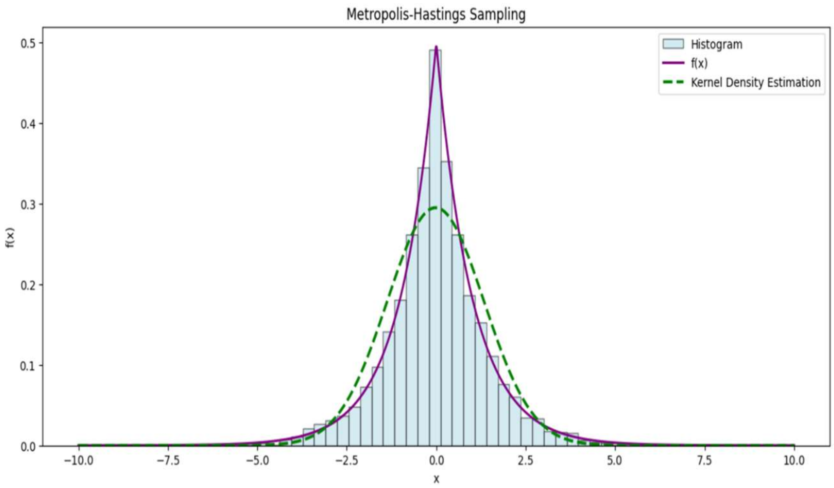
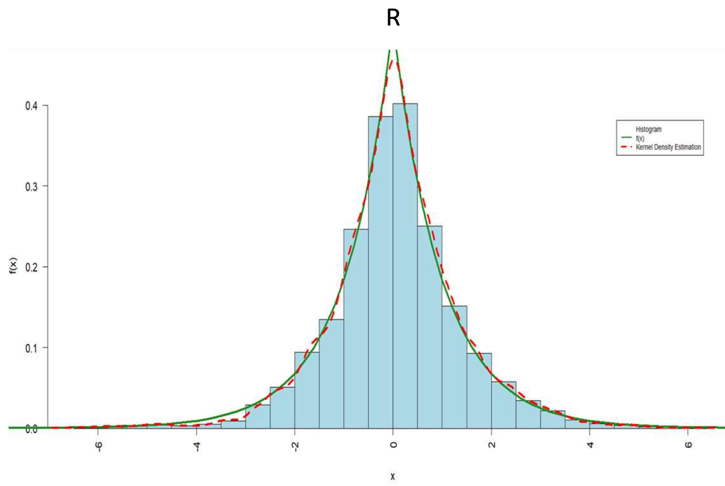
PART 1

(a) The Metropolis-Hastings algorithm is used to simulate random numbers from complex probability distributions. The target probability density function was defined as f(x)=0.5exp(|x|), representing a distribution that decays exponentially as |x| increases. To apply the algorithm, parameters were set, with the number of samples to be generated(N) = 10000, the standard deviation (s)= 1 and the initial value (x0) =0. Hence for each iteration from 1 to N (10,000 times), a new sample x* was proposed from a Normal Distribution, centred at the current sample x(i-1) and standard deviation s. Then the acceptance ratio r= f(x*)/ f(x(i-1)) was calculated, which determines the likelihood of accepting the proposed sample based on the target distribution f(x). Next, a random number 'u' is generated from a Uniform Distribution, between 0 and 1. If u<r, the proposed sample x* is more probable under f(x) compared to the current sample x(i-1). Thus, we accept the proposed sample and set xi = x*. Otherwise, xi = x(i-1). A histogram with 40 bins is drawn to illustrate the density of samples across a range of x. It is overlaid with the sample pdf f(x) for visual comparison. The Kernel Density plot, computed using the normal distribution, is a smooth curve and aligns closely with f(x), which indicates the effectiveness of the sampling algorithm. The sample mean = -0.0195 is close to zero, which aligns with the shape of f(x). The sample standard deviation = 1.35 is relatively large and indicates variability in the sample distribution. Hence, in conclusion, the Metropolis-Hastings algorithm has effectively explored the target distribution and generated samples that capture its central tendency while also reflecting its variability.





(b) The Gelman-Rubin statistic (R^) is employed to evaluate the convergence of the Metropolis-Hastings algorithm, by running multiple chains with different initial values (to ensure the reliability of the generated samples). First, a function is defined to compute the R^ statistic. It calculates the sample mean (Mj) and within-sample variance (Vj) for each chain. It then computes the overall within-sample variance (W), overall sample mean (M) and between-sample variance (B). Finally, it calculates the R^ statistic using the formula (B+W/W) ^0.5. Next, a function is defined to execute multiple chains of the Metropolis-Hastings algorithm and stores the generated chains in an array. It generates J=4 chains, each consisting of N= 2000 iterations. Chains are initialized with an initial value x0=0 and standard deviation s=0.001. A range of s values from 0.001 to 1 is then generated, and the R^ value is calculated for each s value. In order to visualise how the convergence varies with different step sizes, the R^ values are plotted against a grid of s values. The calculated R^ values using the initial parameters (N=2000, s=0.001, J=4) is 1.1875. This suggests potential lack of convergence, as it exceeds the desirable threshold of 1.05 (ideally, lower R^ values closer to 1 indicate better convergence). Through the graph, it can be observed that for certain (larger) s values, R^ is closer to 1 and depicts better convergence. Therefore, based on the analysis, adjusting/increasing the step-size s could potentially improve convergence and increase the reliability of the generated samples from the Metropolis-Hastings algorithm, as indicated by variations in the R^ values.