*\*Please save this file as “LAST NAME\_Assignment 1.docx”*

***Open-Ended Responses***

1. **I.1** If you are defining an object called **vec.x**, you can do so by typing it into a script file first and then executing or by typing it directly into the console. Which way is better for reproducibility and why?

|  |
| --- |
| It is better to type it into a script file first then execute it because the script will keep track of each line of code that you run. This way you will be able to go back to see the exact script you ran for the anticipated output. This is more reproducible for someone to be able to get the same R output as you did. |

1. **I.5** Explain how R came up with the following result:

x <- 1:10

y <- 1:3

x-y

[1] 0 0 0 3 3 3 6 6 6 9

Warning message:

In x - y : longer object length is not a multiple of shorter object length

|  |
| --- |
| R subtracted the vector 1:3 from the vector 1:10 three times by repeating the vector 1:3 after it is subtracted each time. For example, subtracting 1-1, 2-2, 3-3, 4-1, 5-2, etc. R produces the output vector this way, where the last element of it is produced by subtracting 10-1. The warning is given because the subtracted vector y has three elements which does not divide evenly into vector x which has 10 elements. Therefore, vector y is not subtracted an even number of times. |

1. **I.6** Explain the behavior of the **round()** function observed below where 0.5 is rounded down, but 1.5 is rounded up.

round(.5)

[1] 0

round(1.5)

[1] 2

|  |
| --- |
| The round function default rounds to the even digit. For 0.5 it rounds to 0 because 0 is considered the even number, while 1 would be considered the odd number. For 1.5 R rounds to 2 because 2 is the even number, whereas 1 would be the odd number. |

1. **I.7** A researcher wants to create a data set by sampling 100 integers ranging from 50 to 75 with replacement, center those data (subtract the mean from each data point), and then calculate the centered mean divided by the centered standard deviation. Spot the silent error in the following code written to do this:

1 data <- sample(50:75, 100, replace = TRUE) # sample the data

2 data.cen <- data - mean(data) # center data on mean

3 mean(data) / sd(data) # calculate mean / sd

|  |
| --- |
| In step three this calculates the mean of the data divided by the standard deviation of the data. If the intention is to calculate the centered mean from step 2 divided by the centered standard deviation, then the R code needs to call mean(data.cen) / sd(data.cen). |

1. **I.7** Why does the following generate an error? Special note: if you copy/paste from this word document, it brings what are called “smart quotes” which R can’t recognize. You might need to type the quotes manually.

x <- c(1, 5, 3, 4, “3”) # runs fine

sum(x) # error generated

|  |
| --- |
| By putting 3 in quotations in the original x vector, R classifies the elements as characters instead of numbers. So there is a type argument generated when you try to run the sum(x) function. Sum only works with numeric elements, and not character (or text) arguments. |

1. **1.8** Below are two sections of code that accomplish the same thing. Which one would you think is the “best” way? There isn’t necessarily a right answer here, I just want to hear your rationale.

x <- sample(-50:50)

# Code Section 1

x.fil <- x[x > 0] # filter out all negative values

x.fil.sq <- x.fil^2 # square results

mean(x.fil.sq) # compute mean

# Code Section 1

mean((x[x > 0])^2) # filter, square, compute mean

|  |
| --- |
| I think that doing the second set of code is the best way because it does not convolute your environment with multiple unnecessary elements created in the first section of code. It also requires less code for this specific example. |

***Coding Section***

To complete this section, start a new script file with the following layout:

# YOUR NAME

# Assignment 1 Introduction to R

# #1 ---------------- (new section: CTRL + SHIFT + R)

here’s my code # with adequate commenting

# #2 ---------------- (new section: CTRL + SHIFT + R)

here’s my code # with adequate commenting

1. **I.2** Suppose I wrote 2 lines of code and then said the following: “Here, I defined an object x as a numeric vector that contains 5 numeric elements, the numbers 1 through 5. Then I told R to add 1 to each element and print the result.” What are the two lines of code?
2. **I.3** Cohen’s d is a metric that computes the effect size in a comparisons test (if you don’t know what I’m talking about, it’s okay). The formula is:

Graphical user interface

Description automatically generated with medium confidence

, where...

* d is Cohen’s d
* ME, MC are the means of experimental (E) and control (C) groups
* Sample SD pooled is the pooled standard deviation:

Diagram

Description automatically generated

Calculate Cohen’s d in R comparing the two simulated groups below. I would recommend using mean() and sd() to compute means and standard deviations (as opposed to doing them “by hand”). Note: there are functions in other packages that compute Cohen’s d for you, but do not use one of these functions here.

set.seed(42)

exp <- rnorm(100, 1.1, .1) # experimental

set.seed(42)

con <- rnorm(100, 1, .1) # control

1. **I.5** Create an object called **data** and define it as a numeric sequence that starts at 0 and goes to 200 in increments of 0.5 (*i.e.* 0, 0.5, 1, 1.5... ...199, 199.5, 200). Then, take a random sample of 50 points without replacement (cannot sample the same set of points more than once) and assign it to an object called **dat.sample**. Set the seed to 42 prior to sampling so we get the same result. Calculate a 5-number summary of **dat.sample** (minimum, 1st quartile, median, 3rd quartile, and maximum).
2. **I.8** Binning numbers is a pretty common task in research that entails taking a numeric vector and binning them into categories. The following will simulate this process commonly found in educational testing.
   1. Simulate a dataset that contains 200 students’ scores that follow a normal distribution (?rnorm) that have a class average of 80% and a standard deviation of 20%. Use a seed of 42 so we get the same data. You will notice that many of the sampled scored go above 100, which is not possible. Replace any number over 100 with 100.
   2. What is the average and standard deviation of just the students in the top third of the class?
   3. Create a vector that bins scores according to a common grade distribution (below) and tally how many of each grade were given
      * A >= 90%
      * B >= 80%, < 90%
      * C >= 70%, < 80%
      * D >= 60%, < 70%
      * F < 60%