ON SOME METHODS OF GENERATING RANDOM VARIABLES: THE ACCEPTANCE-REJECTION METHOD.

BY

DIBIE NGOZICHUKWUKA PRECIOUS

PSC1607770

DEPARTMENT OF STATISTICS

FACULTY OF PHYSICAL SCIENCES

UNIVERSITY OF BENIN

BENIN CITY

EDO STATE

JANUARY, 2023

ON SOME METHODS OF GENERATING RANDOM VARIABLES: THE ACCEPTANCE-REJECTION METHOD.

BY

DIBIE NGOZICHUKWUKA PRECIOUS

PSC1607770

A PROJECT WORK SUBMITTED TO THE DEPARTMENT OF STATISTICS, FACULTY OF PHYSICAL SCIENCES, UNIVERSITY OF BENIN, BENIN CITY, EDO STATE IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELOR IN SCIENCE (B.Sc. HONS) DEGREE IN STATISTICS.

JANUARY 2023

UNDERTAKING

This project work was carried out by me, **DIBIE NGOZICHUKWUKA PRECIOUS** with matriculation number **PSC1607770.** I have not plagiarized the work of any other author(s). All published works used in this project have been duly cited and acknowledged.

______________________________________ ____________________

DIBIE NGOZICHUKWUKA PRECIOUS Signature and Date

Name of student

CERTIFICATION

This is to certify that this project was carried out by **DIBIE NGOZICHUKWUKA PRECIOUS** with matric number **PSC1607770** in the Department of Statistics, University of Benin, Benin City, Edo State.

______________________ ________________

Mr. ODIJIE C. O. Date

(Project Supervisor)

________________________ _____________

PROF. C. C. ISHIEKWENE Date

(Head of Department)

DEDICATION

To God Almighty who in his infinite mercies saw me through this project in good health. God's grace gave me the strength to be able to successfully accomplish this project. I equally dedicate this project to my parents, late Mr. SUNNY DIBIE and Mrs. R. DIBIE.

ACKNOWLEDGEMENT

I bow my head before the Almighty God, the omnipotent, the omnipresent, the merciful, the most gracious, the compassionate, the beneficent, who is the entire and only source of every knowledge and wisdom endowed to mankind and who blessed me with the ability to do this work.

I would like to take this opportunity to convey my cordial gratitude and appreciation to my worthy, and zealous supervisor Mr. ODIJIE C. O, without whose constant help, deep interest and vigilant guidance, the completion of this study would not have been possible. I am indebted to him for his accommodating attitude, thought provoking guidance, immense intellectual input, patience and sympathetic behavior.

I am thankful to FAVOUR DIBIE for her unrelenting effort, assistance and advice concerning my academics and also to all staff of the department of statistics.

A big thank you to my Mum for her everlasting support financially and in every area I need her.

Thank you to myself also for not giving up even when situation looks it.

God bless us all.

TABLE OF CONTENT

TITLE

PAGES

TABLE OF CONTENTS

ABSTRACT

In this study, we looked at how we can generate random variables using beta distribution as our target distribution with the acceptance-rejection method. We also used the uniform distribution as our proposal distribution. The inability to invert the CDF is when acceptance-rejection method comes in place. Some simulations were made and results were shown.

CHAPTER ONE

INTRODUCTION

1.0. BACKGROUND OF STUDY

In the real world data is not readily available sometimes and so simulation of random variable becomes necessary. We now return to discuss how to generate sample value (i.e random variable or observation) of specific random variables. We will consider few form One of the form of generating random variables is the physical sources. This method is the most basic way to generate random variables. The observation of the flip of a real coin, shuffle actual cards mixed numbered balls. This source are outside the computer science and therefore not practical or useful in the modern computer age.

We can also generate random variable using the empirical resampling method, this method is used sometimes to run simulation you need access to instance of random variable which are not distributed in a very precise way but don't have usable description of the desired distribution.

PROCEDURE

Suppose we were given the sample number 5,5,10,5,5, which has the mean equal to 6 furthermore suppose we have no description on how these number were generated but we want to know if the mean of at least 8 is likely or unlikely for five more number drawn the same way. This can be approximated by drawing many samples of size 5 from the original sample (allow the sample number to be in our sample multitude times)

Pseudo random generator is another method of generating random variables, to avoid need for external table and slow peripherals we tend to use this computer age method. This is to say that the output of deterministic procedures that is same or equivalent to true random sources. The most basic form of a sequential pseudo random generator is a sequence of state S(1), S(2), S(3)… where S(i+1)=g(S(i)), where g() is the deterministic function that map from state to state. The linear congruential generator is an example under this form.

The acceptance rejection method which is the area of specialization in this project is another way to convert one sequence of random variable into another. This method is a way to reproduce random sample from an unknown distribution (target distribution) by using random samples from a similar more suitable or appropriate probability distribution. The aim is for the accepted sample to be distributed as if they from the target distribution. If we assume we can generate a random variable according to the distribution P(x) we can 'reject sample' to a new distribution using an 'acceptance function' q(x) which returns as number in the interval [0,1].

ALGORITHM

1. Obtain a sample y from distribution Y and a sample from Uniform (0,1)
2. Check whether or not $U<f(y)/Mg(y)$.

   If this holds, accept y as a sample drawn from f; if not, reject the value of y and return to the sampling step.

The algorithm will take an average of M iterations to obtain a sample

The inverse transform method is used when we have the ability to generate instance of a random variable according to one distribution and we would like instances according to another distribution. This method is used when we have access to inverse of the cumulative distribution function of the distribution we are trying to generate. This method is a primary method for generating samples number at random from any probability given its cumulative distribution function. It uses uniform distribution to generate random variables of any given distribution E.G: Normal, Bernoulli, Exponential distribution E.T.C

ALGORITHM

1. Compute the inverse f^-1(f)
2. Generate independent random observation U1,U2,…,Un from a uniform number between zero and one.
3. Compute X1=f^-1(U1), X2=f^-1(U2), …, Xn=f^-1(Un).
   Then X1,X2,…,Xn are independent random observations of variables X.

AIM AND OBJECTIVES

The aim of the study is to generate random variables from selected distribution using Acceptance-Rejection method. The objectives is as follows:

1. To petition Acceptance-Rejection method in generating random variables from selected distribution
2. To reveal or show the procedure of generating random sample by acceptance-rejection method.

SCOPE OF THE STUDY

The study covers the methods of generating random variable using the acceptance – rejection method.

LIMITATION OF THE STUDY

1. Rejection sampling can lead to a lot of unwanted samples being taken if the function being sampled is highly concentrated in a certain region.
2. Rejection sampling is the smallest number of non-conforming items in a sample that would lead to the rejection of the entire lot. In most cases (beside reduced sampling) this value is equal to the acceptance.

DEFINTION OF TERMS

1. **Close form**: A form that does not use limits implicitly or explicitly.
2. **Data**: Facts and statistics collection together for reference or analysis.
3. **Random Variables**: This is described informally as a variable whose values depend on outcomes of a random phenomenon.
4. **Linear Congruential Method(LCM):** This is an algorithm that yields a sequence of pseudo randomized numbers calculated with a discontinuous piecewise linear equation.
5. **Sample**: A set of individuals or object collection or a defines procedures.

6. **PDF**: This is a statistical expression that defines a probability distribution for a discrete random variable.
7. **CDF**: This is the probability that the variable takes a value less than or equal to X.
8. **ALGORITHM**: This is a process or set of rules to be followed in calculations or other problems-solving operations especially by a computer.

CHAPTER TWO

LITERATURE REVIEW

In reality it may be difficult to have actual data on a special distribution and so it becomes necessary to generate random variables from such distribution. The generate of random variable can be traceable to John Von Neumann in the year 1946. Over the year, several scholars have researched and developed method of generating random variable for several distributions. However, in this study we shall concentrate mainly on Acceptance-Rejection method.

John Von Neumann(1946) discovered a novel method for generating Pseudo random number called middle square method. In this method, each successive number is the middle digits of the previous number.

Neumann(1945) in his first draft of a report on EDVAC (a planned successor machine to the ENIAC, one American first computers), proposed the stored program concept the idea grew out of discussions he had with several other computer pioneers.

D. h Lehmer(1949) proposed Linear Congruential Generator. This method is one of the oldest and best known Pseudo random number generator algorithm. In this generator, each single number determine its successor by means of a simple linear function followed by means by a simple linear function followed by a modular reduction. The formula of this generator is given as:

LCG: $X_i=(aX_{i-1}+c)$

Where: 'a' is the multiplier

'm' is the modulus

'X_0'($0<X_0<m$)' is the start value or the seed

'X_i' is the sequence of random number generated.

This process usually generates uniform pseudo random integers and when each integer id divided by the mod m, the result is sequence of uniformly distribution random number between 0 and 1 i.e uniformly distributed (0,1), which has the similarities which the CDF value of a given distribution F(X) for any given x since $0<F(X)<1$ also. The next step is usually to transform the uniform random numbers in to a random variable of the distribution in focus (target distribution) by inverting the CDF at the given point U. Unfortunately however the CDF of some distribution cannot be inverted into a closed form other methods can then be applied such as Acceptance-Rejection methods, Numerical method e.t.c. Acceptance-Rejection method was proposed by John Von Neumann.

Neumann(1951) proposed the method which is known as the rejection sampling (R.S). It is a classical Monte Carlo Technique for universal sampling that can be used to generate samples. The basic idea is to generate random samples from another distribution whose density 'major rises' or lie above the target distribution everywhere in the domain of definition of the two densities.

Jasbir S. Arora(2012) said that acceptance rejection method are modification of multistart algorithm to improve its efficiency by using ideas from statistical mechanics. In multistart method, a local minimization is started from each randomly generated point. Thus, the number of local minimization is very large. Acceptance rejection method modify this tunneling procedure. The basic idea of this is to sometimes start local minimization from a randomly generate point even if it has a high cost function value than that at a previously obtained local minimum. This is referred to as the acceptance phase, which involves the calculation of certain probabilities.

CHAPTER ONE

INTRODUCTION

1.0 Background of the Study

In the real world data is may not be readily available sometimes and so simulation of random variable becomes necessary. In this study, we shall discuss some methods of generating random variables and specifically concentrate on acceptance rejection method. One of the forms of generating random variables is the physical source. This method is the most basic way to generate random variables. Examples are the observations of the flip of a fair coin, drawing a card from a shuffled pack, drawing a ball from an urn or a container. These are method outside the computer application and therefore not practical or useful in the modern computer age, where large samples are required.

Using computer has made the generation of random variables simpler to carry out. This is usually done by sampling pseudo random numbers from probability distributions of known random variables with appropriate methods or techniques such as inverse transform method, convolution method, physical sources, empirical resampling method, acceptance-rejection method and so on.

The basic thing is usually to generate uniform random numbers via any of the methods available such as linear congruential generator. The random

numbers are then transformed into random variables of the distribution of interest through an appropriate method listed earlier. The most basic form of a pseudo random generator is a sequence of state $S(1), S(2), S(3)$...
where

$$S(i+1) = g\big(S(i)\big),$$

and $g(.)$ is the deterministic function that map from state to state.

For instance, in the inverse transform method, the cumulative density function of the distribution of interest is directly inverted with the uniform random number into the desired random variables. This method is used when we have access to inverse of the cumulative distribution function of the distribution we are trying to generate. This method is a primary method of generating samples at random from a probability given its cumulative distribution function (CDF). It uses random variables from uniform distribution to generate random variables of any given distribution. e.g. exponential distribution, Weibull distribution, etc. The procedure is first to compute the inverse $F^{-1}(x)$, then generate independent random observations, $u_1, u_2, u_3, \ldots, u_n$ from a uniform random distribution between zero and one. Then compute $x_1, x_2, \ldots, x_n$ which are independent random observations of the variable X. When it is however impossible to invert the CDF $F(x)$ of the random variable X, we resort to other methods such as acceptance-rejection, convolution, numerical methods, lambert W, etc.

The acceptance-rejection method which is the area of concentration in this project, is another way to convert a sequence of random variables of a known distribution to another. This method is a way to generate random sample from distribution (target distribution) of a random variable X with probability density function $f(x)$ by using a proposal distribution of another random variable with density $g(y)$. The idea is that one can generate a sample value from X by sampling from Y instead and accepting the sample from Y with probability

$u < \frac{f(y)}{Mg(y)}$ repeating the draws from Y until a value is accepted. 'M' here is a constant, infinitely bounded on the likelihood ratio $\frac{f(x)}{g(x)}$, satisfying $1 < M < \infty$ over the support (or domain) of X. The procedure is first to obtain a sample y from the known proposal distribution $g(y)$ and generate a uniform number $u \sim uniform\ (0,1)$. Then check for the inequality $u < \frac{f(y)}{Mg(y)}$. The procedure will take an average of M iteration(s) to obtain a sample.

1.1 Aim and Objectives

The aim of the study is to generate random variables from beta distribution using Acceptance-Rejection method. The objectives are as follows:

1. To demonstrate the procedure of generating random sample by acceptance-rejection method.

2. To compare the result of our study with theoretical distribution of the selected distribution(s) for similarities.

1.2 Scope of the Study

The study covers some methods of generating random variables, with concentration on acceptance –rejection method.

1.3 Limitation of the Study

1. In acceptance rejection method, it is often difficult to compute the ratio of the target distribution to the proposal distribution $\frac{f(y)}{g(y)}$ if it does not have simple closed form expression.
2. More points are usually rejected if the proposal density is too far from the target distribution and hence acceptance rate will be low which means more iterations will be needed to generate a single point. Hence the choice of proposal can be a difficult task in the method.

1.4 Definition of Terms

1.4.1. Closed form: A mathematical expression or equation that does not involve special functions such as gamma function, psi function, etc, but contain simple algebraic operators such as $+, \div, -$, brackets, exponents, etc.

1.4.2. Data: Facts and statistics collected together for inference or analysis.

1.4.3. Random Variables: This is described informally as a variable whose values depend on outcomes of a random phenomenon.

1.4.4 Linear Congruential Method (LCGM): This is an algorithm that yields a sequence of pseudo randomized numbers calculated with a discontinuous piecewise linear equation.

1.4.5. Sample: A sample is a subset containing the characteristics of a larger population.

1.4.6. PDF: This is a statistical expression that defines a probability distribution for a continuous random variable. It is the short form for Probability Density Function.

1.4.6. CDF: This is the probability that a random variable X takes a value less than or equal to a particular value, x. It is the short form for Cumulative Distribution Function.

1.4.7. Algorithm: This is a process or set of rules to be followed in calculations or other problems-solving operations especially by a computer

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

In this chapter, we review necessary literature on our study. We also present a brief review of the basic methods of generating random variable besides the one of our interest such as linear congruential generator, convolution method, empirical method, inverse transform method, and then the acceptance-rejection method of interest.

2.1 A Brief History of Random Variables Generation with Computers

The generation of random variable can be traceable to John Von Neumann in the year 1946. Over the years, several scholars have researched and developed method of generating random variable for several distributions.

John Von Neumann (1946) discovered a novel method for generating Pseudo random number called middle square method. In this method, each successive number is the middle digits of the previous number.

Neumann (1945) in his first draft of a report on EDVAC (a planned successor machine to the ENIAC, one American first computers), proposed the stored program concept. The idea grew out of discussions he had with several other computer pioneers.

D. H Lehmer (1949) proposed Linear Congruential Generator. This method is one of the oldest and best known pseudo random number generator algorithm. In this generator, each single number determines its successor by means of a simple linear function, followed by a simple linear function, followed by a modular reduction. The formula of this generator is given as:

$X_i = (aX_{i-1} + c) \; mod \; m, \qquad i = 1, 2, ..., m$

where: a is the multiplier

m is the modulo

$X_0, \; (0 < X_0 < m)$ is the start value or the seed.

X_i is the sequence of random integers generated.

This process usually generates uniform pseudo random integers and when each integer is divided by the modulo m, the result is sequence of uniformly distribution random numbers between 0 and 1 i.e. $U \sim uniform \; (0,1)$, which has the similarities which the CDF value of any given distribution $F(x)$, since $0 < F(X) < 1$ also.

Neumann (1951) proposed the method which is known as the rejection sampling (R.S). It is a classical Monte Carlo Technique for universal sampling that can be used to generate samples. The basic idea is to generate random samples from another distribution whose density 'majorizes' or lie above the target distribution everywhere in the domain of definition of the two densities.

Arora (2012) opined that acceptance-rejection method is a modification of multistart algorithm to improve its efficiency by using ideas from statistical mechanics. In multistart method, a local minimization is started from each randomly generated point. Thus, the number of local minimization is very large. Acceptance-rejection method modifies this tunneling procedure. The basic idea of this is to sometimes start local minimization from a randomly generate point even if it has a high cost function value than that at a previously obtained local minimum. This is referred to as the acceptance phase, which involves the calculation of certain probabilities.

2.2 Some Methods of Generating Random Variables from Known Distributions

After successfully obtaining a suitable uniform random number generator such as the LCG as outlined in some parts of the foregoing section, the next step is usually to transform the uniform random numbers into a random variable of any distribution in focus (target distribution) by inverting the CDF at a given point of the uniform random numbers generated, $U \sim Uniform\,(0,1)$. This is the method known as Inverse Transform method. It involves solving the equation

$$u = F(x), \quad u \in (0, 1)$$

for x in terms of u. That is,

$$x = F^{-1}(u)$$

where $F(x)$ is the CDF of the target distribution of the random variable X.

Unfortunately, however, the CDF of some distributions cannot be inverted into a closed form expression. Hence, other methods can then be applied such as Acceptance-rejection method, Numerical method, Convolution method, Lambert W method, etc., as pointed out in the introduction.

Inverse Transform method can be used to generate samples from known distributions whose CDF have simple closed form expressions such as exponential distribution, Weibull distribution, geometric distribution, etc. Some examples can be found in Harrison (2011), Thomopoulos (2013), Okwuokenye and Peace (2016). Specifically, Okwuokenye and Peace (2016) employed the Inverse Transform method iteratively on Lindley distribution and compared it with composition method of generating samples from the same distribution.

Ghitany *et al.* (2008) gave an algorithm form composition method of generating random variables from Lindley distribution.

Boon (2013) presented a convolution method of generating random variables from the Erlang distribution which is a special case of the gamma distribution when the shape parameter is an integer.

Acceptance-rejection method has been used by various authors from the inventor (Neumann, 1951) to others (Dieter and Ahrens, 1974; Tadikamalla, 1979; Rubinstein, 1981) and more recent authors (Harrison, 2010; Kroese *et al.,* 2011; Ekhosuehi and Odijie, 2021). Specifically, Ekhosuehi and Odijie (2021) generated random samples from gamma distribution using acceptance-rejection method. They compared the method with the method used in MATLAB 2010 and found out their method outperformed the MATLAB method on several comparisons made.

Opone and Ekhosuehi (2017) also generated random numbers from a distribution known as feco distribution using a numerical method.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

In this chapter, we present the method known as acceptance-rejection method which is employed in the procedure of generating random variables from a distribution whose CDF is not invertible due to lack of closed form mathematical expression for the CDF.

3.1 Acceptance-Rejection Method

The Acceptance-rejection (A-R) method is a classical method of random variables generation that allows one to sample from a distribution (known as the target) that is difficult or impossible to simulate by an inverse transformation. Instead, draws are taken from an instrumental density (also known as the proposal) and accepted with a carefully chosen probability. The resulting draw is a draw from the target density.

First, we assume that the distribution we wish to simulate has a probability density function $f(x)$; that is, the continuous case. Later we will give a discrete version too, which is very similar. The basic idea is to find an alternative probability distribution with density function $g(x)$, from which we already have an efficient algorithm for generating random samples (e.g., inverse transform method), with the additional condition that the function $g(x)$ is "close" enough to $f(x)$. In particular, we assume that the ratio $f(x)/g(x)$ is bounded by a constant $c > 0$ such that $\sup_x\{f(x)/g(x)\} \leq c$. We would want c as close to 1 as possible in practice.

Then, the following is the algorithm for generating X distributed as $f(x)$:

3.1.1 *Acceptance-Rejection Algorithm for continuous random variables.

1. Generate a random variable Y distributed as $G(y)$.

2. Generate U (independent from Y) from $Uniform(0, 1)$.

3. If $U \leq \frac{f(y)}{cg(y)}$, then set $X = Y$ ("accept"); otherwise go back to 1 ("reject").

Before we prove this and give examples, some things are noteworthy:

• $f(y)$ and $g(y)$ are functions of random variables, hence so is the ratio $\frac{f(y)}{cg(y)}$; and this ratio is independent of U in Step (2).

• The ratio is bounded between 0 and 1; $0 < \frac{f(y)}{cg(y)} \leq 1$, since $f(y) \leq cg(y)$ for all y.

• The number of times N that steps 1 and 2 need to be called (i.e., the number of iterations needed to successfully generate X) is itself a random variable and has a geometric distribution with "success" probability $p = P(U \leq \frac{f(y)}{cg(y)})$;

with probability mass function given by

$P(N = n) = (1-p)^{n-1}p, n \geq 1.$

Thus on average the number of iterations required is given by

$E(N) = 1/p.$

• In the end we obtain our X as having the conditional distribution of Y given that the event $\{U \leq \frac{f(y)}{cg(y)}\}$ occurs.

A direct calculation yields that $p = 1/c$, by first conditioning on

$Y : P(U \leq \frac{f(y)}{cg(y)} \mid Y = y) = \frac{f(y)}{cg(y)}$

thus, unconditioning and recalling that Y has density g(y) yields

$p = \int_{-\infty}^{\infty} \frac{f(y)}{cg(y)} * g(y)\, dy$

$= \frac{1}{c} \int_{-\infty}^{\infty} f(y)\, dy$

$= \frac{1}{c}$,

where the last equality follows from that fact that f is a density function (hence integrates to 1, by definition). Thus $E(N) = c$, the bounding constant. This indeed shows that it is desirable to choose our alternative density g so as to minimize this constant $c = \sup_x \left\{\frac{f(x)}{g(x)}\right\}$. Of course, the optimal function would be $g(x) = f(x)$ but that is not what we have in mind since the whole point is to choose a different (easy to simulate) alternative from f as F, the CDF of X

cannot be inverted. “In short, it is a bit of an art to find an appropriate g" Chib, S. and E. Greenberg (1995).

However, we summarize that the expected number of iterations of the algorithm required until an X is successfully generated is exactly the bounding constant c.

3.1.2 Proof of Algorithm in 3.1.1

Proof: We must show that the conditional distribution of Y given that

$U \leq \frac{f(y)}{cg(y)}$, is indeed F; that is, that $P(Y \leq y \mid U \leq \frac{f(y)}{cg(y)}) = F(y)$.

Letting $B = \{U \leq \frac{f(y)}{cg(y)}\}, A = \{Y \leq y\}$, recalling that $P(B) = p = 1/c$,

and then using the basic fact that $P(A|B) = P(B|A)P(A)/P(B)$ yields

$$P(U \leq \frac{f(y)}{cg(y)}|Y \leq \text{y}) * \frac{G(y)}{\frac{1}{c}} = \frac{F(y)}{cG(y)} * \frac{G(y)}{\frac{1}{c}} = F(\text{y}),$$

where we used the following computation:

$$P(U \leq \frac{f(y)}{cg(y)}|Y \leq \text{y}) = P(U \leq \frac{f(y)}{cg(y)}, Y \leq \text{y}$$

$$= \int_{-\infty}^{y} \frac{P(U \leq \frac{f(y)}{cg(y)}|Y = \omega \leq \text{y})}{G(y)} g(\omega)d\omega$$

$$= \frac{1}{G(y)} \int_{-\infty}^{y} \frac{f(\omega)}{cg(\omega)} g(\omega)d\omega$$

$$= \frac{1}{cG(y)} \int_{-\infty}^{y} f(\omega)d\omega$$

$= \frac{F(y)}{cG(y)}$.

3.1.3 Discrete Case

The discrete case is analogous to the continuous case. Suppose we want to generate an X that is a discrete random variable with probability mass function (pmf) $p(k) = P(X = k)$. Further suppose that we can already easily generate a discrete random variable Y with pmf $q(k) = P(Y = k)$ such that $\sup_{k} \{p(k)/q(k)\} \leq c < \infty$. The following algorithm yields our X:

3.1.4 *Acceptance-Rejection Algorithm for discrete random variables

1. Generate a random variable Y distributed as q(k).

2. Generate U (independent from Y).

3. If $U \leq \frac{p(y)}{cq(y)}$, then set $X = Y$; otherwise go back to 1.

However, we shall only consider the continuous case in simulation studies since both are similar in procedure.

3.1.5 Properties of the generated sample

We shall examine the following properties of the samples generated in comparison with their theoretical values. They include the mean and variance, of the sample, bias and mean squared error (MSE) of the estimates of the mean and variance. The statistics are given, respectively by:

Mean: $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Variance: $s^2 = \frac{1}{n-1}\sum_{i=i}^{n} (x_i - \overline{x})^2$

Bias: $E(\hat{\theta}) - \theta$ where $\hat{\theta}$ is the statistic of interest (mean or variance in our case) and θ is the theoretical (population) parameter (mean or variance in our case).

MSE $= \frac{1}{n}\sum_{i=i}^{n} \left(\hat{\theta} - \theta\right)^2$

where n is the sample size in all the cases above.

CHAPTER FOUR

SIMULATION STUDIES AND RESULTS

4.0 Introduction

In this chapter we present simulation studies based on our methodology. Brief discussion of results are also given along with the results of each simulation study.

4.1 The Target Distribution

Our target distribution for simulation studies is the beta distribution. The PDF is given by

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, 0 < x < 1, \quad \alpha, \beta > 0$$

Where $\Gamma(.)$ is the gamma function. The distribution can be written in short form as beta(α, β).

The CDF does not exist in closed form expression and is given by:

$$I_x(\alpha, \beta) = \frac{B(x, \alpha, \beta)}{B(\alpha, \beta)}$$

where

$B(x, \alpha, \beta) = \int_0^x t^{\alpha-1}(1-t)^{\beta-1}\, dt$ is the incomplete beta function.

and

$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\, dt$ is the complete beta function.

The plots of the PDF for selected values of the parameters (alpha = α, beta =β) is shown in Figure 1.

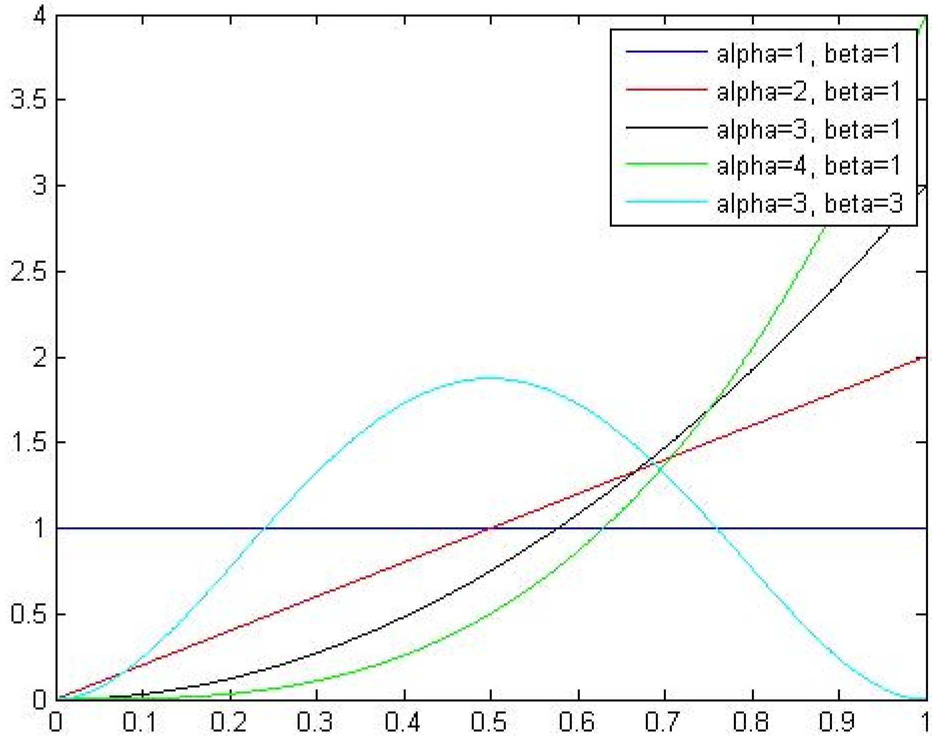

Figure 1: Shapes of beta PDF for various combinations of the shape parameters α and β

4.2 The Chosen Proposal Distribution

The proposal distribution chosen for the simulation is the uniform distribution with PDF given by

$$f(y) = \begin{cases} \frac{1}{b-a}, & a \le y \le b \\ 0, & otherwise \end{cases}$$

and CDF given by

$$F(y) = \begin{cases} 0, & y < a \\ \frac{y-a}{b-a}, & 0 \le y \le b \\ 1, & y > b \end{cases}$$

In our case, we shall use the uniform distribution on (0, 1) since it has a similar domain or support as the beta distribution (our target). In this case the PDF reduces to (with $a = 0$ and $b = 1$)

$$G(y) = 1,\ 0 \le y \le 1$$

And the CDF,

$$F(y) = \begin{cases} 0, & x < 0 \\ y, & 0 \le y \le 1 \\ 1, & y > 1 \end{cases}$$

4.3 Simulation Study 1: For $\alpha = 2,$ and $\beta = 1$, the target beta distribution PDF reduces to

$$f(x) = 2x,\ 0 \le x \le 1 \qquad (1)$$

The code for the generation of 1000 variables using MATLAB is given as follows:

4.3.1 A-R code for generating random samples from beta distribution:

```
clc    % clear the output screen (i.e. command window)

a = 2; % given/chosen parameter value of alpha
b = 1; % given/chosen parameter value of beta

%   The normalizing constant of beta distribution:
 const = gamma(a+b)/(gamma(a)*gamma(b));

%   Domain or support of the beta distribution:
x = linspace(0, 1, 10000);

%   The PDF of beta distribtuion:
f_x = const*(x.^(a-1)).*((1-x).^(b-1)); % as a function of dummy x
syms y
f_y = const*(y.^(a-1)).*((1-y).^(b-1))  % as a function of dummy y

%   The selected proposal density [also uniform(0,1) in this case]:
g_x = 1;     % as a function of x
g_y = 1;     % as a function of y

% Sample size to generate:
n = 500;

% The scaling constant
c = max(f_x./g_x);

plot(x, f_x), hold on     % plot the given beta PDF for display
later

% THE MAIN ALGORITHM
beta_rv = zeros(n, 1);     % to store the r. variables accepted
v1_values = zeros(n, 1);   % to store the corresponding vertical axis
values
rejected_x = zeros(n, 1); % to store rejected ones
v2_values = zeros(n, 1);   % to store vertical axis values of
rejected ones
count_rv = 1;              % the accepted counter/increment dummy
count_rej = 1;             % the rejected counter/increment dummy
while count_rv <= n
    y = rand; % generate a rand uniform number from g(y) the
proposal
    u = rand; %generate a the usual uniform(0, 1) for comparison
    if u <= eval(f_y/(c*g_y))       % if the condition is met
        beta_rv(count_rv) = y;      % accept y as x
        v1_values(count_rv) = u*c;  % save the corresponding plot
value
        count_rv = count_rv + 1;
    else
        rejected_x(count_rej) = y;
```

```
        v2_values(count_rej) = u*c;
        count_rej = count_rej + 1;
    end
end
scatter(beta_rv, v1_values), hold on
scatter(rejected_x, v2_values,'*'),
legend('target density','accepted', 'rejected')
hold off

figure
hist(beta_rv)
```

The first 10 values of the result of the random variable X generated are given as follows

0.3450 0.2882 0.6612 0.2479 0.2296 0.7666

0.3931 0.9147 0.4375 0.1341.

Of course, actual results will vary at each run due to randomness but the distribution (histogram) of the sample will be similar.

The histogram of 1000 variates (in comparison with the theoretical PDF) is given in Figure 2 (a) & (b) as follows.

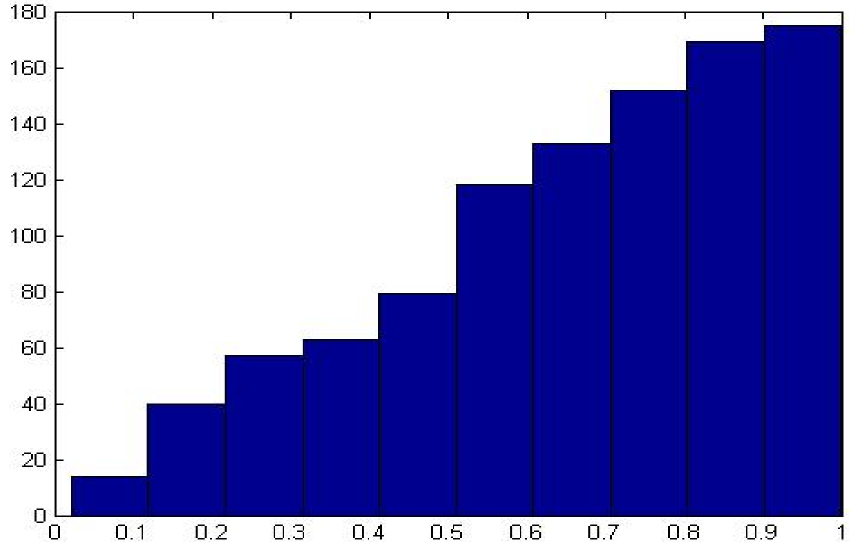

Figure 2(a): Histogram of 1000 beta variables for $\boldsymbol{\alpha = 2}$ and $\boldsymbol{\beta = 1}$

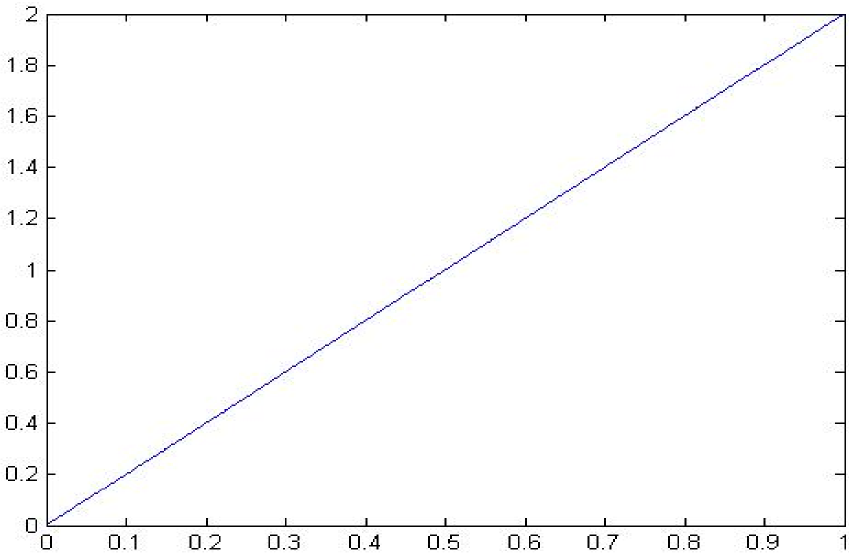

Figure 2(b): PDF of beta distribution with shape parameters $\boldsymbol{\alpha = 2}$ and $\boldsymbol{\beta = 1}$.

From Figure 2 (a) and (b), we can clearly see that the histogram of the random beta variables generated is similar to the theoretical distribution which is the PDF of beta distribution for $\alpha = 2$ and $\beta = 1$ given in equation 1.

The accepted beta variates and rejected variates (those that did not meet the condition in the step two of the algorithm in sub-section 3.1.1) are displayed in

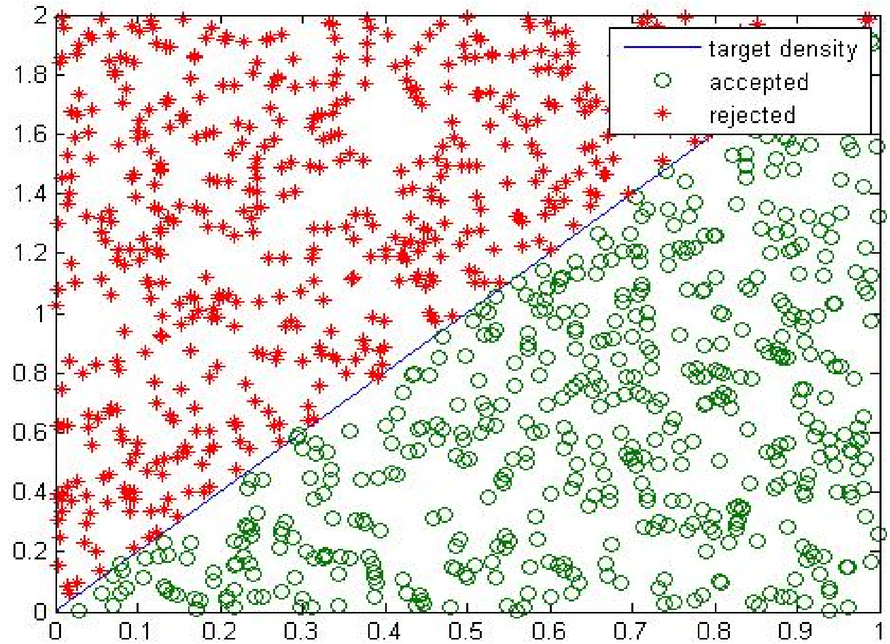

Figure 3.

Figure 3: Accepted beta variate (marked "o" under beta PDF curve) and rejected variates (marked "*" above the beta PDF curve) for $\boldsymbol{\alpha = 1}$ & $\boldsymbol{\beta = 3}$

4.3.2. Simulation Study 2: For $\alpha = 3,$ and $\beta = 3$, the beta function reduces to

$$f(x) = 30x^2(1-x)^2,\ 0 < x < 1 \tag{2}$$

The code for the generation of 500 variables using MATLAB is similar to that in sub-section 4.3.1 with n = 500.

The first 12 values of the result of the random variable X generated is give as

0.4853 0.4881 0.7165 0.5756 0.5454 0.5793

0.4819	0.2914	0.2380	0.8087	0.5751	0.2140

The histogram of the simulated sample and the theoretical PDF of the beta distribution with $\alpha = \beta = 3$ are given in Figure 4(a) & (b).

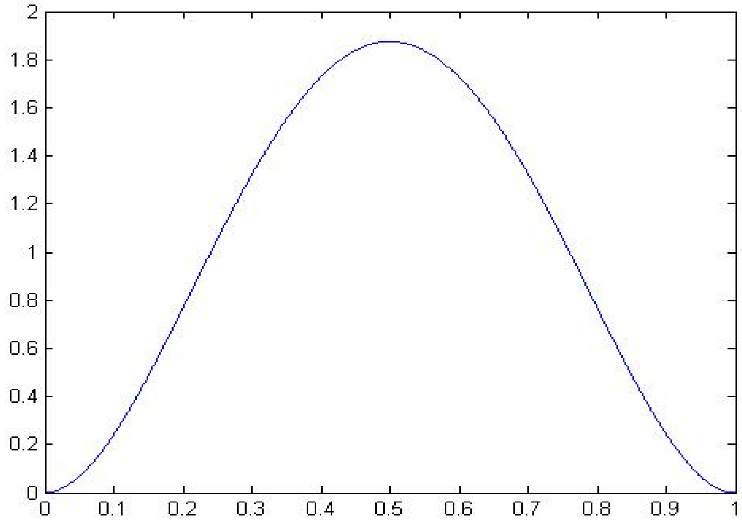

Figure 4(a): PDF of beta (3, 3)

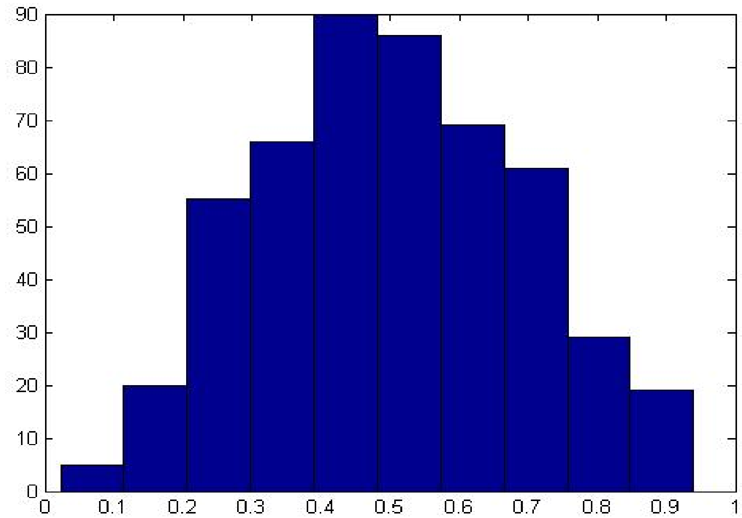

Figure 4(b): Histogram of simulated random variables from beta (3, 3)

Again we can clearly see that the histogram and the theoretical PDF of beta (3, 3) are very similar. The accepted and the rejected points are shown the next figure (Figure 5).

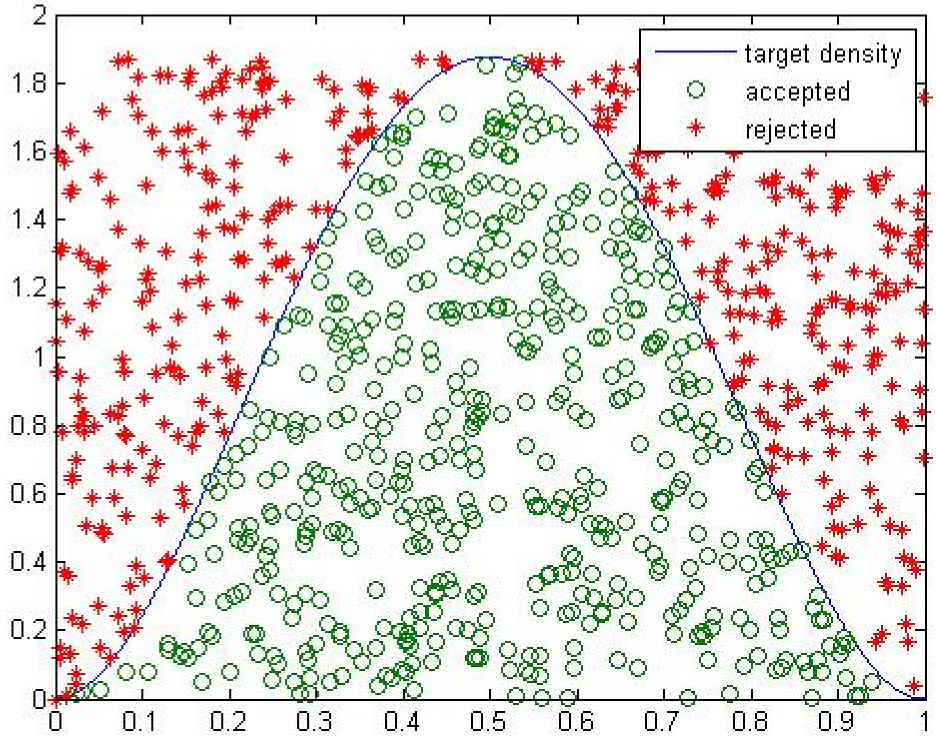

Figure 5: Accepted beta variables (marked "o") and the rejected variables (marked "*") for $\alpha = \beta = 3$

CHAPTER FIVE

SUMMARY AND CONCLUSION

5.1 SUMMARY

In this work, we have been able to study some methods of generating random variables. Specifically, we examined the Acceptance-Rejection method. We did simulation studies and presented some results which shows that the method we applied were appropriate and well implemented.

5.2 CONCLUSION

Acceptance-rejection method is applicable when the inverse transform fails due to inability to invert the CDF. This was the major focus of the research and beta distribution was used as our target distribution while the uniform distribution was also used as our proposal distribution. Simulation studies that were carried out showed that the acceptance-rejection method we applied simulated the data with high degree of accuracy since the plotted histograms were similar to the theoretical distributions. It is our hope that other researches will also be carried out using acceptance-rejection when the CDF is not conforming to a closed form when inverted, using this study as reference or guide.

References

1. Arora (2012) In introduction of Optimum Design (Third, Edition)

2. Boon (2013). Generating random variables. *Dept. Math & Comp. Sc. Eidhoven University.* Lecture 8, January 7, 2013 (slides 8-9).

3. Chib, S. and E. Greenberg (1995). Understanding the Metropolis Hastings Algorithm. American Statistics Journal 49, 327-335.

4. Ahrens, J. H; and Dieter, U (1974). Computer methods for sampling from gamma, beta, Poisson and binomial distributions. *Computing.* 12, 223-246.

5. Ekhosuehi, N and Odijie, C. O. (2021). A comparative Efficient Method of Generating Random Samples from the Gamma Distribution using Acceptance-Rejection Method. *Transactions of the Nigeria Assoc. of Math. Physics,* 15 (April-June, 2021 Issue), pp 81-90

6. Ghitany et al (2008). Lindley distribution and its application. *Math.Comput. Simul.* 78, 493-506.

7. Harrison R. L. (2010). Introduction to Monte Carlo Simulation. *AIR Confproc.* 1204: 17-21

8. Jodra, P. (2010). Computer generation of random variables with Lindley or Poisson-Lindley distribution via the Lambert *W* function. *Maths. Comp. Simul*. 81, 851-859.

9. Kroese et al (2011). Handbook of Monte Carlo Methods, ISBN: 978-0-470-17793-8.

10. Lehmer D. H. (1951): https://www.cs.ru.ac.za/research/g02c2954/final%20writeup.htm.

11. Neumann J. V. (1945): http://www.web.mit.edu/STS.035/www/pdf/edvac.pdf.

12. Neumann J. V. (1946): http://www.cis.usouthal.edu/falculty/daigle/project1/46john.htm.

13. Okwukenye, M. and Peace, K. E. (2016): A comparison of inverse transform and composition methods of data simulation from the Lindley distribution. *Comm. For Stat. App. And Methods.* 23, 517-529

14. Opone and Ekhosuehi (2017) On Some method of generating random variables. NAS *conference proceeding* (2017)

15. Online sources: *Acceptance-Rejection algorithm for continuous and discrete random variables, and the proof of the algorithm. http://www.colubia.edu/~ks20/4703-sigman/4703-07-Note-ARM.pdf