

# SDGB 7844 HW 1: R Syntax and Control Structures

Instructor: Prof. Nagaraja

Due: 9/27

Submit two files through Blackboard: (a) .Rmd R Markdown file with answers and code and (b) Word document of knitted R Markdown file. Your file should be named as follows: “HW1-[Full Name]-[Class Time]” and include those details in the body of your file.

Please submit your solutions only once! Complete your work individually and comment your code for full credit. For an example of how to format your homework see the files related to the Lecture 1 Exercises and the RMarkdown examples on Blackboard.

1. The vectors `state.name`, `state.area`, and `state.region` are pre-loaded in R and contain US state names, area (in square miles), and region respectively.
  - (a) Identify the data type for `state.name`, `state.area`, and `state.region`.
  - (b) What is the longest state name (including spaces)? How long is it?
  - (c) Compute the average area of the states which contain the word “New” at the start of the state name. Use the function `substr()`.
  - (d) Use the function `table()` to determine how many states are in each region. Use the function `kable()` to include the table in your solutions. (Notes: you will need the R package *knitr* to be able to use `kable()`. See the RMarkdown example in the Assignments folder on Blackboard for an example.)

2. Perfect numbers are those where the sum of the proper divisors (i.e., divisors other than the number itself) add up to the number. For example, 6 is a perfect number because its divisors, 1, 2, and 3, when summed, equal 6.

- (a) The following code was written to find the first 2 perfect numbers: 6 and 28; however, there are some errors in the code and the programmer forgot to add comments for readability. Debug and add comments to the following:

```
num.perfect <- 2
count <- 0
iter <- 2

while(count <= num.perfect){
  divisor <- 1

  for(i 2:(iter-1)){
    if(iter%%i==0) divisor <- c(divisor, i)
  } # end for loop

  if(sum(divisor)=iter){
    print(paste(iter, " is a perfect number", sep=""))
    count <- count + 1
  } # end if

  iter <- iter + 1

} # end while loop
```

- (b) Use the function `date()` at the start and at the end of your amended code. Then compute how long the program approximately takes to run (you can do this subtraction by hand). Find the run time when you set `num.perfect` to 1, 2, 3, and 4. Create a table of your results (Note: use the first table format in the RMarkdown Example file in the Assignments folder on Blackboard) . What are the first four perfect numbers?
- (c) Let `x <- 1:4` and define `y` to be the vector of run times. Plot `y` vs `x` using the code below. Is the relationship between the discovery of perfect numbers and run times on your computer linear? Justify your answer.

```
plot(x, y, pch=20, type="b",
     xlab="number of perfect numbers discovered",
     ylab="cumulative length of time (in seconds)",
     main="Cumulative Run Times to Discover Perfect Numbers",
     las=TRUE)
```

3. The geometric mean of a numeric vector  $x$  is computed as follows:

$$\tilde{x} = \left( \prod_{i=1}^n x_i \right)^{1/n}.$$

- (a) Using a **for** loop, write code to compute the geometric mean of the numeric vector  $x < -c(4, 67, 3)$ . Make sure your code (i) removes any **NA** values and (ii) prints an error message if there are any non-positive values in **x**.
- (b) Test your code on the following cases and show the output: (i)  $\{\text{NA}, 4, 67, 3\}$ , (ii)  $\{0, \text{NA}, 6\}$ , (iii)  $\{67, 3, \infty\}$ , and (iv)  $\{-\infty, 67, 3\}$ .