WFC 198 “Sampling Animal Populations” – Homework Week 2

Due April 18th, 9am

Please complete the following exercises in R, “knit” your script into a html report (as you learned in lab 2) and submit it on Canvas under Assignments.

The point of this assignment is for you to practice writing R code. You will get points for providing the correct R code that produces the correct answer. For example, the correct answer to the task “Add 1 and 1 together” would not be “2”, but rather “1+1”, which, when run in R, will produce the outcome of 2.

Please preface every piece of code with a comment that states to which question the code belongs, for example: “#Task 3a” before the code used to tackle Task 3a. If we cannot follow your code, we cannot grade it. If you are unsure of how to format your script, see the short example homework script on Canvas (Files – Homework – Examplescript.R).

Work through the questions in order, as they build on each other.

Question 1: Missing data (5 Pts)

Create the following numerical vector in R (copy-paste is ok):

vec<-c(0,NA,NA,9,13,5,6,7,17,NA,3,19,2,NA,12)

1. Using the appropriate R command, calculate the sum of all elements in the vector (note: do not use arithmetic operators, use a single command) (1 Pt)
2. Using appropriate R commands, determine the number of elements in this vector that are not NA (2 Pts)
3. Create a new vector, vec2, that is a subset of vec, specifically, that contains the first five non-NA elements of vec. Do so using subsetting and R commands. (2 Pts)

Question 2: Building a two-population model using a for-loop (10 pts)

In class we talked about meta-populations, which consist of multiple (sub-)populations that are connected to each other by dispersal. Demographic rates can differ between populations. We built on the simple population model to develop the following mathematical representation of a two-population model:

In these equations, *N1* and *N2* are the sizes of population 1 and 2, respectively; *t* indicates the time step; *r1* and *r2* are the per-capita rates of change for both populations; and the parameters indicate the proportion of the population dispersing to the other population. is the proportion of population 1 that moves to population 2 (note that , the proportion of population 1 that remains in population 1, is equal to ). Conversely, is the proportion of population 2 that moves to population 1.

1. Translate these mathematical expressions into R code and combine them with a for-loop to simulate the size of population 1 and 2 for 50 time steps. To do so, follow the same general steps we used in lab to simulate an age-structured population. Use the following input parameters and starting population sizes (9 Pts):

*N1* at time 1: 20

*N2* at time 1: 50

*r1*: 0.1

*r2*: 0.1

: 0.1

: 0.2

1. Determine the size of population 1 and 2 in year 50 (0.5 Pt)
2. Calculate the size of the metapopulation in year 50 (0.5 Pt)

Question 3: Plotting population trajectories (10 Pts)

1. In R, make a plot of total population size against time steps, using the plot() function. Set the “type” argument to produce a line (“l”); use the “xlab” and “ylab” arguments to label your axes. To properly produce the full plot, you will also have to include the following argument in the plot() command: ylim=c(0,80). This will force the y axis to be displayed in the range from 0 to 80. (5 Pts)
2. Now, using the points() command, add the trajectories of the two subpopulations to the plot. In lab we used points() to add a single point (defined by one x and one y value) to a plot, but you can also use that command to add multiple points (i.e., multiple pairs of x and y values). Use the “type” argument to produce a line, just as you did under a). Note that because you are adding two sets of population sizes, you need to execute the points() command twice, once for each subpopulation. Within the points() command, use the “col” argument to give each subpopulation a different color (some colors that R recognizes: red, green, blue, grey). (5 Pts)

TOTAL POINTS: 25