Homework 1 Graded

Sheridan Grant

Must be uploaded to Canvas under "Homework 1 Graded" by
Monday, April 6 at 11:59pm

Instructions

Format your code using the style shown on the course website. Any time I ask you to demonstrate something, show something, generate something, etc., you must provide the code that does so. The grader will be running your code and verifying that it solves the problems presented below. Your code should not produce errors; if it does so, we will be lenient in grading this first assignment, but not in the future.

1 R Basics

(a) Write a line of code that generates a vector of length 47 and does not contain the number 1. [1pt]

(b) Write a line of code that assigns a vector of length 47 to a variable. [1pt]

(c) Write a line of code that generates a vector of length 47 with a mean of 0 and assigns it to a variable. Write another line of code that demonstrates that its mean really is 0. [2pts]

(d) Using no more than 5 lines of code, generate a vector of length 47 named "standardized" with a sample standard deviation of 1. It might be helpful to figure out what happens to the standard deviation of a mean-zero vector when it is multiplied by a constant. [3pts]

(e) Write code that computes the sum of all the multiples of 3 between 100 and 200. [3pts]

2 Data Types

We've seen a few data types in class so far, including numerics, integers, logicals, and vectors. You'll learn about a few more on your own for this problem.

The class of functions `is.[data type]` return `TRUE` if their argument is of type `[data type]` and `FALSE` otherwise. For instance, `is.numeric(x)` returns `TRUE` if `x` is numeric and `FALSE` otherwise.

(a) Assign the variable `notLogical` a value such that `is.logical(notLogical)` returns `FALSE`. [1pt]

(b) Write a single line of code using `is.logical` and `notLogical` (and some other stuff) that returns `TRUE`. [1pt]

(c) Write two lines of code that demonstrate that `is.numeric` and `is.integer` are not the same function. [2pts]

(d) One data type we did not talk about is the character type. Characters (or chars) begin and end with `"` or `'`. For example, `"Hello, world!"` and `'foo! bar!'` are both chars. The opening and closing quotes must match—`"incorrect'` is not a char. Assign your full name (as a character, including spaces) to the variable `name`. Write a line of code that verifies that `name` is a char type. [2pts]

(e) Write a line of code that returns your name in all uppercase. You will need to find a function that does this, and apply it to `name`. [2pts]

3 Normal Distribution Functions

For many distributions, R includes 4 functions that do useful things. For the Normal distribution, they are `rnorm`, `pnorm`, `dnorm`, and `qnorm`. You should read about these functions on your own so you understand what they do.

(a) Using `pnorm` (which we saw in class), demonstrate that there is approximately a 95% chance that a Normal random variable lies between -1.96 and 1.96. [2pts]

(b) Using `qnorm`, demonstrate that there is exactly a 95% chance that a Normal random variable lies between approximately -1.96 and approximately 1.96. [2pts]

(c) Using `rnorm`, generate a vector of $10,000$ samples from the standard Normal distribution. What percentage of them are between -1.96 and 1.96? [2pts]

(d) Using `dnorm` and the vector from the previous problem, show that the value in the vector with the highest density (under the standard Normal distribution) is the one that is closest to zero. [3pts]

(e) Generate a histogram of the vector from part (c). You can do this very easily with R, but you'll need to figure out how on your own. You do not need to turn in an image file of the histogram—just write code that generates a histogram.

[2pts] For an extra point, give the histogram a title and x-axis label of your choosing that are different from the defaults. [1pt]

4 Binomial Distribution

You should be familiar with the Binomial distribution from a previous stats class. A Binom(n, p) random variable can be generated by obtaining a coin that comes up heads with probability p, flipping it n times, and counting the number of heads.

(a) Generate two vectors each of length $10,000$, named `b1` and `b2`,[1] from Binom$(100, p_1)$ and Binom$(100, p_2)$ distributions. Choose p_1 and p_2 so that the distribution of `b1` is symmetric, and the distribution of `b2` is asymmetric. (`b1` need not be *exactly* symmetric, but it should be close. You can easily Google how to make a symmetric binomial distribution, also.) Write code that generates histograms of each vector (`b1`'s histogram should clearly be symmetric, and `b2`'s should clearly be asymmetric). [4pts]

(b) Using `rnorm`, generate $10,000$ samples from a Normal distribution with mean 50 and standard deviation 5. Write code that generates a histogram of this vector. Write a comment that compares this histogram to the histogram of `b1`. [3pts]

(c) Using `rnorm`, generate $10,000$ samples from a Normal distribution with the same mean and standard deviation as the distribution you used to generate `b2` (you will again need to rely on your knowledge from a previous stats class). Write code that generates a histogram of this vector. Write a comment that compares this histogram to the histogram of `b2`. [3pts]

[1] Yes, you can include numbers in a variable name, as long as the very first character is a letter.