Statistics 4861/6861

Assignment 1 (100 points) Due: Wednesday, October 22

Email me one file. Do not use a “reply” to email your file. Use .R as the extension. The file name submitted by a student named Albert Einstein is AlbertEinsteinAssign1.R I should be able to load your code into the R environment by using the “Open Script” feature of R. Then I should be able to run your code. Put your name as the first line of your .R file. This will be a line that is executable, and look this: name = “Albert Einstein”. Of course your own name will be substituted for Albert Einstein. When I run tests on your code, I will want to know whose code is running. Your code should be well-commented, and indented so a reader can easily follow the code. Bring a print out of your file to class. You one file should contain two functions.

1. Write a function that merges two already sorted vectors into a third sorted vector. The input vectors are sorted in ascending order. The function prototype is:

   merge.sort <- function(in1,in2).

   Example: x = c(1,2,3,4) and y = c(1.5,3,5). Then z = merge.sort(x,y) results in z = c(1,1.5,2,3,3,5)

2. Write a function that bins data. This is the kind of thing one does when making a histogram. We are given a data vector x, and a vector containing the boundary of the bins. This vector is called bins.
   Function prototype is:
   bin.data <- function(x,bins)

   Check that bins is strictly increasing. The bins are open on the left and closed on the right (except for the last bin). For example, if bins = c(2.5,5,7.8,9) you are to determine if an element of x falls into the bin (-Inf,2.5], the bin (2.5,5], the bin (5,7.8], the bin (7.8,9], or the bin (9,Inf). Here x is a numeric data vector, and we want to bin the data. If bins has length m, then we return a vector of length (m+1). We do not allow –Inf or Inf values in bins.

   The purpose of the function is to return a count of how many elements of x fall into each bin. Using the bins vector as defined above, If x = c(8.3, -2, 2.3, 7.9, 2.5, 2.51, 8.5, -8.9, 9.2) we return the vector c(4,1,0,3,1). The explanation of the output vector is given below.

   There are four values of i such that x[i] <= bins[1]
   There is one value of i such that bins[1] < x[i] <= bins[2]
   There are zero values of i such that bins[2] < x[i] <= bins[3]
   There are three values of i such that bins[3] < x[i] <= bins[4]
   There is one value of i such that bins[4] < x[i]