Biostatistics 615 Learning Exercise #11 (10 pts)

Due by November 12th 2024 (Tuesday) 11:59pm. Use Gradescope (via Canvas) to submit an R file.

- Your submission should only contain one `R` file named `rejectionSampling.R` that contains a function named `rejectionSampling(n, a, b)`.
- 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 - Rejection Sampling (10 pts)

Write an R script `rejectionSampling.R` that contains function `rejectionSampling(n, a, b)` based on the following specification.

$$\pi(x) = C(a,b)\exp(-x^2)x^{a-1}(1-x)^{b-1}, \text{for } x \in (0,1),$$

where $a > 1$, $b > 1$ and $C(a,b) = \left\{\int_0^1 \exp(-x^2)x^{a-1}(1-x)^{b-1}dx\right\}^{-1}$. Choose Beta$(a,b)$ to construct the envelope density.

Given two real numbers a and b as arguments, the function simulates n random numbers, which should be at least **five million** in the evaluation, from $\pi(x)$. Using those random numbers, your function should return a `list()` object containing the following three attributes:

- `attempted` : The total number of attempted simulations, which should be equal or greater than `n`.
- `accepted` : The total number of accepted simulations, which should be equal to `n`, as well as `length(values)`.
- `values` : The vector of accepted values. The length should be equivalent to `n`

The evaluation code provided computes and prints the acceptance probability in the first line and quantiles at $0.01, 0.25, 0.5, 0.75$ and 0.99 separated by white spaces in the second line. Note that all the evaluation metrics requires only have one valid precision digit (you do not need to calculate these metrics or round the values yourself).

Example output of running the test code is given below:

```
> samp <- rejectionSampling(5000000,18.1,21.0)
> samp$accepted  # number of accepted samples
[1] 5e+06
> print(samp$accepted/samp$attempted, digits=1)  # print acceptance ratio
[1] 0.8
> print(quantile(samp$values,probs=c(0.01, 0.25, 0.5, 0.75, 0.99),na.rm=TRUE),
    digits=1)
 1% 25% 50% 75% 99%
0.3 0.4 0.5 0.5 0.6
```

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` and `stats` package in your implementation. Use `help(...)` to check whether a function you want to use belongs to the allowed packages or not.
- Your answer should be accurate up to 1 significant digits for each output values.
- Each test case must finish within 5 seconds.

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