

R- Exercise: Logic Overview and Assumptions

Pavani Satish Kalepalli

sat37@outlook.com 716-289-6839

Problem 1

- The function f1 takes a vector of numeric values and a 1-element vector N as inputs
- The function does not coerce other data types into vectors: It is assumed that the user tests the function only with valid numeric vectors
- If the user passes other data types, no further processing is done and the function returns a null value with the error message displayed to the user
- The user input N must be divisible with the number of elements in the input vector for constructing a matrix out of the input vector of numbers.
- If N doesn't exactly divide the number of input elements, the elements of the input vector are repeated during the matrix creation
- Since the matrix function displays a warning when repeating the elements of the vector, it is suppressed from being displayed to the end user.
- The difference between the actual number of elements in the input vector and the number of elements in the reconstructed matrix is computed so that the additional items can be removed later without being displayed as part of the output.
- An 'lapply' function is used to convert all the columns of the matrix into a list of vectors. All the additional values that were repeated during matrix creation are removed from the output. The object returned by this function is a list of vectors and those vectors could be accessed using `op1[[1]]`, `op1[[2]]` etc. where 1,2 and so on are the vector numbers in the output list.

Problem 2

- The function f2 accepts a vector of integers as input and returns a data frame as output. This function doesn't coerce other data types into vectors and displays an error message if other data types are passed as an input
- It is assumed that the end user passes a valid vector of integers as input. However, the function also works with a vector of real numbers.
- A table is created on the input vector. This table captures the frequency of each element in the input vector
- A data frame is created from the above table to display the values and frequencies as its two columns
- As the problem statement requires an output of only those values which occur more than once, only a subset of the above data frame meeting this condition is returned as the output

Problem 3

- The function f3 accepts a vector of strings as input. It is assumed that the user supplies valid dates as the strings of the input vector. And if the user fails to supply valid date strings as the input, the corresponding fields in the output are marked by NA values.
- As a first step in this function, the input strings are converted into valid dates. Moreover, it is assumed that the input strings are of the type m/d/ y as given in the problem statement. All other formats of dates are treated as invalid values and again marked by NA fields in the corresponding output.
- As a second step, the difference between given date and current date is calculated. An absolute difference is used here as the problem statement didn't require us to output negative values.
- The 'quarter' value is assigned based on the divisibility of difference value computed in the previous step with 91.0. Once the quarter value is assigned, an output frame is constructed from the input, difference and quarter values
- It is to be noted that the 'InputDate' field in the data frame is exactly same as the input that was provided to the function and not how the function interprets the dates into a valid format.

Problem 4

- The function accepts a vector of strings as input and outputs a list of lists as output and each sub list consists of two vectors corresponding to amounts and dates
- In the pattern matching of dollar amounts, non-dollar numeric values and dates, it is assumed that all these values are surrounded by spaces.
- If the numeric values are concatenated with other alphabets or punctuations, and not separated from them by spaces, they are not treated as numbers or dollar amounts. They are considered as user IDs or complex variable names and hence not extracted from the string.
- During the extraction of numeric values and dates, NO other punctuation such as braces, parentheses, arithmetic operators, colons, semi-colons, and special characters are handled. During the extraction of numeric values, the following cases were also considered: comma-formatted dollar values, decimal, non-decimal values, and the first digit never being zero.
- The date format mentioned in the problem statement is the only valid format for pattern matching. Here, it is assumed from the given example that the year is a 4-digit value and only '/' is used as a separator between the fields of the date.
- It is assumed that the input strings are as simple and meaningful as the one given in the problem statement and no other fancy characters are handled during the pattern match.
- An 'lapply' function is used to create a list out of the two output vectors for each string. And the function returns a list of these lists.
- The vectors in the output list can be accessed using these syntaxes:
op4[[1]]\$amt[[1]] or op4[[1]]\$date[[1]] where 1 corresponds to the first string of the input vector.