Hooper\_ESM211\_HW3

Tess Hooper

2/18/2020

### 1. Make a matrix for Loggerhead Turtle population

**1.1. Make and print 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))  
  
# Eggs per subadult and eggs per adult  
  
print(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 subets of A.**

## Matrix are first presented by ROW then COLUMN  
  
  
# Print subset for row 4, column 3  
print(A[4, 3])

## [1] 0.019

# Print subset for column 3  
print(A[ , 3])

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

# Print subset for row 4  
print(A[4, ])

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

# Print subset for (3,3) and (4,3)  
print(A[3:4, 3])

## Lg Juv Subadult   
## 0.657 0.019

# Print subset for columns 3 and 5  
print(A[ , c(3,5)])

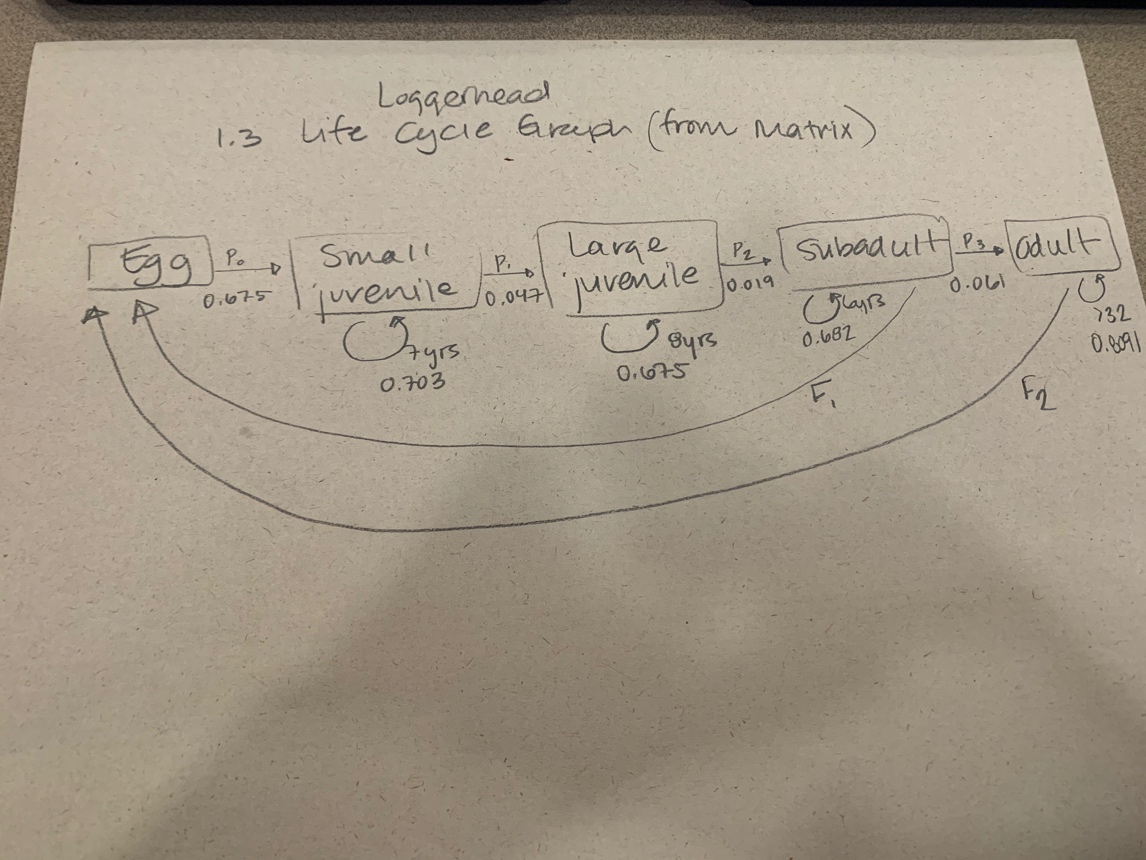
## Lg Juv Adult  
## Egg 0.000 61.896  
## Sm Juv 0.000 0.000  
## Lg Juv 0.657 0.000  
## Subadult 0.019 0.000  
## Adult 0.000 0.809

# Print subset for everything but the first row  
print(A[-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’ve gotten the values I expect given how the matrix works.

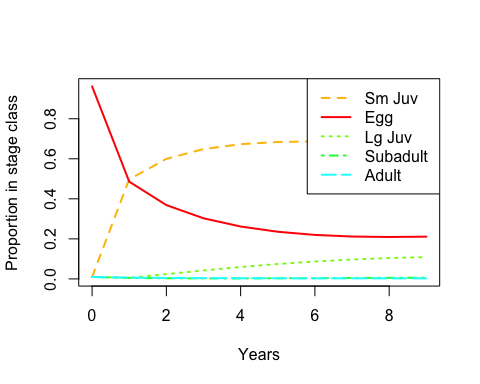
**1.3. Draw the life cycle graph for Loggerhead Sea Turtles**



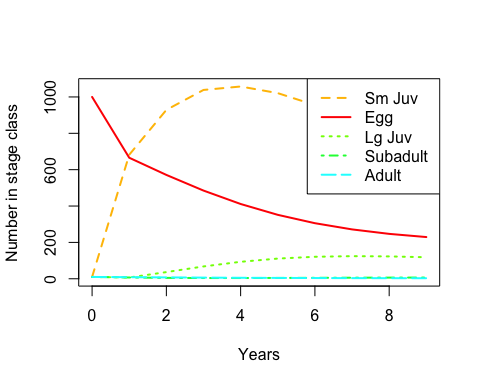
### 2. Project the population matrix

**2.1. Examine outputs of pop.projection and stage.vector.**

n\_0 <- c(1000, 10, 10, 10, 10) # Initial abundance  
  
pop <- pop.projection(A, n\_0, iterations = 10) # Project the matrix forward. Package also provides plotting routine  
  
stage.vector.plot(pop$stage.vector) # Plot each stage through time (proportion of each stage). Quickly population comes to be dominated by small juveniles.



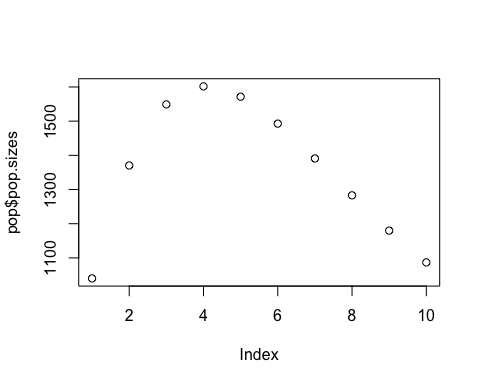
#By default, the above code plots the proportions in each stage through time. To see the actual abundances, use  
stage.vector.plot(pop$stage.vector, proportions = FALSE) # We see dcline in egg, big increase in small juveniles.



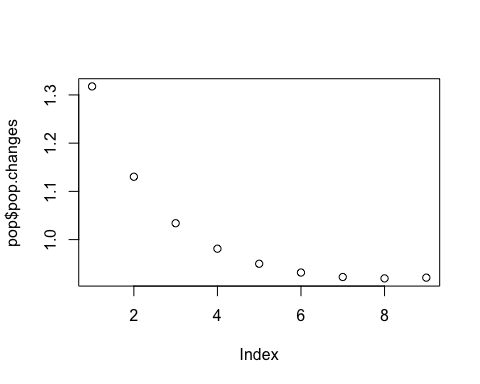
# This is N0 - to get N1 take A %\*% n\_0 - can do this in console  
  
n\_1 <- A %\*% n\_0  
  
n\_2 <- A %\*% n\_1

**2.2. Plot pop$pop.size and pop$pop.changes through time. What do these plots tell us?**

# Total number of individauls across all the stages (pop$pop.sizes)  
plot(pop$pop.sizes)



help("pop.projection")  
  
# ALl stages would be increasing or decreasing at same rate if we were at stable stage distriubtion (SSD)  
  
# Proportional change in population size  
plot(pop$pop.changes)

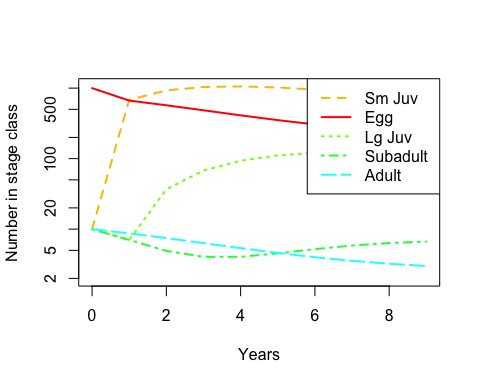


pop$pop.size tells us the total number of individuals across all the stages of the life cycle. When we plot the population for Loggerhead sea turtles, we see that the number of individuals increases over time until the population reaches about 1600 individuals. Then, the population decreases over time until the population reaches about 1100 individuals.

pop$pop.changes tells us the proportional change in population size. According to our plot of Loggerhead sea turtles, the proportional change in population size starts just above 1.3, then decreases over time to stabilize around 0.2.

**2.3. Examine the stage vector plot. Has stable stage distribution (SSD) been achieved by the end of your simulated time series?**

# 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)  
  
  
# Look at stage classes using log scale using actual abundance and not proportions:  
stage.vector.plot(pop$stage.vector, proportions = FALSE, log = "y")  
  
# Log scale versus non-log scale: Changing exponentially - on log scale that is a straight line. This is visually more easy to assess. Hard to eyeball exponential curves. Proportions of each stage will tell you if we've reached SSD but won't tell us the numbers (?)   
  
pop\_project <- pop.projection(A, n\_0, iterations = 20) # project matrix  
  
stage.vector.plot(pop$stage.vector, proportions = FALSE, log = "y")

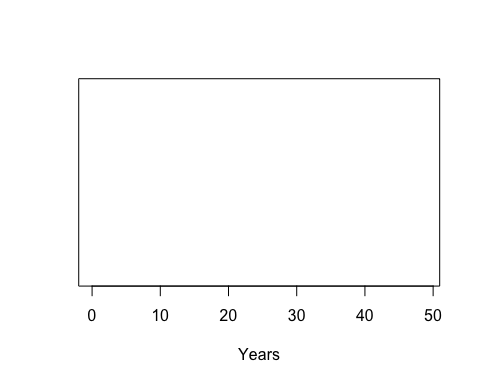


Stable stage distribution (SSD) has not been achieved in this time series because the sub-adult and adult stages are not declining at the same growth rate.

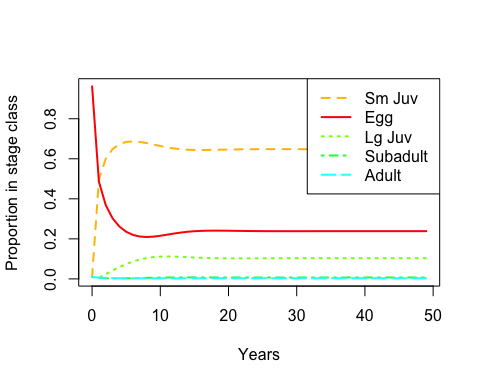
**2.4. Run simulation for longer time series to see when population appears to be at SSD.**

#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?  
  
# Increase # of iterations from 10 to 50 to see if population reaches the SSD  
pop\_SSD <- pop.projection(A, n\_0, iterations = 50)  
  
# Plot stage classes using log scale using actual abundance and not proportions:  
stage.vector.plot(pop\_SSD$stage.vector, proportions = FALSE, log = "y")

## Warning in plot.window(...): nonfinite axis limits [GScale(-inf,  
## 3.02449,2, .); log=1]



# Plot each stage through time using proportions  
stage.vector.plot(pop\_SSD$stage.vector)



The population has reached SSD when the lines on the plot are parallel. Looking at the plot for proportions of population change, it takes around 15-20 years before population reaches SSD.

### 3. Analyzing the population matrix.

Eigenvalues and eigenvectors give a lot of information about the “asymptotic” (once the population reaches the SSD) dynamics and structure. You can calculate them yourself (see the code in SPsandpiper.R), but the popbio package has two functions to extract the key information:  
lambda(A)

stable.stage(A)

**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?**

lambda(A) #true asymptotic growth rate

## [1] 0.9515489

# 0.951  
  
stable.stage(A) # corresponding eigenvector expressed as proportion

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

# Lambda is less than one but we need it to be greater than one for the population to increse.   
  
  
# Egg Sm Juv Lg Juv Subadult Adult   
#0.279209989 0.626678506 0.084945390 0.005016370 0.004149745

Lambda gives us the stable slope for the entire time series (10 iterations). All stages are stable. SSD gives us the stable slope per stage across the time series.

**3.2. Use the model to evaluate two different management strategies to improve the status of the loggerhead sea turtle population. The two best options are to manage the nesting beaches to increase egg/hatchling survival or to reduce the bycatch of adult turltes in shrimp trawling nets.**

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

A[2,1] #survival of eggs to small juveniles --> 2nd row, 1st column

## [1] 0.675

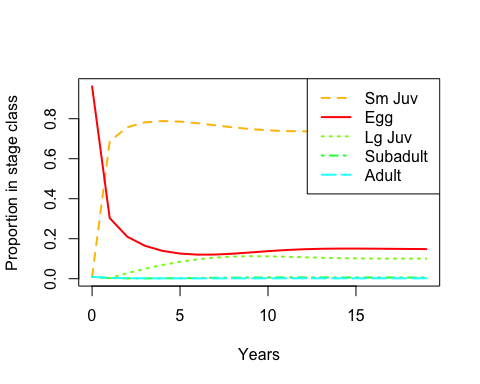
# 0.675  
  
  
A[5,5] #survival of adults (from adults, to adults) --> 5th row, 5th column

## [1] 0.809

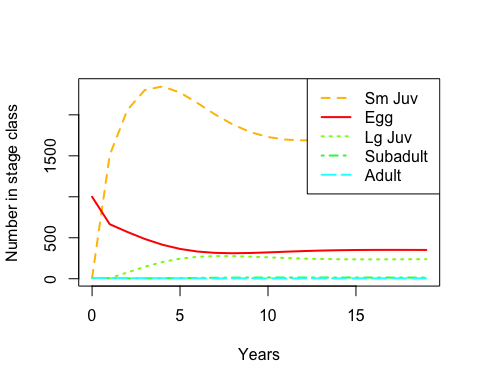
# 0.8091

*b. 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?*

# Change egg/hatching survival rate by increasing A[2,1]  
#A[2,1] <- 1.5  
# Check if asymptotic growth rate is now >1  
#lambda(A)  
  
# Make new matrix for increased survivability of hatchlings  
class\_names <- c("Egg", "Sm Juv", "Lg Juv", "Subadult", "Adult")  
A\_juvenile <- 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))  
  
# Plot   
n\_0 <- c(1000, 10, 10, 10, 10) # Initial abundance  
  
pop\_juvenile <- pop.projection(A\_juvenile, n\_0, iterations = 20) # Project the matrix forward. Package also provides plotting routine  
  
stage.vector.plot(pop\_juvenile$stage.vector) # Plot each stage through time (proportion of each stage). Quickly population comes to be dominated by small juveniles.



#By default, the above code plots the proportions in each stage through time. To see the actual abundances, use  
stage.vector.plot(pop\_juvenile$stage.vector, proportions = FALSE)



lambda(A\_juvenile)

## [1] 1.000698

# Have to increase survivability to 1.5 to get lambda greater than one  
# Is this attainable?   
# Also need to reduce bycatch to help adults   
  
# Management practices can be more or less affective depending on which life cycle they are targeting

In order to increase survivability for hatchlings, we would need to increase survivability to 1.5 to acheive lambda greater than one. This is not attainable.

*c. 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?*

# Change egg/hatching survival rate back to its original value. Since I made a new matrix for juveniles, I can simply use A. But want to make sure that all values are correct.   
A[2,1] <- 0.675  
  
# Increase adult survival rate  
A[5,5] <- 0.95  
  
# Check if asymptotic growth rate is now >1  
lambda(A)

## [1] 1.015079

# Lambda = 1.015079  
  
  
# Increase adult survival rate  
A[5,5] <- 0.93  
  
# Check if asymptotic growth rate is now >1  
lambda(A)

## [1] 1.003739

# 1.003739  
  
# 0.93 is the lowest we can go to get it >1

Lambda must be greater than 1 in order for the population to increase. We need to increase adult survival rate to 0.93 or greater in order to achieve this.

*d. 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?*

We should prioritize increasing adult survivability as this is more attainable than increasing hatchling survivability (increasing 0.93 vs. 1.5). One way we could achieve this is by focusing our management efforts on reducing adult bycatch of sea turtles in order to increase adult survivability. However, we would want to run the same tests on each individual life stage as well.

### 4. Sensitiviy and elasticity in R

Like the asymptotic growth rate and the stable stage structure, the sensitivities and elasticities can be calculated from the eigenvalues and eigenvectors of the matrix. The primer library (which accompanies the book A Primer of Ecology in R) bundles the calculations together to produce the biologically relevant output:

# Already have loggerhead matrix created as "A"  
  
#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))  
  
# Set adult survivability back to original setting:  
A[5,5] <- 0.809  
  
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

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

demo <- DemoInfo(A)  
  
help(DemoInfo)  
# Calculates lambda, stable age/stage structure, reproductive value, sensitivity, and elasticity   
# lambda = a scalar, long term asymptotic growth rate  
# SSD = vector of stable age/stage structure  
# RV = vector of reproductive value  
# Sensitivities = A matrix of sensitivities --> shows how much lambda will increase given a unit increase of that matrix element  
# Elasticities = a matrix of elasticities --> proportional sensitivity  
# PPM = population projection matrix   
  
  
# lambda = 0.9515489  
  
# 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?**

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

When we look at the sensitivity output, [5, 2] causes the greatest change to λ (value of 68.3). However, the sensitivity matrix gives us values for life cycle stage transitions that the turtle does not undergo. Given this, the actual sensitivity matrix element that is most influential on λ for loggerhead is [4,3]. Looking at the elasticity matrix, the element that will be 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 Crowder et al. (1994) article presents the elasticity of λ to changes in reproductive output, survival while remaining in the same stage, and survival with growth into the next life stage.

**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?**

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

# [5,1] = 21.15

The sensitivity element [5, 1] represents an egg directly becoming an adult. This is not a transition that the turtles undergo. Therefore, having a non-zero value here (25.15) in the sensitivity matrix does not make sense. We should expect to see a zero value.

***Miscellaneous Notes from Class***

+ Population is declining - lambda is less than one

+ Every year you multiply the population by lambda to get next years population –> 5% decline every year

+ We want to get lambda greater than one so we don’t have decline. What do we do to change lambda?

+ Anthropogenic impact on beaches (poaching, other threats to hatchlings, like artifial light, etc). Reducing these threats could increase survival of egg stage. Bycatch is reducing survival of adult sea turtles, so reducing bycatch would increase survival of adults and increase reproducibility

+ What part of population matrix represents survival of egg stage? —> 0.675 - from egg to small juveniles. If we wanted to simulate reducing threats to nesting beach, we could increase this term to represent higher survival from egg to small juvenile.

+ What about adult survival? —> 0.8091. Increase this term