

BSDS 100: Intro to Data Science with R

Assignment 4

by Abbie M. Popa

Directions: For all questions in this assignment, write complete sentences and fully answer any question that is asked. Provide all R code and solutions by *knitting* your final RStudio file into a single file named `[your_name]_CA4.pdf`. This assignment is due next Tuesday by the beginning of class (by 2:39 PM, Tuesday, September 11). The assignment is worth 30 points. Late assignments will automatically have 10 points deducted. Remember to include all `/textttR` code and results (except when noted) to receive full credit.

1. Create the vector

```
> myAtomicVector <- c(1, 4, 3, 2, NA, 3.22, -44, 2, NA, 0, 22, 34)
```

Now, create code that runs to answer each of the following questions.

- (a) (2 points) How many positive numbers (> 0) are there in this vector (do not count NAs)?
 - (b) (2 points) How many negative numbers (< 0) are there in this vector (do not count NAs)?
 - (c) (2 points) How many 0's are there in this vector (do not count NAs)?
 - (d) (2 points) How many NAs are there in this vector?
 - (e) (2 points) How many numbers in the vector are non-zero **and** not NAs?
 - (f) (2 points) What is the sum of the positive numbers in this vector?
 - (g) (2 points) What is the sum of the negative numbers in this vector?
2. Consider a vector of length 1000, where F_n is the n th number in the sequence. Then the [Fibonacci sequence](#) is the vector where the following recursion holds:

$$F_n = F_{n-1} + F_{n-2}$$

That is, the n th number in the sequence will be the sum of the previous two numbers.

Create a vector, `Fib.vec`, that has the first 1000 numbers in the Fibonacci sequence using the following code (which includes a `for` loop that we'll talk more about later in this course):

```
#initialize the vector for memory
Fib.vec <- rep(0, 1000)

#store the first two entries to be 1
Fib.vec[1] <- 1
Fib.vec[2] <- 1

#iterate to get the remaining values
for(i in 3:1000){
  Fib.vec[i] <- Fib.vec[i-1] + Fib.vec[i-2]
}
```

- (a) (2 points) What are the first 8 and last 8 entries of `Fib.vec`?
- (b) (2 points) Using the Fibonacci numbers generated above, generate a vector (of length 999) with values (again, don't print these out)

$$Z_n = \frac{F_{n+1}}{F_n}$$

For this sequence, one could use a `for` loop as used in part (a), or better yet note that dividing two vectors of the same length will return a vector whose entries contain the division of entries in each vector. This is very useful for such calculations and this vector operation is fairly unique to `R`!

- (c) (2 points) Plot the first 10 entries of the vector Z_n using the command `plot(Z_n[1 : 10])`. Then add a line to the plot using the following command `abline(h = (1 + sqrt(5))/2)`. This value is known as the [golden ratio](#) in mathematics.
- (d) (2 points) Comment on the plot that you obtain. What do you observe?
- (e) (2 points) What is wrong with typing the following code?

```
x <- Fib.vec(1:5)
```

3. Using the Fibonacci vector above, create the following data structures. Remember that using the `?foo` will provide documentation on the function `foo` as needed. And be careful about the use of arguments here to get the data structure you want.

- A matrix of size 100×10 named `Fib.matrix1` whose columns, when stacked on top of one another will return the original vector.
- A matrix of size 100×10 named `Fib.matrix2` whose rows, when stacked side by side will return the original vector.
- An array of dimension $10 \times 10 \times 10$ names `Fib.array` where each 10×10 matrix in the array is such that when its columns are stacked on top of one another would generate a Fibonacci vector of length 100.

Answer the following questions

- (a) (2 points) What is the mean of the 18th row of `Fib.matrix1`?
- (b) (2 points) What is the standard deviation of the 8th column of `Fib.matrix2`?
- (c) (2 points) What is the entry in the 5th row of the 2nd column of the 8th matrix in `Fib.array`?