Biostatistics 615 Learning Exercise #8 (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 `piecewiseLinearInterpolation.R` that contains a function named `piecewiseLinearInterpolation(X, Y, Z)`.
- 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 - Piecewise Linear Interpolation (10 pts)

Suppose you observe n data points $\{x_i, y_i\}_{i=1}^n$ with $x_i, y_i \in \mathbb{R}$ from a noise free model:

$$y_i = f(x_i),$$

where $f(x)$ is an unknown **quadratic** function and the coefficients are all integers. Your goal is to perform a piecewise linear interpolation on a set of data points $\{z_j\}_{j=1}^m$.

Write an R script `piecewiseLinearInterpolation.R`, containing a function `piecewiseLinearInterpolation(X, Y, Z)` that performs piecewise linear interpolation and returns a data frame containing the following two attributes in the given order:

- `f.hat` : A vector containing $\hat{f}(z_j)$ for $j \in \{1, \ldots, m\}$.
- `f.true` : A vector containing $f(z_j)$ for $j \in \{1, \ldots, m\}$ for the quadratic function f.

The input arguments are vectors x, y of the same size, and another vector z containing the values to be evaluated. These values will be provided and loaded outside the function and passed as arguments. No 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:

```
> source('piecewiseLinearInterpolation.R')
> X = readRDS('test.1.X.rds')
> Y = readRDS('test.1.Y.rds')
> Z = readRDS('test.1.Z.rds')
> rst = piecewiseLinearInterpolation(X,Y,Z)
> dim(rst)
[1] 1000    2
> head(rst)
```

```
      f.hat     f.true
1 -5235.1849 -5235.1577
2 -4161.6171 -4161.5799
3  -592.6289  -592.6287
4 -4323.0815 -4323.0793
5 -2602.5209 -2602.5208
6 -2361.2254 -2361.2247
```

The input RDS files relevant to this example is provided on the Canvas homework page as `ex8testcase1.zip`.

Note that, for this problem, it is particularly encouraged to reuse the existing functions in lecture slides rather than implementing everything from scratch. For all problems, you can include as many functions as you want as long as your code contains the required function.

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 5 significant digits for each output values.
- Each test case must finish within 1 second.

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