

Please submit all R code

You may put your answers after the code as a comment

Remember to include useful comments. They will be graded

1. The precision of a floating point number is determined by its mantissa. R uses a double floating point number which has a mantissa of length 52 bits plus one “implied” bit for a total of 53 bits. To explore this concept, complete the following:

- a. Use R code to determine how large of a number can be expressed with 53 bits. Express your number in scientific notation with base 10 (this is how R will read out your number so no conversion is needed).
- b. R cannot store numbers with perfect precision that are larger than the answer to 1.a. To demonstrate this, create an object called `bigNum1` and assign a value of 99999....9 where the total number of 9’s in `bigNum1` is equal to the exponent from the answer to 1.a. Next, create a second object called `bigNum2` and assign a value of 999999....98 where the number of digits in `bigNum2` is the same as the number of digits in `bigNum1`. Subtract `bigNum2` from `bigNum1`. If R stored the numbers correctly, the result should be 1. Did R calculate the result correctly?

NOTE: Do not create `bigNum2` with the code: `bigNum1 - 1`. This will defeat the purpose of the exercise. `bigNum1` and `bigNum2` can be typed in manually but it would be better to create these numbers using a for loop.

- c. Update the value of `bigNum1` by multiplying it by 10 and adding 9. Update the value of `bigNum2` by multiplying it by 10 and adding 18. Subtract `bigNum2` from `bigNum1`. Again, the answer should be 1. Did R calculate the result correctly? Why or why not?
2. Use the `as.integer()` function to determine the integer versions of the following numbers: 1.0, 1.2, 1.5, 1.9, 1.99999. What do the answers tell you about how R treats rounding when converting a float into an integer?
 3. Create an object called `longFloat` and assign a value of 1.9999...9 where the total number of 9’s in `longFloat` is equal to the exponent from the answer to 1.a. Use the `as.integer()` function to convert to an integer. Did you receive the answer you expected? Now add one more 9 to the end of `longFloat` by dividing it by 10 and adding 1.8. Use the `as.integer()` function to convert the new value to an integer. Did you get the same result? Why do you think R behaves this way?
 4. Import the data file called “Students Data” as a data frame and do the following:
 - a. Using the `qplot()` function from the “ggplot” package, create a histogram of the grades of the students in the class. Label the x-axis “Grades”.
 - b. The ‘study session’ column indicates whether or not a student attended the optional study session, with TRUE indicating that the student did attend and FALSE indicating

that they did not attend. Use R code to determine what percentage of the students attended the study session.

- c. Extract the “gender” column from “Students Data” as a vector of character strings and store in an object called `genders`. Use this new vector to calculate how many males are in the class.