International Journal of Statistical Sciences *ISSN 1683-5603*
Vol. 21(2), 2021, pp 185-212

Application of Quantile Transform Method while Simulating Random Variates from Densities with Non-Closed Form of Cumulative Distribution Function

Anamul Haque Sajib*
Department of Statistics, Dhaka University, Dhaka-1000, Bangladesh

Syed Ehsan-Ar- Rafi
Ph.D. Research Fellow, Department of Mathematics and Statistics, Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA

Sabina Sharmin
Department of Statistics, Dhaka University, Dhaka-1000, Bangladesh

*Correspondence should be addressed to Anamul Haque Sajib
(Email: sajibstat@du.ac.bd)

[Received July 14, 2021; Revised August 20, 2021; Accepted September 7, 2021]

Abstract

Computer-generated random variates from different densities have been used in various simulation applications as stochastic inputs. The quantile transform method is one of the most common and efficient variates generation techniques, which requires the inverse of the cumulative distribution function (cdf) of a desired density. However, the application of the quantile transform method is limited to variates for which the cdf or inverse of the cdf is in simple closed form. This paper discusses the applicability of quantile transform method to generate variates for non-closed form of the cdf or inverse of the cdf and numerically solves the non-closed form of the cdf or inverse of the cdf. The application of the quantile transform method is then investigated through a simulation study where normal, gamma and non-central χ^2 densities are considered. From the simulation study, it is observed that generating variates from such densities via quantile transform method is not precluded because of the non-closed form of the cdf or inverse of the cdf. After comparing the quality of the sample produced and the computational cost of this study with other standard methods used to generate variates, it is concluded that the quantile transform method is equally efficient as other standard methods

Keywords: Quantile transform method, non-closed form of the cdf, direct method of data generation.

AMS Subject Classification: 62P25.

1. Introduction

Conducting an empirical study of a random variable $X \sim f(x)$ requires sample generation from its probability density function (pdf) $f(x)$. Many methods are available in the literature to generate sample from $f(x)$. Most importantly, quantile transform (direct), rejection sampling (indirect), Gaussian mixture and Markov Chain Monte Carlo methods are commonly used to generate sample from $f(x)$. However, the choice of data generation technique for a particular pdf $f(x)$ is contextual, i.e. the choice of data generation technique for a particular pdf $f(x)$ depends on: (i) whether its distribution function $F(x)$ is in simple closed form or not (ii) computing time and implementation simplicity of the chosen algorithm (iii) the dimension of the desired density. When the cdf of the desired density has a closed form, the quantile transform method is usually used with high efficiency. Having a non-closed form of the cdf renders us to use other alternative methods such as rejection sampling and Gaussian mixture. Apart from the Markov Chain Monte Carlo method, all the methods mentioned here are used for one dimensional probability densities. Generalizing these methods for high dimensional probability densities is quite challenging. The Markov Chain Monte Carlo method is commonly used for simulating from the high dimensional probability densities. However, we are interested in generating only from the one dimensional probability densities, and this is why the Markov Chain Monte Carlo method is not covered in this paper.

In an article, Tadikamalla et al. (1981) stated that the quantile transform method becomes inefficient due to numerical solution of non-closed form of cdf. Furthermore, Casella et al. (2002) asserted that, in practice, quantile transform method could be prohibitively long and complicated due to non-closed form of cdf and they recommended to explore other methods in such situation.

Ghitany et al. (2008) also asserted that the quantile transform method fails to generate variates from Lindley density as the inverse transform of Lindley cdf has non-closed form. Horger et al. (2018) argued that solving non-closed form of cdf or non-closed form of inverse transform of cdf numerically requires high computational effort and the quality of the sample decreases.

Unfortunately, we have not found any rigorous numerical study in the literature in favor of assertions made by Tadikamalla et al. (1981), Casella et al. (2002), Ghitany et al. (2008)}, and Horger et al. (2018).

On the contrary, Okwuokenye et al. (2016) showed in their study that generating Lindley variates by numerically solving the inverse transform of its non-closed form of cdf is not precluded. Furthermore, they generalized their findings to any probability density functions whose cdf have non-closed form i.e. solving non-closed form of cdf numerically does not preclude variate generation from their pdfs under quantile transform method and it is equally efficient as other standard methods as far as the quality of the sample is concerned.

To the best of our knowledge, there is no numerical study to date that justified the generalized assertion made by Okwuokenye et al. (2016). This paper demonstrates an extensive numerical study regarding the efficiency of the quantile transform method for generating variates from the densities with non-closed form of the cdf or inverse of the cdf by using the current state of the art computing facilities.

We organize the rest of the paper as follows: Section 1.1 discusses the algorithm of quantile transform method while some important terminologies used in this paper are discussed in section 1.2. In section 2, we show how the quantile transform method generates variates from the normal, gamma, and non-central χ^2 densities. Simulation results and discussions are presented in section 3 which are followed by conclusion and future work presented in section 4.

1.1. Quantile Transform Method

Let $f(x)$ be the pdf of a continuous random variable X. Then cdf of this random variable X can be defined as

$$F_X(x) = \Pr(X \leq x) = \int_{-\infty}^{x} f(t)dt,$$

where $F_X(x)$ is assumed as a strictly increasing or non-decreasing distribution function. The function $F_X^{-1}(x)$ denotes the inverse of $F_X(x)$ which can be defined as

$$y = F_X(x) \Leftrightarrow x = F_X^{-1}(y)$$

The inverse function defined above, F_X^{-1}, is valid only when F_X is strictly increasing distribution function. The F_X which is not strictly increasing distribution function i.e. the distribution function is constant on some interval the above equation needs to be defined as follows Casella et al. (2002)

$$F_X^{-1}(y) = \inf\{x: F_X(x) \leq y\}$$

Theorem 1. Let $F_X(x)$ be the cdf of a continuous random variable X and define another random variable $Y = F_X(x)$. Then Y follows uniform (0,1) i.e. $F_Y(y) = \Pr(Y \leq y) = y, 0 < y < 1$.

The proof of theorem 1 is available in the textbook written by Casella et al. (2002). Theorem 1 can be used to generate random variate X from a population with cdf $F_X(x)$. It works in a two steps procedures to generate random variate X from a population with cdf F_X: (i) draw $U \sim uniform(0,1)$ (ii) return X by solving the equation $F_X(x) = u$.

Example 1 (Generating Exponential Variate). Suppose we wish to generate random variate X from exponential density, $f(x;\ \lambda) = \lambda\, e^{-\lambda x}; x, \lambda \geq 0$, using quantile transform method. For exponential density the form of the cdf $F_X(x)$ is $1 - e^{-\lambda x}$ which is in simple closed form. Solving the equation $1 - e^{-\lambda x} = u$ yields $x = -\frac{1}{\lambda}\ln(1-u) = F_X^{-1}(u)$ which is an explicit form. Quantile transform method which is commonly used to generate exponential variate as F_X^{-1} for exponential density has an explicit form.

Example 2 (Generating Lindley Variate). To analyze failure time data Lindley density is used which was introduced by Lindley (1958). The Lindley distribution can be written as a mixture of exponential and gamma distributions which belongs to an exponential family. It represents a good alternative to the exponential failure time distributions that suffer from not exhibiting uni-modal and bathtub shaped failure rates. The probability density function of Lindley variate X (survival time) is defined as

$$f(x;\theta) = \frac{\theta^2}{1+\theta}(1+x)e^{-\theta x}, \qquad x, \theta > 0,$$

where θ is the scale parameter. The corresponding cdf for Lindley variate is

$$F(x;\theta) = 1 - \frac{1+\theta+\theta x}{1+\theta}e^{-\theta x},,\ x, \theta > 0,$$

which is in simple closed form. However, the solution of the equation $1 - \frac{1+\theta+\theta x}{1+\theta}e^{-\theta x} = u$ yields non explicit form of F_X^{-1} i.e. x can not be expressed explicitly in terms of u and θ. The quantile transform method is not considered to simulate random variate in such cases (as F_X^{-1} has no explicit form). The

composition method proposed by Ghitany et al. (2008) can be used to generate Lindley variate.

Example 3 (Generating Normal Variate). The normal or Gaussian density is one of the most widely used probability density functions in Statistics. The probability density function of normal random variate X is defined as

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}, -\infty < x < \infty,$$

where $-\infty < x < \infty$ and $\sigma^2 > 0$ are the population mean and variance respectively. The corresponding cdf for normal random variate is

$$F(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{x} e^{-\frac{1}{2\sigma^2}(y-\mu)^2}\, dy,$$

which is not in simple closed form as it requires to evaluate intractable integral. As a consequence, the solution of the equation $F_X(x) = u$ yields non explicit form of F_X^{-1} i.e. X can not be evaluated explicitly. As the cdf $F_X(x)$ for normal density involves intractable integral, hence $F_X^{-1}$is also intractable. Therefore, instead of using the quantile transform method, another efficient method such as Box-Muller transformation is used to simulate normal random variate. The form of the cdfs for gamma and chi-square densities also have intractable integral (gamma first kind) like the cdf of normal density. Therefore, the quantile transform method cannot be considered to generate gamma and chi-square variates in the literature.

In this paper, we mainly show how random variate X can be generated via the quantile transform method when $F_X(x)$ or F_X^{-1} has no simple closed-form (may involve intractable integral) discussed in Examples 2-3. After generating sample from the distribution function $F_X(x)$, we investigated the quality of the generated sample and required computational efforts to fill up the current lacks in the existing literature regarding this phenomenon. In the next subsection, some important terminologies are discussed that will be used throughout this paper.

1.2. Some Related Terminologies

This section discusses some important terminologies such as average bias, average mean square error, coverage probability and Wald confidence interval which will be used to assess the quality of a generated sample. Suppose we have a probability density function $f(x;\Theta)$ which is parameterized by unknown constant Θ and let $\widehat{\Theta}$

be an estimator of Θ which is a function of observed sample. Then average mean square error, coverage probability and Wald confidence interval are defined as follows:

Average Mean Square Error: Average mean square error (MSE) of an estimator $\widehat{\Theta}$ is defined as

$$MSE(\widehat{\Theta}) = \frac{1}{k}\sum_{i=1}^{k}(\widehat{\Theta}_i - \Theta)^2$$

Usually, two or more estimators can be compared using their MSEs. The smaller the MSE value, the better the estimator. In this paper, MSE of an estimator will be calculated based on two samples of the same size generated through two different data generation techniques to evaluate the quality of the samples produced by them.

Wald-Based Confidence Interval:

The Wald based 95% confidence interval (CI) is intrinsically related to Wald test. To assess the quality of the samples, generated through two different data generation techniques, we use Wald based 95% confidence interval but in a slightly different way. From the Wald-based confidence interval the average width of the confidence interval and coverage probability are defined respectively as follows

Average width $= \frac{1}{k}\sum_{i=1}^{k}(\widehat{U}_i - \widehat{L}_i)^2$ and Coverage probability $= \frac{1}{k}\sum_{i=1}^{k} I_{(0,1)}$,

where $\widehat{U}_i$ and $\widehat{L}_i$ are the upper and lower limits of Wald based 95% confidence interval and $I_{(0,1)}$ is an indicator function which takes value one when Θ is contained in the confidence interval otherwise it takes zero. Coverage probability calculated from $100\,(1-\alpha)\%$ CI is expected to be $(1-\alpha)$ while average width of CI is expected to be as small as possible for a sample generated through an efficient method.

Ljung-Box Test: Ljung-Box test, jointly developed by Ljung and Box (1978)}, is widely used to test the randomness of a time series in econometric and other applications of time series analysis. According to them, the algorithm of Ljung-Box test is: (i) H_0: the data are independently distributed against H_a: the data possess some serial correlation up to a certain lag h, (ii) The quantity $Q =$

$n(n+2)\sum_{k=1}^{h}[(n-k)^{-1}r_k^2]$ which is a function of sample auto-correlation r at lag k and sample size n, denotes the test statistic, (iii) $Q \sim \chi^2_{(h)}$ under H_0 and reject the null hypothesis if $Q > \chi^2_{(1-\alpha,\ h)}$ where $\chi^2_{(1-\alpha,\ h)}$ is the $(1-\alpha)$ quantile of the χ^2 distribution with h degrees of freedom.

2. Quantile Transform Method for Non-Explicit F_X^{-1}

This section discusses how the quantile transform method generates sample from the distribution $F_X(x)$ whose F_X^{-1} has no explicit form. The normal, gamma and non-central chi-square densities are considered here to illustrate how the quantile transform method works when F_X^{-1} no explicit form has.

2.1. Generating Normal Variate via Quantile Transform Method

In the subsection 1.1, we have presented the form of cdf of normal density which has non closed form theoretical expression (intractable integral). Recall that the cdf of normal density

$$F(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{x} e^{-\frac{1}{2\sigma^2}(y-\mu)^2}\,dy,$$

This cdf needs to be approximated as it involves intractable integral, and approximation can be made via three different approaches: (i) construction of approximation formulas (ii) using distribution functions which are close to normal distribution (iii) construction of bounds. There is no concrete study regarding the most efficient approach among these three approaches, and determining the most efficient approach is beyond the scope of this study. In this study, we consider a simplistic approximation formula proposed by Choudhury (2014) to approximate the cdf of normal density which is

$$\phi(x) \approx 1 - \frac{1}{\sqrt{2\pi}}\frac{e^{-x^2}}{0.226 + 0.64x + 0.33\sqrt{x^2+3}}\ , x > 0.$$

The right hand side of equation 8 approximates the cdf of normal density for $x > 0$, and it has a minimum accuracy of three digits shown in their study. The inverse transform method requires to solve the following equation to generate standard normal variate

$$1-\frac{1}{\sqrt{2\pi}}\frac{e^{-x^2}}{0.226+0.64x+0.33\sqrt{x^2+3}}=u,\qquad x>0,$$

where $u\sim Uniform\,(0.5,1)$. Equation 9 has no explicit solution for x. The rootSolve package in R which finds the root of a non-linear equation using Newton-Raphson method was used to solve equation 9. Solution of equation 9 only produces $|z|$ i.e. positive standard normal variate. However, using symmetry property of standard normal variate we can have z from $|z|$ by simply setting $z=-|z|$ when $u<0.5$ ootherwise $z=-|z|$ where $u\sim Uniform\,(0,1)$. The transformation $x=\mu+\sigma x$ is required to generate $X\sim N(\mu,\sigma^2)$ from z.

2.2. Generating Gamma Variate via Inverse Transform Method

Suppose the random variable $X\sim gamma(\alpha,\beta)$, where $\alpha>0$ and $\beta>0$ are the shape and rate parameters, respectively. Then the cdf of X is defined as

$$F(x;\,\alpha,\beta)=\int_0^x\frac{\beta^\alpha y^{\alpha-1}e^{-\beta y}}{\Gamma(\alpha)}dy=\frac{\gamma(\alpha,\beta x)}{\Gamma(\alpha)},$$

where $\gamma(\alpha,\beta x)$ and $\Gamma(\alpha)$ are the lower incomplete gamma and the gamma functions respectively. The cdf $F(x;\,\alpha,\beta)$ defined in equation 10 has no explicit form as it involves lower incomplete gamma function, and hence quantile transform method was not considered to generate gamma variate in the literature. For the gamma family, quantile transform method can only be applied to generate exponential variate for which F_X^{-1} has explicit closed form. There are several efficient methods available in the literature to generate gamma variate. However, the purpose of this paper is to investigate the performance of quantile transform method compared to other available methods used in the literature regarding gamma variate generation. For the shape parameter$\alpha=k$, where $k$is an integer, Papoulis et al. (1965) suggested an approximation formula for equation 10 which is

$$F(x;\,\alpha,\beta)\approx 1-\sum_{i=0}^{\alpha-1}\frac{\beta x^i}{i!}e^{-\beta x},x>0.$$

Gamma density becomes Erlang density when α is an integer, and the cdf for Erlang density is also intractable due to the lower incomplete gamma function. However, like gamma variate there is a simple generation technique (variable transformation method) available in the literature on Erlang variate. To generate

gamma variate when α an integer (Erlang variate) is using quantile transform method the following equation needs to be solved for x

$$1 - \sum_{i=0}^{\alpha-1} \frac{\beta x^i}{i!} e^{-\beta x} = u, \quad x > 0,$$

where $u \sim Uniform(0,1)$. Like equation 9, the above equation does not have an explicit solution for x. Therefore, numerical technique such as the Newton-Raphson method is required to solve for x, and we use rootSolve package in R like earlier to solve it. To the best of our knowledge, there is no approximation formula for gamma cdf in the literature when α is not an integer. Therefore, this paper does not cover data generation from gamma density via quantile transform method when α is not an integer.

2.3. Non-central χ^2 Variate via Quantile Transform Method

Let $Y_1, Y_2, \cdots, Y_k$ be k independent normally distributed random variates with mean $\mu_i$and unit variances. Then the random variable $X = \sum_{i=1}^{k} Y_i^2$ follows non-central χ^2 density with parameters k and $\lambda = \sum_{i=1}^{k} \mu_i^2$, which are the degrees of freedom and non-centrality parameter, respectively. Non-central χ^2 square density becomes central χ^2 density when $\lambda = 0$. Recall that the cdf of non-central χ^2 which is

$$F(x; k, \lambda) = e^{-\lambda/2} \sum_{j=0}^{\infty} \frac{(\lambda/2)^j}{j!} Q(x; k + 2j), \quad x, \lambda, k > 0,$$

where $Q(x; k + 2j)$ is the cdf of the central χ^2 density with k degrees of freedom which can be defined as

$$Q(x; k + 2j) = \frac{\gamma((k + 2j)/2, x/2)}{\Gamma(k + 2j)/2}, \qquad x, k > 0,$$

where $\gamma((k + 2j)/2, x/2)$ is the lower incomplete gamma function. There are a number of approximations available in the literature to approximate the cdf of non-central χ^2 denoted in equation 13. Abdel-Aty (1954) proposed a normal approximation to the non-central χ^2 distribution which is defined as

$$F(x; k, \lambda) \approx \phi\big(z(x, k, \lambda)\big),$$

where $F(x; k, \lambda)$ and $\phi(z)$ are the distribution functions of non-central χ^2 and standard normal variates, respectively and

$$z(x,k,\lambda)=\frac{\left(\frac{x}{k+\lambda}\right)^{1/3}-\left(1-\frac{2(k+2\lambda)}{9(k+\lambda)^2}\right)}{\sqrt{\frac{2(k+\lambda)}{9(k+\lambda)^2}}}.$$

To approximate the $p, 0 \le p \le 1$, quantile of x amounts to solving $\phi\big(z(x,k,\lambda)\big)=p$ which yields

$$x=F_X^{-1}(p,k,\lambda)\approx(k+\lambda)\left\{1-\frac{2(k+2\lambda)}{9(k+\lambda)^2}+\phi^{-1}(p)\sqrt{\frac{2(k+\lambda)}{9(k+\lambda)^2}}\right\}^3,$$

where $\phi^{-1}(p)$ is the quantile function (inverse of the cdf) of standard normal density that can be determined by the equation 9. This approximation is also known as Abdel-Aty (1954) first approximation where $Y=\left(\frac{X}{k+\lambda}\right)^h, h=\frac{1}{3}$, is considered to be normally distributed with mean about to $1-\frac{2(k+\lambda)}{9(k+\lambda)^2}$ variance $\sqrt{\frac{2(k+\lambda)}{9(k+\lambda)^2}}$. Munuswamy (1959) had modified Abdel-Aty (1954) approximation formula by considering $Y'=\left(\frac{X}{k+\lambda}\right)^h$ where h was not considered to be fixed to $\frac{1}{3}$. The value of h was determined in such way that the third cummulant of Y'vanishes. The objective of this modification was to make the distribution of Y' more normal than that of Y. Sankaran (1963) also proposed another variant of Y'which was defined as $Y''=\left(\frac{X-b}{k+\lambda}\right)^{1/2}$. The value of b was determined such that the leading term in the expansion for one of the cumulants of Y'' vanishes. From the findings of Munuswamy (1963) study, it was evident that there is no approximation which is the best for all situations (different combination of k and λ values), and every approximation method has an accuracy issue. In other words, it is not certain that approximation which is the best at one percentage point will also be best at another percentage point. Furthermore, it was also stated that approximation proposed by Abdel-Aty (1954) is the easiest to apply (does not require to calculate cummulants) although generally it is less accurate than that of approximation proposed by Munuswamy (1963). For the sake of simplicity, we choose Abdel-Aty (1954) approximation formula to approximate the cdf of non-central χ^2 in this study.

3. Results and Discussion

In this section, we evaluate the performance of the quantile transform method for generating random variates from the densities with non closed-form of cdf through simulation study. In addition to this, we also compare the performance of the quantile transform method with the method used as an alternative of it to generate random variates from such densities, respectively. More specifically, we compare quality of the samples generated by the quantile transform method and its alternative method, building block to form different R functions such as rnorm, rgamma, and rchisq to generate random variates from such densities. For notational convenience, we use the notation q_rnorm, q_rgamma and q_rchisq in this paper to represent the quantile transform method which is used to generate random variates from the densities with non-closed form of cdf or inverse of it.

Table 1: Sample RMSE* of estimates $\hat{\mu}$ and $\hat{\sigma}^2$ based on 1000 simulated data sets.

$[\mu, \sigma^2]$	Method	Sample size (n)				
		20	40	80	160	360
[0,1]	q_rnorm	0.4000	0.2742	0.1927	0.1365	0.0948
	rnorm	0.3787	0.2813	0.1927	0.1373	0.0959

*Root mean square error (RMSE)= $\sqrt{\frac{1}{1000}\sum_{i=1}^{1000}\left[(\hat{\mu}_i-\mu)^2+\left(\hat{\sigma}_i^2-\sigma^2\right)^2\right]}$

Table 1 represents the average root mean square error (RMSE) of $\hat{\mu}$ and $\hat{\sigma}^2$ both for q_rnorm and rnorm methods. The values of RMSE under both methods are very close to each other, and both of these decrease as the sample size increases. The estimates from the q_rnorm method and rnorm method are essentially the same when sample size is $n \geq 80$. The estimated $(1-\delta)\times 100\% = 95\%$coverage probability of $\hat{\mu}$ and $\hat{\sigma}^2$under both methods are presented in Table 2. The average estimated coverage probability values of $\hat{\mu}$ and $\hat{\sigma}^2$ under both methods are very close to target 95% value. Based on the simulation results presented in Tables 1 and 2, we can conclude that quality of the samples generated using q_rnorm method and rnorm are almost the same.

Table 2: Average coverage probability of $\hat{\mu}$ and $\hat{\sigma}^2$ (%) based on 1000 simulated data sets.

Parameter	Method	Sample size (n)				
		20	40	80	160	360
$\mu = 0$	q_rnorm	95.1	94.7	94.9	94.8	94.8
	rnorm	94.7	94.9	94.3	95.0	95.0
	q_rnorm	94.8	95.7	94.5	94.4	95.0
	rnorm	95.5	93.9	95.4	94.7	96.0

Table 3 presents the average estimates of type 1 error $\hat{\delta}$ rates which is obtained by comparing the two empirical data sets generated using the q_rnorm method and rnorm methods, respectively. For all the sample sizes estimated $\hat{\delta}$ values are less than 0.05 which means two empirical data sets generated through q_rnorm method and rnorm methods are not different. As computing time is one of the core components to measure the efficiency of any method, CPU time to generate sample of different sizes for q_rnorm method and rnorm is computed using the system.time command in R which is presented in Table 4. From Table 4, we see that irrespective of all sample sizes q_rnorm method requires higher computing time compared to rnorm method. For example, to generate two hundred thousand observations q_rnorm requires 28.12 seconds while it is less than one second for rnorm method. However, required time in q_rnorm is not too big to think about it.

Finally, to test the randomness of the generated samples produced by q_rnorm method we have used both the ACF (auto-correlation function) plot and the Ljung-Box test. The first two rows of Figure 1 show the ACF plots of generated samples produced by q_rnorm method for different sample sizes $n=20, 200, 2000,$ and 20000. From the ACF plot produced for n=20, it is observed that lag at 3 is beyond the 95% significance band. Similarly, for the ACF plots for n=200, 2000 and 20000, lags at 12 and 20, lags at 20, 37, 87 and 95 and lags at 4, 30, 31, 49, 54, 63 and 92 are beyond the 95% significance band, respectively. However, it does not confirm the presence of auto-correlation and it could happen because of sampling error. The Ljung-Box test is carried out to confirm the presence of auto-

correlation, and the P values for all the tests are greater than 0.05 which means that the sample evidences fail to reject the null hypothesis (there is no auto-correlation in the generated sample). The P values for Ljung-Box test is not shown here but are available in the R codes. All the results are reproducible as a specific seed number is used to generate the samples.

Table 3: Estimated level of significance $\hat{\delta}$ for comparing empirical distribution functions based on 1000 simulated data sets.

$[\mu, \sigma^2]$	δ	Sample size (n)				
		20	40	80	160	360
[0,1]	$\hat{\delta}$	2.4	2.4	3.0	3.9	3.8

$\hat{\delta}$ values are reported as in percent (%) where represents the proportion of times the null hypothesis is rejected out of 1000 replicates. The null hypothesis is $H_0: F_1 = F_2$, where F_1 and F_2 be the empirical distribution functions of data simulated using the q_rnorm and the rnorm methods respectively.

Table 4: Average computing time in seconds based on 20 simulated data sets.

$[\mu, \sigma^2]$	Method	Sample size (n)				
		20	40	80	160	360
[0,1]	q_rnorm	0.00163	0.01639	0.17767	3.8780	28.172
	rnorm	0.00001	0.00002	0.00013	0.003	0.0181

The last two rows of Figure 1 show the overlying of standard normal distribution on the sample histograms. From these plots, it is clear that fitted line for the standard normal distribution appears to follow the histogram bars adequately for all sample sizes, and it is getting more accurate as the sample size increases.

Tables 5-8 and Tables 9-12, placed in the supplementary materials in section 4.1, represent the estimated average RMSE, coverage probability, rate of level of significance and required computing time for q_rgamma versus rgamma and q_rchisq versus rchisq, respectively. Very similar patterns of estimated average RMSE, coverage probability, rate of level of significance and required computing

time observed in Tables 1-4 for q_rnorm versus rnorm can also be seen in Tables 5-8 and Tables 9-12. However, estimated coverage.

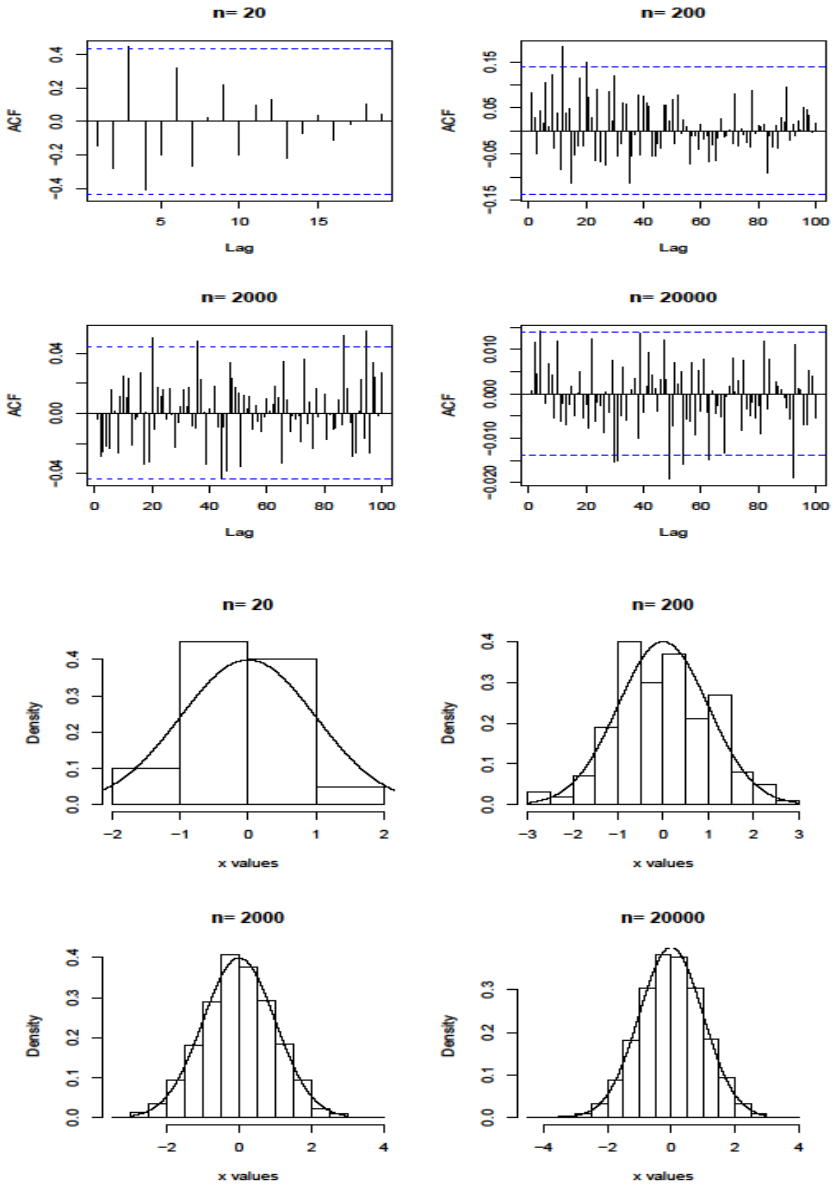

Figure 1: ACF and overlying the true pdf of standard normal distribution on sample histogram plots for different sample sizes. Samples are generated using the q_rnorm method.

probability both for q_rgamma and rgamma for n=20 does not reach to targeted 95% coverage probability.

Like the first two rows of Figure 1, the first two rows of both Figures 2 and 3 are also used to check the randomness of gamma and χ^2variates generated through q_rgamma and q_rchisq, respectively. The Ljung-Box test confirms that both the gamma and χ^2 variates generated through q_rgamma and q_rchisq, respectively are free from auto-correlation although some of the lags at different orders for different sample sizes are beyond the 95% confidence band.

4. Conclusion

From the results of the simulation study, it is observed that the quality of the variates generated from the densities with non-closed from of cdf by numerically solving its cdf or inverse of its cdf via quantile transform method are equally good (statistical properties) as the variates produced by other standard methods used in the current literature. From the simulation study, we concluded that the existing assertions regarding the generation from the density with non-closed from of cdf via quantile transform method made by Tadikamalla et al. (1981), Casella et al. (2002), Ghitany et al. (2008), and Horger et al. (2018) are not supported by the results of the simulation study but the results of the simulation study supported the assertion made by Okwuokenye et al. (2016). However, generating variates from the densities with non-closed form of cdf via quantile transform method requires higher computing time compared to alternative methods although required time is insignificant. The success of the quantile transform method in the case of non-closed from of cdf depends on the availability and difficulty level of the cdf approximation formulas. The cdf approximation formulas need to be chosen efficiently to get the quality samples from a desired density.

Reference

[1] Abdel-Aty, S. H. (1954). Approximate formulae for the percentage points and the probability integral of the non-central χ^2 distribution. Biometrika, 41(3/4), pp. 538-540.

[2] Casella, G. and Berger, R. L. (2002). Statistical inference. Cengage Learning. Chapter 2, page 53.

[3] Choudhury, A. (2014). A simple approximation to the area under standard normal curve. Mathematics and Statistics, 2(3), pp. 147-149.

[4] Ghitany, M. E., Atieh, B. and Nadarajah, S. (2008). Lindley distribution and its application. Mathematics and computers in simulation, 78(4), pp. 493-506.

[5] Horger, F., Würfl, T., Christlein, V. and Maier, A. (2018). Deep learning for sampling from arbitrary probability distributions. arXiv preprint arXiv: 1801.04211.

[6] Lindley, D. V. (1958). Fiducial distributions and Bayes' theorem. Journal of the Royal Statistical Society: Series B (Methodological), 20(1), pp. 102-107.

[7] Ljung, G. M. and Box, G. E. (1978). On a measure of lack of fit in time series models. Biometrika, 65(2), pp. 297-303.

[8] Okwuokenye, M. and Peace, K. E. (2016). A comparison of inverse transform and composition methods of data simulation from the Lindley distribution. Communications for Statistical Applications and Methods, 23(6), pp. 517-529.

[9] Papoulis, A. and Pillai, S. U. (1965). Probability Random Variables. Stochastic Proc.

[10] Munuswamy, S. (1959). On the non-central χ^2 distribution. Biometrika, 46, 235-7.

[11] Munuswamy, S. (1963). Approximations to the non-central chi-squared distribution Biometrika, 50(1-2), 199–204.

[12] Tadikamalla, P. R. and Johnson, M. E. (1981). A complete guide to gamma variate generation. American Journal of Mathematical and Management Sciences, 1(3), pp.213-236.

Supplementary Materials

Table 5: Sample RMSE* of estimates $\hat{\alpha}$ and $\hat{\beta}$ based on 1000 simulated data sets.

$[\alpha, \beta]$	Method	Sample size (n)				
		20	40	80	160	360
$[3, 8.5]$	q_rgamma	4.46	2.77	1.89	1.29	0.91
	rgamma	4.54	2.73	1.86	1.23	0.87
[6, 4.5]	q_rgamma	4.71	2.89	1.97	1.34	0.68
	rgamma	4.87	2.82	1.98	1.29	0.66
[12, 2.5]	q_rgamma	5.51	3.37	2.27	1.56	1.05
	rgamma	5.67	3.38	2.35	1.51	1.03
[24, 1.5]	q_rgamma	10.24	6.24	4.17	2.87	2.02
	rgamma	10.37	6.30	4.37	2.79	1.95

Table 6: Average coverage probability of $\hat{\alpha}$ and $\hat{\beta}$ (%) based on 1000 simulated data sets.

Parameter	Method	Sample size (n)				
		20	40	80	160	360
$\alpha = 3$	q_rgamma	91.9	93.1	94.3	94.9	95.7
	rgamma	92.6	93.7	94.1	95.0	94.4
$\beta = 8.5$	q_rgamma	92.0	93.9	95.0	94.7	95.3
	rgamma	92.8	93.4	94.2	95.2	95.3
$\alpha = 6$	q_rgamma	92.5	93.5	94.3	94.6	95.6
	rgamma	93.5	94.6	94.3	95.6	94.7
$\beta = 4.5$	q_rgamma	92.6	94	94.8	94.6	95.4
	rgamma	93.4	93.1	93.8	95.5	94.6
$\alpha = 12$	q_rgamma	92.3	94.1	94.9	95	95.7
	rgamma	93.5	94.7	93.8	95.5	94.2
$\beta = 2.5$	q_rgamma	92.2	94.3	94.9	94.5	95.6
	rgamma	93.5	94.3	93.5	95.1	94.1
$\alpha = 24$	q_rgamma	92.9	94.2	94.7	94.9	95.8
	rgamma	93.8	94.1	94	94.9	94.3
$\beta = 1.5$	q_rgamma	92.2	94.4	95.3	94.6	95.7
	rgamma	93.4	94.4	94.2	95	94.4

Table 7: Estimated level of significance $\hat{\delta}$ for comparing empirical distribution functions based on 1000 simulated data sets.

$[\alpha, \beta]$	δ	Sample size (n)				
		20	40	80	160	360
[3, 8.5]	$\hat{\delta}$	2.3	3.1	2.7	4.6	3.6
[6, 4.5]	$\hat{\delta}$	2.5	3.0	3.0	3.5	4.3
[12, 2.5]	$\hat{\delta}$	2.8	3.0	3.2	3.6	3.7
[24, 1.5]	$\hat{\delta}$	2.6	3.2	3.3	3.6	3.8

$\hat{\delta}$ values are reported as in percent (%) where $\hat{\alpha}$ represents the proportion of times the null hypothesis is rejected out of 1000 replicates. The null hypothesis $H_0: F_1 = F_2$, where F_1 and F_2 be the empirical distribution functions of data simulated using the $Q\_\texttt{rgamma}$ and the `rgamma` methods respectively.

Table 8: Average computing time in seconds based on 20 simulated data sets.

$[\alpha, \beta^2]$	Method	Sample size (n)				
		20	200	2000	20000	200000
[3,8.5]	q_gamma	0.003	0.02750	0.23350	1.78350	35.2805
	rgamma	0.00002	0.00003	0.00028	0.003	0.0040

Table 9: Sample RMSE* of estimates $\hat{k}$ and $\hat{\lambda}$ based on 1000 simulated data sets.

$[k, \lambda]$	Method	Sample size (n)				
		20	40	80	160	320
$[3, 20]$	q_rchisq	21.39	14.91	10.87	7.78	5.57
	rchisq	19.87	14.48	10.16	7.13	5.05
[12, 10]	q_rchisq	15.44	10.77	7.79	5.58	4.00
	rchisq	15.16	10.77	7.53	5.45	3.85
[15, 5]	q_rchisq	11.94	8.33	6.00	4.31	3.09
	rchisq	11.75	8.37	5.99	4.24	3.02

Table 10: Average coverage probability of $\hat{k}$ and $\hat{\lambda}$ (%) based on 1000 simulated data sets.

Parameter	Method	Sample size (n)				
		20	40	80	160	320
$k = 3$	q_rchisq	96.1	95.6	95.3	95.1	94.8
	rchisq	95.5	95.2	95	95.3	95.5
$\lambda = 20$	q_rchisq	96.6	95.5	95.4	95.1	95.1
	rchisq	95.4	95.3	95.1	95.6	95.8
$k = 12$	q_rchisq	96.2	95.5	95.1	95.3	95
	rchisq	96.6	95.4	95.2	95.1	95.4
$\lambda = 10$	q_rchisq	96.3	95.6	95.2	95.2	94.7
	rchisq	96.6	96	95.3	95.5	95.4
$k = 15$	q_rchisq	95.9	95.3	94.9	95.6	95.1
	rchisq	96.1	95.9	95.5	95.1	94.9
$\lambda = 5$	q_rchisq	96.4	95.6	95.2	95.4	94.8
	rchisq	95.6	96.1	95.5	95.2	95.1

Table 11: Estimated level of significance (δ) for comparing empirical distribution functions based on 1000 simulated data sets

$[k, \lambda]$	δ	Sample size (n)				
		20	40	80	160	320
[3, 20]	$\hat{\delta}$	2.8	2.4	3.8	5.0	5.0
[12,10]	$\hat{\delta}$	4.2	2.3	3.3	4.0	4.0
[15,5]	$\hat{\delta}$	3.6	2.7	3.2	3.6	4.6

$\hat{\alpha}$ values are reported as in percent (%) where $\hat{\alpha}$ represents the proportion of times the null hypothesis is rejected out of 1000 replicates. The null hypothesis $H_0: F_1 = F_2$, where F_1 and F_2 be the empirical distribution functions of data simulated using the q_rchisq and the rchisq methods respectively.

Table 12: Average computing time in second based on 20 simulated data sets

$[k, \lambda]$	Method	Sample size (n)				
		20	200	2000	20000	200000
[3,20]	q_rchisq	0.00005	0.001	0.005	0.0430	0.3350
	rchisq	0.00001	0.001	0.0005	0.007	0.0595

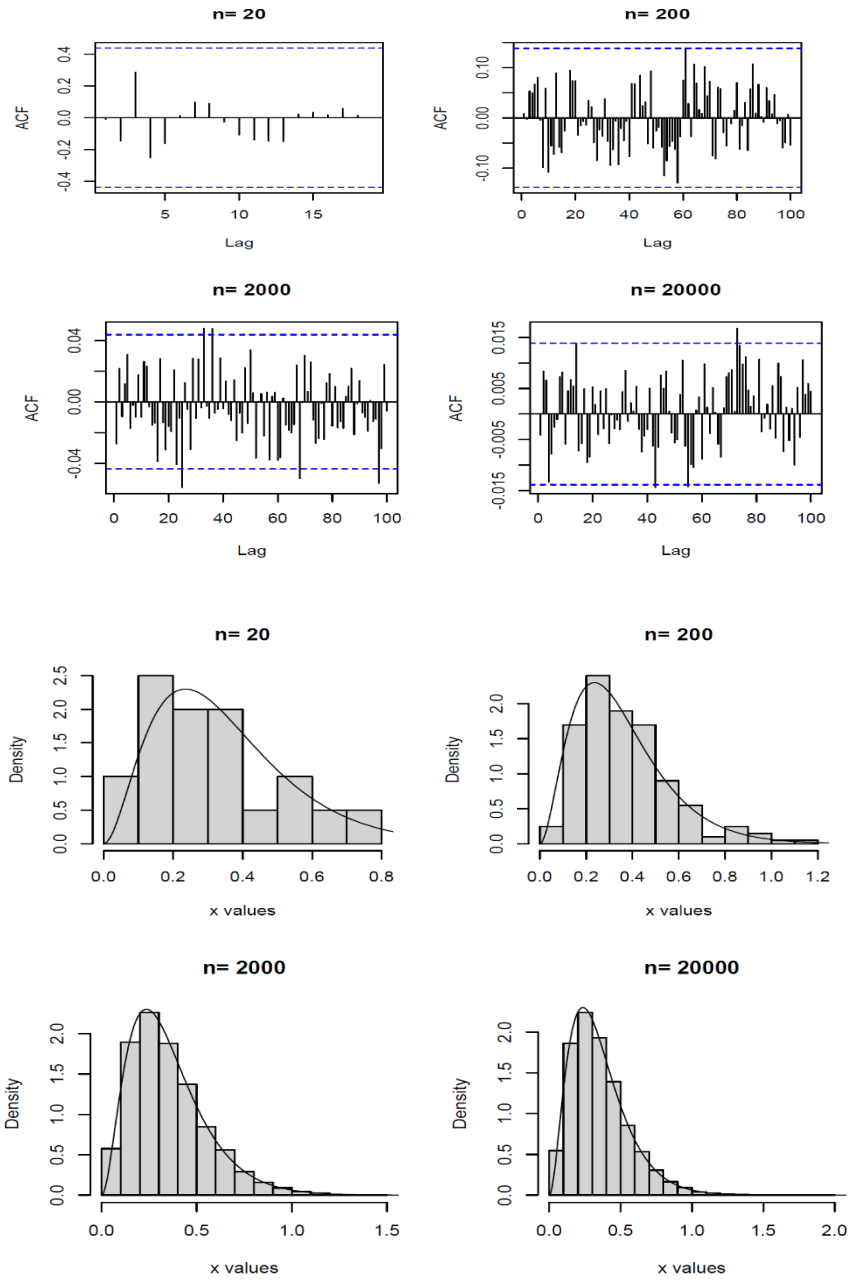

Figure 2: ACF and overlying the true pdf of gamma distribution on sample histogram plots for different sample sizes. Samples are generated using the q_rgamma method.

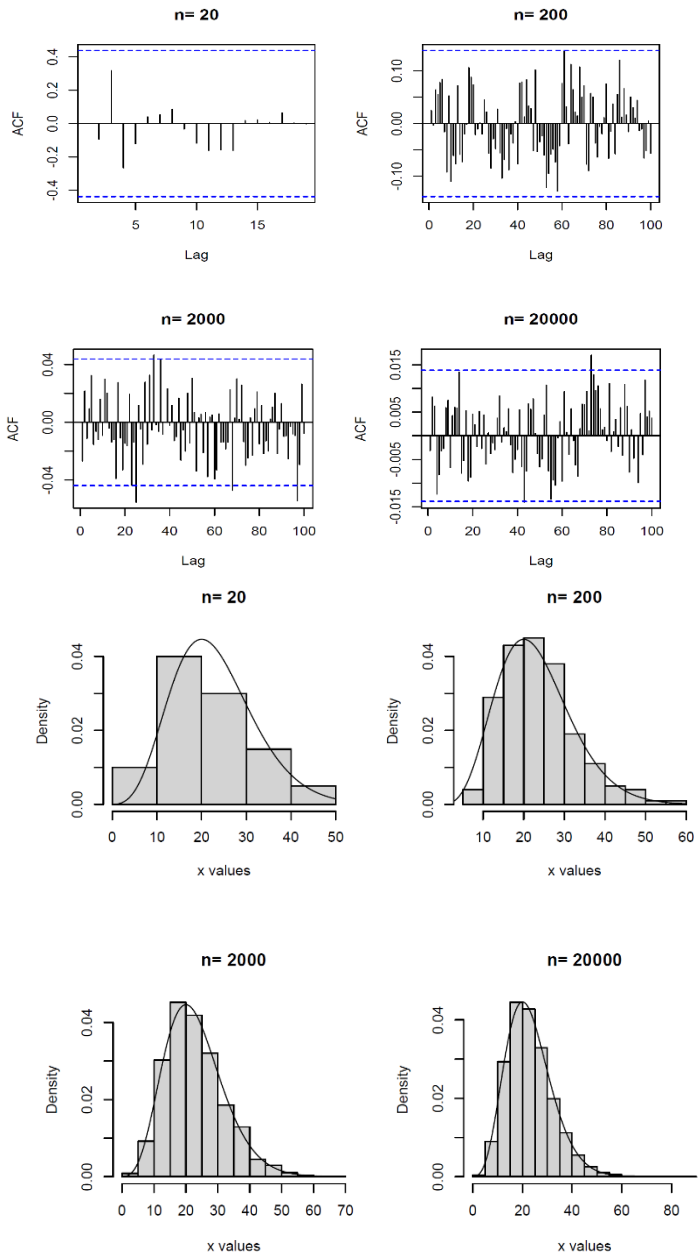

Figure 3: ACF and overlying the true pdf of chi-square distribution on sample histogram plots for different sample sizes. Samples are generated using the q_rchisq method.

R Codes

```
## Standard Normal Variate Generation by Quantile Transform Method (q_rnorm)
## Generate one data set which contains n observations ##
library(rootSolve)
rinorm<-function(n)
   {
     u<-runif(n,.5)
     a<-b<-numeric(n)
     for (i in 1:n)
       {
        f<-function(x)
         {
            1-((1/sqrt(2*pi))*(exp(-
           (x^2)/2)/(0.226+0.64*x+0.33*sqrt(x^2+3))))-u[i]
         }
        x<-uniroot.all(f,c(0,10))
        v=runif(1)
        a[i]=ifelse(v<=.5,x,-x)
       }
     return(a)
   }
## Generate k number of data sets (X and Y) for both methods where each data set contains n observations ##
k=1000; n=320
## Use seed number to generate each sample from the same place (useful to reproduce results)
seed=50
X<-matrix(0,k,n)
Y<-matrix(0,k,n)
for (i in 1:k)
  {
    set.seed(seed+i)
    X[i,]<-rinorm(n)
    Y[i,]<-rnorm(n)
  }
```

```
## Calculating Type 1 Error (whether two empirical distributions are the same)
size_test<-function(X,Y)
             {
                r<-nrow(X)
                count<-0
                for (j in 1:r)
                 {
                 ks<-ks.test(X[j,],Y[j,])
                  ind<-ifelse(ks$p>=0.05,0,1)
                  count<-count+ind
                }
               return(count/r)
             }
size_test(X,Y)
## Calculate average bias, MSE and coverage probability of mean and variance parameter
statis<-function(X,Y)
          {
             r<-nrow(X)
             col<-ncol(X)
             n<-(col-1)
         # mean and variance for sample generated by inverse transform method
              ime_x<-apply(X,1,mean)
             iva_x<-apply(X,1,var)
          # mean and variance for sample generated by available method
                    me_y<-apply(Y,1,mean)
                    va_y<-apply(Y,1,var)
           # bias for mean (mb), variance (vb), mse for mean and variance (mmse, vmse) under inverse method
              imb<-ivb<-immse<-ivmse<-imcp<-ivcp<-numeric(r)
              mb<-vb<-mmse<-vmse<-mcp<-vcp<-numeric(r)
              sd_ime_x<-sd(ime_x)
              sd_me_y<-sd(me_y)
```

```
z=qnorm(.975)
chi_up<-qchisq(0.025,n)
chi_ll<-qchisq(0.975,n)
for (j in 1:r)
   {
     # statistics under inverse method
            imb[j]<-ime_x[j]-0
           ivb[j]<-iva_x[j]-1
           immse[j]<-imb[j]^2
           ivmse[j]<-ivb[j]^2
     # do for icp as well for mean and variance
     # coverage probability for mean
        iul<-ime_x[j]+z*sd_ime_x
        ill<-ime_x[j]-z*sd_ime_x
        if (0>ill & 0<iul) imcp[j]<-1
     # coverage probability for variance for inverse
       transform method
     iv_ul<-(n*iva_x[j])/chi_up
     iv_ll<-(n*iva_x[j])/chi_ll
     if (1>iv_ll & 1<iv_ul) ivcp[j]<-1
     # statistics under available method
         mb[j]<-me_y[j]-0
      vb[j]<-va_y[j]-1
      mmse[j]<-mb[j]^2
      vmse[j]<-vb[j]^2
     # do for cp as well
     ul<-me_y[j]+z*sd_me_y
     ll<-me_y[j]-z*sd_me_y
     if (0>ll & 0<ul) mcp[j]<-1
    # Coverage probability for variance for available
      method
    v_ul<-(n*va_y[j])/chi_up
    v_ll<-(n*va_y[j])/chi_ll
     if (1>v_ll & 1<v_ul) vcp[j]<-1
}
```

```
      list(iamb=mean(imb),iavb=mean(ivb),iammse=mean(immse),iavmse
      =mean(ivmse),iamcp=mean(imcp),irmse=sqrt(mean(immse)+mean(iv
      mse)),iavcp=mean(ivcp),amb=mean(mb),avb=mean(vb),ammse=mean(
      mmse),avmse=mean(vmse),rmse=sqrt(mean(mmse)+mean(vmse)),amcp
      =mean(mcp),avcp=mean(vcp))
      }
## Return all the outputs
statis(X,Y)
## code for computing time to generate sample under q_rnorm and
rnorm ##
isimu<-function(n)
        {
          rinorm(n)
        }
simu<-function(n)
        {
          rnorm(n)
        }
n=20000
system.time(isimu(n*1000))/1000
system.time(simu(n*1000))/1000
## Code for checking randomness visually ##
library(TSA); par(mfrow = c(2, 2))   # Set up a 2 x 2 plotting
space
# Create the loop.vector (all the columns)
loop.vector <- 1:4
sample.size<-c(20,200,2000,20000)
seed=5
for (i in loop.vector)
            {
              # Loop over loop.vector
                   set.seed(seed+i)
                   x<-isimu(sample.size[i])
               # Plot acf of x
                   j=sample.size[i]
                   acf(x,lag=100,main=paste("n=", j))
              }
```

Code for testing randomness through Ljung-Box Test

```
n=20; set.seed(6)
x1<-isimu(20)
Box.test (x1, lag = 3, type = "Ljung")

n=200; set.seed(7)
x1<-isimu(200)
Box.test (x1, lag = 12, type = "Ljung")
Box.test (x1, lag = 20, type = "Ljung")

n=2000; set.seed(8)
x1<-isimu(2000)
Box.test (x1, lag = 20, type = "Ljung")
Box.test (x1, lag = 37, type = "Ljung")
Box.test (x1, lag = 87, type = "Ljung")
Box.test (x1, lag = 95, type = "Ljung")

n=20000; set.seed(9)
x1<-isimu(20000)
Box.test (x1, lag = 4, type = "Ljung")
Box.test (x1, lag = 30, type = "Ljung")
Box.test (x1, lag = 31, type = "Ljung")
Box.test (x1, lag = 49, type = "Ljung")
Box.test (x1, lag = 54, type = "Ljung")
Box.test (x1, lag = 63, type = "Ljung")
Box.test (x1, lag = 92, type = "Ljung")
```

Overlaying the true pdf on sample histogram

```
par(mfrow = c(2, 2))  # Set up a 2 x 2 plotting space
set.seed(12)
x1<-isimu(20)
Box.test (x1, lag = 100, type = "Ljung")
hist(x1,freq=F,breaks=4,main=paste("n=", 20),xlab="x values")
xx<-seq(-3,3,0.01)
lines(xx,dnorm(xx),type="l")

set.seed(7); x2<-isimu(200)
Box.test (x2, lag = 100, type = "Ljung")
hist(x2,breaks=15,freq=F,main=paste("n=", 200),xlab="x values")
xx<-seq(-3,3,0.01)
lines(xx,dnorm(xx),type="l")
```

```
set.seed(8); x3<-isimu(2000)
Box.test (x3, lag = 100, type = "Ljung")
hist(x3,breaks=15,freq=F,main=paste("n=", 2000),xlab="x values")
xx<-seq(-3,3,0.01)
lines(xx,dnorm(xx),type="l")

set.seed(9); x4<-isimu(20000)
Box.test (x4, lag = 100, type = "Ljung")
hist(x4,breaks=15,freq=F,main=paste("n=", 20000),xlab="x values")
xx<-seq(-3,3,0.01)
lines(xx,dnorm(xx),type="l")
```

##

Code for generating sample from Gamma density using quantile transform method

```
library(rootSolve)
```**## R code for shape parameter 3 ##**

```
g3=function(n,b)
    {
      v=c(); u=runif(n)
      for(i in 1:n)
         {
                 gc=function(x)
               {
                  1-exp(-x*b)*((1+(b*x/1)+((b*x)^2)/2))-u[i]
               }
            v[i]=uniroot.all(gc,c(0,1000))
         }
     return(v)
   }
## For other shape parameter values code will be very similar to previous one ##
## Code for producing Summary statistics and graphs are very similar to r_norm ##
## Original codes will be given to reader whenever authors are requested ##
```

Chi square variate generation by quantile Transform Method (q_rchisq)

Generate one data set which contains n observations

```
library(rootSolve)
ichisq<-function(n,k,l=0)
        {
          x = c()
          u = runif(n)
          for (i in 1:n)
            {
                  x[i]<-(k+l)*(1-
                  (2*(k+2*l))/(9*(k+l)^2)+qnorm(u[i])*sqrt((2*(k+2
                  *l))/(9*(k+l)^2)))^3
            }
           return(x)
         }
## For other parameter values code will be very similar to
previous one ##
## Code for producing summary statistics and graphs are very
similar to r_norm ##
## Original codes will be given to reader whenever authors are
requested ##
```