Part 1 In this part, you are asked to work with the Markov Chain Monte Carlo algorithm, in particular the Metropolis-Hastings algorithm. The aim is to simulate random numbers for the distribution with probability density function given below

$$f(x) = \frac{1}{2}\exp\left(-|x|\right),$$

where x takes values in the real line and $|x|$ denotes the absolute value of x. More specifically, you are asked to generate $x_0, x_1, \ldots, x_N$ values and store them using the following version of the Metropolis-Hastings algorithm (also known as random walk Metropolis) that consists of the steps below:

Random walk Metropolis

Step 1 Set up an initial value x_0 as well as a positive integer N and a positive real number s.

Step 2 Repeat the following procedure for $i = 1, \ldots, N$:

- Simulate a random number x_* from the Normal distribution with mean x_{i-1} and standard deviation s.
- Compute the ratio
$$r\left(x_*, x_{i-1}\right) = \frac{f\left(x_*\right)}{f\left(x_{i-1}\right)}.$$
- Generate a random number u from the uniform distribution between 0 and 1.
- If $u < r\left(x_*, x_{i-1}\right)$, set $x_i = x_*$, else set $x_i = x_{i-1}$.

(a) Apply the random walk Metropolis algorithm using $N = 10000$ and $s = 1$. Use the generated samples $(x_1, \ldots x_N)$ to construct a histogram and a kernel density plot in the same figure. Note that these provide estimates of $f(x)$.Overlay a graph of $f(x)$ on this figure to visualise the quality of these estimates. Also, report the sample mean and standard deviation of the generated samples (Note: these are also known as the Monte Carlo estimates of the mean and standard deviation respectively).

Practical tip: To avoid numerical errors, it is better to use the equivalent criterion $\log u < \log r\left(x_*, x_{i-1}\right) = \log f\left(x_*\right) - \log f\left(x_{i-1}\right)$ instead of $u < r\left(x_*, x_{i-1}\right)$.

(b) The operations in part 1(a) are based on the assumption that the algorithm has converged. One of the most widely used convergence diagnostics is the so-called $\widehat{R}$ value. In order to obtain a valued of this diagnostic, you need to apply the procedure below:

- Generate more than one sequence of $x_0, \ldots, x_N$, potentially using different initial values x_0. Denote each of these sequences, also known as chains, by $(x_0^{(j)}, x_1^{(j)}, \ldots, x_N^{(j)})$ for $j = 1, 2, \ldots, J$.
- Define and compute M_j as the sample mean of chain j as

$$M_j = \frac{1}{N}\sum_{i=1}^{N} x_i^{(j)}.$$

and V_j as the within sample variance of chain j as

$$V_j = \frac{1}{N}\sum_{i=1}^{N}(x_i^{(j)} - M_j)^2.$$

- Define and compute the overall within sample variance W as

$$W = \frac{1}{J}\sum_{j=1}^{J} V_j$$

- Define and compute the overall sample mean M as

$$M = \frac{1}{J}\sum_{j=1}^{J} M_j,$$

and the between sample variance B as

$$B = \frac{1}{J}\sum_{j=1}^{J}(M_j - M)^2$$

- Compute the $\widehat{R}$ value as

$$\widehat{R} = \sqrt{\frac{B+W}{W}}$$

In general, values of $\widehat{R}$ close to 1 indicate convergence, and it is usually desired for $\widehat{R}$ to be lower than 1.05. Calculate the $\widehat{R}$ for the random walk Metropolis algorithm with $N = 2000$, $s = 0.001$ and $J = 4$. Keeping N and J fixed, provide a plot of the values of $\widehat{R}$ over a grid of s values in the interval between 0.001 and 1.