MONTE CARLO SIMULATIONS

UNIVERSITY OF SUSSEX

Contents

1. Random Number Generation

1.1 Introduction – Random Number Generation

To simulate random events using computer programs, we often use probability distributions. In this task, we will sample random variable X from **Log-Normal distribution**. The generated numbers will be then plotted on a histogram, which will be compared to the theoretical distribution by overlaying a theoretical curve on the histogram. Subsequently, to quantitatively assess the goodness of fit between the empirical and theoretical distributions, we will use a goodness-of-fit method as a function of sample size.

1.2 Literature Review – Random Number Generation

Lognormal Distribution:

According to Johnson, Kotz, and Balakrishnan (1994), the lognormal distribution is a continuous probability distribution containing random variables whose logarithm is normally distributed. In other words, X is lognormally distributed, if the natural log of X has normal distribution.

Ross (2010) states that the lognormal distribution is commonly used to model variables that are strictly positive and skewed to the right, such as stock prices, incomes, and particle sizes. The lognormal distribution can be defined by two parameters: the mean and the standard deviation of the logarithmic values of the variable.

The probability density function (PDF) of a lognormal distribution can be expressed as:

$$\mathbf{f(x) = (1 / (x * \sigma * sqrt(2\pi))) * exp(-((ln(x) - \mu)\hat{}2) / (2 * \sigma\hat{}2))}$$

where x is the variable, μ is the mean of the log(x), and σ is the standard deviation of the natural log(x).

The cumulative distribution function (CDF) of a lognormal distribution can be obtained by integrating the PDF, and CDF can be employed to calculate the probability of a variable falling within a certain range.

The lognormal distribution has several useful properties, such as the ability to model strictly positive variables and generate a wide range of skewness and kurtosis values. These properties make lognormal distribution useful in various fields, such as finance, economics, and engineering. [Bickel and Doksum, (2001)].

Box-Muller Method:

Devroye (1986) discusses the generation of random numbers from the lognormal distribution using various algorithms. One such algorithm is the Box-Muller method. The Box-Muller method is a popular algorithm for generating independent, standard normal random variables using uniform random variables. The algorithm converts a pair of independent uniformly distributed random variables U1 and U2, into a couple of independent normal random variables, Z1 and Z2, leveraging trigonometric functions.

Below are the steps of the Box-Muller:

Step 1: Firstly, two independent uniform random variables, U1 and U2, in the interval (0,1) are sampled.

Step 2: Next step is to calculate two variables, namely A and B, using the equations:

$$\mathbf{A = sqrt\ (-2 * ln(U1))}$$

$$\mathbf{B = 2 * \pi * U2}$$

Step 3: Next step is to compute two new variables, namely Z1 and Z2, using the equations:

$$\mathbf{Z1 = A * cos(B)}$$

$$\mathbf{Z2 = A * sin(B)}$$

The final output of the algorithm, Z1 and Z2, are independent and standard normal random variables. This method can be easily extended to generate normal random variables with any mean and variance by applying suitable transformations to the standard normal variables Z1 and Z2.

This algorithm was first introduced by Box and Muller in 1958. Despite the limitation that the method requires pairs of uniform random variables it has become a widely used algorithm for generating normal random variables in various application areas including but not limited to the sectors such as finance, engineering, and statistics [Knuth, 1997].

The following section describes how lognormal distribution is obtained from a normally distributed random variables and statistical tests to compare the empirically sampled random variables and theoretical distribution.

1.3 Solution – Random Number Generation

The PDF of the log-normal distribution with parameters μ (mean of log of x) and σ (standard deviation of log of x) is given by:

$$\mathbf{f(x) = (1\ /\ (x * \sigma * sqrt(2\pi))) * e\hat{}(-(log(x)-\mu)\hat{}2\ /\ (2\sigma\hat{}2))}$$

here μ is the mean of the logarithm of the random variable and σ is its standard deviation.

To compute random numbers from the lognormal distribution, following algorithm is implemented:

Step 1: Select μ and σ to the desired values.

Step 2: Produce a random number Z from the standard normal distribution. This is done using Box-Muller method as described in the above section.

Step 3: Calculate **X = e^(μ + σ * Z)**. Here X is the random number from the Log-Normal distribution.

The theoretical mean E(X) and variance V(X) of the lognormal distribution with parameters μ and σ are given by:

$$\mathbf{E[X] = e\textasciicircum(\mu + \sigma\textasciicircum 2/2)}$$

$$\mathbf{Var(X) = (e\textasciicircum(\sigma\textasciicircum 2) - 1) * e\textasciicircum(2\mu + \sigma\textasciicircum 2)}$$

We can use these values to check the correctness of the generated samples.

By setting the number of samples, n, to 10,000 and the parameters **μ** and **σ** to 0 and 0.5 respectively, we sample the random variables from the log normal distribution. Upon plotting the histogram for the sampled values, we can observe, by overlaying the theoretical distribution using R's library function dlnorm, that the generated samples closely follow the distribution. Below is the plot demonstrating the same.

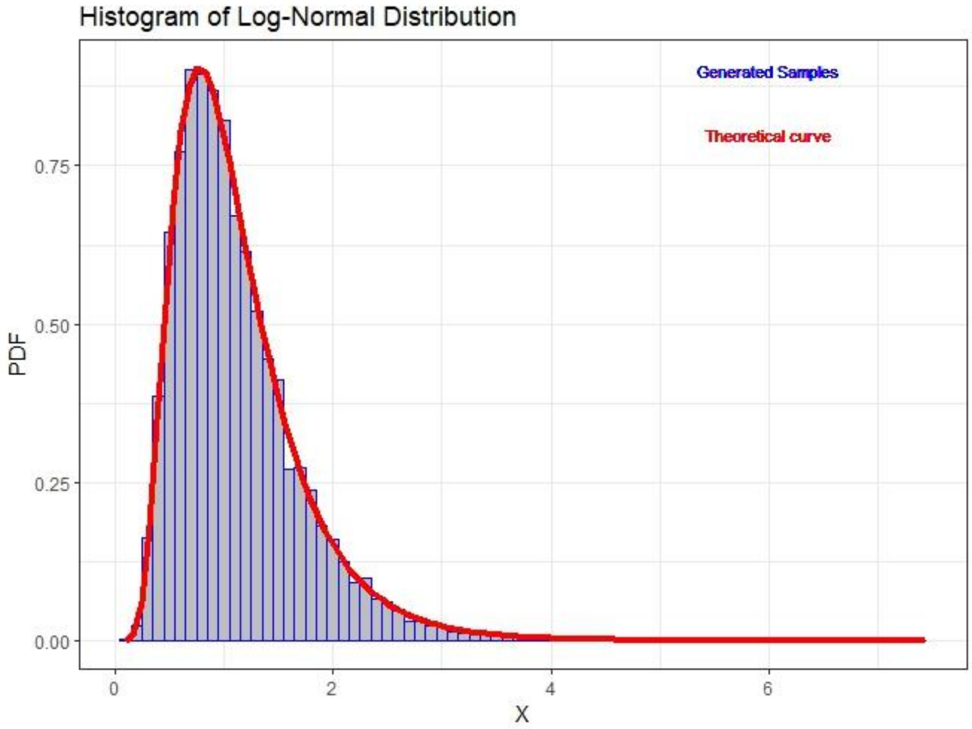


Figure 1: Comparing theoretical and empirical distributions

The theoretical curve in red closely follows the frequency distribution of the empirically sampled log-normal distribution. To further verify whether the two distributions belong to same family we conducted two statistical tests:

Chi-Square test:

According to Lehmann and Romano (2005), chi-square test is a popular statistical test for analyzing the relationship between discrete variables. It involves comparing the observed and expected frequencies for each category, with an underlying assumption that the variables are independent. The test statistic follows a chi-square distribution where the degrees of freedom is equal to the number of classes minus one. To conduct the test, a null hypothesis must be formulated which states the two variables are independent, calculate observed and expected frequencies, and subsequently compute the chi-square test statistic using the below formula.

$$X^2 = \Sigma ((O - E)^2 / E)$$

Where O and E are observed and expected values in a particular bin.

If the value calculated using the above formula is greater than the critical value, the null hypothesis can be safely rejected, and it can be concluded that there is a significant relationship between the variables. Montgomery and Runger (2018) discuss the chi-square test and its applications in various fields, such as engineering and quality control.

For our case, we computed a P-value of 0.029 corresponding to a test statistic of 69.33. Therefore, we can reject the null hypothesis with 95% confidence level. Hence the two distributions are identical.

KS-Test:

Massey Jr. (1951) presents the original paper by Kolmogorov and Smirnov introducing the KS test for goodness of fit. The Kolmogorov-Smirnov (KS) test is a non-parametric statistical test which can be used to ascertain if two samples come from the same distribution. This test is based on comparing the cumulative distribution functions (CDFs) of the two samples. The D statistic, which is calculated as the maximum difference between the two CDFs, is used as the test statistic. The null hypothesis is that the two samples are derived from the same distribution.

We get a p-value of 0.268 for the corresponding D-statistic of 0.01. This means that null hypothesis cannot be rejected and therefore there is evidence that the two samples belong to the same distribution.

2. Markov Chain Monte Carlo methods (MCMC)

2.1 Introduction - MCMC

In this task we are asked to simulate a three state Markov chain using Monte-Carlo simulations. We will use MCMC techniques to derive the invariant distribution of the transition probabilities associated with the three states.

2.2 Literature Review - MCMC

A Markov chain is a stochastic process that involves a series of events or states in which the probability of migrating from one state to another is dependent strictly on the current state and no prior states. Markov chains find applications in many real-world problems, such as financial markets, weather patterns, and biological systems. The concept of the Markov Property is used to construct and understand these chains [13]. The importance of Markov chains is on the probability of transitioning between states [14].

Markov Chain Monte Carlo is an alternative to Monte Carlo sampling methods. It is more reliable than the latter in some cases. It can be employed for the random sampling of a probability distribution in a high-dimensional space, where the selection of the next sample is dependent on the current sample that was drawn.

Markov Chain Monte Carlo (MCMC) algorithms work by defining a Markov chain with the required distribution as its stable-state distribution. The Metropolis-Hastings algorithm is a commonly used MCMC algorithm that uses a proposition and an acceptance-rejection step. The Ehrenfest urn model is another simple example of a discrete Markov chain that can be used to illustrate the basic principles of MCMC.

In the Ehrenfest urn model, we assume that there are two urns, each containing a predetermined number of balls. At each incremental time step, a ball is chosen at random and placed in the other urn. This process can be modeled as a Markov chain, where the state of the system can be given by the number of balls in each urn. The equilibrium distribution in this case will be the binomial distribution.

The Metropolis algorithm is a modification of simpler random walk algorithm which is used in the Ehrenfest urn model. In the Metropolis algorithm, a proposal distribution is used to generate a new state of the system. The suggested distribution used in the sampling process can be any easily sampled distribution, such as a Gaussian distribution. The acceptance-rejection step is then used to determine whether to accept or reject the proposed state depending on the ratio of the probabilities of the proposed state and the current state. This ratio is known as the acceptance probability.

The ergodic theorem and strong law of large numbers guarantee the convergence of MCMC algorithms to their equilibrium distribution. These are fundamental tools to prove the convergence of MCMC algorithms to their equilibrium distribution.

The Ergodic theorem states that, under some assumptions, a stationary and irreducible Markov chain will spend most of its time in the regions where density is high. In other words, the Markov chain will converge to its stable-state distribution as the number of iterations tend to infinity.

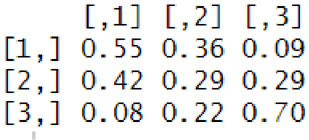
On the other hand, the Strong Law of Large Numbers ensures that the MCMC algorithm's empirical distribution will almost certainly converge to the true distribution if a sufficiently large number of iterations are performed.

There are some underlying conditions for ensuring the convergence of MCMC algorithms to their equilibrium distribution, such as irreducibility, aperiodicity, and positive recurrence of the Markov chain. Irreducibility ensures that the Markov chain will be able to reach any state in the state space in a finite number of steps. Aperiodicity condition guarantees that any state can be achieved by the chain at any point in time. The positive recurrence condition ensures that the chain will eventually return to high-density regions.

The convergence of MCMC algorithms has been extensively studied and proven in the literature. One of the most important references is the book "Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference" by Gamerman and Lopes (2006), which provides a comprehensive introduction to MCMC methods and their convergence properties. Another important reference is the article "Markov chain Monte Carlo convergence diagnostics: a comparative review" by Brooks and Gelman (1998), which presents a thorough review of the various methods to diagnose and assess the convergence of MCMC algorithms.

2.3 Solution - MCMC

Firstly, define the matrix containing transition probabilities with three possible states (1,2,3).



```
     [,1] [,2] [,3]
[1,] 0.55 0.36 0.09
[2,] 0.42 0.29 0.29
[3,] 0.08 0.22 0.70
```

This choice of transition probabilities ensures the following properties of the resulting Markov chain:

- Irreducibility: An irreducible Markov chain is the one in which every state in the state space can be reached from every other state. In this case, the transition probability matrix P satisfies this condition.

- Aperiodicity: A Markov chain is aperiodic if the chain can return to any state at irregular intervals and thus not follow a cyclic pattern. In this case, the transition probability matrix P also satisfies this condition since there are no fixed periods in the chain.

Therefore, the choice of the transition probability matrix ensures that the Markov chain is irreducible and aperiodic.

The invariant distribution of a Markov chain with transition matrix P is a probability distribution π such as if π is used as the initial distribution, the distribution of Markov chain after each step remains π itself. Mathematically, this can be shown as:

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$$

where π is a row vector representing the invariant distribution and P is the transition matrix.

For calculating the invariant distribution of a Markov chain, we must solve the system of linear equations given by:

$$\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$$

we need to note the constraint that the elements of π should sum to 1.

We can programmatically solve this equation using the eigen decomposition method. Firstly, we need to calculate the transpose of matrix P. Then, find the eigenvalues and eigenvectors of the transpose matrix using the eigen() function in R. Next, we select the eigenvector corresponding to the eigenvalue with the largest absolute value and normalize it to obtain the invariant distribution. Below is our invariant distribution:

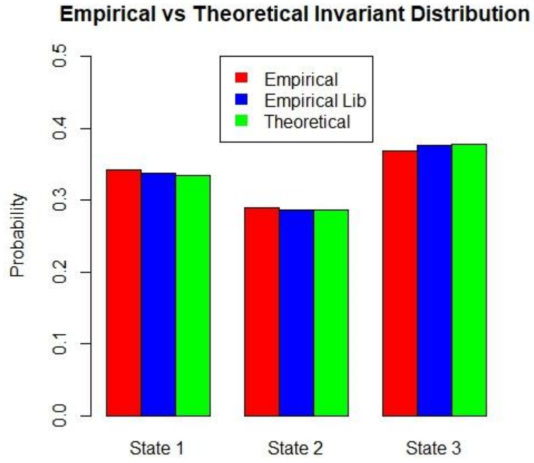
$$\boldsymbol{\pi} = [0.3350550 \ 0.2869975 \ 0.3779475]$$

Next, we simulate the Markov chain and compare the empirical distribution we obtain by Monte Carlo simulations with the theoretical invariant distribution shown above. We have used two methods to simulate. One using quantile transformation method and another by using the inbuilt r library function just for comparison. Below are the simulation results for the two:

$\boldsymbol{\pi 1}$ = [0.33342 0.28424 0.38234] – Using the quantile transformation method

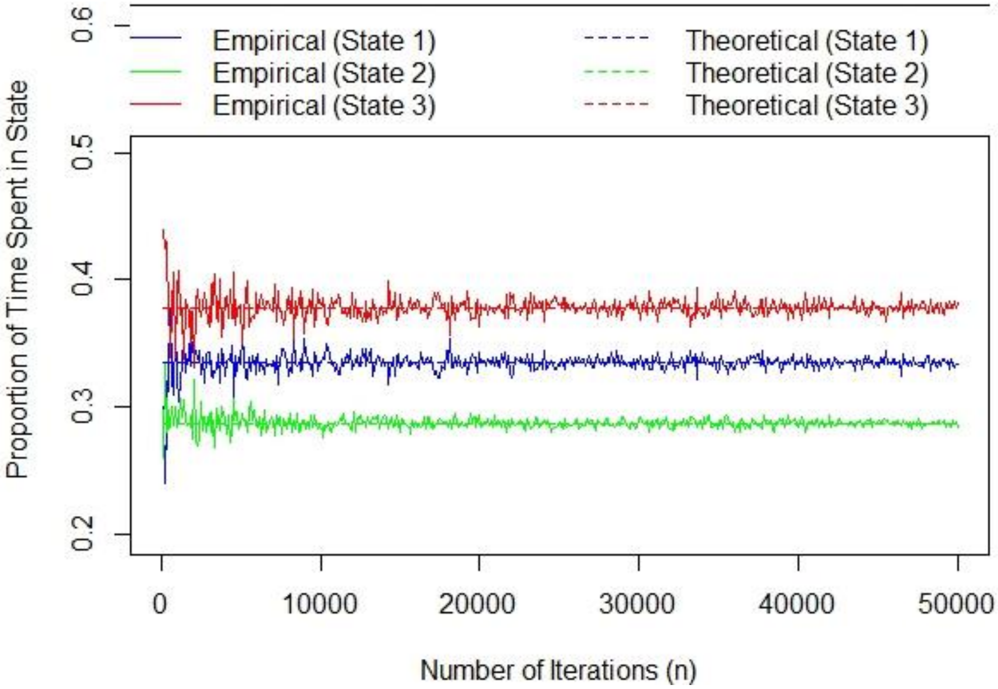
$\boldsymbol{\pi 2}$ = [0.33774 0.28567 0.37660] – Using the library function

The plot below compares the three distributions:



The plot shows that the simulated values are close to the theoretically calculated invariant distribution.

We also tried to visualize how the state varies as a function of number of steps/iterations:



We can observe here that as we increase the number of steps the simulated distribution moves closer to the theoretically computed invariant distribution. Therefore, we had chosen a large number of steps (100,000).

3. Random Walk

3.1 Introduction - Random Walk

In this section we will define a random walk using the Log-Normal distribution selected in the first section. The random walk is defined as the sum N random variables drawn from the Log-normal distribution, i.e., **ZN =** $^{N}\sum_{i=1}$ **Xi.** After that, we will calculate the cumulative distribution function for ZN theoretically and compare it with the empirically derived CDF which will be generated using Monte Carlo simulations [15].

3.2 Literature Review - Random Walk

As defined above a random walk is a stochastic process specified by summing N independent random variables. Random walks are commonly employed to simulate financial time series, and the addition of independent and identically distributed random variables is a common technique to generate such processes.

Random walk with log-normal distribution finds application in many domains such as finance and economics, for example it can be used to model stochastics properties of return in intraday trading in speculative financial products [21]. The log-normal distribution has been widely used in finance, particularly for modeling price changes, distribution of income, and firm sizes [22]. The random walk with a log-normal distribution has also been applied in the context of options pricing [23], portfolio optimization

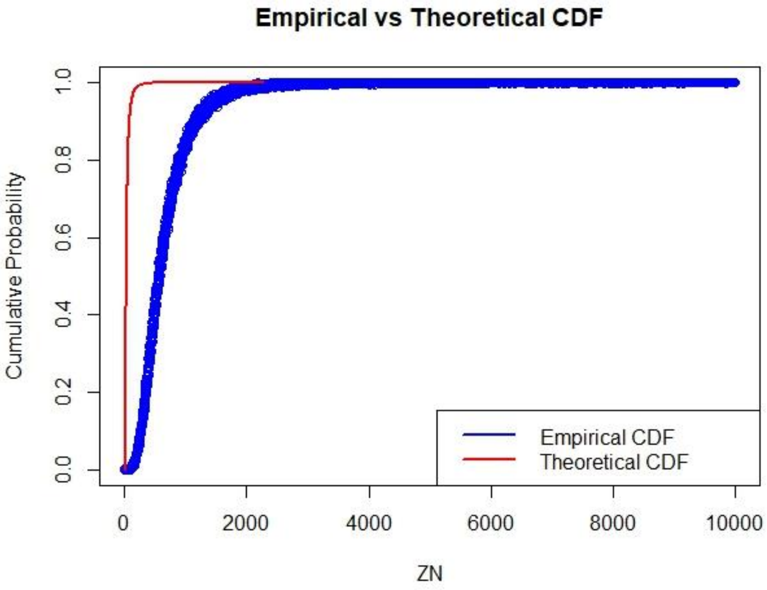
[24], and risk management [25]. In addition, the log-normal distribution has been used to model other types of data, such as the distribution of income and wealth [26].

In a nutshell, the random walk with a log-normal distribution has proven to be a versatile and powerful tool in modeling and analyzing various phenomena in finance and beyond.

3.3 Solution - Random Walk

Firstly, we define a function to generate lognormal distribution using Box-Muller method. Subsequently, we sample n random variables, ZN, which is a sum of N random variables sampled from a lognormal distribution. Next, we calculate the CDF theoretically using the built-in dlnorm() function.

Next step involves a monte-carlo program that samples random variables from the above distribution ZN. This is our empirically generated CDF. Now we can compare the theoretical and empirical CDF using plot.



It can be observed that the two distributions seems to be similar, however there are some notable differences.

Finally, we conclude by testing the normality of the below variable using Jarque-Bera test:

$$U_N = \frac{Z_N - N\mathbb{E}(X)}{\sqrt{N\mathrm{Var}(X)}}$$

Where, E(X) = exp(mean_dist + 0.5*sd_dist^2)

V(X) = sqrt((exp(sd_dist^2)-1)*exp(2*mean_dist+sd_dist^2))

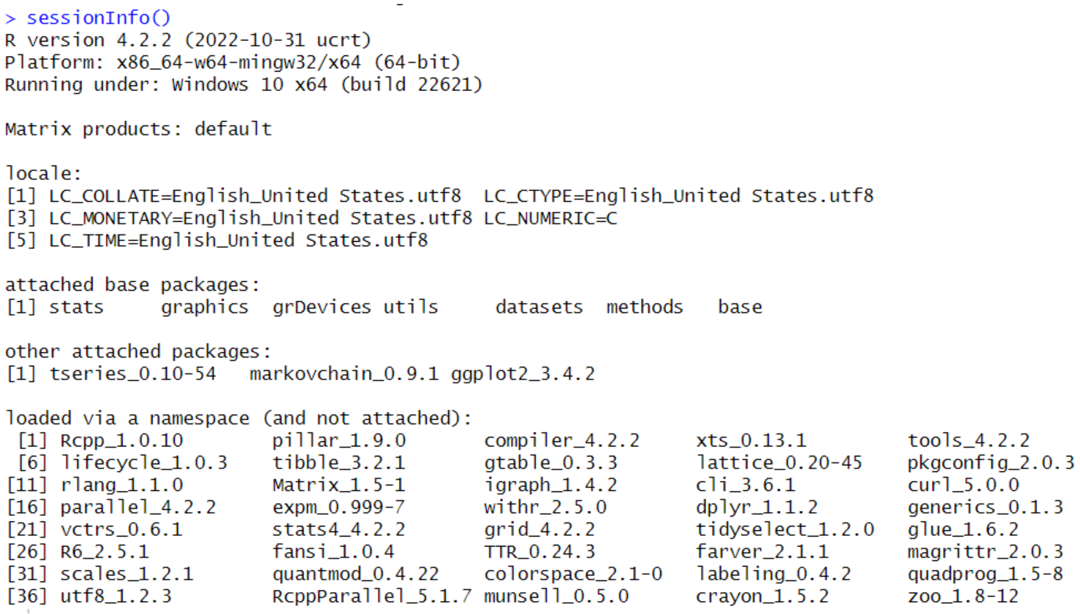
mean_dist is the mean of the log of the lognormally distributed variable X

sd_dist is the standard deviation of the log of the lognormally distributed variable X

The null hypothesis of the Jarque-Bera test is that the data is normally distributed.

We get a p-value of 0.6599 hence, we fail to reject the null hypothesis and conclude that U_N is normally distributed.

4. R Version Information



```
> sessionInfo()
R version 4.2.2 (2022-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 22621)

Matrix products: default

locale:
[1] LC_COLLATE=English_United States.utf8  LC_CTYPE=English_United States.utf8
[3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
[5] LC_TIME=English_United States.utf8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] tseries_0.10-54   markovchain_0.9.1 ggplot2_3.4.2

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.10        pillar_1.9.0       compiler_4.2.2     xts_0.13.1         tools_4.2.2
 [6] lifecycle_1.0.3    tibble_3.2.1       gtable_0.3.3       lattice_0.20-45    pkgconfig_2.0.3
[11] rlang_1.1.0        Matrix_1.5-1       igraph_1.4.2       cli_3.6.1          curl_5.0.0
[16] parallel_4.2.2     expm_0.999-7       withr_2.5.0        dplyr_1.1.2        generics_0.1.3
[21] vctrs_0.6.1        stats4_4.2.2       grid_4.2.2         tidyselect_1.2.0   glue_1.6.2
[26] R6_2.5.1           fansi_1.0.4        TTR_0.24.3         farver_2.1.1       magrittr_2.0.3
[31] scales_1.2.1       quantmod_0.4.22    colorspace_2.1-0   labeling_0.4.2     quadprog_1.5-8
[36] utf8_1.2.3         RcppParallel_5.1.7 munsell_0.5.0      crayon_1.5.2       zoo_1.8-12
```

5. References

1) Devroye, L. (1986). Non-uniform random variate generation. Springer.
2) Johnson, N. L., Kotz, S., & Balakrishnan, N. (1994). Continuous univariate distributions (Vol. 1). Wiley. Chapter 24
3) Bickel, P. J., & Doksum, K. A. (2001). Mathematical statistics: basic ideas and selected topics (Vol. 1). CRC Press. Section 3.3.3
4) Ross, S. M. (2010). Introduction to probability models (Vol. 10). Academic press. Chapter 5
5) Box, G. E. P., & Muller, M. E. (1958). A note on the generation of random normal deviates. Annals of Mathematical Statistics, 29(2), 610-611.
6) Knuth, D. E. (1997). The art of computer programming: Seminumerical algorithms (Vol. 2). Addison-Wesley. Section 3.4.1 covers the Box-Muller method and its limitations.
7) Gentle, J. E. (2003). Random number generation and Monte Carlo methods. Springer. Chapter 3 provides a detailed discussion of the Box-Muller method and its extensions.
8) https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Lognormal.html
9) Lehmann, E. L., & Romano, J. P. (2005). Testing statistical hypotheses. Springer Science & Business Media. Chapter 6
10) Montgomery, D. C., & Runger, G. C. (2018). Applied statistics and probability for engineers. John Wiley & Sons. Chapter 15
11) https://en.wikipedia.org/wiki/Log-normal_distribution
12) Massey Jr, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. Journal of the American Statistical Association, 46(253), 68-78.
13) https://towardsdatascience.com/markov-chains-simply-explained-dc77836b47e3
14) https://study.com/learn/lesson/markov-chain-example-applications.html

15) Robert, C. P., & Casella, G. (2010). Monte Carlo statistical methods. Springer Science & Business Media.
16) Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). Markov chain Monte Carlo in practice. Chapman and Hall/CRC.
17) Gamerman, D., & Lopes, H. F. (2006). Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference. Chapman and Hall/CRC
18) Brooks, S. P., & Gelman, A. (1998). Markov chain Monte Carlo convergence diagnostics: a comparative review. Journal of computational and graphical statistics, 7(4), 434-455
19) https://www.statslab.cam.ac.uk/~james/Markov/s110.pdf
20) https://stats.libretexts.org/Bookshelves/Computing_and_Modeling/RTG%3A_Simulating_High_Dimensional_Data/The_Monte_Carlo_Simulation_Method#:~:text=The%20Law%20of%20Large%20Numbers,be%20close%20to%20their%20mean.&text=The%20strong%20law%20of%20large,to%20the%20expected%20value%E2%80%8B.
21) Barndorff-Nielsen, O. E., & Shephard, N. (2004). Econometric analysis of realized volatility and its use in estimating stochastic volatility models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 65(2), 253-280. doi: 10.1111/j.1467-9868.2003.05014.x
22) Mandelbrot, B. (1963). The variation of certain speculative prices. The Journal of Business, 36(4), 394-419. doi: 10.1086/294632
23) Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. Journal of Political Economy, 81(3), 637-654. doi: 10.1086/260062
24) Markowitz, H. M. (1952). Portfolio selection. The Journal of Finance, 7(1), 77-91. doi: 10.1111/j.1540-6261.1952.tb01525.x
25) Alexander, C. (2008). Market risk analysis: Practical financial econometrics. John Wiley & Sons.
26) Reed, W. J., & Jorgensen, M. (2004). The double Pareto-lognormal distribution—a new parametric model for size distributions. Communications in Statistics-Theory and Methods, 33(8), 1733-1753. doi: 10.1081/STA-120037533
27) ProQuest Ebook Central - Reader
28) What is Monte Carlo Simulation? | IBM
29) 🔗 📑 Introduction to Markov Chain w/ R 🔎 | Kaggle
30) Log-normal Distribution | Brilliant Math & Science Wiki
31) R: The Log Normal Distribution
32) https://canvas.sussex.ac.uk/courses/22420/files/3587629?module_item_id=1179819
33) https://canvas.sussex.ac.uk/courses/22420/files/3587631?module_item_id=1179820