BIOS 545 Spring 2017 Homework 1

Due 11:59 PM on January 30, 2017

Instructions

There are 20 questions at 5 points each. Send responses via email in a file named using the convention of `BIOS545_LastName_FirstName_HW1.R`. You should use RStudio to create the file. Email to BOTH `dvandom@emory.edu` and `wsp@emory.edu`. Submissions arriving after the indicated due date and time will incur a 10 percent penalty for each day late.

All of these problems can be solved using material presented in class and in the labs. Unless otherwise indicated in the problem, you may use other R functions to help you find a solution although you cannot download additional packages to solve a problem. In some cases you might have to use the help mechanisms described in Week 1 to locate an appropriate function. We will run your R commands at the R console to verify the statements.

Problems

1) Create a vector called `x1` that contains every 5th number between 1 and 101, i.e. 1, 6, . . . , 96, 101.

2) Create a vector called `x2` that contains the sum of the squares of all even numbers/integers between -50 and 50

3) Execute the following commands that will generate 100 numbers between 0 and 10. Some of the numbers will be repeated more than once. Write a one line R statement that will present the percentage associated with the most frequently occurring number. Your answer will be a character string such as "19%". It might be helpful to review lecture notes for functions that relate to tabulation. Also recall that in R it is encouraged to use composite functions which makes it easy to do something on one line.

```
set.seed(123)
(myr <- rpois(100,4))
```

```
##   [1]  3  6  3  6  7  1  4  7  4  4  8  4  5  4  2  7  3  1  3  8  7  5  5
##  [24] 10  5  5  4  4  3  2  8  7  5  6  1  4  5  2  3  2  2  3  3  3  2  2
##  [47]  2  4  3  6  1  4  6  2  4  2  2  5  7  3  5  2  3  3  6  4  6  6  6
##  [70]  4  5  5  5  0  4  2  3  4  3  2  3  5  3  6  2  4  9  7  6  2  2  5
##  [93]  3  5  3  2  5  2  4  4
```

4) A codon is a sequence of three nucleotides that appear together and typically serve a particular biological purpose. The typical start codon is ATG and one possible stop codon is TAA. Create a 4-element vector of character strings called `dna2`, where each element is a 3-nucleotide codon. Create the vector so that it starts with the start codon, has GAA and CAC in the middle, and ends with the stop codon.

5-7) Run the code below to generate a single character string containing a sequence of 100 nucleotides.

```
dna3 <- paste(sample(c("A", "C", "G", "T"), 100, replace = T), collapse = "")
```

5) Create a new version of the `dna3` vector called `dna3.expanded` that is a 100-element character vector with each element either A, C, G, or T.

6) Because we sampled A, C, G, and T with equal probabilities, each should appear roughly 25 times in our sequence. Use an R function to tabulate the actual frequencies of each nucleotide.

7) Drop the first 50 nucleotides in `dna3.expanded` and then convert the remaining nucleotides back to compressed format, in a variable named `dna3.subset`.

8-11) The `quantmod` package allows you to download stock market prices from Yahoo! Finance. Run the code below to download `quantmod` from CRAN and load 10 years of stock prices for Apple, then answer the following questions.

```
install.packages("quantmod")
library(quantmod)
apple.prices <- as.matrix(getSymbols("AAPL", from = "2007-01-20", to = "2017-01-20",
                                     auto.assign = FALSE))
```

8) If you run `head(apple.prices)` you will see the structure of the matrix. The last column, labeled AAPL.adjusted, has the adjusted closing price for each trading day. Use bracket notation to extract the closing price for the first and last trading day. Save these values into variables called `first.price` and `last.price`.

9) Calculate the total percent return over the 10-year period by dividing `last.price` by `first.price`, subtracting 1, and then multiplying by 100. Notice that it's extremely high; regret not investing in Apple 10 years ago; and then move on to the next question.

10) Notice that the `apple.prices` matrix has dates for row names. Using a certain function covered in lecture, extract these row names into a vector called `apple.dates`.

11 If you run `class(apple.dates)`, you will see that despite looking like dates, R is treating them like character strings. Convert the `apple.dates` vector to date class, and call it `apple.realdates`.

12) Determine the highest closing price for Apple over the last 10 years, on what date that price was achieved, and how many calendar days have passed from then until today (use Jan. 20, 2017 for "today").

13) Notice that the `apple.prices` matrix also gives the highest and lowest price that the stock reached on each day. Using vector arithmetic, calculate the differences between the high and low price for each day, and save these values into a vector named `daily.swing`. Calculate the mean, median, and standard deviation for the values in this vector.

14) Normal theory tells us that if a variable follows a normal distribution, then approximately 68% of values fall within one standard deviation of the mean. Suppose we want a more precise estimate, and we can't find a z-table. Generate 1 million values from a standard normal distribution, store them in a vector called `x.normal`, and calculate the percent that are in the interval (-1, 1). (Note: the true value is 68.27%)

15-17) Bike sharing is a relatively new idea whereby people can rent out city bikes from various stations around a city, ride them around for a few hours, and then return the bike to the same station or a different one. The bike share program in San Francisco actually has a website with data on the trips people take.

Go to http://www.bayareabikeshare.com/open-data and click YEAR 1 DATA to download a zip file. Unzip it and move the file named `201402_trip_data.csv` to your working directory in R. Then load it into R by running:

```
read.csv("201402_trip_data.csv")
```

15) write R code to print the 3 most common start stations.

16) Determine the total number of trips summarized in this dataset, and how many of the trips had duration less than 120 minutes.

17) Create a vector called `start.dates` that has the dates indicated by Start.Date, but stored as a date/time object. Calculate the range of dates included in this bike share dataset.

18-19) Body mass index is calculated as body weight in kilograms divided by squared height in meters. Run the following code to generate some made-up weight and height values for a group of people.

```
height <- c(1.63, 1.84, 1.49, 1.73, 1.80)
weight <- c(80.1, 83.0, 62.4, 69.3, 74.2)
```

18) Create a vector called `bmi` that contains the BMI values for each of the five subjects.

19) Create a logical vector called `overweight` that is TRUE for participants with BMI greater than 25 and FALSE otherwise. Calculate the number that are overweight.

20) Create the matrix J so that

$$J_{i,j}$$

is equal to -1 where i + j $<$ 8 and zero otherwise

$$\mathbf{J} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 0 \\ -1 & -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \end{bmatrix}$$