WFC198\_Bolas\_Hwk2

Ellie Bolas

April 17, 2018

## Question 1: Missing data

Dealing with NAs

vec<-c(0,NA,NA,9,13,5,6,7,17,NA,3,19,2,NA,12)  
  
#a) calculate the sum of all elements in the vector   
sum(vec) #returns NA

## [1] NA

sum(vec, na.rm = TRUE)

## [1] 93

#b) determine the number of elements in this vector that are not NA   
#find out how many are NA  
vecNA <- sum(is.na(vec))  
vecNA

## [1] 4

#length tells you how many elements total  
length(vec)- vecNA #tells you elements that aren't NA

## [1] 11

#c) 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)   
head((na.exclude(vec)), 5)

## [1] 0 9 13 5 6

## Question 2: Building a two-population model using a for-loop

N\_1 (t)=N\_1 (t-1)*(1+r\_1 )*pi\_1,1+N\_2 (t-1)*(1+r\_2 )*pi\_2,1 N\_2 (t)=N\_2 (t-1)*(1+r\_2 )*pi\_2,2+N\_1 (t-1)*(1+r\_1 )*pi\_1,2

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 pi indicate the proportion of the population dispersing to the other population.

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. Input parameters: N1 at time 1: 20 N2 at time 1: 50 r1: -0.1 #per captia rates of change r2: 0.1 #per capita rates of change pi\_1,2: 0.1 #migration from 1 -> 2 pi\_2,1: 0.2 #migration from 2-> 1

#a  
tsp <- 50 #50 times steps  
n1 <- rep(NA,tsp) #pop. at each time step, we need a placeholder that is empty for all time steps  
n2 <- rep(NA,tsp)  
n1[1] <- 20 #number in pop. 1 at tsp 1  
n2[1] <- 50 #number in pop. 2 at tsp 1  
r1 <- -0.1 #rate change pop. 1  
r2 <- 0.1 #rate change pop. 2  
immpop1 <- 0.1 #immigration from pop1 -> pop2  
immpop2 <- 0.2 #immigration from pop2 -> pop1  
staypop1 <- 0.9 #number in pop.1 who stay in pop.1  
staypop2 <- 0.8  
poptotal <- n1 + n2 #will use this to determine population at a given time set  
  
  
#need for-loop to start at time 2 bc we already have time 1  
#i is the index for different time steps  
  
for(i in 2:tsp){  
 n1[i] <- (n1[i-1]\*(1 + r1)\*(staypop1)) + (n2[i-1]\*(1+r2)\*(immpop2))  
 n2[i] <-(n2[i-1]\*(1 + r2)\*(staypop2)) + (n1[i-1]\*(1+r1)\*(immpop1))  
 poptotal[i] <- n1[i] + n2[i]  
} #n1 is a vector, so you always want to use [] to store numbers or fill them, where as () indicates a function or equation  
  
  
#b) Determine the size of population 1 and 2 in year 50 (0.5 Pt)  
n1[50]

## [1] 27.81637

n2[50]

## [1] 22.75885

#c) Calculate the size of the metapopulation in year 50 (0.5 Pt)  
n1[50] + n2[50]

## [1] 50.57522

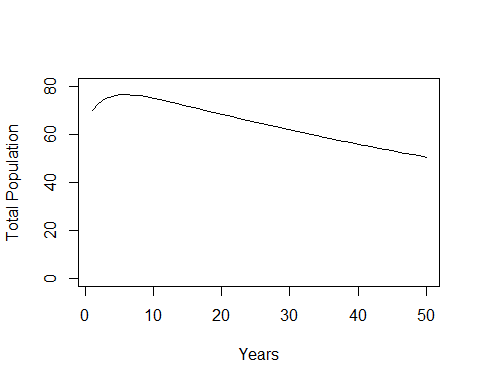
## Question 3: Plotting population trajectories (10 Pts)

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)

poptotal

## [1] 70.00000 73.00000 74.86000 75.92440 76.43426 76.55828 76.41456  
## [8] 76.08572 75.62957 75.08657 74.48501 73.84468 73.17939 72.49881  
## [15] 71.80967 71.11667 70.42307 69.73114 69.04243 68.35803 67.67867  
## [22] 67.00484 66.33685 65.67493 65.01919 64.36971 63.72651 63.08959  
## [29] 62.45894 61.83452 61.21630 60.60422 59.99823 59.39829 58.80434  
## [36] 58.21631 57.63416 57.05783 56.48726 55.92239 55.36317 54.80954  
## [43] 54.26145 53.71884 53.18165 52.64983 52.12333 51.60210 51.08608  
## [50] 50.57522

time.index <- 1:tsp  
plot(time.index, poptotal, xlab = "Years", ylab = "Total Population", type = "l", ylim=c(0,80))



1. 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)

plot(time.index, poptotal, xlab = "Years", ylab = "Total Population", type = "l", ylim=c(0,80))  
points(time.index, n1, col="blue")  
points(time.index, n2, col="red")

