BISO560R - Function Case Study

Pittard

02/03/14

Newton's Method

Let's consider the problem of computing the square root of a number. On the surface this isn't something we worry about too much since there are functions available to us in every major programming language that will do this for us. The field of numerical analysis is concerned with ways to efficiently find quantities such as roots. These methods present excellent opportunities for functional programming. We'll look at implementing Newton's method for computing a square root. R makes this easy.

Vectors

4) Given a vector x that contains all the numbers between 1 and 100, present an R statement that will return all numbers evenly divisible by 2 and 3 but not by 7.

5) Given the vector below, normvec, present R statement(s) that will return the "middle value" (median) of the vector. Note you cannot use the median function as part of your solution although you can use it to verify that you have the correct answer. As an example if you had this vector:

```
set.seed(123)
( somevec = rnorm(11) )
 [1] -0.56047565 -0.23017749  1.55870831  0.07050839  0.12928774  1.71506499
 [7]  0.46091621 -1.26506123 -0.68685285 -0.44566197  1.22408180
```

The middle value would be 0.07050839. Verify by using the median function. Note that your code should work for any vector of odd length. Look at some of the functions for rearranging vectors and rounding. Here is the vector you will operate on.

```
set.seed(123)
somevec = rnorm(1001)

# Your answer should match the following:
```

```
median(somevec)
[1] 0.00729009
```

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 statements 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 a one line R statement 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

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

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

9) Using the “which()” function, find the position of Russell in the queue.

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 statement that does NOT involve a specific element number.

Sampling and DNA Manipulation

11) DNA nucleotide sequences are represented by combinations of A,C,G, and T. Use the sample function to create a fifty four 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 C’s and T’s each occur 30 percent of the time. Store the resulting sample into a vector called my.dna.

```
my.dna = <your R expression>
```

12) In molecular biology, a reading frame is a way of dividing the sequence of nucleotides in a nucleic acid (DNA or RNA) molecule into a set of consecutive, non-overlapping triplets. While you may or may not know about reading frames you have enough skill with R to generate them. As an example given the following vector of DNA:

```
somedna = c("C","A","G","T","A","T","A","C","C","A","T","C")
```

the first reading frame could be represented by a matrix wherein each column represents a "triplet". So given this vector of DNA find a way to select each group of 3 characters, starting with the first three characters (nucleotides) into a structure resembling the following.

```
     RF1  RF2  RF3  RF4
[1,] "C"  "T"  "A"  "A"
[2,] "A"  "A"  "C"  "T"
[3,] "G"  "T"  "C"  "C"
```

Your job is to take the sequence you generated in problem 11 and present the R statement(s) necessary to produce the structure above.

13) In molecular biology, the "GC" content of a string of DNA is the percentage of nitrogenous bases, G or C, in a DNA molecule. Please present R commands that will compute the percent GC content of the entire sequence that you generated in step 11. Your R code should include use of regular expressions to do the computation.

Matrix Manipulation

14) Given a square matrix K produce a one line R statement that will set K_{ij} = 1 where i >= j. Produce another one line R statement that will set K_{ij} = 0 where i < j. Your statements should work in general for any NxN matrix but here is an example using a 4x4 matrix.

```
K            # Given this matrix.
     [,1] [,2]
[1,]    1    3
[2,]    2    4

# After applying the above rules it will look like:
K
     [,1] [,2]
[1,]    1    0
```

```
[2,]    1    1
```

15) Given the following matrix, write an R statement that, for each column in the matrix, presents the range of values, (minimum and maximum).

```
set.seed(123)
mymat = matrix(rnorm(100),10,10)
```

grep

Read in the following information relative to country names. When you finish reading this info you will have the names of various countries contained within the character vector ”countries”.

```
url = "http://nestor.sunderland.ac.uk/~cs0her/Statistics/therbook/worldfloras.txt"
hold = read.table(url,header=T,sep="\t")
countries = as.character(hold[,1])
```

16) Present an R statement that searches this vector and finds all country names that contain either a C or E.

17) Present an R statement that searches this vector and finds all country names that contain the letter R preceeded by a single space character.

Factors

18) The PH scale runs from 0 to 14, with values from [0 , 7) being acid, values from [7, 8) are neutral, and values [8, Inf) are alkaline. Note that the “[” character means “inclusive”. The “)” means non inclusive. As an aside, note that “Inf” is a valid quantity in R. Write an R statement that cuts the following vector into a factor with the appropriate labels of acid, neutral, and alkaline as defined above. Note that you will need to with the arguments to the cut command to insure that the intervals are defined as above. To help insure 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))
```

19) Create the two following vectors (cut and paste if you wish)

```
mywt = c(2.62,2.875,2.32,3.215,3.44,3.46,3.57,3.19,3.15,3.44,3.44,4.07,3.73,
         3.78,5.25,5.424,5.345,2.2,1.615,1.835,2.465,3.52,3.435,3.84,3.845,
         1.935,2.14,1.513,3.17,2.77,3.57,2.78)

mygears = c(4,4,4,3,3,3,3,4,4,4,4,3,3,3,3,3,3,4,4,4,3,3,3,3,3,4,5,5,5,5,5,4)
```

Provide a one-line statement that prints the standard deviation of mympg values per gear group. (Gears in this case refers to the number of gears that a given car has).

paste

20) Use the paste command to create the following output. This should be a one line statement:

```
[1] "label_10_5" "label_11_4" "label_12_3" "label_13_2" "label_14_1" "label_15_5"
```

Extra Credit

No partial credit 5 points Note you may not use the functions, “which.max”, “which.min”, or “match” to solve this problem. Given a vector find the element number corresponding to the value that is closest to the value of 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 R statement(s) should return “2”.

```
2.150310 4.153221 2.635908 4.532070 4.761869
```

So given the following vector present R statements to find the element number that is closest to the value of 4.

```
set.seed(123)
my.vec = runif(1000,1,6)
```