DOI: https://doi.org/10.46354/i3m.2019.mas.020

NONPARAMETRIC FREQUENCY POLYGON ESTIMATION FOR MODELING INPUT DATA

Stephen Hague[(a)], Simaan AbouRizk[(b)]

[(a),(b)]Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Canada

[(a)]steve.hague@ualberta.ca, [(b)]abourizk@ualberta.ca

ABSTRACT

To construct valid probability distributions solely from input data, this paper compares three nonparametric density estimators: (1) histograms, (2) Kernel Density Estimation, and (3) Frequency Polygon Estimation. A pseudocode is implemented, a practical example is illustrated, and the *Simphony.NET* simulation environment is used to fit the nonparametric frequency polygon to a set of data to recreate it as a posterior distribution via the Metropolis-Hastings algorithm.

Keywords: nonparametric density estimation, input modeling, frequency polygon

1. INTRODUCTION

Input modeling in simulation studies can be divided into two broad approaches. A classic approach, whereby standard statistical distributions are used to model underlying input data using a standard approach of:

1) Selecting one of the standard statistical distributions,
2) Parameterizing the distribution, (e.g., using the method of moments or the method of maximum likelihood),
3) Examining the goodness-of-fit (e.g., using a standardized test such as the Kolmogorov-Smirnov or Chi Square tests), and
4) Repeating as necessary until an acceptable fit is found.

Another approach for input modeling is to use nonparametric modeling techniques to construct valid probability distributions directly by defining a probability density function (PDF) and cumulative distribution function (CDF) solely from input data.

The advantages of the classical approach are numerous:

- A wide array of probability distributions exists.
- Efficient algorithms for the evaluation of the PDF and CDF, as well as the generation of random deviates, are readily available for a variety of platforms.
- Storage requirements are minimal as, once fit to the data, only parameters are required for simulation.
- Many random processes are known to follow certain distributions.

While these advantages have resulted in the widespread use of the classical approach for input modeling, certain datasets (e.g., multi-modal, Monte Carlo simulation outputs and Markov chain Monte Carlo (MCMC) algorithm outputs) are not well-suited for this approach. Multimodal data sets are a good example of what happens when classical approaches become limited.

Consider the example adapted from Scott (1985), in which 400 samples are generated from the bimodal mixture density

$$\frac{3}{4}Normal(0.00, 1.00) + \frac{1}{4}Normal(1.75, 0.25) \quad (1)$$

where 75% of the samples are generated from a normal distribution with a mean of 0 and a standard deviation of 1, and the remaining 25% of the samples are generated from a normal distribution with a mean of 1.75 and a standard deviation of 0.25. A histogram and CDF of the samples versus the theoretical distribution are illustrated in Figure 1 (*top left and right, respectively*).

The bimodal nature of the distribution renders the fitting of a standard probability distribution difficult in practice. Indeed, using the method of maximum likelihood, a triangular distribution with a low value of −2.9769, a high value of 2.4815, and a most likely value of 1.7790 was selected; see Figure 1 (*bottom left and right*).

Despite the triangular distribution being selected as the best fit, the fit was rejected when tested using the Kolmogorov-Smirnov goodness-of-fit test (Table 1), demonstrating that the classical input modeling approach is not well-suited for modeling this dataset.

Nonparametric input models are not limited in form the same way that classic statistical distributions are. The main limitations are related to 1) sampling during simulation studies from distributions that have been identified as nonparametric, and 2) using nonparametric models when data used for characterizing the input model are not sufficient to properly identify the underlying distribution of a given phenomenon (e.g. when limited data are collected to model machine breakdowns and have been known to follow an exponential distribution, but do not reveal such).

In this paper we discuss how to address the first limitation.

2. NONPARAMETRIC DENSITY ESTIMATORS

Nonparametric density estimators construct a density function directly from a given set of samples $\{x_1, \cdots, x_n\}$. Three commonly applied methods are compared here, namely: the histogram, kernel density estimation, and the frequency polygon.

2.1. Histograms

The most basic nonparametric estimator is the histogram, which is constructed by choosing an origin, x_0, and a bin width of $h > 0$. The bins of the histogram are the intervals

$$[x_o + mh, x_0 + (m+1)h) \quad m \in Z \tag{2}$$

and the histogram itself is defined by

$$\hat{f}(x) = \frac{1}{nh}(number\ of\ x_i\ in\ same\ bin\ as\ x) \tag{3}$$

(Silverman 1986). The choice of bin width h is subjective, greatly affecting the usefulness of the resulting histogram. Readers are referred to Scott (1979) for a detailed discussion of the challenges associated with the use of histograms.

2.2. Kernel Density Estimation

A well-known improvement of the histogram is the kernel density estimate, based on the concept of Rosenblatt (1956) and Parzen (1962). Intuitively, the kernel density estimate surrounds each data point in a sample with a small "bump" of density. The density estimate is the sum of these "bumps."

The kernel density estimator is defined as

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{4}$$

where $h > 0$ is a smoothing parameter—known as the bandwidth—and K is a non-negative function—known as the kernel—that integrates to one.

The bandwidth, h, is typically chosen to be as small as the data allow. The kernel is commonly represented by the density function of a probability distribution, including any one of the following:

- Uniform(-1, 1);
- Triangular(-1, 0, 1);
- Beta(2, 2, -1, 1), also referred to as Epanechnikov or parabolic; or
- Normal(0, 1).

Using a bandwidth of h = 0.12 and Normal(0, 1) as the kernel, the kernel density estimator for the data sampled from Equation 1 and the corresponding CDF are illustrated in Figure 2 (*top left and right, respectively*).

In contrast to the triangular distribution selected using the classical approach, the CDF generated using the nonparametric kernel density approach resulted in an acceptable fit (Table 1) when examined using the Kolmogorov-Smirnov test.

In cases where the kernel is a probability density, generation of a random deviate from the kernel density estimator is straightforward. First, a random deviate, u,

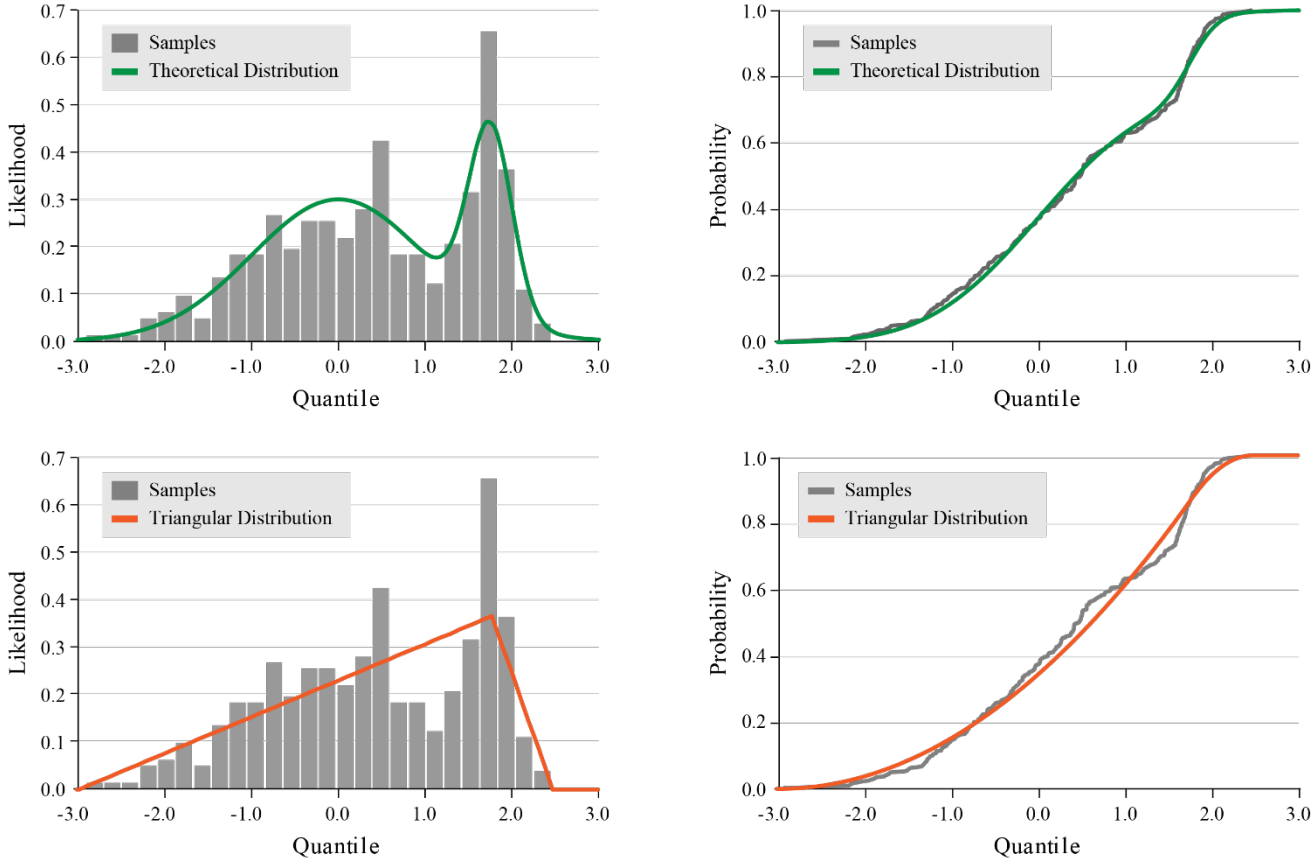

Figure 1: Histograms (*left*) and Cumulative Distribution Functions (*right*) of Generated Samples versus Theoretical Distribution (*top*) and Triangular Distribution (*bottom*).

is generated from the kernel. Next, an element x_j is randomly chosen from the sample set $\{x_1, \cdots, x_n\}$. Then

$$hu + x_j \tag{5}$$

is a random deviate from the kernel density estimator.

2.2.1. Limitations of Kernel Density Estimation

Although kernel density estimation is available in a variety of software packages, including *R* (R Core Team 2019) and MATLAB (MathWorks 2019), its functionality is limited in conditions where n is large because (i) evaluation of the PDF and CDF becomes computationally intensive, as the kernel must be evaluated at every sample point, and (ii) the estimate requires considerable amounts of storage, as every sample point must be available.

2.3. Frequency Polygon Estimation

The frequency polygon can be thought of as a generalization of the triangular distribution. Its PDF is constructed "from a histogram by connecting with straight lines the mid-bin values of the histogram" (Scott 1985). Its CDF is then a piecewise quadratic function.

As with the histogram, the frequency polygon is dependent on the choice of bin width ($h > 0$). Scott (1985) notes that the bin width for an optimal frequency polygon will generally differ from that of the optimal histogram.

Using a bin width of $h = 0.21$, the frequency polygon for the data sampled from Equation 1 and the corresponding CDF are illustrated in Figure 2 (*bottom left and right, respectively*). As with the kernel density estimate, the Kolmogorov-Smirnov test determined that the CDF generated using the frequency polygon resulted in an acceptable fit (Table 1).

Table 1: Results of the Kolmogorov-Smirnov Test

	Test Statistic	Fit*
Triangular Distribution	0.07501	Rejected
Kernel Density Estimator	0.02861	Accepted
Frequency Polygon	0.02353	Accepted

*At significance α = 0.05, with critical value = 0.06791.

Once constructed, computation of the frequency polygon does not require the availability of the original data points, rendering it efficient even in conditions when n is large.

3. PSEUDO CODE IMPLEMENTATION

For the pseudocode conventions used herein, see Cormen et al. (2009). From an implementation perspective, the frequency polygon consists of three

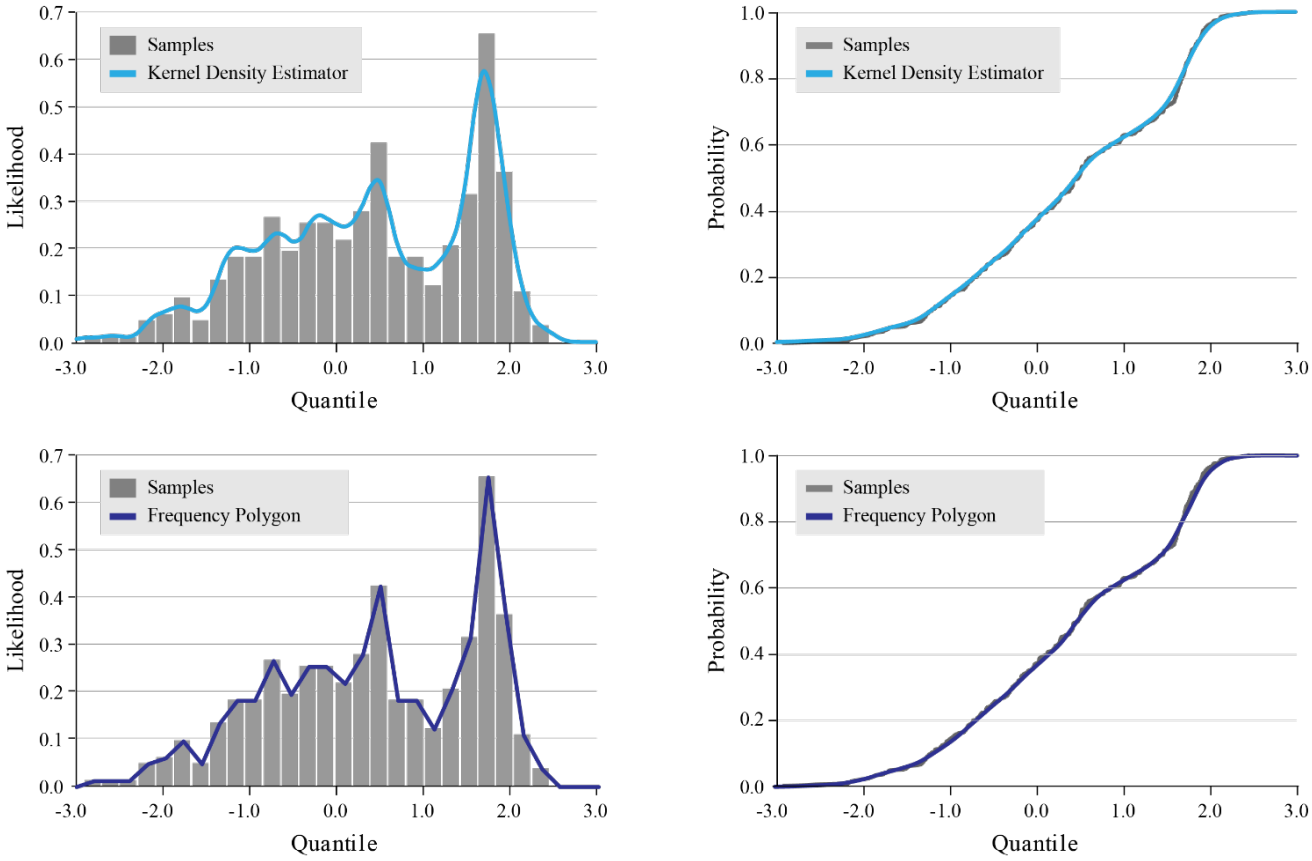

Figure 2: Histograms (*left*) and Cumulative Distribution Functions (*right*) of Generated Samples versus the Kernel Density Estimator (*top*) and Frequency Polygon (*bottom*).

arrays:

$$X[0 \cdots k+1]$$
$$L[0 \cdots k+1] \quad (6)$$
$$C[0 \cdots k+1]$$

where k is the number of bins in the original histogram. The array X is defined by

$$X[i] = \text{mid-point of bin } i \qquad 0 \le i \le k+1 \quad (7)$$

the array L (the PDF values corresponding to the entries in X) is defined by

$$L[i] = \begin{cases} 0 & \text{if } i = 0 \\ \text{mid-point value of bin } i & \text{if } 1 \le i \le k \\ 0 & \text{if } i = k+1 \end{cases} \quad (8)$$

and the array C (the CDF values corresponding to the entries in X) is defined recursively by setting $C[0] = 0$ and

$$C[i] = C[i-1] + \frac{h}{2}(L[i-1] + L[i]) \quad (9)$$

for $i > 0$, where h is the bin width of the original histogram.

Note that for persistence purposes, only arrays X and L need to be stored, as C can be reconstructed if necessary.

3.1. Binary Search Procedure

The existence of a procedure, termed BINARY-SEARCH(x, T, p, r), is then assumed. This procedure takes a key, x, together with a sorted subarray, $T[p \cdots r]$, and returns one of the following:

- If $T[p \cdots r]$ is empty $(r < p)$, then the index p is returned.
- If $x \le T[p]$ and, therefore, less than or equal to all of the elements of $T[p \cdots r]$, then the index p is returned.
- If $x > T[p]$, then the largest index q in the range $p < q \le r+1$ is returned, such that $T[q-1] < x$.

Example pseudocode for a binary search procedure is detailed in Cormen et al. (2009).

3.2. Initialize Procedure

In this procedure, the data are first binned, and the values for the three arrays are then constructed. Specifically, the INITIALIZE procedure initializes the X, L, and C arrays given the array of data T, to which the polygon is to be fit.

```
INITIALIZE(T, X, L, C)
 1 determine the number of bins (k)
 2 initialize X[0 ⋯ k+1] as a new array
 3 initialize L[0 ⋯ k+1] as a new array
 4 initialize C[0 ⋯ k+1] as a new array
 5 min = min(T)
 6 max = max(T)
 7 h = (max − min)/k
 8 for i = 0 to T.length − 1 // bin the data
 9     j = min(⌊(T[i] − min)/h⌋ + 1, k)
10     L[j] = L[j] + 1
11 for i = 0 to k + 1 // construct the arrays
12     X[i] = min + h(i − 0.5)
13     L[i] = L[i]/T.length/h
14     if i > 0
15         C[i] = C[i − 1] + h(L[i − 1] + L[i])/2
```

3.3. Slope Procedure

The SLOPE procedure is a helper method that calculates the slope of the PDF line segment between $X[i]$ and $X[i+1]$.

```
SLOPE(X, L, i)
 1 return (L[i + 1] − L[i])/(X [i + 1] − X [i])
```

3.4. Probability-Density Procedure

The PROBABILITY-DENSITY procedure evaluates the PDF of the frequency polygon for real argument, x. It begins by determining the largest index, i, such that $X[i] < x$, and then uses linear interpolation to calculate the value of the PDF at x.

```
PROBABILITY-DENSITY(X, L, x)
 1 if x ≤ X[0] or X[X.length − 1] ≤ x
 2     return 0
 3 else
 4     i = BINARY-SEARCH(x, X, 0, X.length − 1) − 1
 5     m = SLOPE(X, L, i)
 6     h = x − X[i]
 7     return L[i] + mh
```

3.5. Cumulative-Distribution Procedure

This procedure evaluates the CDF of the frequency polygon for real argument, x. It begins by determining the largest index, i, such that $X[i] < x$. The procedure then adds the area of the trapezoid formed by the PDF between $X[i]$ and x to the cached CDF value $C[i]$.

```
CUMULATIVE-DISTRIBUTION(X, L, C, x)
 1 if x ≤ X[0]
 2     return 0
 3 elseif X[X.length − 1] ≤ x
 4     return 1
 5 else
 6     i = BINARY-SEARCH(x, X, 0, X.length − 1) − 1
 7     m = SLOPE(X, L, i)
 8     h = x − X[i]
 9     y = L[i] + mh // value of PDF at x
10     return C[i] + h(L[i] + y)/2
```

3.6. Quantile Procedure

The QUANTILE procedure evaluates the inverse CDF (i.e., quantile function) of the frequency polygon for real argument y. It begins by determining the largest

index, i, such that $C[i] < y$. If the slope of the PDF line segment between $X[i]$ and $X[i+1]$ is non-zero, then the quantile is the solution to a quadratic equation. Otherwise, linear interpolation is used to calculate the quantile.

QUANTILE(X, L, C, y)
1 **if** $y < 0$ **or** $1 < y$
2 **return** NaN
3 **elseif** $y = 0$
4 **return** $X[0]$
5 **elseif** $y = 1$
6 **return** $X[X.length - 1]$
7 **else**
8 $i =$ BINARY-SEARCH$(y, C, 0, C.length - 1) - 1$
9 $m =$ SLOPE(X, L, i)
10 $h = y - C[i]$
11 **if** $m \neq 0$ // solve quadratic
12 **return** $X[i] + (\sqrt{(L[i]^2 + 2mh)} - L[i])/m$
13 **else** // linear interpolation
14 $m = (X[i+1] - X[i])/(C[i+1] - C[i])$
15 **return** $X[i] + mh$

3.7. Sample Procedure

Using the inverse transform method, this procedure generates a random deviate from the frequency polygon.

SAMPLE(X, L, C)
1 generate random number $y \in [0,1]$
2 **return** QUANTILE(X, L, C, y)

4. PRACTICAL EXAMPLE

As discussed previously, the frequency polygon is best suited for density estimation of large data sets, such as those acquired from automated sensors, the outputs of Monte Carlo simulations (e.g., risk analyses), or outputs of Markov chain Monte Carlo (MCMC) algorithms (e.g., posterior distributions generated using Bayesian statistics).

To demonstrate the functionality of the proposed approach, the frequency polygon method was applied to generate the CDF of the outputs of a MCMC algorithm obtained by Ji and AbouRizk (2017). In their study, Ji and AbouRizk (2017) modeled the number of nonconforming (i.e., failed) welds in a pipe weld inspection process using a binomial distribution $B(n, p)$, where n was the sample size, and p was the probability of nonconformance (i.e., weld failure). The prior distribution $P(p)$ of parameter p was modeled as Beta(0.5, 0.5) and, after observing D nonconforming welds in n inspections, the posterior distribution, $P(p|X)$, was determined to be

$$Beta(D + 0.5, n - D + 0.5) \tag{10}$$

In particular, if $n = 100$ and $D = 10$, the posterior distribution is

$$Beta(10.5, 90.5) \tag{11}$$

It is important to note that, in this particular case, a closed-form (i.e., analytical) solution for the posterior distribution exists; however, a closed-form solution is often difficult to derive or does not exist for many posterior distributions. Therefore, for the purposes of demonstrating the functionality of the frequency polygon approach, a numerical solution was instead determined using the Metropolis-Hastings algorithm—a common MCMC method (Metropolis et al. 1953, Hastings 1970)—together with the frequency polygon. From Bayes' Theorem, the posterior distribution is

$$P(X) = \frac{L(X|p)P(p)}{P(X)} \tag{12}$$

where $L(X|p)$ is the likelihood function. As $P(X)$ is independent of p,

$$P(X) \propto L(X|p)P(p) \tag{13}$$

The Metropolis-Hasting algorithm is then used to generate random samples from the probability distribution when a function is proportional to its PDF. For the distribution denoted in Equation (12),

$$L(p) = p^D(1-p)^{n-D} = p^{10}(1-p)^{90} \tag{14}$$

and $P(p)$ is the PDF of $Beta(0.5, 0.5)$, therefore

$$P(X) \propto p^{10}(1-p)^{90}\frac{p^{0.5}(1-p)^{0.5}}{Beta(0.5, 0.5)} \tag{15}$$

Using Equation 17 and a starting p value of 0.1, the Metropolis-Hastings algorithm was used to generate 10,000 samples from the posterior distribution. A histogram consisting of 42 bins was constructed from the samples, and a frequency polygon was generated in *Simphony.NET* (AbouRizk et al. 2016) using the pseudo code approach detailed above. The resulting histogram and CDF are illustrated in Figures 3 and 4, respectively.

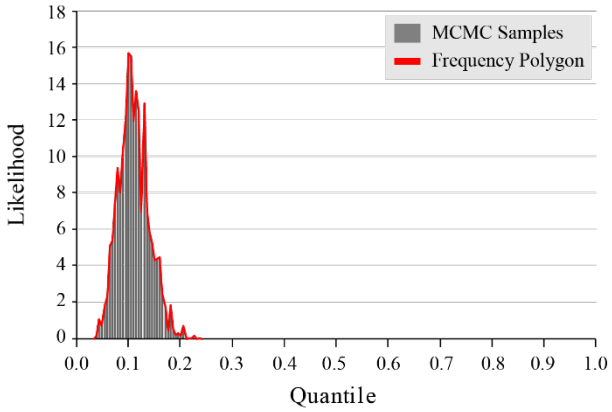

Figure 3: Histogram of Samples Generated using an MCMC-Based Method versus the Frequency Polygon

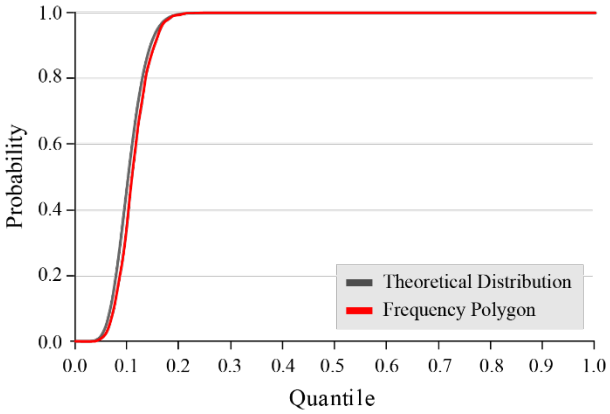

Figure 4: Cumulative Distribution Function of Theoretical Distribution versus the Frequency Polygon

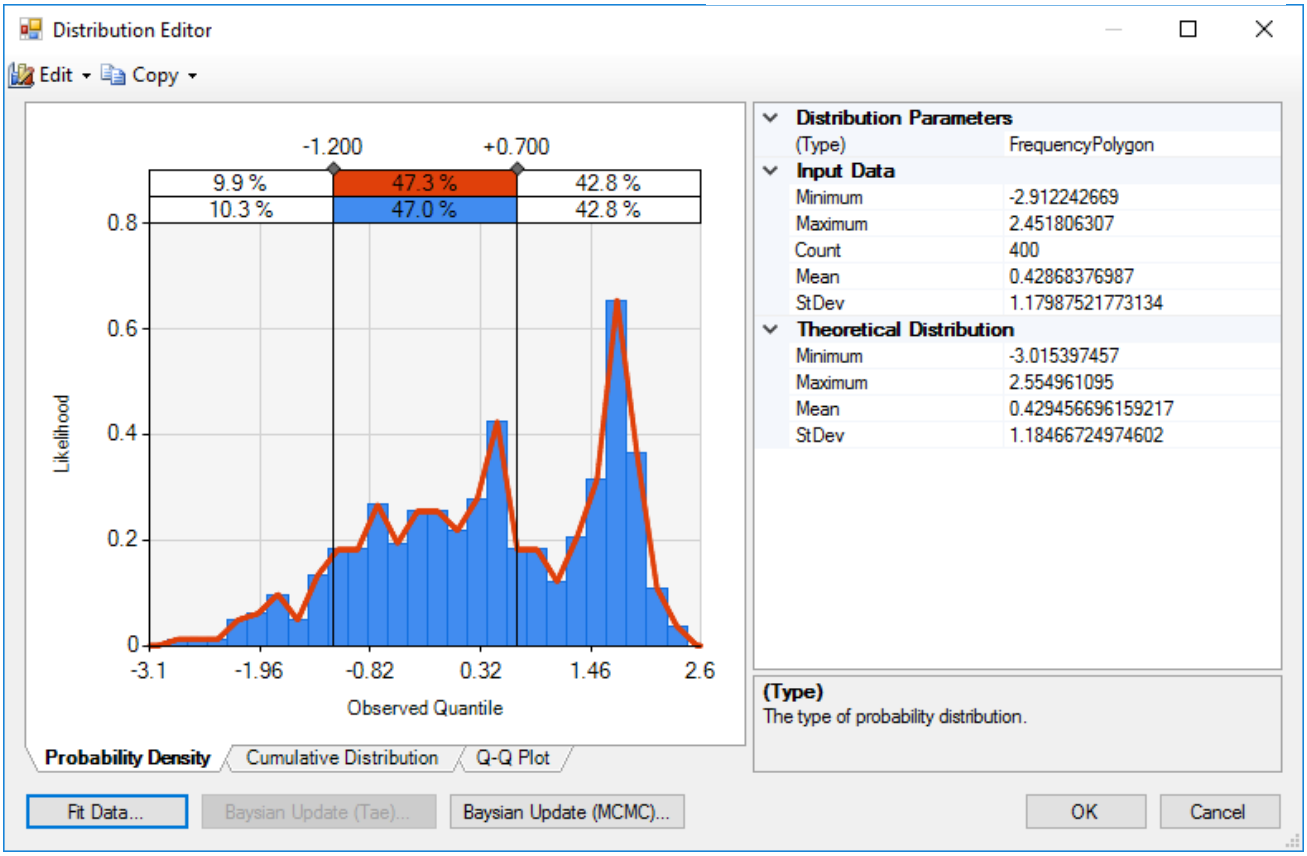

Figure 5: Output Results of the Implementation of the Frequency Polygon Pseudocode Approach in *Simphony.NET*.

5. IMPLEMENTATION IN *SIMPHONY.NET*

The frequency polygon has been implemented within the *Simphony.NET* simulation environment (AbouRizk et al. 2016). *Simphony.NET* is a modeling environment comprised of simulation services and a modeling user interface. Based on modular and hierarchical concepts, *Simphony.NET* provides a medium for developing and deploying simulation modeling templates that are general purpose by design, while featuring a number of special purposes templates.

The *Simphony.NET* environment now supports the fitting a frequency polygon to a set of data and the creation of a frequency polygon as a posterior distribution via the Metropolis-Hastings algorithm. A screenshot of the output results of the frequency polygon approach in *Simphony.NET* are illustrated in Figure 5.

6. CONCLUSIONS

The paper demonstrates how to implement nonparametric input modeling techniques to augment classic methods during simulation studies. The approach covers defining a probability density function

from random data and estimating both the cumulative density function and the inverse distribution. The latter facilitates sampling during simulation. The implementation was summarized in pseudo code to facilitate its use by others within their own systems.

ACKNOWLEDGMENTS

This project was supported by a Collaborative Research and Development Grant (CRDPJ 492657) from the Natural Sciences and Engineering Council of Canada.

REFERENCES

AbouRizk S., Hague S., Ekyalimpa R., Newstead S., 2016. Simphony: A next generation simulation modelling environment for the construction domain. Journal of Simulation, 10(3), 207–215.

Cormen T.H., Leiserson C.E., Rivest R.L., Stein C., 2009. Introduction to algorithms. 3rd ed. Cambridge, MA: The MIT Press.

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

Ji W., AbouRizk S., 2017. Credible interval estimation for fraction nonconforming: analytical and numerical solutions. Automation in Construction, 83(2017), 56–67.

MathWorks, Inc., 2019. MATLAB. Available from: https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab [Accessed 03 April 2019].

Metropolis N., Rosenbluth A.W., Rosenbluth M.N., Teller A.H., 1953. Equation of state calculations by fast computing machines. The Journal of Chemical Physics, 21(6), 1087–1092.

Parzen E., 1962. On estimation of a probability density function and mode. The Annals of Mathematical Statistics 33(3), 1065–1076.

R Core Team, 2019. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Available from: https://www.R-project.org [Accessed 03 April 2019].

Rosenblatt, M., 1956. Remarks on some nonparametric estimates of a density function. The Annals of Mathematical Statistics, 27(3), 832–837.

Scott D.W., 1979. On optimal and data-based histograms. Biometrika, 66(3), 605–610.

Scott D.W., 1985. Frequency polygons: Theory and application. Journal of the American Statistical Association, 80(390), 348–354.

Silverman B.W., 1986. Density estimation for statistics and data analysis. London, UK: Chapman & Hall.

AUTHOR BIOGRAPHIES

Simaan AbouRizk, PhD, PEng, FRSC, FCAE, NAC is a Distinguished University Professor, Tier 1 Canada Research Chair in Operations Simulation, and Chair of the Department of Civil and Environmental Engineering at the University of Alberta. As a renowned expert in the development and application of computer simulation for construction planning, productivity improvement, constructability review, and risk analysis, AbouRizk's research focuses on developing innovative information technologies for modeling, analyzing, and optimizing operations in the construction industry. His contributions to academia and industry have been recognized by numerous organizations including the American Society of Civil Engineers and the Canadian Society for Civil Engineering.

Stephen Hague is a System Analyst in the Department of Civil and Environmental Engineering at the University of Alberta. Steve has been instrumental in the development of numerous simulation and software tools that have been implemented throughout the industrial construction sector in Alberta, Canada. He has also played an integral role in the development of *Simphony.NET*, which continues to be extended at the University of Alberta through Dr. AbouRizk's research program.