Biostatistics 615 Mastery Assignment #5 (10 pts)

Due by October 22nd 2024 (Tuesday) 11:59pm. Use Gradescope (via Canvas) to submit an R file.

- Your submission should only contain one `R` file named `solveMatrixEquation.R` that contains a function named `solveMatrixEquation(A, B, C)`.
- Your code will be evaluated in Gradescope using 10 different test cases using an automated script. Full credit will be given if your code passes all test cases.
- You are allowed to submit multiple times before the deadline, but only the last submission will be graded. Automated feedback will be provided for each submission.
- You need to implement the function to work with arbitrary (valid) input values beyond the 10 cases tested. If you tweak your implementation so that your functions works specifically for the test cases, you will not receive any credit.
- You may test the function in the Google Colab page at https://bit.ly/615hw5xtra to test your code on a subset of test cases.
- Implement your function as efficient as you can. If your program does not finish after running for 10 seconds for case 1-7, 12 seconds for case 8, 15 seconds for case 9, and 20 seconds for case 10, respectively, you will lose the points for those test cases. Note that the official solution finishes much faster for any test case, so this should be a reasonable time limit. Also, note that the running time includes the time for reading the input files. The time reported in the Google Colab test page will be much faster because the input files are already loaded in the memory.
- All the returned values should have at least 8 correct significant digits, so make sure that your implementation retains sufficient numerical precision.

Problem 1 - Solve Matrix Equation (10 pts)

Consider a matrix linear equation

$$AX + XB = C,$$

where A, B, C and X are all of dimension $n \times n$. A, B, C are given matrices and X is an unknown sparse matrix only taking the integer values (i.e. rounded to nearest integer).

Write an R script `solveMatrixEquation.R` containing a function `solveMatrixEquation(A, B, C)` to solve this equation to obtain the estimate of X.

The input arguments are squared sparse matrices A, B, C of the same size, provided outside the function as exemplified in the evaluation code. The return value is a sparse matrix of X, stored in triplet (or COO) format (i.e. `Tsparsematrix`).

Using `library(Matrix)` is expected, and no other packages except for the `base` package may be used for this problem. No error handling for malformed argument is needed. Example output of running the test code is given below:

```
> A = readRDS("args/test.1.A.rds")
> B = readRDS("args/test.1.B.rds")
> C = readRDS("args/test.1.C.rds")
> rst = solveMatrixEquation(A, B, C)
> print(data.frame(i=rst@i+1, j=rst@j+1, x=rst@x),row.names=FALSE)
 i j x
 1 1 9
 3 1 8
 2 2 3
 3 2 1
 1 3 4
 2 3 6
```

The example input files `test.1.A.rds`, `test.1.B.rds`, and `test.1.C.rds` are provided within the file `hw5_xtra_examples.tar.gz` accessible in https://bit.ly/615hw5xtra.

The actual examples that will be used for grading will contain larger matrices, so be sure to implement your function to work with arbitrary valid input values.

As noted above, for this problem, you are allowed to use the `Matrix` library. Note that the `Matrix` library is already loaded, and you should NOT include `library(Matrix)` in your code. Calling `library(...)` in your submitted cause will cause errors. Note that you are NOT allowed to use any functions outside the `base` and `Matrix` package in your implementation. Use `help(...)` to check whether a function belongs to the `base` or `Matrix` package or not.

The cloud machine that will be used for grading has a very small (0.5GB) memory, so make sure that you do not use too much memory in your implementation. If you code runs out of memory, "Killed" error messages will appear in the user output.

No error handling for malformed arguments is needed.