Mathematical Theory and Modeling
ISSN 2224-5804 (Paper) ISSN 2225-0522 (Online)
Vol.10, No.3, 2020
www.iiste.org

Emulation of the Central Limit Theorem Using a Monte-Carlo Based Approach

Ronald Ekyalimpa[1*]

1. College of Engineering, Art, Design, and Technology, Makerere University, PO box 7062, Kampala, Uganda

* E-mail of the corresponding author: rekyalimpa@gmail.com

Abstract

The majority of simulation experiments fulfill the central limit theorem particularly those that are stochastic and warrant the execution of multiple iterations during the process of their experiment execution. This class of simulation models can benefit from the existence of this theorem by utilizing it as a verification approach that certifies the accuracy in which the simulation experiment has been carried out. This paper formalizes this process and proposes a framework for achieving this given that thus far, the simulation community has not put forward a standard way for doing this. The systematic behaviors of freshmen at a University (particularly related to lectures), were abstracted and studied such that the cycle length for the time that a freshman commits daily towards their lectures was simulated using a Monte-Carlo based approach. The simulation of the academic behavior of freshmen was set up in a fashion that was consistent with the proposed framework so that it was possible to showcase the strategies in which the central limit theorem can be utilized in the verification of a simulation experiment.

Keywords: Simulation Experiment, Monte-Carlo Simulation, Central Limit Theorem

1. Introduction

For several years, the world has been operating based on systems, both natural and artificial (man-made) ones. These systems are comprised of several components that interact in a holistic and harmonize fashion. The spectrum of such systems include: natural biological systems, artificial systems such as manufacturing industries, transportation systems, logistics systems, financial systems, etc. Over the years, there have been a number of domains that have emerged with the sole goal of gaining insights into the intricate details of the functioning of these systems, for purposes of analyzing, designing, and improving them. This analysis and design require that at least one of the state variables of the system of interest is systematically tracked. A significant number of systems behave in a dynamic and stochastic fashion. This behavior is usually inherited by most, if not all of the state variables for a given system. Analysis and design that incorporate these dynamics and stochasticity warrant the use of advanced, robust techniques to guarantee reliable results. Data modeling techniques and simulation, are excellent examples of methods that can be used for this purpose. Each of these methods requires the state variables, behavioral logic, and other system constructs to be precisely abstracted in the form of a model, which is subsequently implemented on a computer. Models are preferred to experimenting with the real system because: 1) The system may not exist at the time the analysis/design is being done, 2) it is less risky (cheaper, safer, etc.) to experiment with a replica, i.e., a model, rather than the actual system, 3) there is no interference in the operation of the system. The abstraction of models and their implementation needs to be properly done to guarantee accurate results. There have been several strategies postulated for effective model abstraction and implementation. Good examples include the observance of the central limit theorem and the law of large numbers when conducting stochastic simulation studies. This was the focus of the study presented in this paper.

Fulfilling the central limit theorem in stochastic simulation studies demonstrates that the study was executed reliably and credibly, from a statistical point of view. The central limit theorem also provides a robust framework for researchers performing analytics, for obtaining precise measures (mean and standard deviation) of state variables that they may be tracking related to stochastic systems. A significant portion of, if not all, studies involving rigorous statistical analytics, outside of the simulation domain, strive to demonstrate that they have satisfied the central limit theorem. Demonstrating this is almost becoming a pre-requisite for accepting the results of such studies in these domains. This is not the case within the simulation domain and yet there are several benefits to embracing it. This may partly be due to a lack of appreciation of the theorem, and the fashion in which simulation experiments are set up, i.e., does not directly lend itself to the analytics associated with the central limit theorem. Consequently, showcasing ways in which simulation studies could fulfill such theorems, in

a simplistic fashion, would move the simulation domain steps closer toward fully embracing the practice of ensuring this theorem is explicitly fulfilled in every study. This was the main purpose of this study.

Case study based approaches are effective in achieving objectives both within industry practice and academia/research. As such, this approach was adopted in this study. A stochastic system that represents the daily schedule of a typical freshman at any college, was abstracted, modeled, and experimented with, using a Monte Carlo simulation-based approach. The experiment used a batch setup so that configurations were consistent with other typical studies that easily demonstrate fulfillment of the central limit theorem. A deliberate choice was made for this case study so that it was easy to present and follow. The rest of the paper details work related to the subject, an overview of the case study, the methods used to implement the case study, results and discussions, and the conclusions.

2. Literature Review

2.1 Systems

A system is a collection/group of interrelated, interdependent entities, or components that work interactively together in a seamless fashion (Ckeckland 1997; Backlund 2000). Most systems will have a list of possible states that they can assume at any given point in time. The state within a given system is a function of the values that the system variables have at that time. System variables are also often referred to as state variables. Systems can be static or dynamic in nature. Static systems have a set of parameters that represent the state of the system and these don't change as time passes. On the other hand, dynamic systems have their state variables changing with time. The behavior of these state variables is modeled using differential equations which have time as one of their independent variables. These state variables can be deterministic or stochastic in nature. Deterministic state variables are those that don't have uncertainty associated with them. Stochastic state variables are uncertain in nature. This uncertainty can be random or human-centric (also often referred to as linguistic) in nature. System state variables that are stochastic further sub-categorized as either discrete or continuous. Discrete state variables are those that belong to a discrete domain and are modeled using discrete probability distributions. A discrete domain is one that is bounded and has finite possible values that are known beforehand. Continuous state variables belong to a continuous domain and are abstracted and represented using continuous probability distributions. Continuous domains may or may not be bounded and are comprised of infinite possibilities of values that cannot be envisaged at the outset of an experiment/analysis.

It has always been the interest of analysts to study the behavior of systems under normal operation or their response to a stimulus that they may be exposed to. It is not always possible to learn behavioral patterns by disrupting the real system because of the associated safety, and cost risks from doing this. As such, abstracting these types of systems into computer models which can then be experimented with, is the most viable approach that can be adopted. The computer simulation domain provides inexpensive, robust techniques and tools to implement system abstraction and experimentation. It is for this reason that this domain has rapidly advanced and this paper seeks to further this advancement in a sustainably.

2.2 Computer Simulation

Computer simulation is a mathematical computer-based approach used to abstract a real-world system onto a computer for purposes of experimentation. There are different methods and forms in which this can be accomplished, e.g. Monte Carlo simulation, Discrete Event Simulation (DES), Continuous Simulation (CS). At a higher level, System Dynamics (SD), and Agent-Based Modeling (ABM) modeling paradigms, make use of those low-level simulation implementation schemes. In this paper, a Monte-Carlo type simulation was used. An overview of this type of simulation is presented in the following section.

2.3 Monte-Carlo Simulation

According to Rugen and Callahan (2008), Monte Carlo simulation is a probabilistic analytical process that is widely used in areas such as engineering, science (physics, biology, etc.), finance, insurance, health, and environmental risk assessment. It differs from the traditional simulation in that the model parameters have to strictly be treated as stochastic or random variables, rather than as fixed values (Bonate 2001). Regardless of the application area, the goal of using Monte Carlo analysis is to precisely define values associated with a particular state variable and a level of risk, i.e., profitability corresponding to each value. There is no doubt that Monte Carlo simulation is an extremely flexible and useful analytical approach with vast application areas. Nonetheless, this technique has several pitfalls associated with it (Ferson 2008). Four of these are discussed here. (1) It is data-intensive and usually cannot produce results unless a considerable body of empirical information has been

collected, or unless the analyst is willing to make several assumptions in the place of such empirical information. (2) Although appropriate for handling variability and stochasticity, Monte Carlo methods cannot be used to propagate partial ignorance under any frequentist interpretation of probability. (3) Monte Carlo methods cannot be used to conclude that exceedance risks are no larger than a particular level. (4) Finally, Monte Carlo methods cannot be used to effect deconvolutions to solve back-calculation problems such as often arise in remediation planning.

Given that Monte-Carlo simulation is a mathematical technique that involves performing analytics on deviates that are randomly drawn from their respective unique probability distributions, there is a need for several iterations to be performed, which warrants the use of computers in its implementation. Computer implementations are supported via commercial software such as @Risk, Crystal Ball, etc. There are also generic computer programming environments that have custom mathematics libraries that support writing Monte Carlo simulations such as Matlab, R, Mathematica, etc. Mathematica was utilized for writing the Monte Carlo simulation implementations for the case study presented within this paper.

2.4 Central Limit Theorem

The *Central Limit Theorem* (CTL) was first proposed by a French mathematician, Abraham De-Moivre, in 1733 (Henk 2004). Henk (2004) claims that at the time, Abraham used this theorem to demonstrate that the number of heads obtained from tossing a fair coin several times approximately followed a normal distribution. Abraham's scholarly contributions went silent and only resurfaced in 1812 when another French mathematician, Pierre-Simon Laplace used it to demonstrate how normal distributions can be used to approximate Binomial probability distributions (Henk 2004). The Central Limit Theorem also didn't get fully appreciated after Laplace's work. It's not until 1901 that a Russian Mathematician, Aleksandr Lyapunov revisited this theorem and demonstrated how it works in more general simplistic terms (Henk 2004). Subsequently, other scholars made this an area of active research resulting in its formal name and definitions (Polya 1920; Bernstein 1945; Le Cam 1986; Galton 1989; Hald 1998; Fischer 2011).

The essence of most statistical studies is to draw inferences about a specific population. Two popularly tracked statistics in such studies include the mean and standard deviation. *Central Limit Theory* (CLT), provides a credible framework for achieving precise estimates regarding characteristics of study populations. The central limit theorem states that if large enough samples are randomly drawn from a study population, the mean values of these samples will be normally distributed about the true mean value of the population. This normal probability distribution represents the mean and variance of the state variable being tracked in the study population. However, in order to fulfill the *central limit theorem*, there are a number of requirements that need to be met; these include: 1) each sample need to be sizable, i.e., of size 30 or greater, 2) the samples need to be drawn randomly, 3) the sampling is done with replacement, and 4) a large number of samples needs to be used. Once these requirements are met, the samples can be said to be independent and identically distributed (IID) hence making the statistical study credible.

Central Limit Theorem is an extremely useful phenomenon that facilitates data scientists to accurately predict the characteristics of a particular population, especially the mean and standard deviation of the population. There are four essential components within the CLT, all hinged on the mean and standard deviation. The first makes a statement about the relation between population and sample mean. The second makes a statement about the relationship between population standard deviation and sample standard deviation. The third makes a statement on the relation between sample mean values and population standard deviation. The last component makes mention of the distribution of sample mean and standard deviation values. In summary, CLT states that if large enough samples are drawn from a given population, then the average of the sample means will be equal to the population mean. Similarly, the average of the sample standard deviations will be equal to the standard deviation of the population (LaMorte 2016). The last aspect of the CLT states that the sample mean values and the sample standard deviation values will also be normally distributed. It has been stated that these three aspects of CLT hold true regardless of whether the population from which the samples are drawn follows a normal distribution or not. However, it is also mentioned that in case the population is not normally distributed, then the sample size should be large enough, i.e. greater than or equal to 30 (LaMorte, 2016). For normally distributed populations, this requirement for a large sample size does not need to be fulfilled for the CLT to hold true (LaMorte 2016). Equations 1, 2, and 3 summarize the first three components of the CLT mathematically.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \bar{x}_i \tag{1}$$

$$\sigma = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{2}$$

$$\frac{\sigma}{\sqrt{n}} = \sqrt{\frac{1}{n}\sum_{i}^{n}\left(Average\ of\ \ sample\ mean - \bar{x}_i \right)^2} \tag{3}$$

The *law of large numbers* is closely related to the *central limit theorem* because any statistical analytics performed and needs to be consistent with the *central limit theorem* would have to satisfy the *law of large numbers*. In this study, data samples were randomly drawn from population probability distributions in batches. Batches were taken to represent samples. Consequently, the number and size of batches were experimented with while trying to demonstrate the consistency of a dynamic cyclic system with the *central limit theorem*.

2.5 Simulation Model Verification and Validation

It is always desirable to have a simulation study (i.e., model development, experimentation process, results, etc.) certified as valid and reliable. Others can make use of the deliverables with confidence. Verification is a systematic process of making sure that every aspect of the study is being done the right way, i.e., things are being done right. This is consistent with the definitions provided by several scholars in the simulation domain. On the other hand, efforts directed towards making sure that the right things are being done in the study, i.e., the right thing is being done, would qualify as validation. The quest for simple and more effective verification and validation techniques is an issue that is actively being pursued in the simulation domain because of the need for credible and reliable models and results. This paper represents such an effort but from a verification perspective.

3. A CTL-Based Simulation Framework

Frameworks are proposed to provide a robust, consistent, and easy way for practitioners within a specific domain to accomplish certain tasks that are usually large scale and complex in nature or often create confusion and inconsistencies in their execution. A framework to facilitate simulation modelers to utilize the CTL in the verification of their simulation experiments has been proposed for these same reasons.

3.1 A Framework for Monte-Carlo Simulation

When a modeler is faced with a challenge of performing analytics on state variables that are stochastic in nature, their best bet is to design and implement/perform a Monte-Carlo simulation experiment. Monte-Carlo simulation is an abstract concept to a lot of scholars and practitioners that need and make use of it. In most cases, they make use of software that perform the required analytics behind the scenes, and simply treat it as a black-box process. This paper attempted to demystify the process of Monte-Carlo simulation by presenting a simplistic framework in which it can be performed. This framework is presented in the form of a Table. It assumes that we have got a total of "t" stochastic variables abstracted for a system or operation or process that needs to be diagnosed, analyzed, optimized, or designed. It is assumed that analytics are performed on these state variables using regular arithmetic operations (see the following equation). Examples of these operations could be addition, subtraction, multiplication, and division. The operations could be one of these enumerated ones but different as we move from the left to the right, i.e., from one state variable to another. Given that each of these state variables is stochastic in nature, they will each be represented by an appropriate probability distribution fitted using either empirical data or expert knowledge. The fact that these arithmetical operations cannot be directly applied between the probability distributions that represent each state variable, warrants the use of Monte-Carlo simulation. Rather than operate on the distributions themselves, Monte-Carlo simulation acts on random deviates or variates drawn from each of the respective probability distributions. This is done several times, i.e., for multiple iterations, so that a representative result is obtained. In each iteration, a random deviate (RD_i) is drawn/sampled from the probability distribution (PD_i) for the respective state variable (SV_i). The arithmetic operations defined between the state variables can then been applied to the drawn random deviates for that iteration and a result (R_i) obtained. This would represent one row in the tabulated framework for performing Monte-Carlo simulations. This is repeated for several rows, i.e., many iterations so that there are multiple values of the result. The set of values that are obtained as results (i.e., $\{R_1, R_2, R_3, ..., R_n\}$) are distributed in a particular fashion. This is summarized in Table 1.

Table 1. Schematic summarizing sampling and arithmetic done on deviates in a Monte-Carlo simulation

Iteration	**SV_1**	**SV_2**	**.**	**.**	**SV_t**	**Result**
	PD_1	**PD_2**	**.**	**.**	**PD_t**	
1	RD_{11}	RD_{12}	.	.	RD_{1t}	R_1
2	RD_{21}	RD_{22}	.	.	RD_{2t}	R_2
3	RD_{31}	RD_{32}	.	.	RD_{3t}	R_3
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
n	RD_{n1}	RD_{n2}	.	.	RD_{nt}	R_n

3.2 A Framework for CLT-based Monte-Carlo Simulation

The central limit theorem states that if several iterations of a simulation have been performed to generate several result values, these results tend to be normally distributed. Compliance of Monte-Carlo simulation results to the central limit theorem can be verified by performing output analysis on the values. This could include performing a distribution fit to the simulation results to see if a normal distribution comes up as an excellent fit. Also, p-p and q-q plots can be generated and visual inspection performed to check the conformance of results to a normal distribution. Conformance to the normal distribution confirms the fulfillment of the central limit theorem.

There is a variation to this experimental setup which includes performing the experiment in batches, with each batch having several iterations. Basic statistics would then be performed on batch results to obtain one mean value per batch. It is these batch mean values, i.e., $\{\mu_1, \mu_2, .., \mu_k\}$ that are then tested for normality. A tabular schematic for this experimental setup is shown below.

Table 2. Schematic summarizing a proposed CLT-based Monte-Carlo Simulation framework

Batch #	**Iteration #**	**SV_1**	**SV_2**	**.**	**.**	**SV_t**	**Iteration Result**	**Batch Mean Result**
		PD_1	**PD_2**	**.**	**.**	**PD_t**		
		RD_1	**RD_2**	**.**	**.**	**RD_t**		
1	1	RD_{111}	RD_{112}	.	.	RD_{11t}	R_{11}	μ_1
	2	RD_{121}	RD_{122}	.	.	RD_{12t}	R_{12}	
	.	RD_{131}	RD_{132}	.	.	RD_{13t}	R_{13}	
	.	.	.	.	.	.	.	
	n	RD_{2n1}	RD_{2n2}	.	.	RD_{2nt}	R_{2n}	
2	1	RD_{211}	RD_{212}	.	.	RD_{21t}	R_{21}	μ_2
	2	RD_{221}	RD_{222}	.	.	RD_{22t}	R_{22}	
	.	RD_{231}	RD_{232}	.	.	RD_{23t}	R_{23}	
	.	.	.	.	.	.	.	
	n	RD_{2n1}	RD_{2n2}	.	.	RD_{2nt}	R_{2n}	
.	.	.	.	.	.	.	.	.
	.	.	.	.	.	.	.	.
k	1	RD_{k11}	RD_{k12}	.	.	RD_{k1t}	R_{k1}	μ_k
	2	RD_{k21}	RD_{k22}	.	.	RD_{k2t}	R_{k2}	
	.	RD_{k31}	RD_{k32}	.	.	RD_{k3t}	R_{k3}	
	.	.	.	.	.	.	.	
	n	RD_{kn1}	RD_{kn2}	.	.	RD_{knt}	R_{kn}	

4. A Case Study

A case study based approach was adopted within this study because it was deemed the most effective strategy for showcasing how to configure typical simulation studies to be compliant with the *law of large numbers* and the *central limit theorem*. The case involved tracking the time associated with the arrival, residence, and return of college students on a typical school day. The choice of this case study was deliberate and meant to prevent the reader from getting distracted with the complex logic of a model of any other system that would otherwise have been chosen (e.g. a transportation system, logistic system, construction operation, industrial process, etc.). Its liner/cyclic nature with just three activities makes it easy to understand hence freeing the mind of the reader to focus on other verification aspects that are key to the simulation modeling process.

The system which tracks the different states of a college student on a typical school day ignored the time they are away from school. It only considers there state when inbound to the college, at the college, and outbound. These were abstracted as three liner/cyclic activities with the duration being the main parameter tracked/measured. A schematic layout (abstraction) showing the logical flow sequence and interrelation of the three state variables, is indicated in Figure 1. A third composite state variable, cycle time, was also tracked from the data models of the three basic state variables, as an outcome of the Monte Carlo simulation computations.

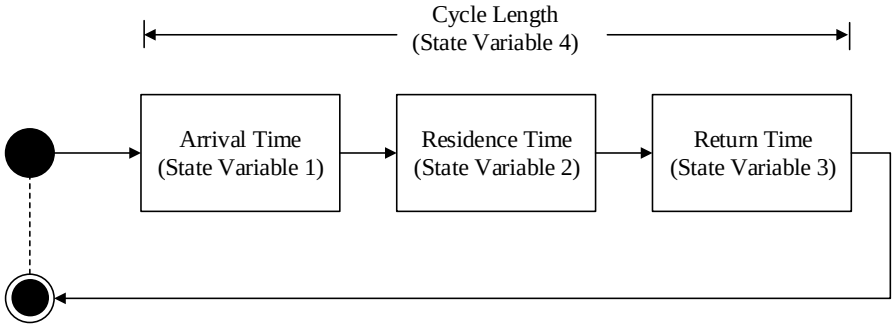


Figure 1. Cyclic schematic of the system

Each of the three state variables (arrival time, residence time, and return time) are stochastic in nature hence making the entire system stochastic and warranting the use of Monte Carlo simulation methods for its emulation. The stochasticity in the travel times can be attributed to a number of factors such as: the state of the person (i.e., their mood, level of urgency), disruptions encountered along the way (e.g. greeting friends), weather (sunny, overcast, rainy), natural variations in the travel speed from person to person. In order to ensure consistency in the data collected for each subject person and amongst all subjects, a number of assumptions were made, for example, the same transportation mode (i.e., walking) was assumed to be used all the time by all subjects in the study, and no major disruptions were assumed to occur when en-route to or from the college.

The study was comprised of different aspects, i.e., data collection, input modeling, simulation experimentation, and output analysis. A schematic diagram showing the interrelation between these components is summarized in Figure 2.

4.1 Data Collection

The study carried out was an empirical one and therefore had a data collection component/aspect. Travel and residence times for each subject were measured in minutes using a stop clock application on a smartphone. Daily records were then transferred and archived in an excel file. Landmarks were conveniently chosen and used as start or endpoints when doing timing with the stop clock, in order to ensure consistency in the data collection process. The data collection was carried out for just over one month and a half. Records for each subject were initially kept separate to facilitate front-end scrutiny and cleaning of the data, but these were subsequently aggregated together.

4.1 Data Analysis and Results

4.1.1 Descriptive Statistics

The analytics of descriptive statistics are the front-end of most traditional and state-of-the-art statistical studies. These statistics give the data analyst insights into trends in the data and overall quality of the data. As such, basic statistics were computed and results summarized for each state variable in Table 3.

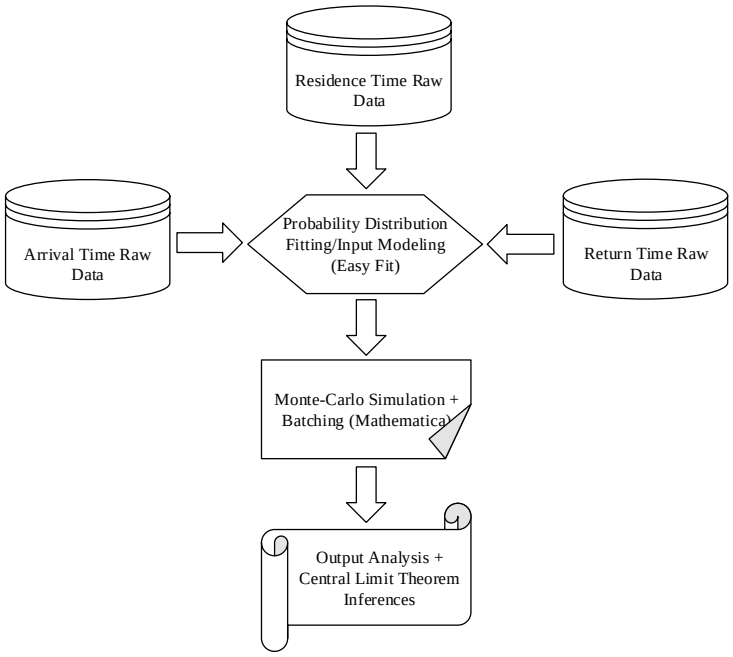
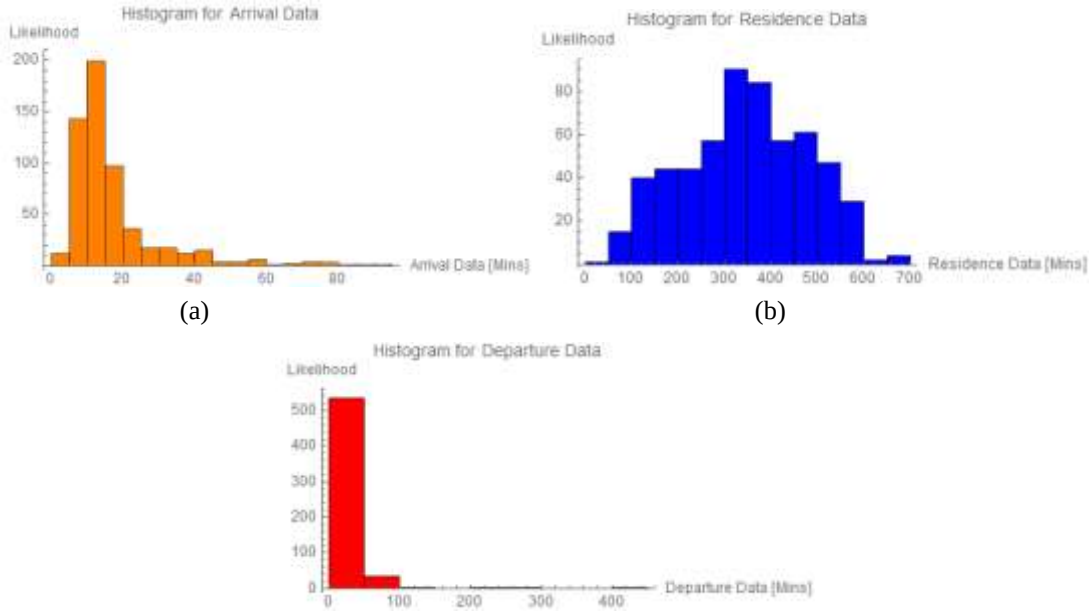


Figure 2. Schematic diagram of interrelation between different components

Table 3. Basic statistic values for the datasets of each of the state variables

Descriptive Statistic	**State Variable**		
	Arrival Time	**Residence Time**	**Departure Time**
Count	575	575	575
Minimum	3.00	15.00	4.00
Maximum	91.00	692.00	449.12
Mean	16.80	343.87	22.97
Standard Deviation	13.22	134.47	29.27
Skewness	2.59	-0.09	8.29
Kurtosis	10.84	2.32	99.93

The distribution of the values for each of the state variables was assessed by plotting histograms using the data for each variable. The plots generated are summarized in Figure 3.



(a) (b)

(c)

Figure 3. Histogram for (a) arrival, (b) residence, and (c) departure state variables

A visual inspection of the histogram for the "Arrival" state variable reveals that most of the data is skewed to the right. That for the "Residence" state variable indicates a near-symmetric distribution of the data. The "Departure" state variable is highly skewed to the right. These observations are consistent with the values of the skewness computed in the descriptive statistics.

Analytics were performed to gain further insights into the distribution of the data and establish the existence or non-existence of outliers. A box-whisker plot was generated for this purpose. The data for the Residence state variable showed the greatest spread/variation. The data for the Arrival and Departure state variables were tight. The plots indicate that the data collected for the Arrival and Residence state variables indicate that there are no outliers present in the data. However, the box-whisker plot indicates that the data for the Departure state variable had outliers present within it. The box-whisker plot indicates these outliers on the higher side, i.e., they are large values.

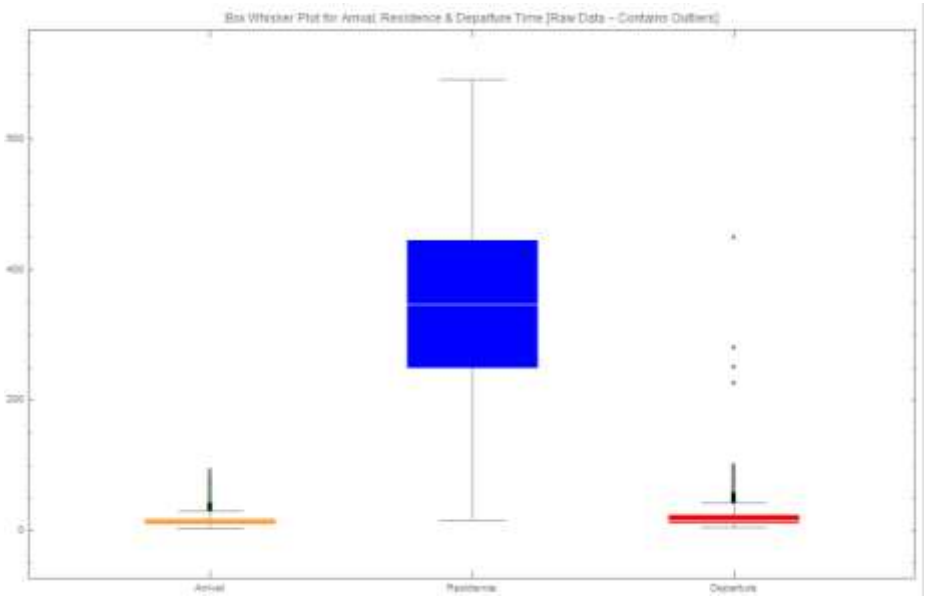


Figure 4. Box-Whisker plots for the study state variables

The metrics used in the generation of the box plots, i.e., the maximum, minimum, median, upper, and lower quartiles, are summarized in Table 4.

Table 4. Quartile statistic values for the datasets of each of the state variables

Statistic	Arrival Time	Residence Time	Departure Time
Maximum	91.00	692.00	449.12
Upper Quartile	17.86	445.50	24.00
Median	13.00	347.00	15.68
Lower Quartile	9.26	249.25	12.00
Minimum	3.00	15.00	4.00

The box-whisker plots generated indicate the presence of outliers in the data for the "Arrival" and "Departure" state variables. The data for the "Residence" state variable does not contain outliers. This is consistent with the relatively high values obtained for the kurtosis for the "Arrival" and "Departure" state variables and low kurtosis value for the "Residence" state variable. The reason for the presence of the outliers in the arrival and departure data lies in the fact that there are significant variations in the distances traveled by the students from their points they reside to the college and from the college to the points they reside. The residence time did not indicate the presence of outliers because students have similar schedules since they are doing identical courses hence variations don't go to extremes. Standard Jack-knifing operations were applied to the data for the "Arrival" and "Departure" state variable. The following equations, 4 and 5 were used to compute the threshold values used in the identification of the outlier values.

$$Upper\ outlier\ threshold = Upper\ quartile + \left((Upper\ quartile - Lower\ quartile) \times 1.5\right) \quad (4)$$

$$Upper\ outlier\ threshold = Upper\ quartile + \left(Inter - quartile\ range \times 1.5\right) \quad (5)$$

Once the outliers were removed from the Departure data, the subsequent analytics, such as probability distribution fitting, etc., were performed.

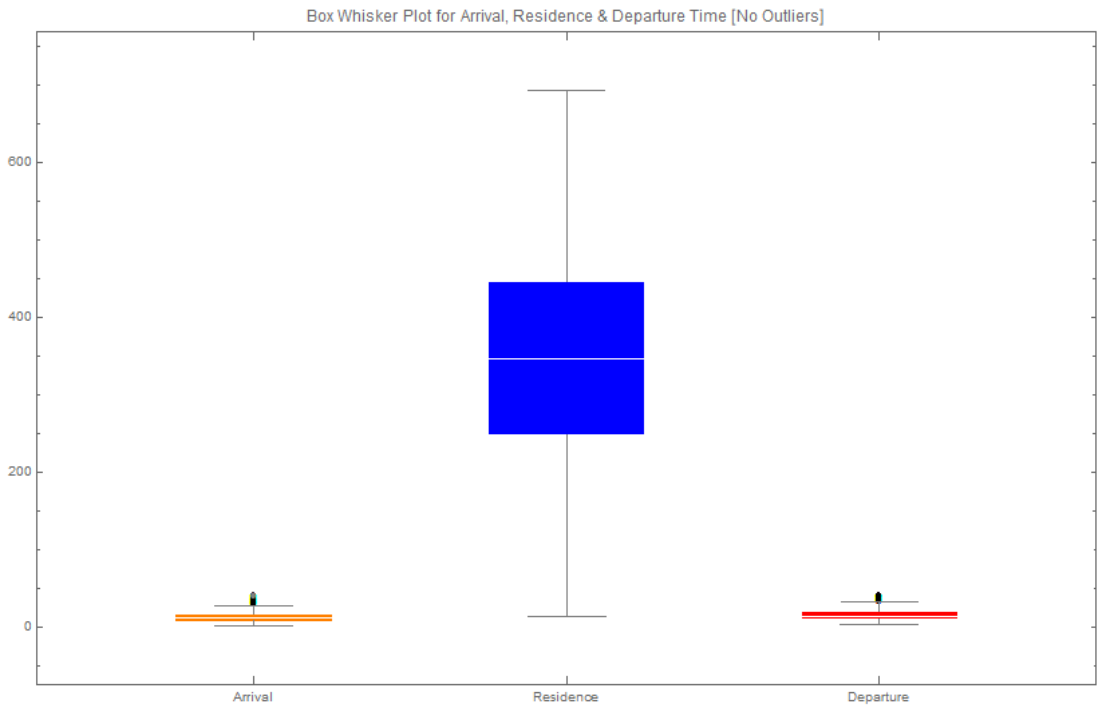


Figure 5. Box-Whisker Plots for the Study State Variables (No Outliers)

4.1.2 Input Modeling – Fitting Distributions

Fitting probability distributions is one of the crucial steps undertaken in any stochastic analytics study. A data-driven fitting process was adopted for this study because of the availability of data for the state variables of interest. There are several kinds of software that can be directly or indirectly used in distribution fitting. Some software provides for all the required distribution fitting services while others provide for some of them. "EasyFit" software was used to perform the probability distribution fitting in this study because it explicitly provides for the fitting of parameters and goodness of fit rankings in the same synthetic environment. Once the data is inserted into the software, probability distributions are fitted and ranked based on different criteria, i.e., Kolmogorov-Smirnoff, Anderson-Darling, and Chi-square (See Table 5).

Table 5. Top 10 probability distributions fitted and ranked based on different criteria

State Variable								
Arrival Time			**Residence Time**			**Departure Time**		
K-S	**A-D**	**Chi-Square**	**K-S**	**A-D**	**Chi-Square**	**K-S**	**A-D**	**Chi-Square**
Burr	Burr	Log-Logistic (3P)	Log-Pearson 3	Gen. Gamma (4P)	Kumaraswamy	Dagum	Dagum	Log-Logistic
Log-Logistic (3P)	Dagum	Burr (4P)	Johnson SB	Johnson SB	Gen. Gamma (4P)	Log-Logistic (3P)	Log-Logistic (3P)	Gumbel Max
Dagum	Log-Logistic (3P)	Burr	Triangular	Kumaraswamy	Johnson SB	Burr	Burr	Gen. Gamma
Burr (4P)	Burr (4P)	Dagum	Error	Error	Gen. Extreme Value	Gen.Extreme Value	Frechet (3P)	Frechet
Dagum (4P)	Dagum (4P)	Frechet	Beta	Gen. Extreme Value	Pert	Frechet (3P)	Gen.Extreme Value	Log-Pearson 3
Gen.Extreme Value	Gen.Extreme Value	Pearson 5	Gen. Gamma (4P)	Weibull (3P)	Fatigue life (3P)	Pearson 6 (4p)	Pearson 5 (3p)	Lognormal (3P)
Pearson 6 (4p)	Pearson 5 (3p)	Lognormal (3P)	Kumaraswamy	Triangular	Erlang (3P)	Pearson 5 (3p)	Pearson 6 (4P)	Log-Logistic (3P)
Pearson 5 (3p)	Pearson 6 (4p)	Log-Pearson3	Gen. Extreme Value	Dagum	Weibull (3P)	Pearson 6	Pearson 6	Inv. Gaussian
Pearson 6	Pearson 6	Dagum (4P)	Gen. Pareto	Normal	Beta	Lognormal (3P)	Burr (4P)	Fatigue Life (3P)
Log-Pearson 3	Log-Logistic	Gen.Extreme Value	Pearson 6 (4P)	Fatigue Life (3P)	Normal	Log-Pearson 3	Log-Pearson 3	Dagum

The probability distributions which are fitted to the data and ranked by the software are those supported by the fitting software. The distribution selected to model a particular state variable depends on its average ranking from all the criteria and the fact that the probability distribution is supported by the environment in which the stochastic experimentation is to be done. The overall rankings of the fitted distributions assuming equal importance of the three ranking criteria were then summarized in Table 6. Only the top five probability distributions for each state variable are presented.

Table 6. The top five ranked probability distributions for each state variable from the fitting process

Rank	State Variable		
	Arrival Time	Residence Time	Departure Time
1	Burr	Johnson SB	Log-Logistic (3P)
2	Log-Logistic (3P)	Gen. Gamma (4P)	Dagum
3	Dagum	Kumaraswamy	Frechet
4	Burr (4P)	Gen. Extreme Value	Burr
5	Dagum (4P)	Error	Gen. Extreme Value

Ratings for the probability distributions for each distribution fitting ranking criteria were calculated using the following formula. The subscript "j" represents the ranking criteria for distribution fitting, i.e., Kolmogorov-Smirnov (K-S), Anderson-Darling (A-D), and Chi-square. This Equation 6 is set up in such a way that distributions ranked high, are assigned a high rating while those ranked low are assigned low ratings.

$$Rating_j = (Maximum\ ranking + 1) - ranking_j \tag{6}$$

Total ratings were calculated (using the following Equation) for each probability distribution and then used as a basis for generating overall rankings for the probability distributions for all the fitting criteria. Each distribution fitting criterion was given equal importance in the total rating computations (Equation 7). The subscript "i" represents the specific probability distribution being dealt with.

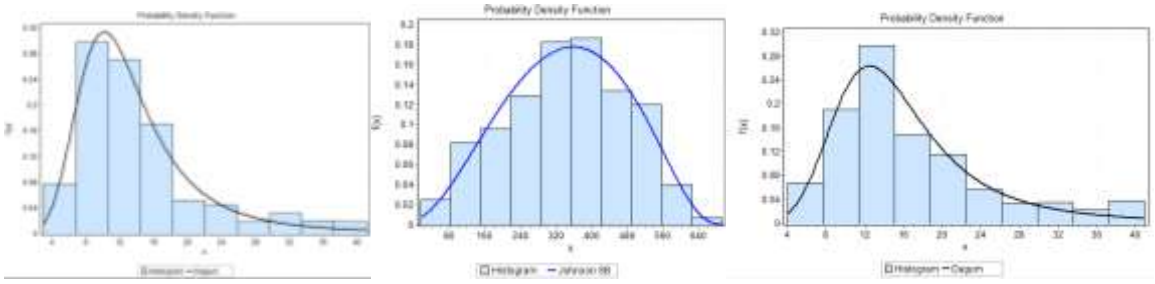
$$Total\ rating_i = \sum_{j=1}^{3} Rating_j \tag{7}$$

Distributions ultimately selected had to be bounded on the lower and upper side because of the nature of the system and its state variables that were being modeled. Also, these probability distributions had to be supported within the software in which the Monte Carlo simulation experimentation was to be done, i.e., Mathematica. Consequently, the probability distributions finally used in the modeling of the different state variables were summarized in Table 7.

Table 7. Fitted probability distributions for the different input state variables

State Variable	Fitted Probability Distribution
Arrival Time	Dagum [k=1.4242, α=3.2738, β=10.646]
Residence Time	Johnson SB [ϒ=-0.15439,δ=1.2529,λ=767.35,ξ=-60.513]
Departure Time	Dagum [k=1.3324, α=3.5146, β=13.251]

The Probability Density Functions (PDFs) for the probability distributions that were finally chosen to represent each of the state variables, were then presented in Figure 6. These fitted probability distributions were then made use of in the Monte-Carlo simulation experimentation work that was done. Details of this are presented in the following section.



(a) (b) (c)

Figure 6. PDF for (a) arrival time, (b) residence time, and (c) departure time

4.1.3 Monte-Carlo Simulation

The main purpose of this study was to demonstrate aspects of the central limit theorem from a stochastic simulation perspective. The system idealized was one intended to model cycle length based on three state variables – arrival time, residence time, and departure time. The cycle length was taken as the arithmetic sum of the three state variables. If these variables were deterministic in nature, the cycle length computation would have been straight forward. However, since the variables are stochastic, the simple arithmetic addition of the variables would not give the correct result. Consequently, random variates need to be sampled from the respective probability distributions and arithmetic, i.e., addition, performed on these variates. This is repeated several times until a pre-set number of iterations is reached. This process is referred to as Monte Carlo Simulation. The Monte Carlo Simulation in this study was reconfigured to accommodate the performance of the experiment in batches. This modification was made to allow for the different aspects of the central limit theorem.

A simulation experiment was run with a random number seed set to a value of 1,000,000. The number of iterations simulated was 10,000 together with a batch size of 10. The mean values for these batched cycle lengths were computed together with their variance. The code snippet written within the Mathematica environment to achieve this is summarized in Figure 7.

```
(*Monte-Carlo Simulations*)
Clear[ExperimentIndexes, BatchSize, TotalIterations, ListBatchCycleLengthValues, ListBatchCycleLengthMeanValues, ListBatchCycleLengthVarianceValues, ListBatchCycleLengthStandardDeviationValues,
  ListExperimentCycleLengthValues, ListStateVariable1, ListStateVariable2, ListStateVariable3];
BatchSize = 10; TotalIterations = 10000; ListBatchCycleLengthValues = {}; ListBatchCycleLengthMeanValues = {}; ListBatchCycleLengthVarianceValues = {};
ListBatchCycleLengthStandardDeviationValues = {}; ListExperimentCycleLengthValues = {}; ListStateVariable1 = {}; ListStateVariable2 = {}; ListStateVariable3 = {};
SeedRandom[1000000];
For[
  ExperimentIndexes = 1,
  ExperimentIndexes <= TotalIterations, ++ExperimentIndexes,
  (*Sample variates and sum them up. Add the sum to the list of variates*)
  Clear[sv1, sv2, sv3, sum];
  sv1 = RandomVariate[BeganDistribution[1.4242, 3.2739, 10.046]]; sv2 = RandomVariate[JohnsonDistribution["SU", -0.15439, 1.2529, -40.513, 787.35]];
  sv3 = RandomVariate[BeganDistribution[1.3324, 3.3146, 13.251]];
  AppendTo[ListStateVariable1, sv1]; AppendTo[ListStateVariable2, sv2]; AppendTo[ListStateVariable3, sv3];
  sum = sv1 + sv2 + sv3;
  AppendTo[ListBatchCycleLengthValues, sum]; AppendTo[ListExperimentCycleLengthValues, sum];
  If[
   ExperimentIndexes ≠ 1,
   (*Check to see if we are at the end of the batch*)
   If[
    Mod[ExperimentIndexes, BatchSize] == 0 || ExperimentIndexes == TotalIterations,
    (*We are at the end of the batch*)
    (*Find the average values*)
    AppendTo[ListBatchCycleLengthMeanValues, Mean[ListBatchCycleLengthValues]];
    AppendTo[ListBatchCycleLengthVarianceValues, Variance[ListBatchCycleLengthValues]];
    AppendTo[ListBatchCycleLengthStandardDeviationValues, StandardDeviation[ListBatchCycleLengthValues]];
    (*Clear Batch data*)
    ListBatchCycleLengthValues = {},
    (*We are not at the end of the batch*)
    Y = 1;
    ],
   Y = 1;]];
Histogram[ListBatchCycleLengthMeanValues, AxesLabel → {"Mean Batch Cycle Length (Mins)", "Likelihood"}, PlotLabel → "Histogram for Mean Batch Cycle Length", ChartStyle → Red]
Histogram[ListBatchCycleLengthVarianceValues, AxesLabel → {"Variance Batch Cycle Length (Mins)", "Likelihood"}, PlotLabel → "Histogram for Variance Batch Cycle Length", ChartStyle → Red]
Histogram[ListBatchCycleLengthStandardDeviationValues, AxesLabel → {"StDev. Batch Cycle Length (Mins)", "Likelihood"}, PlotLabel → "Histogram for Standard Deviation Batch Cycle Length",
 ChartStyle → Red]
```

Figure 7. Mathematica code snippet for the CLT-based Monte Carlo simulation experiment

The Mathematica code snippets were written based on sound logic that was first formalized and presented in a flow chart. This logic is presented in Figure 8.

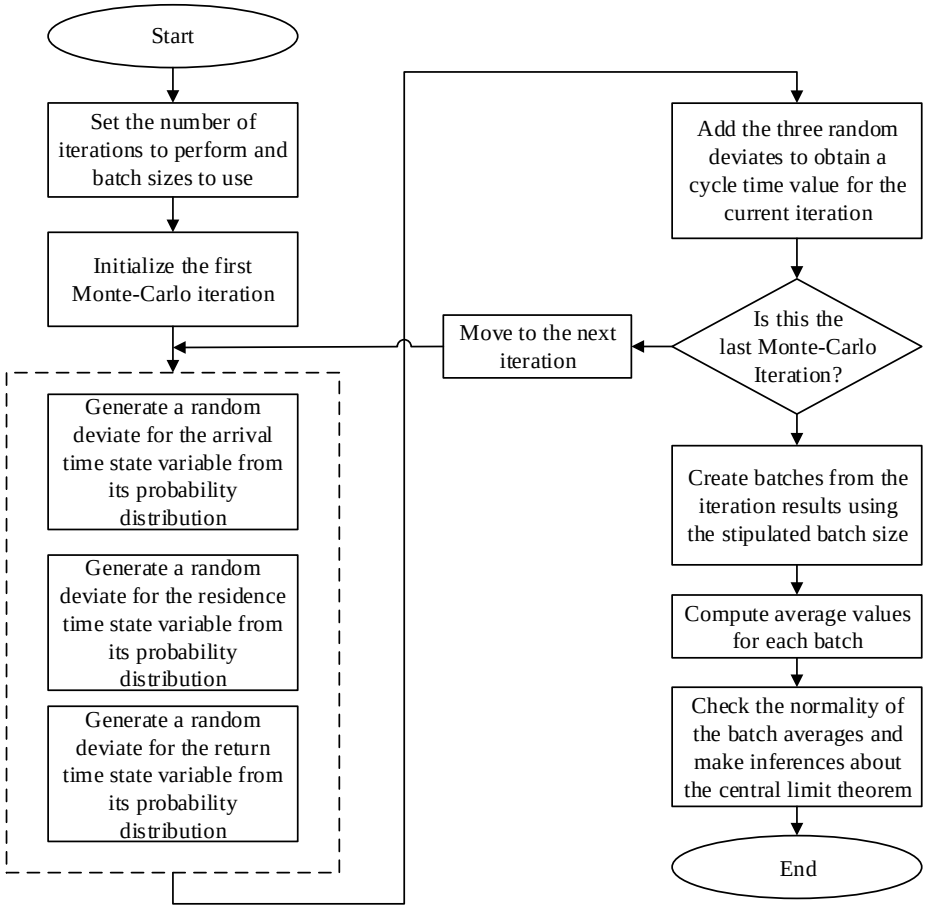
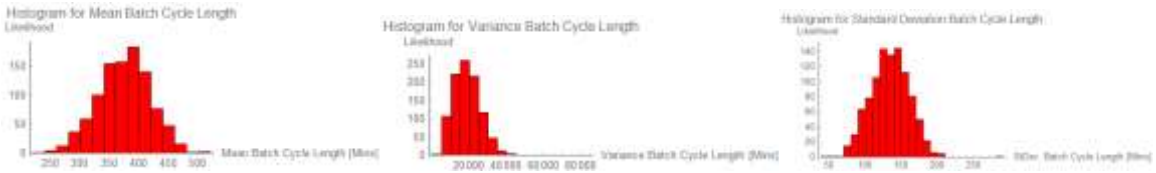


Figure 8. A flow chart of the Monte-Carlo simulation program logic

4.1.4 Output Analysis

4.1.4.1 Histograms

Histograms were plotted for the mean values, variances, and standard deviation of the batch values collected for cycle length during the simulation experiment. This was done in order to visually confirm whether or not the values appear to be normally distributed. The diagrams obtained were summarized in the following Figures (Figure 9).



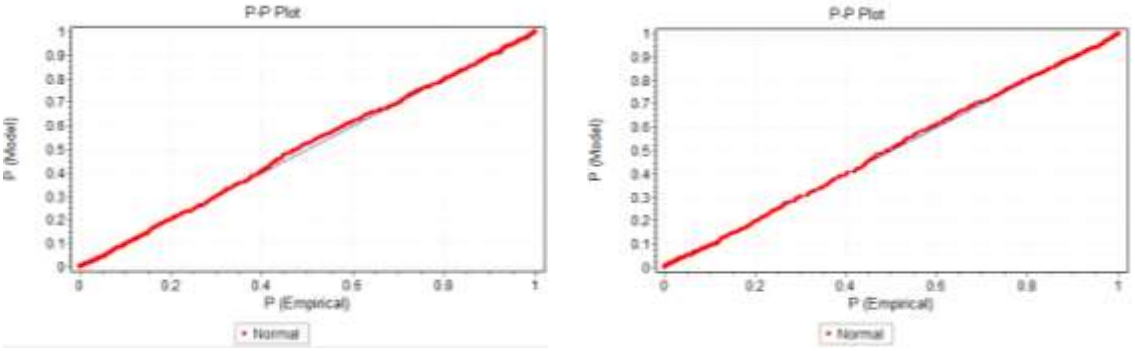
(a) (b) (c)

Figure 9. Histograms for (a) mean, (b) variance, and (c) standard deviation experiment values

Visual inspection of these histograms indicates that the values are normally distributed. These findings were consistent with the central limit theorem because the output values for the means and standard deviation tend to closely follow a normal distribution and were not dependent on the type of probability distributions for the inputs for the simulation experiments. There is no mention of the variance of the means following a normal distribution in the central limit theorem. Results obtained in this experiment are consistent with this because the variances are not normally distributed. They seem to be skewed to the right.

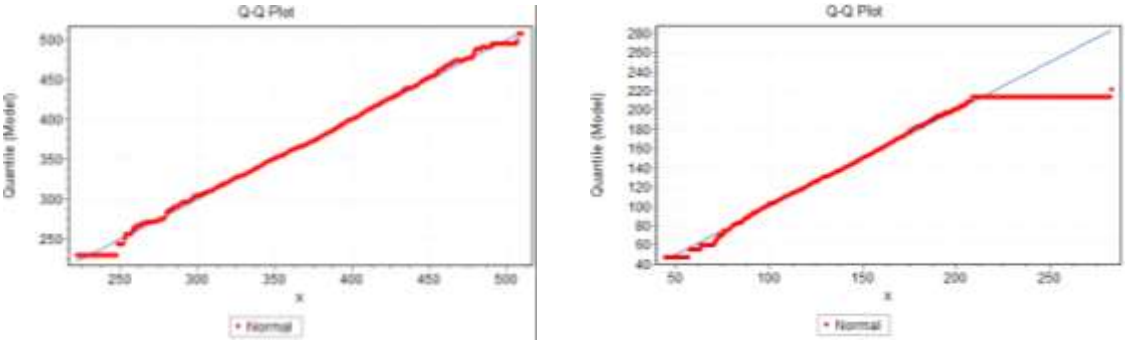
4.1.4.2 P-P and Q-Q Plots

To confirm inferences drawn from the visual inspection done on histograms plotted from the simulation experiments, P-P and Q-Q plots were made of the mean and standard deviation values. The results obtained were summarized in Figures 10 and 11.



(a) (b)

Figure 10. P-P plots to illustrate normality of (a) mean and (b) standard deviation values



(a) (b)

Figure 11. Q-Q plots to illustrate normality of (a) mean and (b) standard deviation values

The P-P and Q-Q plots confirm the normal nature of the mean and standard deviation values obtained from the simulated batches. The deviations of the theoretical normal distribution at the tail ends are typical and not an anomaly that would put the normality into question.

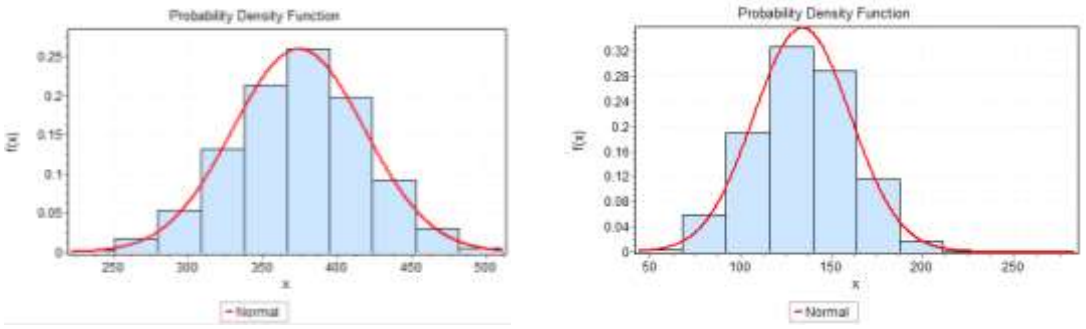
4.1.4.3 Fitted Normal Distributions

Following the confirmatory tests that indicated the normality of the mean and standard deviation values, theoretical normal distributions were fitted to the mean and standard deviation datasets respectively. The parameter values obtained were summarized in Table 8. Empirical mean and standard deviation values were obtained for each dataset and summarized in the same Table 8.

Table 8. Mean, standard deviation results of the Monte-Carlo simulation

Variable	Empirical Mean	Empirical Std. Dev.	Fitted Normal Distribution
Mean Batch Values	374.42	44.28	Normal [μ=374.41, σ=44.28]
Standard Deviation Batch Values	134.10	26.65	Normal[μ=134.10, σ=26.65]

A plot of the fitted normal distributions and the histogram of the datasets were generated and presented in Figure 12. This was done for purposes of confirming/illustrating the normality of the mean and standard deviation values generated from the simulation. These figures demonstrate the conformance of the Monte-Carlo simulation experiment to the Central Limit Theorem.



(a) (b)

Figure 12. Theoretical PDFs overlaying empirical PDFs for (a) means values and (b) standard deviation values

4.1.4.3 Mean and Standard Deviations Confidence Intervals

Confidence intervals are a mathematical way to express the range within which the true value of a parameter that is being estimated, lies, with a specified length of confidence. It is the closest that modelers can come to give the best guess as to what the actual value for a given parameter will be. Confidence intervals can be generated for different statistics, i.e., mean values, variances, quantiles, probability values, etc., using different mathematical formulations. In this case study, intervals were determined for the mean and variances based on a 95% confidence and results summarized in Table 9.

The confidence interval for the population's mean cycle time is based on the computed sample mean value. It has been demonstrated that the sample values from which the mean was computed are normally distributed. However, the standard deviation for the population is unknown. Consequently, a confidence interval formulation applied was that based on the t-distribution (formulation summarized in Equation 8).

$$\bar{x} \pm t_{\alpha/2}\frac{s}{\sqrt{n}} \tag{8}$$

The formulations used for obtaining the confidence interval for the standard deviation were based on that for the variance (the following Equation).

$$\frac{(n-1)s^2}{\chi^2_{\alpha/2}} \leq \sigma^2 \leq \frac{(n-1)s^2}{\chi^2_{1-\alpha/2}} \tag{9}$$

$$\sqrt{\frac{(n-1)s^2}{\chi^2_{\alpha/2}}} \leq \sigma \leq \sqrt{\frac{(n-1)s^2}{\chi^2_{1-\alpha/2}}} \tag{10}$$

The variance equation (Equation 9) is based on the relation between standard deviation and variance. When this relation is applied to the confidence interval formulations, Equation 10 is obtained. This was used to compute the confidence interval for the standard deviation. Results for confidence interval computation are summarized in Table 9.

Table 9. Confidence intervals for the mean and standard deviation results

Statistic	95% Confidence Interval
Mean	[372.21,376.63]
Standard Deviation	[132.27,135.98]

5. Conclusions and Recommendations

It has been demonstrated in this paper that the Central Limit Theorem (CTL) is a sound mathematical theorem that can be made use of within the computer simulation domain. It has been proposed to utilize CTL for simulation verification, particularly the Monte-Carlo type simulation studies. The paper proposed a framework that can be applied in configuration experiments for such kind of simulations so that it is easy to perform the verification. However, it's envisaged that the extent of use of the CTL for this purpose will vary significantly with the nature of the problem domain. The aspects of CTL that refer to the distribution of the sample means and standard deviations, can be applied indiscriminately in the verification of Monte-Carlo type studies. However, the use of aspects of the CTL which relate the mean of samples and standard deviation of samples to the mean and standard deviation of the population are conditioned on prior knowledge of the population mean and standard deviation. It should be noted that this may not always be the case for most typical engineering systems problems. This will likely be the only pitfall with the proposed simulation verification approach, but only for this class of problems.

It is recommended that a study be done to establish the effect of the total number of simulation runs and batch sizes on the fulfillment of the Central Limit Theorem.

References

Backlund, A. (2000). The Definition of System. *Kybernetes*, Vol.29, No. 4.

Bernstein, S. N. (1945). *The Scientific Legacy of P.L. Chebyshev. Part I: Mathematics.* Academiya Nauk USSR, Moscow & Leningrad.

Bonate, P.L. (2001). A Brief Introduction to Monte Carlo Simulation. *Clinical Pharmacokinetics*, Vol. 40, No.1. https://doi.org/10.2165/00003088-200140010-00002

Ckeckland, P. (1997). *Systems Thinking, Systems Practice.* Chichester: John Wiley & Sons, Ltd.

Ferson, S. (1996). What Monte Carlo Methods cannot do; Human and Ecological Risk Assessment. *An International Journal*, Vol. 2, No. 4, DOI: 10.1080/10807039609383659

Fischer, H. (2011). *A History of the Central Limit Theorem: From Classical to Modern Probability Theory: Sources and Studies in the History of Mathematics and Physical Sciences.* New York: Springer, ISBN: 978-0-387-87856-0.

Galton, F. (1989). *Natural Inheritance.* Basingstoke: Macmillan.

Hald, A. (1998). *A History of Mathematical Statistics from 1750 to 1930.* ISBN: 978-0471179122.

Henk, T. (2004). *Understanding Probability: Chance Rules in Everyday Life.* Cambridge: University Press. ISBN 0-521-54036-4.

Le Cam, L. (1986). The Central Limit Theorem around 1935. *Statistical Science*, Vol. 1, No. 1, Doi: 10.2307/2245503

Polya. G. (1920). On the Central Limit Theorem of Probability Calculation and the Problem of Moments. *Mathematical Journal*, Vol. 8, No. 3-4, Doi: 10.1007/BF01206525

Rugen, P., and Callahan, B. (1996). An overview of Monte Carlo, a Fifty Year Perspective, Human and Ecological Risk Assessment: *An International Journal*, Vol. 2, No. 4, DOI: 10.1080/10807039609383647