Statistical Computing - Exercises 08 - Floating Point Numbers

"There are only 10 kinds of people in this world: those who understand binary, and those who don't."

– Nerdy T-shirt

We are going to explore the floating point number system. The main objective of these exercises is to learn about the distribution of randomly selected floating point numbers, and learn about the spacing between floating point numbers. For all problems, you should include a short demonstration (one or two examples) showing that you coded your function correctly.

1. Write a function that has a logical vector (TRUE/FALSE) as its argument. The function should treat the logical vector as a binary integer in that FALSE = 0 and TRUE = 1. The function should calculate and return the corresponding integer. For example, for argument c(TRUE,FALSE), the function should return 2, since 10 in base 2 is 2 in base 10 (hence the joke above). For argument c(FALSE, TRUE, FALSE, TRUE), the function should return 5 because 0101 in base 2 is 5 in base 10.
2. Write a function that takes in a logical vector of length 64 and returns the sign (-1 or 1), the exponent (integer between -1023 and +1023), and fraction (number between 1 and 2) according to the IEEE double precision format. Assume that the first digit is the sign, the next digits are for the exponent, and the last digits are for the fraction.
3. Write a function that takes in the sign, exponent, and fraction, and returns the floating point number in decimal format. You do not need to worry about the special cases -Inf, Inf, and NaN.
4. Write a function that takes in an integer-power-of-two (e.g 4, 16, 64) and produces the next largest floating point number.
5. Take a random sample of 1 million floating point numbers. To do this, randomly sample length-64 logical vectors, and for each vector, calculate the corresponding floating point number.
6. Subsample your floating point numbers to look at only the positive numbers. Make a histogram of the logarithm of the positive values. What distribution does it resemble?
7. How many individual double precision floating point numbers are there? Do you think every individual number has been stored on some computer at some point in history? In other words, do you think there are any floating point numbers that have never appeared anywhere on any computer? Use outside sources to inform your answer.