Homework 3: Matrix Models

Sidney Gerst

February 25, 2020

# Modeling Sea Turtle Age Structure

### 1. Enter the Model

#Create the Matrix  
  
class\_names <- c("Egg", "Sm Juv", "Lg Juv", "Subadult", "Adult")  
A <- matrix(c(0, 0, 0, 4.665, 61.896,  
 0.675, 0.703, 0, 0, 0,  
 0, 0.047, 0.657, 0, 0,  
 0, 0, 0.019, 0.682, 0,  
 0, 0, 0, 0.061, 0.809),  
 nrow = 5, ncol = 5, byrow = TRUE, dimnames = list(class\_names, class\_names))

1.1. Print the matrix you have just created, and ensure that it matches the one in Table 2 of Crowder et al.(1994)

####1.1  
  
A

## Egg Sm Juv Lg Juv Subadult Adult  
## Egg 0.000 0.000 0.000 4.665 61.896  
## Sm Juv 0.675 0.703 0.000 0.000 0.000  
## Lg Juv 0.000 0.047 0.657 0.000 0.000  
## Subadult 0.000 0.000 0.019 0.682 0.000  
## Adult 0.000 0.000 0.000 0.061 0.809

1.2. Print out the subsets of A described in the list above. Do you get the values you expect? Do youcunderstand how matrix subsetting works? If not, what don’t you understand?

###1.2  
  
A[4,3]

## [1] 0.019

A[3:4, 3]

## Lg Juv Subadult   
## 0.657 0.019

A[,3] #just column

## Egg Sm Juv Lg Juv Subadult Adult   
## 0.000 0.000 0.657 0.019 0.000

A[4,] #just row

## Egg Sm Juv Lg Juv Subadult Adult   
## 0.000 0.000 0.019 0.682 0.000

A[, c(3,5)] #columns 3 and 5

## Lg Juv Adult  
## Egg 0.000 61.896  
## Sm Juv 0.000 0.000  
## Lg Juv 0.657 0.000  
## Subadult 0.019 0.000  
## Adult 0.000 0.809

A[-1,] #everything except row 1

## Egg Sm Juv Lg Juv Subadult Adult  
## Sm Juv 0.675 0.703 0.000 0.000 0.000  
## Lg Juv 0.000 0.047 0.657 0.000 0.000  
## Subadult 0.000 0.000 0.019 0.682 0.000  
## Adult 0.000 0.000 0.000 0.061 0.809

I understand how subsetting works. The left side of the bracket represents rows and the right side represents columns. Putting a (-) before a number takes away that row.

1.3. From the matrix you have just created, draw the life cycle graph, putting in the values for each transition. This can be hand-drawn.

*See Attachment*

### 2. Projecting the Population Matrix

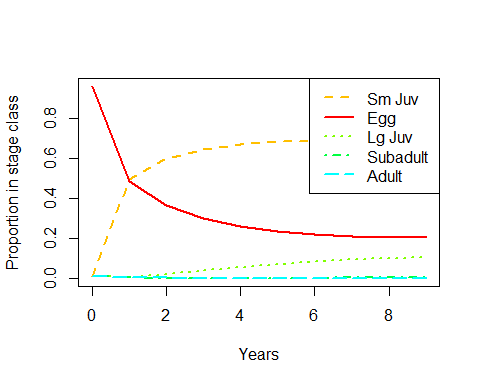
n\_0 <- c(1000, 10, 10, 10, 10) # Initial abundance  
#pop projection  
n\_1 <- A %\*% n\_0  
A %\*% n\_1

## [,1]  
## Egg 571.19685  
## Sm Juv 928.75384  
## Lg Juv 36.68069  
## Subadult 4.91458  
## Adult 7.46591

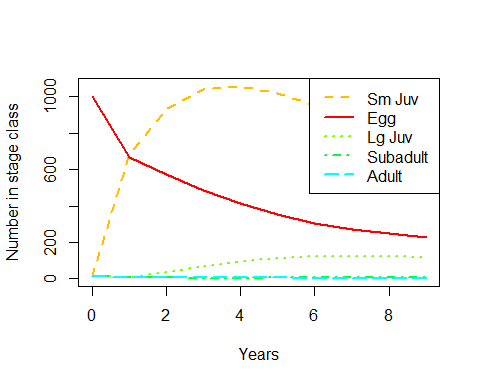
#do it in a function  
pop <- pop.projection(A, n\_0, iterations = 10) # Project the matrix  
pop

## $lambda  
## [1] 0.9211549  
##   
## $stable.stage  
## Egg Sm Juv Lg Juv Subadult Adult   
## 0.211083402 0.671087214 0.108936804 0.006133496 0.002759084   
##   
## $stage.vectors  
## 0 1 2 3 4 5 6  
## Egg 1000 665.61 571.19685 485.036481 411.289802 351.626280 305.624171  
## Sm Juv 10 682.03 928.75384 1038.471823 1057.445316 1021.004674 955.114025  
## Lg Juv 10 7.04 36.68069 67.750644 93.320349 111.011399 120.921709  
## Subadult 10 7.01 4.91458 4.048677 4.048460 4.534136 5.201497  
## Adult 10 8.70 7.46591 6.339711 5.375795 4.595974 3.994726  
## 7 8 9  
## Egg 271.522516 246.937023 229.398165  
## Sm Juv 877.741474 800.329955 729.314449  
## Lg Juv 124.335922 122.942550 118.388763  
## Subadult 5.844934 6.348627 6.665672  
## Adult 3.549024 3.227702 2.998477  
##   
## $pop.sizes  
## [1] 1040.000 1370.390 1549.012 1601.647 1571.480 1492.772 1390.856 1282.994  
## [9] 1179.786 1086.766  
##   
## $pop.changes  
## [1] 1.3176827 1.1303438 1.0339800 0.9811646 0.9499152 0.9317268 0.9224490  
## [8] 0.9195569 0.9211549

#Plotting routine  
stage.vector.plot(pop$stage.vector) # Plot each stage through time



###2.1  
#if we are interested in abundances  
stage.vector.plot(pop$stage.vector, proportions = FALSE)



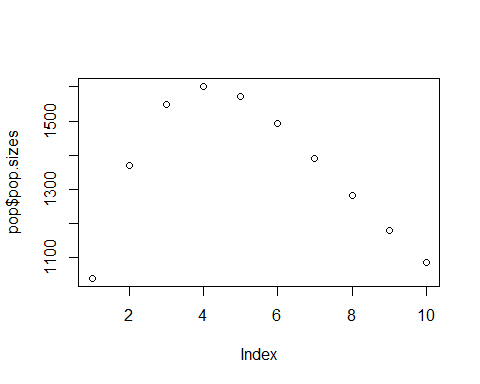
2.1. The output of pop.projection has a number of other elements besides stage.vector. Look at all of the elements of pop and make sure that you understand them (referring to the help page if needed).

#do it in a function  
pop <- pop.projection(A, n\_0, iterations = 10) # Project the matrix  
pop

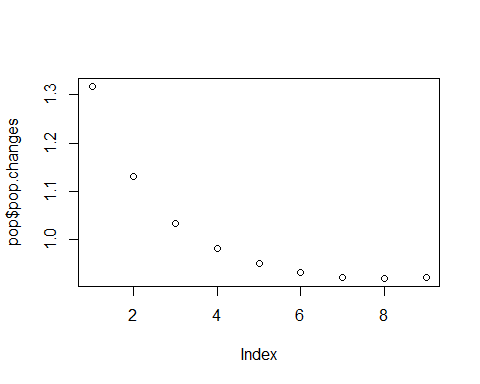
## $lambda  
## [1] 0.9211549  
##   
## $stable.stage  
## Egg Sm Juv Lg Juv Subadult Adult   
## 0.211083402 0.671087214 0.108936804 0.006133496 0.002759084   
##   
## $stage.vectors  
## 0 1 2 3 4 5 6  
## Egg 1000 665.61 571.19685 485.036481 411.289802 351.626280 305.624171  
## Sm Juv 10 682.03 928.75384 1038.471823 1057.445316 1021.004674 955.114025  
## Lg Juv 10 7.04 36.68069 67.750644 93.320349 111.011399 120.921709  
## Subadult 10 7.01 4.91458 4.048677 4.048460 4.534136 5.201497  
## Adult 10 8.70 7.46591 6.339711 5.375795 4.595974 3.994726  
## 7 8 9  
## Egg 271.522516 246.937023 229.398165  
## Sm Juv 877.741474 800.329955 729.314449  
## Lg Juv 124.335922 122.942550 118.388763  
## Subadult 5.844934 6.348627 6.665672  
## Adult 3.549024 3.227702 2.998477  
##   
## $pop.sizes  
## [1] 1040.000 1370.390 1549.012 1601.647 1571.480 1492.772 1390.856 1282.994  
## [9] 1179.786 1086.766  
##   
## $pop.changes  
## [1] 1.3176827 1.1303438 1.0339800 0.9811646 0.9499152 0.9317268 0.9224490  
## [8] 0.9195569 0.9211549

2.2. Plot poppop.changes through time. What do these tell you?

###2.2  
#pop.sizes  
plot(pop$pop.sizes)



#pop.changes  
plot(pop$pop.changes)

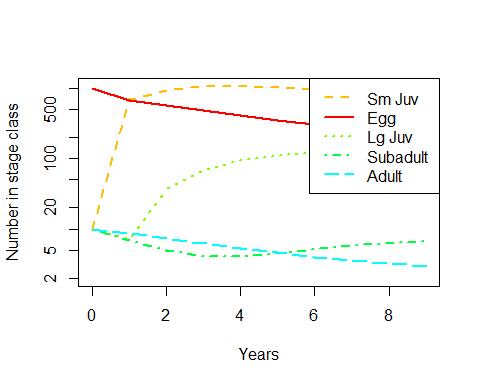


Pop sizes shows the population size through all life stages.

Pop changes shows how the total population changes through time.

2.3. Once the population has reached the stable stage distribution (SSD), all stages will grow or decline exponentially with the same growth rate. Looking at the stage vector plot, has this been acheived by the end of your simulated time series? (Tip: this might be easier to determine if you make the plot with abundance on a log scale. You can do this by including log = “y” in the call to stage.vector.plot)

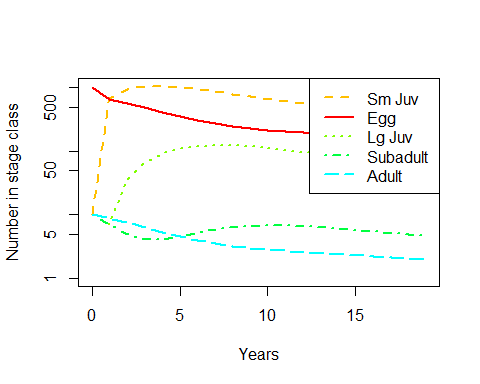
###2.3  
stage.vector.plot(pop$stage.vector, proportions = FALSE, log = "y")



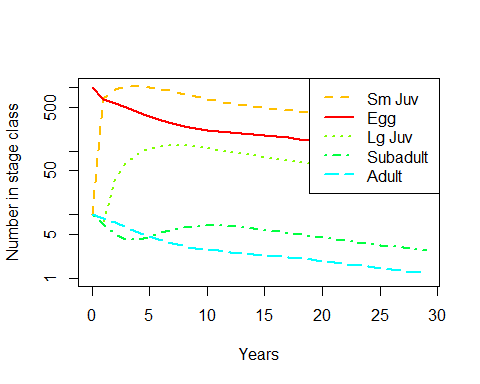
The population is not stable by year 10, the lines are not at the same slope.

2.4. If the population has not reached the SSD, run the simulation for longer. How many years are required before the population appears to be at the SSD?

#do it longer  
pop\_20 <- pop.projection(A, n\_0, iterations = 20) # Project the matrix  
stage.vector.plot(pop\_20$stage.vector, proportions = FALSE, log = "y")



pop\_30 <- pop.projection(A, n\_0, iterations = 30) # Project the matrix  
stage.vector.plot(pop\_30$stage.vector, proportions = FALSE, log = "y")



#  
#if you haven't run the model long enough, the labels are misleading  
#these values can be calculated through the eigan value and eigan value analysis

All seem to be declining at the same rate starting at 15 years and becoming stable.

### 3. Analyzing Population Matrix

3.1. Compare the values of lambda and SSD with the equivalent outputs of pop.projection from the initial run (with only 10 years of simulation). Why are they different?

#Where the population becomes stable  
lambda(A)

## [1] 0.9515489

stable.stage(A)

## Egg Sm Juv Lg Juv Subadult Adult   
## 0.238508404 0.647732505 0.103356123 0.007285382 0.003117586

Lambda is different than the SSD because it is the stable slope for the entire length of age and all stages are stable, while the SSD is the slope stable slope per age stage.

3.2. You want to improve the status of the population so that it is no longer declining. You think that your best options are to manage the nesting beaches to increase egg/hatchling survival (e.g., controlling poaching, motorized vehicles, dogs, bright lights that disorient hatchlings) or to reduce the bycatch of adult turtles in shrimp trawling nets (e.g., by requiring a modified design with a “turtle excluder device” or by reducing fishing effort). Use the model to evaluate the effects of these two strategies:

1. Which element of the projection matrix represents egg/hatchling survival? Which represents adult survival?

The egg element is in the first column, second row [1,2]. The adult element is in the fifth column, fifth row [5,5].

1. Increase egg/hatchling survival in the model, and re-calculate λ1. By how much does it increase? Experiment with different values of this survival term until you get an asymptotic growth rate of 1 or more. How large does egg survival need to be to achieve this?

#Want the Lambda to be one for all egg survival  
class\_names <- c("Egg", "Sm Juv", "Lg Juv", "Subadult", "Adult")  
A\_1 <- matrix(c(0, 0, 0, 4.665, 61.896,  
 1.5, 0.703, 0, 0, 0,  
 0, 0.047, 0.657, 0, 0,  
 0, 0, 0.019, 0.682, 0,  
 0, 0, 0, 0.061, 0.809),  
 nrow = 5, ncol = 5, byrow = TRUE, dimnames = list(class\_names, class\_names))  
  
pop\_egg <- pop.projection(A\_1, n\_0, iterations = 10)  
  
lambda(A\_1)

## [1] 1.000698

#impossible to save the eggs rates

The egg element has to be greater than 1 in order for λ to be one. More specifically, it has to be 1.5. At this rate it is not possible to save the population by saving eggs.

1. Put the egg survival back to its original value, increase adult survival in the model, and re-calculate λ1. By how much does it increase? Experiment with different values of this survival term until you get an asymptotic growth rate of 1 or more. How large does adult survival need to be to achieve this?

#Want the Lambda for the adults  
A\_2 <- matrix(c(0, 0, 0, 4.665, 61.896,  
 0.675, 0.703, 0, 0, 0,  
 0, 0.047, 0.657, 0, 0,  
 0, 0, 0.019, 0.682, 0,  
 0, 0, 0, 0.061, 0.924),  
 nrow = 5, ncol = 5, byrow = TRUE, dimnames = list(class\_names, class\_names))  
pop\_adult <- pop.projection(A\_2, n\_0, iterations = 10)  
  
lambda(A\_2)

## [1] 1.000497

#Have to save the adults

The adult only needs to be 0.924 to have a λ greater than 1.

1. Based on this analysis, which life stage seems the more promising one to target managment at? What else would you need to know to reach a final conclusion?

Adults are more promising to target managment. A sensitivity anaylsis should be performed to reach a final conclusion.

### 4. Sensitivity and elasticity analysis in R

class\_names <- c("Egg", "Sm Juv", "Lg Juv", "Subadult", "Adult")  
A <- matrix(c(0, 0, 0, 4.665, 61.896,  
 0.675, 0.703, 0, 0, 0,  
 0, 0.047, 0.657, 0, 0,  
 0, 0, 0.019, 0.682, 0,  
 0, 0, 0, 0.061, 0.809),  
 nrow = 5, ncol = 5, byrow = TRUE, dimnames = list(class\_names, class\_names))  
  
  
DemoInfo(A)

## $lambda  
## [1] 0.9515489  
##   
## $SSD  
## [1] 0.238508404 0.647732505 0.103356123 0.007285382 0.003117586  
##   
## $RV  
## [1] 1.000000 1.409702 7.454890 115.569961 434.209028  
##   
## $Sensitivities  
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.05792157 0.1573013 0.02509995 0.001769249 0.0007571032  
## [2,] 0.08165216 0.2217480 0.03538345 0.002494114 0.0010672899  
## [3,] 0.43179894 1.1726640 0.18711737 0.013189557 0.0056441208  
## [4,] 6.69399385 18.1793066 2.90080031 0.204472046 0.0874983860  
## [5,] 25.15006963 68.3016503 10.89862514 0.768223917 0.3287410397  
##   
## $Elasticities  
## Egg Sm Juv Lg Juv Subadult Adult  
## Egg 0.00000000 0.00000000 0.00000000 0.008673803 0.04924777  
## Sm Juv 0.05792157 0.16382640 0.00000000 0.000000000 0.00000000  
## Lg Juv 0.00000000 0.05792157 0.12919579 0.000000000 0.00000000  
## Subadult 0.00000000 0.00000000 0.05792157 0.146550473 0.00000000  
## Adult 0.00000000 0.00000000 0.00000000 0.049247770 0.27949327  
##   
## $PPM  
## Egg Sm Juv Lg Juv Subadult Adult  
## Egg 0.000 0.000 0.000 4.665 61.896  
## Sm Juv 0.675 0.703 0.000 0.000 0.000  
## Lg Juv 0.000 0.047 0.657 0.000 0.000  
## Subadult 0.000 0.000 0.019 0.682 0.000  
## Adult 0.000 0.000 0.000 0.061 0.809

# 0.3287410397 slope of lambda at A5,5 (Sij)  
# 0.08165216 slope of lambda at A1,2

4.1. Referring to the help page and section 2.2 of the Stevens chapter, make sure you understand what each of the outpurs of DemoInfo represents. The “RV” (reproductive value) (another kind of eigenvector) is the only bit we haven’t covered in lecture

Demo <- DemoInfo(A)  
Demo

## $lambda  
## [1] 0.9515489  
##   
## $SSD  
## [1] 0.238508404 0.647732505 0.103356123 0.007285382 0.003117586  
##   
## $RV  
## [1] 1.000000 1.409702 7.454890 115.569961 434.209028  
##   
## $Sensitivities  
## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.05792157 0.1573013 0.02509995 0.001769249 0.0007571032  
## [2,] 0.08165216 0.2217480 0.03538345 0.002494114 0.0010672899  
## [3,] 0.43179894 1.1726640 0.18711737 0.013189557 0.0056441208  
## [4,] 6.69399385 18.1793066 2.90080031 0.204472046 0.0874983860  
## [5,] 25.15006963 68.3016503 10.89862514 0.768223917 0.3287410397  
##   
## $Elasticities  
## Egg Sm Juv Lg Juv Subadult Adult  
## Egg 0.00000000 0.00000000 0.00000000 0.008673803 0.04924777  
## Sm Juv 0.05792157 0.16382640 0.00000000 0.000000000 0.00000000  
## Lg Juv 0.00000000 0.05792157 0.12919579 0.000000000 0.00000000  
## Subadult 0.00000000 0.00000000 0.05792157 0.146550473 0.00000000  
## Adult 0.00000000 0.00000000 0.00000000 0.049247770 0.27949327  
##   
## $PPM  
## Egg Sm Juv Lg Juv Subadult Adult  
## Egg 0.000 0.000 0.000 4.665 61.896  
## Sm Juv 0.675 0.703 0.000 0.000 0.000  
## Lg Juv 0.000 0.047 0.657 0.000 0.000  
## Subadult 0.000 0.000 0.019 0.682 0.000  
## Adult 0.000 0.000 0.000 0.061 0.809

# 0.3287410397 slope of lambda at A5,5 (Sij)  
# 0.08165216 slope of lambda at A1,2

4.2. Looking at the sensitivity and elasticity matrices, what can you conlude about which matrix elements would likely have the biggest impact on λ if they were changed?

####Sensitivity and Elasticities  
  
Demo$Sensitivities

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 0.05792157 0.1573013 0.02509995 0.001769249 0.0007571032  
## [2,] 0.08165216 0.2217480 0.03538345 0.002494114 0.0010672899  
## [3,] 0.43179894 1.1726640 0.18711737 0.013189557 0.0056441208  
## [4,] 6.69399385 18.1793066 2.90080031 0.204472046 0.0874983860  
## [5,] 25.15006963 68.3016503 10.89862514 0.768223917 0.3287410397

Demo$Elasticities

## Egg Sm Juv Lg Juv Subadult Adult  
## Egg 0.00000000 0.00000000 0.00000000 0.008673803 0.04924777  
## Sm Juv 0.05792157 0.16382640 0.00000000 0.000000000 0.00000000  
## Lg Juv 0.00000000 0.05792157 0.12919579 0.000000000 0.00000000  
## Subadult 0.00000000 0.00000000 0.05792157 0.146550473 0.00000000  
## Adult 0.00000000 0.00000000 0.00000000 0.049247770 0.27949327

The sensitvity element that is the most influential on λ is [4,3], while [5,2] is larger, it is not a transition that the turtle undergoes.

The elasticity element that is the most influential on λ is [5,5]

4.3. Compare the elasticity matrix with Fig. 1 in Crowder et al. (1994). Do you understand where the values in the figure come from?

The values from Figure 1 come from three different elasticities of the probability of suriving and remaining in the life stage, the probability of surviving and going to the next stage, and reproductive output.

4.4. Look at the sensitivity matrix produced by DemoInfo. What does the sensitivity for element a5,1 represent? Does it make sense to have a non-zero value here? Why or why not?

[5,1] represents the transition between eggs to adults. This transition cannot exist, so a non-zero answer does not make sense because you would expect to see a zero.