AGU INTERNATIONAL JOURNAL OF SCIENCES
p-ISSN 3093 - 3536

GENERATING RANDOM NUMBERS FOR STANDARD AND NON-STANDARD POSTERIOR DISTRIBUTIONS

Pham Thi Thu Hoa$^{(1)}$ Pham Thi Thu Huong$^{(1)}$
^{1}An Giang University, VNU-HCM

Information:
Received:20/3/2025
Accepted:12/08/2025
Published:12/2025

Keywords:
Generating random numbers, Bayesian statistics, standard posterior, non-standard posterior.

ABSTRACT

Bayesian statistics make inferences for unknown parameters with respect to posterior distributions. Under Bayes' rule, functions of posterior distributions are established using likelihood function and prior distributions. However, these posterior functions usually have non-standard and unpredictable forms. So, generating a sample from posterior distribution is very crucial to make inferences in Bayesian statistics. The two algorithms for generating samples from a posterior distribution are proposed in this paper. We also apply both algorithms to generate samples for standard distributions and non-standard distributions. The Shapiro-Wilk tests are performed to show the accuracy of both algorithms

1. INTRODUCTION

Recently, Bayesian statistics received great attention from statisticians. The ideas for making inferences on interval estimation of Bayesian statistics are more convincing and understandable than hypothesis testing with the help of modern computing machines. A primary idea for Bayesian thinking is that it provides a common-sense interpretation of statistical conclusions (Robert & Casella, 2004). For instance, a Bayesian interval for an unknown quantity of interest can be directly regarded as having a high probability of containing the unknown quantity. Moreover, the information about prior distribution can be used effectively in Bayesian statistics while they have no role in classical statistics (Choi & Hobert, 2013; Wakefield, 2013). This provides a strong impetus to the Bayesian viewpoint.

In Bayesian analysis, unknown parameters are considered as random variables. So, they have distributions. We can draw probability distributions and make inference of unknown parameters. These distributions are called posterior distribution or target distribution in a model performance. There are three elements to perform a standard Bayesian analysis: the observed data y, the likelihood function $\Pr(y \mid \theta)$ with parameter θ, a prior distribution $\Pr(\theta)$ (Hastings, 1970; Wakefield, 2013; Ghosh & Mitra, 2018). The Bayes' rule has the form of

$$\begin{aligned}\Pr(\theta \mid y) &= \frac{\Pr(y \mid \theta)\Pr(\theta)}{\Pr(y)} \\ &\propto \Pr(y \mid \theta)\Pr(\theta).\end{aligned} \qquad (1.1)$$

Here, $\Pr(y \mid \theta)$ is the likelihood function, $\Pr(\theta)$ is the prior distribution, $\Pr(y)$ is the normalizing constant and $\Pr(\theta \mid y)$ is the posterior distribution. The normalizing constant has the form of

$$\Pr(y) = \int_{\theta} \Pr(y \mid \theta)\Pr(\theta) d\theta.$$

This constant is independent with parameter θ and this term is usually ignored in the Bayes' rule.

From (1.1), based on likelihood function (data observations), prior distribution, we can obtain the formula of target distribution or posterior distribution of the interested parameters. The posterior distribution can have any form of distribution such as close form distribution which is the distribution we know about or non-standard form which is the distribution we do not know about. In these cases, the best thing is the capability of generating random samples from our posterior distributions (Robert & Casella, 2004). When we have a sample from the target distribution, we can easily obtain statistical inferences about the unknown parameters. Drawing samples from the posterior distribution is the goal in Bayesian computations. This requires the ability to generate random numbers from a nonstandard density or distribution.

In this paper, we propose two algorithms to generate random variables from standard and non-standard target distributions using fundamental theorem of simulation. The first algorithm is easier to implement than the second algorithm. However, the second algorithm can overcome some drawbacks of the first algorithm about the truncated domain and the acceptance rate. Both of them are applied to generate samples from a normal distribution and a nonstandard distribution.

The paper is organized as follows. The fundamental theorem of simulation has been presented in Section 2.1. The detailed steps of Algorithm 1 and Algorithm 2 are introduced in Section 2.2 and 2.3. In Section 3, the simulation studies are conducted for standard distribution which is normal distribution in Section 3.1 and non-standard distribution in Section 3.2. The conclusion is discussed in Section 4.

2. ALGORITHMS FOR RANDOM NUMBERS GENERATION

2.1 Fundamental theorem of simulation

The general ideas for simulating a random variable, having a certain density function, have been presented in the fundamental theorem (Robert, C. & Casella, G, 2004; Marin, J & Robert, C. P, 2014). For a given density function $f(x)$, we have

$$f(x) = \int_{0}^{f(x)} du. \qquad (2.1)$$

Here, U is a random variable having uniform distribution from 0 to $f(x)$. So, $f(x)$ is also the marginal density function of variable X in the joint distribution

$$(X,U) \sim U\left\{(x,u) : 0 < u < f(x)\right\}.$$

Thus, we can say that simulating for a random variable X with given density function $f(x)$ is equivalent to simulate for a joint distribution $\left(X,U\right)$ in which U has uniform distribution from 0 to $f(x)$ and variable X obtain randomly values in its domain of X when $u < f(x)$.

2.2 Algorithm for generating random numbers from fundamental theorem

From the Fundamental theorem in (2.1), we propose the algorithm for simulating a random variable with density function $f(x)$ as follows:

Algorithm 1:

Step 1: Set a n number of values for generating for random variable X .

Step 2: Generate n values for variable Y having uniform distribution from the domain of variable X .

Step 3: Generate n values for variable U having uniform distribution from 0 to m . Here, $U \sim U(0,m)$ and m are the maximum value of the density $f(x)$.

Step 4: Set $X = y$ for pairs where $u < f(y)$.

Here, the density of uniform distribution in (a,b) has the form of

$$f(x) = \begin{cases} \dfrac{1}{b-a}, & a < x < b \\ 0 & , x \notin (a,b) \end{cases}.$$

This algorithm will result in n values for a random variable X from a density $f(x)$.

2.3 *An improved algorithm for generating random numbers*

There are some problems with Algorithm 1. First problem is that from Step 1, we generate a uniform variable Y which has truncated density function. In reality, the variable X usually does not have finite truncated domain. The second problem is when we accept the value for random variable X from a uniform variable Y , we throw away a lot of points. So, the acceptance rate is low. To minimize the problems from Algorithm 1, we propose an improved algorithm called Algorithm 2 as follows:

Algorithm 2:

Step 1: Set a n number of values for generating for random variable X .

Step 2: Generate n values for variable Y having a known distribution which is cover the tails of the random variable X .

Step 3: Generate n values for variable U having uniform distribution from 0 to $Mg(y)$. Here, $g(y)$ is the density function of Y and M is constant so that $Mg(y) > f(x)$

Step 4: Set $X = y$ for pairs where $u < f(y)$

This algorithm will result in n values for a random variable X from a density $f(x)$ with higher acceptance rate than Algorithm 1.

3. SIMULATION

3.1 *Generating random numbers for normal distribtuion*

Using Algorithm 1, we generate a sample for standard normal distribution. The density function for standard normal distribution has the form of

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \qquad (3.1)$$

We generate 1000 random points for a uniform variable Y having values from -3 to 3 and 1000 random points for a uniform variable U having values from 0 to $1/\sqrt{2\pi}$. The picture of the points for variable Y and normal density function are presented in Figure 1a. After applying Algorithm 1, there are approximately 400 points that are accepted resulting in 40% of acceptance rate. The picture of accepted points for standard normal distribution is presented in Figure 1b.

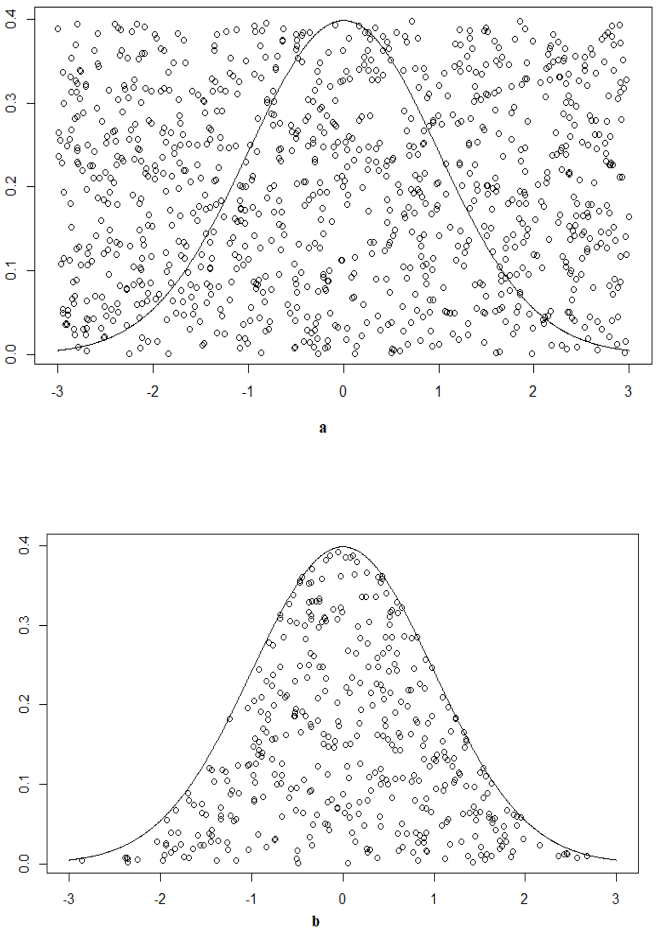


***Figure 1*: The generating points and accepted points for standard normal distribution from Algorithm 1.**

From the selected points, we draw histogram and Q_Q plot for the sample (Figure 2). The histogram from Figure 2a shows similar and close shape of the density function in (3.1). From the Q-Q plot in Figure 2b, we can see that the generated sample follows strictly from standard normal distribution.

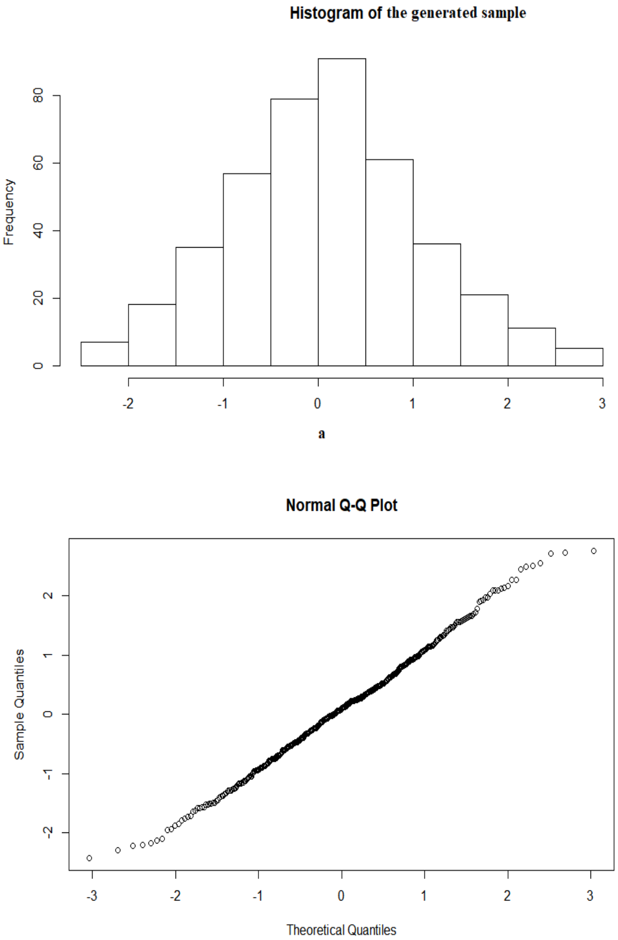
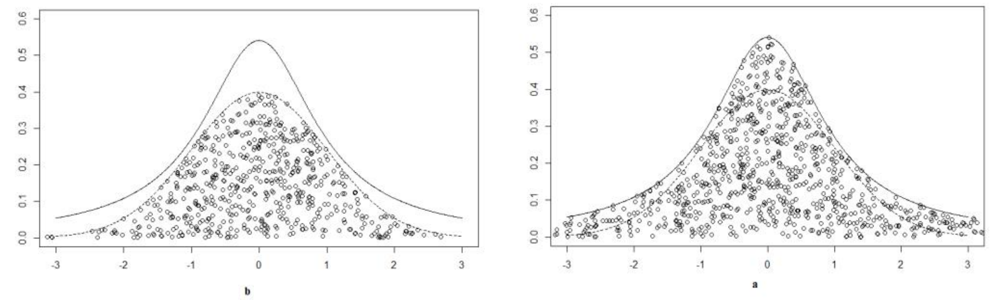


***Figure 2*: Histogram and Q_Q plot for the generated sample from Algorithm 1.**

Using an improved algorithm (Algorithm 2) for generating variation, we generate a sample for standard normal distribution. We generate 1000 random points for a Cauchy variable $Y \sim C(0,1)$ and 1000 random points for a uniform variable U having values from 0 to $M.g(y)$. The value for M can be chose so that $M.g(y) > f(x)$ on the domain of variable X according to Algorithm 2. The magnitude of this value can affect the acceptance rate, and the good acceptance rate is around 70% (Wakefield, J. ,2013). Here, we choose the value of $M = 1.7$ so that the acceptance rate is acceptable and better than the acceptance rate in Algorithm 1. The picture of the points for variable Y and normal density function are presented in Figure 3a.

After applying Algorithm 2, there are approximately 600 points that are accepted resulting in 60 of acceptance rate. The picture of accepted points for normal distribution is presented in Figure 3b. We can see that there are some points of the sample out of the interval [-3,3]. This describes the standard normal distribution more precisely. The histogram and Q_Q plot for the sample are drawn in Figure 4.



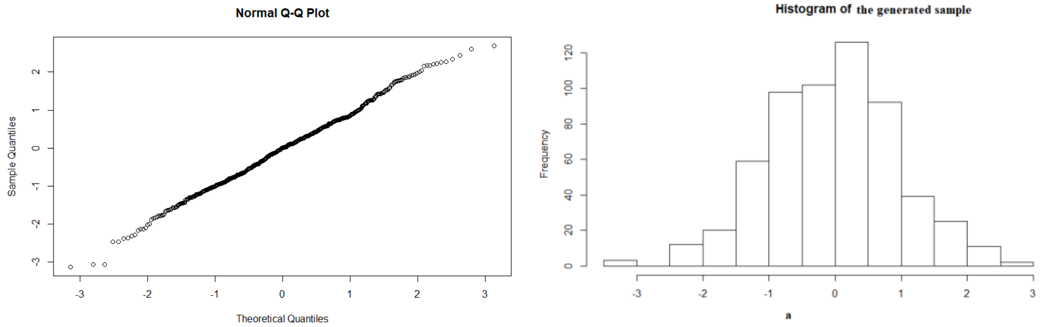


***Figure 3*: The generated points and accepted points for standard normal distribution from Algorithm 2**.

***Figure 4*: Histogram and Q_Q plot for the generated sample from Algorithm 2.**

We use the Shapiro-Wilk normality test for the accepted sample in Algorithm 1 and Algorithm 2. The results are presented in Table 1. The results show that for both algorithms, the generated samples follow the normal distribution with p-value <0.05. However, the sample generated from Algorithm 2 is slightly better.

Table 1: Shapiro-Wilk normality test for Algorithm 1 and Algorithm 2

	p-value for Algorithm 1	P value for Algorithm 2
Shapiro-Wilk normality test	<0.05	<0.01

3.2 Generating random numbers for non-standard distribution

In this section, we will use both Algorithm 1 and Algorithm 2 to generate a sample for a random variable which do not have known standard form density function as normal distribution, exponential …. A simulation study will be performed for the non-standard density function has the form of

$$f(x) = \begin{cases} 3x^2, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases} . \tag{3.2}$$

In this case, we will generate a sample for the chosen distribution using Algorithm 1. We generate 1000 random points for a uniform variable Y having values from 0 to 1 and 1000 random points for a uniform variable U having values from 0 to 3. The picture of the points for variable Y in Figure 5a.

After applying Algorithm 1, the acceptance rate is about 40%. The picture of accepted points for the distribution and histogram is presented in Figure 5b and 5c. We can see that the form of the histogram is close and similar to the form of the density function in (3.2).

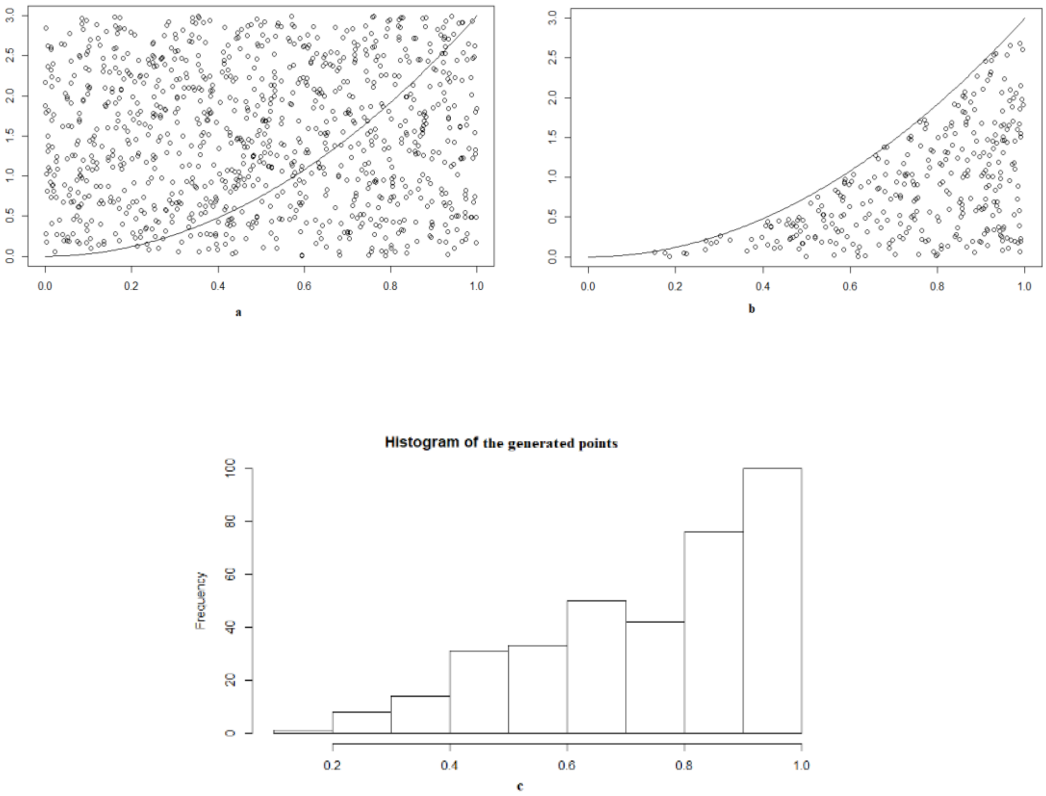


***Figure 5*: The generating points accepted points and histogram for the generated sample from Algorithm 1.**

We also generate a sample for the chosen distribution (1) using Algorithm 2. We generate 1000 random points for a variable Y having density function of x^2 and 1000 random points for a uniform variable U having values from 0 to $M.g(y)$, here we choose $M = 4$. After applying Algorithm 2, the acceptance rate is about 65%. The picture of accepted points for the distribution and histogram is presented in Figure 6. Here, the histogram in Figure 6c shows similar result as in Algorithm 1 and has close form to the density function in (3.2).

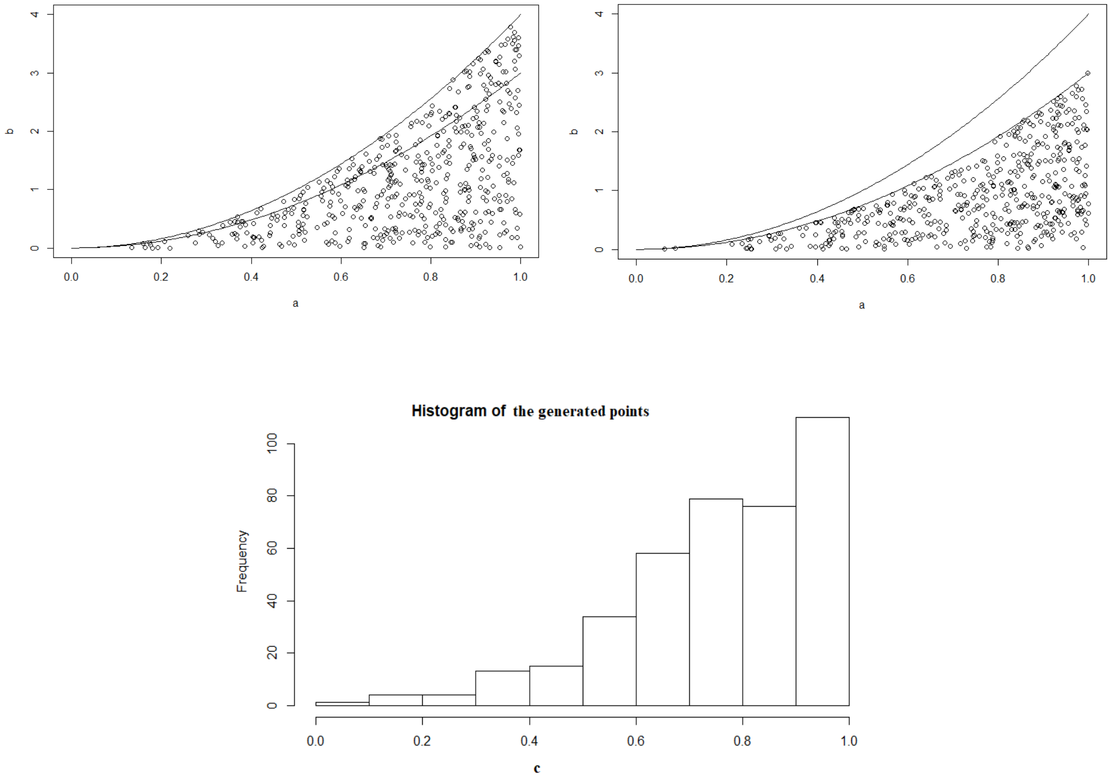


***Figure 6*: The generating points accepted points and histogram for the generated sample from Algorithm 2.**

4. CONCLUSION

Generating a sample for a random variable having a target distribution is crucial work in Bayesian statistics. From likelihood function and prior distribution, we can get a posterior distribution. From a posterior distribution, we generate a sample to make inferences for unknown parameters. In this paper, we propose two algorithms for generating a random variable with known density function. These algorithms can be applied for generating both standard and non-standard variables. The syntax and tactics for Algorithm 1 are easier than Algorithm 2. However, Algorithm 2 can solve the problem of infinite domain of a random variable which can be seen in normal simulation and improve the acceptance rate when we perform accepting points which can be seen in both simulation studies.

REFERENCES

Choi, H. M., & Hobert, J. P. (2013). The Polya-Gamma Gibbs sampler for Bayesian logistic regression is uniformly ergodic. *Electronic journal of Statistics, 7, 2054-2064.*

Cox, D. R., & Hinkley, D. V. (1979). *Theoretical Statistics*. New York: Chapman & Hall.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (1995). *Bayesian Data Analysis*.

London: Chapman & Hall.

Ghosh, J., Li, Y., & Mitra, R. (2018). On the use of Cauchy prior distribution for Bayesian logistic regression. *Bayesian Analysis 13(2)*, 359-383.

Hastings, W. K. (1970). Monte Carlo sampling methods using markov chains and their applications. *Biometrika, 57(1),* 97–109.

Marin, J., & Robert, C. P. (2014). *Bayesian essentials with R*. New York: Springer-Verlag.

Robert, C., & Casella, G. (2004). *Monte Carlo Statistical Methods*. New York: Springer-Verlag.

Wakefield, J. (2013). *Bayesian and Frequentist Regression Methods*. New York: Springer.