Biostatistics 615 Learning Exercise #7 (10 pts)

Due by October 8th 2024 (Tuesday) 11:59pm. Use Gradescope (via Canvas) to submit an R file.

- Your submission should only contain one `R` file named `euclideanDistanceThreshold.R` that contains a function named `euclideanDistanceThreshold(X, Y, thres)`.
- 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.
- Implement your function as efficient as you can. If your program does not finish within the time limit for each test case, you will lose the points for those test cases. Note that the official solution finishes much faster than the test cases, so this should be a reasonable time limit.

Problem 1 - Pairwise Calculation of Euclidean Distance (10 pts)

Write an R function `euclideanDistanceThreshold(X, Y, thres)` in the file named `euclideanDistanceThreshold.R` so that it returns the number of column pairs with Euclidean distance less than or equal to `thres` from two matrices `X` and `Y` with the same number of rows.

Specifically, let X be a $p \times n$ matrix and Y be a $p \times m$ matrix. The Euclidean distance between two columns, $X_{\cdot i}$ and $Y_{\cdot j}$, is defined as

$$D_{ij} = \sqrt{\sum_{k=1}^{p} (X_{ki} - Y_{kj})^2}.$$

Your task is to implement a function `euclideanDistanceThreshold(X, Y, thres)` that takes two matrices `X` and `Y` and a threshold `thres` as input and returns the number of column pairs (i, j) such that $D_{ij} \leq$ `thres`.

The input matrices `X` and `Y` are large matrices with the same number of rows. An example input and corresponding results are provided as follows.

```
> set.seed(1234)
> X = matrix(rnorm(1e5),100,1000)
> Y = matrix(rnorm(2e5),100,2000)
> system.time(rst <- euclideanDistanceThreshold(X, Y, 15))
user  system elapsed
0.150   0.003   0.154
> print(rst)
[1] 1633080
```

There are specific requirements in the implementation:

- You are NOT allowed to use any functions outside the `base` package in your implementation. Use `help(...)` to check whether a function you want to use belongs to the `base` package or not.

- Your answer should be accurate up to the decimal point.

- Each test case must finish within 5 seconds. Each pair of input matrices typically contains millions of elements, and the machine running the test cases has has a small (0.5GB) memory, so it is important to implement your function efficient in both computational cost and memory usage.

- The input matrices are NOT sparse, so you do not need to consider sparse matrix operations.

You do not need to implement error handling for malformed arguments in this function.