**BIOS560R – FALL 2013 – HOMEWORK #1**

\* 20 QUESTIONS @ 5 POINTS EACH

\* Due by 11:59 PM on 26th of September. Send responses in a plain text file (Notepad on windows or TextEdit on Apple). We will run each of your R statements at the R console. You may also use R studio to save your commands into a .R file and send that if you wish. Just make it clear which question you are answering. (e.g.)

\* Send to [dvandom@emory.edu](mailto:dvandom@emory.edu) AND [wsp@emory.edu](mailto:wsp@emory.edu) with a subject line of BIOS560R Homework #1

\* All of these problems can be solved using approaches we have discussed. Check the slides for helpful functions.

\* Partial credit is available for all but the optional extra credit problem.

**VARIABLES and FORMULAE**

**1 -3 )** Create R "expressions" that represent the following formulae. Given the values x = 2, a = 2, b = 3, n = 3 evaluate the forumlae and present the result for each.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| x = 2, a = 2, b = 3 | x = 2, a = 2, b = 3 | a = 2, b = 3, n = 3 |
| z = ? | z = ? | z = ? |

**VECTORS:**

**4)** Given the vector below provide an R expression(s) that gives the sum of all odd numbers that are greater than 20 and less or equal to 25:

set.seed(123)

my.nums = round(runif(1000,15,25))

**5)** Use the paste command to create the following output. This should be a one-line expression.

[1] "label\_1\_5" "label\_2\_4" "label\_3\_3" "label\_4\_2" "label\_5\_1"

**VECTOR INDEXING:**

Suppose that the following character vector represents the names of people waiting in a line at a bank. "Steve" is first in line and "Liam" is at the end of the line.

queue = c("Steve", "Russell", "Alison", "Liam")

The object of this exercise is to provide R expressions to update the queue vector to reflect a sequence of events. As a basic example, the statement “Johnny pushes his way to the front of the queue” could look like:

queue = c("Johnny",queue)

So please start the exercise with the following vector:

queue = c("Steve", "Russell", "Alison", "Liam")

Provide the R expressions to update the queue vector to reflect the following events. Note that you will be changing the queue vector for each event so each expression will be of the form:

queue = <your expression> # A vector manipulation

**6)** Barry arrives to wait at the end of the queue behind Liam

Expression: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7)** Steve is served and, thus, leaves the front of the queue

Expression: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**8)** Pam arrives and convinces the others to let her go to the front of the queue

Expression: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**9)** Using the "which()" function, find the position of Russell in the queue.

Expression: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**10)** Alison gets impatient and leaves. For this case don't assume that you know where she is in the queue. That is, provide an R expression that does NOT involve a specific element number.

Expression: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**THE SAMPLE COMMAND:**

**11)** DNA nucleotide sequences are represented by combinations of "A","C","G", and "T". Use the sample function to create a forty element vector containing these letters. However, we would like "A"s to occur 15 percent of the time and G's to occur 25 percent of the time. The Cs and Ts each occur 30 percent of the time. Store the resulting sample into a vector called my.dna. Note that percentages are probabilities divided by 100.

my.dna = <your R expression>

**12)** In molecular biology, the “GC content” of a string of DNA is the percentage of nitrogenous bases (G or C) in a DNA molecule.Write an R expression(s) that will compute the percent "GC" content of the entire sequence that you generated above. Hint: you need to find the total number of both Gs and Cs as part of this problem.

**MATRICES:**

**13)** Provide an R expression to generate a matrix "my.matrix", which is a 10 row by 10 column matrix from 100 random elements from a Normal distribution. Use this as the basis to answer the following questions.

**14)** Provide an R expression that extracts the diagonal of the matrix.

**15)** Provide an R expression to name the rows "R1, R2, … R10" and the columns C1, … C10

**16)** Given this matrix:

x = matrix(1:16,4,4)

Write an R expression that, for each column in the matrix, determines the number of values less than 4 or greater than 10.

**17) Given** a square matrix K of any size, say 4x4, produce a one line R expression such that (K )ij = 1 where i >= j. Note that i and j correspond to rows and columns. Your expression should work in general for any NxN matrix not just 4x4. Hint: We covered two functions in particular that will make this easy. As an example:

|  |  |
| --- | --- |
| > K  [,1] [,2]  [1,] 1 3  [2,] 2 4 | > K  [,1] [,2]  [1,] 1 3  [2,] 1 1 |

**18)** Given the following system of equations solve for x1 and x2.

2x1 + 10x2 = 16

5x1 + 12x2 = 32

**FACTORS:**

The PH scale runs from 0 to 14, with values:

[0 , 7) – acid

[7, 8) – neutral

[8, Inf) - alkaline

Note that the "[" character means "inclusive". The ")" means that the interval does not include the closing value. As an aside, note that "Inf" is a valid quantity in R.

**19)** Write an R expression that "cuts" the following vector into a factor with the appropriate labels of acid, neutral, and alkaline. Note that you will need to check out some of the arguments to the cut command to make sure the intervals are defined as above. To first make sure that the intervals are being defined correctly. You might want to first leave off the labels.

set.seed(123)

my.ph = round(runif(25,0,14))

**20)** Create these two vectors (cut and paste if you wish)

mympg = c(21,21,22.8,21.4,18.7,18.1,14.3,24.4,22.8,19.2,17.8,16.4,17.3,15.2,10.4,10.4,14.7,32.4,30.4,33.9,21.5,15.5,15.2,13.3,19.2,27.3,26,30.4,15.8,19.7,15,21.4)

mycyl = c(6,6,4,6,8,6,8,4,4,6,6,8,8,8,8,8,8,4,4,4,4,8,8,8,8,4,4,4,8,6,8,4)

Provide an expression that prints the variance of mympg values across all cylinder types.

**\*\* Optional EXTRA CREDIT (No partial credit) 8 points**

Note you may **not** use the function “which.max” to solve this problem. Given a vector find the element corresponding to the value that is closest to the number 4. As an example if you had the following vector, the value that is closest to 4 is the second element with a value of 4.15. Thus your expression should return "2".

2.150310 4.153221 2.635908 4.532070 4.761869

So given the following vector produce an R expression to find the element that is closest to the number 4.

set.seed(123)

my.vec = runif(1000,1,6)

**Consider a single die – it has 6 sides each labeled 1 , 2, 3, 4, 5, 6 If a user rolls a single die then the result will be one of these values**

**Set.seed(123)**

somevec <- rnorm(1000)

length(somevec[c(-1:-10,-100:-110)])

Present a statement that shows the sum of all element numbers, (not values), which are evenly divisible by 3.5 and 2

sum(which(1:1000 %% 3.5 == 0 & 1:1000 %% 2 == 0))

[1] 35784

Create a vector T that contains all numbers greater than zero and less than 100,000 that are evenly divisible by 19 How many numbers are in T ?

How many numbers in T have a square that ends in a 1 ?

myvec <- sample(seq(1,21,3),100000,T)

length(unique(myvec))

This vector contains 100,000 elements although there are repetitions of various numbers contained throughout the vector. As an example you can see the number 1 repeated some number of times as well as the number 3 – there are also others. Present R statements that 1) identify the total number of unique values in the vector and 2) the largest unique value. Your code should work in general – that is – do not attempt to do this visually