df <- century.df %>%
  pivot_longer(!year) %>%
```

```

mutate(name=factor(name, labels=c(1:5)),
       #can't have zeros with a log scale, so approximate zero instead
       value=ifelse(value==0,0.1,value)) %>%
filter(as.numeric(name)>=3)

#make a plot
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  #logtheme+
  labs(title="Fig. 4")

```



The abundances are all growing exponentially, but something is being preserved. Try plotting again but dividing each by the total number of individuals that year.

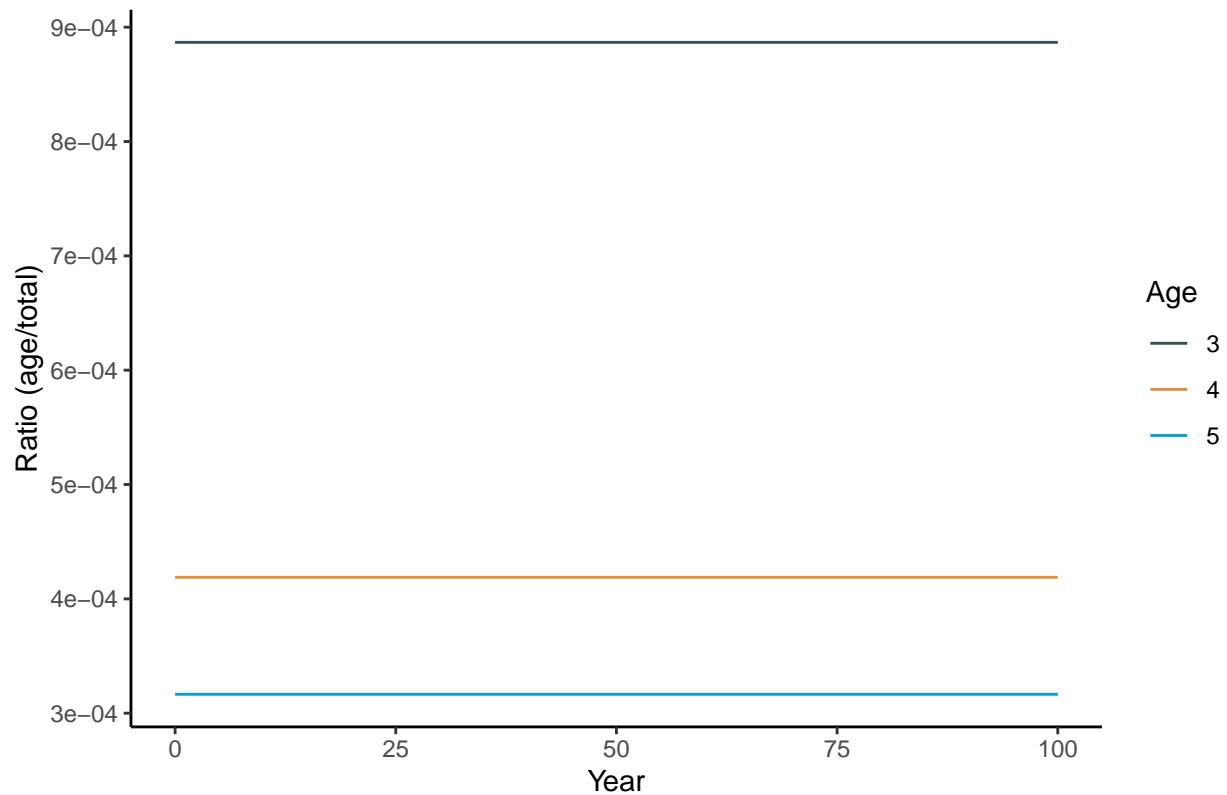
```

#long dataframe for plotting
plot.df <- century.df %>%
  mutate(N_total = N_1+N_2+N_3+N_4+N_5) %>%
  pivot_longer(!c("year", "N_total")) %>%
  mutate(name=factor(name, labels=c(1:5)),
         value=value/N_total)%>%
  filter(as.numeric(name)>=3)

#make a plot
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  labs(title="Fig. 5", y="Ratio (age/total)")

```

Fig. 5



TASK 9: Simulate the population for 100 years starting from at least two other initial distributions not equal to the eigenvector distribution above.

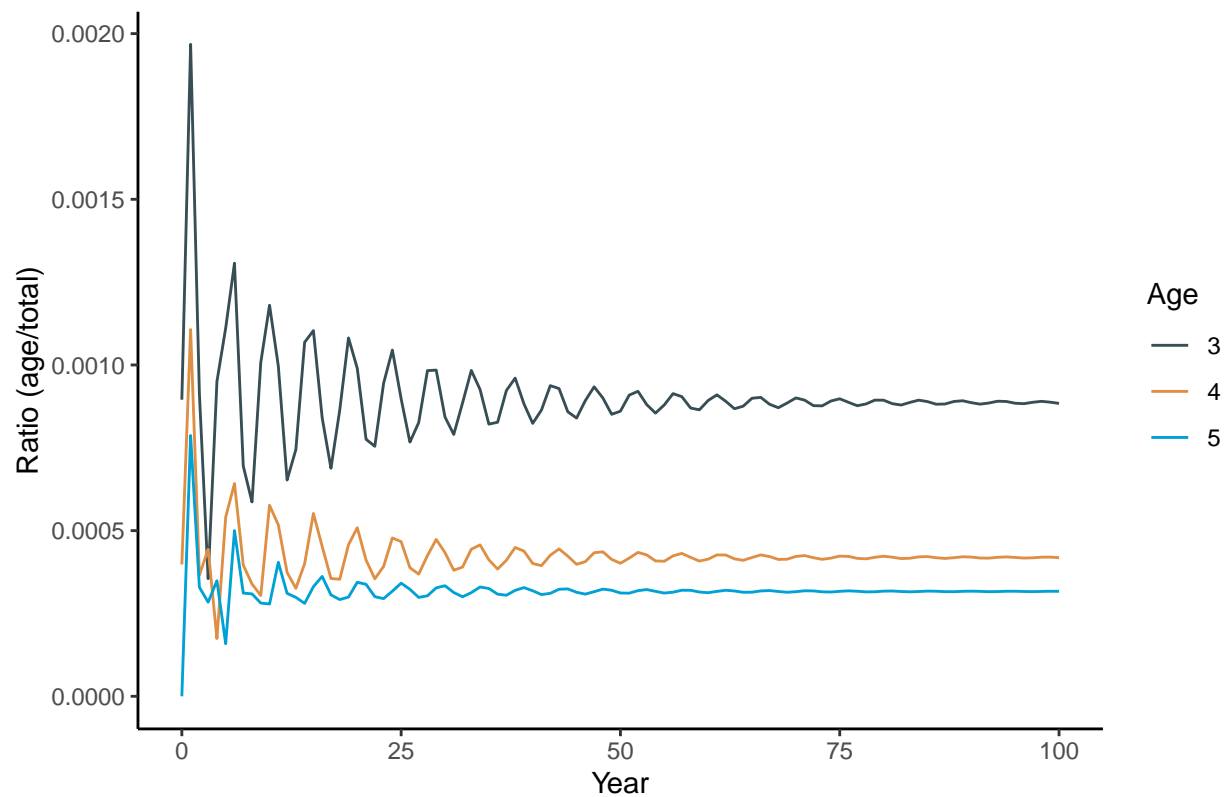
```
#initial vector of age distribution
v <- c(1, 0.004, 0.0009, 0.0004, 0) * 1000

#custom function
century.df <- fishyfunction(M, v)

#long dataframe for plotting
plot.df <- century.df %>%
  mutate(N_total = N_1+N_2+N_3+N_4+N_5) %>%
  pivot_longer(!c("year", "N_total")) %>%
  mutate(name=factor(name, labels=c(1:5)),
         value=value/N_total)%>%
  filter(as.numeric(name)>=3)

#make a plot
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  #logtheme+
  labs(title="Fig. 6", y="Ratio (age/total)")
```

Fig. 6



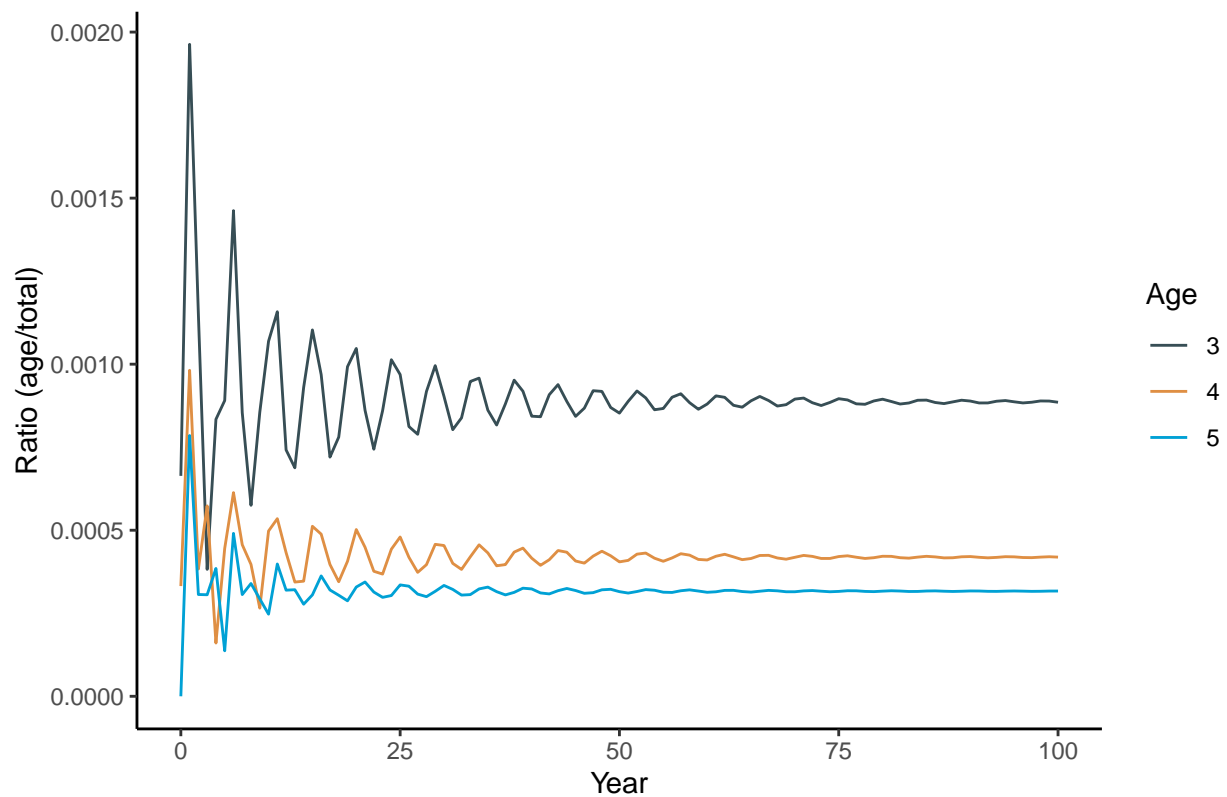
```
#initial vector of age distribution
v <- c(1.5, 0.005, 0.001, 0.0005, 0) * 1000

#custom function
century.df <- fishyfunction(M, v)

#long dataframe for plotting
plot.df <- century.df %>%
  mutate(N_total = N_1+N_2+N_3+N_4+N_5) %>%
  pivot_longer(!c("year", "N_total")) %>%
  mutate(name=factor(name, labels=c(1:5)),
         value=value/N_total)%>%
  filter(as.numeric(name)>=3)

#make a plot
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  #logtheme+
  labs(title="Fig. 7", y="Ratio (age/total)")
```

Fig. 7



Plot the abundances of Age-3, Age-4, and Age-5 through time, but as before divide by the total number of individuals in that year. Is the population distribution trending towards our previous result?

In both cases, the population is trending towards the result predicted by the eigenvectors.

Optional Exercises:

```
#initial matrix
M <- matrix(c(0,0,0,2000,
              0.005,0,0,0,
              0,0.2,0,0,
              0,0,0.5,0),
            ncol=4, nrow=4, byrow=TRUE)

#initial vector of age distribution
v <- c(1000,500,250,125)

#empty list to store results
result.list <- list()

for (year in 0:100){

  #recalculate fecundity
```

```

#i made my own formula, because the one in the lab didn't work
#the idea is that if N_4 is zero F is large, but when N_4 is large F drops to 1
#the original equation didn't work in part because F would drop to 0, causing population crash
M[1,4] <- 1999*exp((1-v[4]))+0.1

#make data frame of results
result <- data.frame(year=year,
                     N_1 = v[1],
                     N_2 = v[2],
                     N_3 = v[3],
                     N_4 = v[4])

#record result
result.list <- append(result.list, list(result))

#perform calculation
v <- M %*% v
}

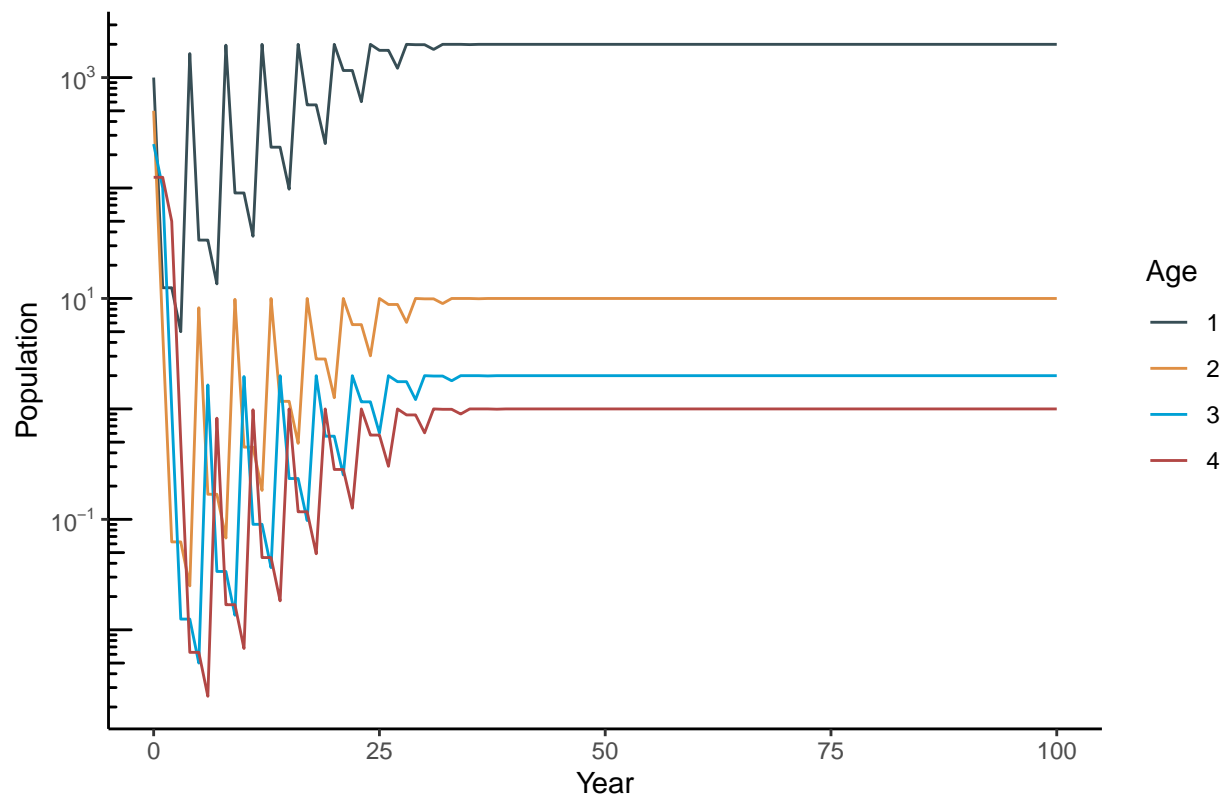
#final dataframe
century.df <- bind_rows(result.list)

#long dataframe for plotting
plot.df <- century.df %>%
  pivot_longer(!year) %>%
  mutate(name=factor(name, labels=c(1:4)),
         #can't have zeros with a log scale, so approximate zero instead
         value=ifelse(value==0,0.1,value))

#make a plot
ggplot(plot.df, aes(year, value, color=name))+
  theme+
  logtheme+
  labs(title="Fig. 7")

```

Fig. 7



Reflection Questions:

Why didn't our work from Sections 2 and 3 create exponential growth or decay? Check the eigenvalues!

```
#make the matrix
M <- matrix(c(0,0,0,2000,
              0.005,0,0,0,
              0,0.2,0,0,
              0,0,0.5,0),
            ncol=4, nrow=4, byrow=TRUE)

#check eigenvalues
Re(eigen(M)$vectors[,1])
```

```
## [1] 0.9999868753 0.0049999344 0.0009999869 0.0004999934
```

The largest eigenvalue from sections 2 and 3 is essentially equal to 1, so there is no exponential growth or decay.

Fisheries generally target adults above a certain length or age. How would you incorporate/represent fishing mortality into a Leslie matrix model? How do you think fishing on only the oldest fish would impact the stable age distribution of the modeled population?

Fishing mortality would be represented by decreasing the values in the fourth row of the Leslie matrix. In the challenge task above, fecundity increased as the adult population decreased, and so in that scenario we would expect the population of younger fish to increase despite fishing mortality on the age-4 population, and the age distribution would become skewed towards younger fish. In reality, the effects of fishing are likely much more complicated. . .