Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

ANSWERS TO THE TOP TEN INPUT MODELING QUESTIONS

Bahar Biller

Dept. of Manufacturing & Operations Management
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A

Barry L. Nelson

Department of Industrial Engineering
& Management Sciences
Northwestern University
Evanston, IL 60208-3119, U.S.A.

ABSTRACT

In this tutorial we provide answers to the top ten input-modeling questions that new simulation users ask, point out common mistakes that occur and give relevant references. We assume that commercial input-modeling software will be used when possible, and only suggest non-commercial options when there is little else available. Detailed examples will be provided in the tutorial presentation.

1 WHY USE INPUT MODELS AT ALL?

This question could be rephrased as, "Why do stochastic simulation?" The premise behind stochastic simulation—simulation that includes randomness—is that the uncertainty in the system cannot be wished away without painting an unrealistic picture of system performance. Input models represent the uncertainty. For the purpose of this tutorial, "input modeling" will mean selecting and fitting a probability distribution (perhaps multivariate) to represent some process whose behavior cannot be predicted with certainty.

For example, suppose you are a supplier of a component that is supposed to last for one year, a component that you know has a mean time to failure of 2 years. A client is willing to pay \$1000 for your component, but wants you to pay a penalty of \$5000 if failure occurs in less than one year. Should you take this contract?

If you ignore the uncertainty in the component's life time and base your decision on the average two-year life, then this is a no-brainer: You will pocket \$1000 for each component you sell. On the other hand, if you know that the distribution of time to failure is well modeled as being exponentially distributed (an input model) with mean 2 years, then it turns out that you can expect to *lose* about \$967 on each component you sell. In this simple example the expected or long-run average profit can be computed so you do not need simulation to estimate it, but the same point applies to simulation experiments: You cannot just plug in mean values for all the uncertainties and expect to discover the mean performance of the system itself.

2 DOES THE PARTICULAR INPUT MODEL MATTER?

Absolutely! Simply injecting *some* uncertainty into the simulation is not enough. The simulation outputs can be quite sensitive to the particular input model chosen, and matching the mean alone is rarely sufficient.

For instance, in the reliability example described in the answer to Question 1, suppose you modeled the component life time as having a uniform distribution between 0 and 4 years, because this distribution has the right mean (2 years) and is easier to work with than the exponential. Under the uniform model, the expected loss on each component is \$3500, rather than \$967. So if you were trying to negotiate a different contract that was profitable, the uniform model would cause you to overprice the component (and lose business to a competitor who has better input models).

Input modeling error is particularly nasty because it is very difficult to quantify. This is in contrast to the *estimation error* in the simulation output performance measures. Estimation error can be measured via a confidence interval or standard error, and reduced by making more replications or longer runs. Unfortunately, you can not simulate your way out of an inaccurate input model.

3 WHY NOT JUST REUSE THE DATA YOU HAVE?

When process data are available, then using that data to drive the simulation model can be a very good idea (we discuss about how to use it appropriately in the answer to Question 9). However, there are a number of reasons why it is often better to fit an input model. These include the following:

- To fill in gaps and smooth the data: A finite sample of data is nearly always an imperfect representation

of the process that produced it. There may be gaps in which values are possible, but none occurred in this particular sample. Or there may be collections of values that are overrepresented, just by chance. One way to think about input modeling is that you are trying to infer characteristics of the true underlying process that are not perfectly represented in the data. In fact, the physics of the process may provide a basis for choosing a particular input model, independent of the data (see Question 4).

- Insure that tail behavior is represented: This is similar to the previous point. By definition, highly unusual events do not occur very often; therefore, they may not be appropriately represented in a sample of data, particularly if the sample size is small. But these rare events often correspond to the extreme conditions (power spikes, long service times, or early equipment failure) that make systems perform badly. A simulation model that does not include the chance of extreme events will not correctly represent the risks to the system. By fitting an input model you can infer the tail behavior that may not be present in the data.
- Reflect dependencies in the inputs: For certain types of data sets, specifically those that exhibit dependence or nonstationary behavior, the data set cannot be naively resampled. Consider, for instance, an input model that represents a customer's behavior on a commercial web site. The customer may undertake a sequence of transactions, such as connecting, logging in, browsing, adding to a shopping list, more browsing, comparative pricing, reading product information, more browsing, checking out, and disconnecting. Although different customers will exhibit different behaviors, certain patterns are more likely than others, and some may even be forced to occur in sequence (one has to connect before logging in, for instance). Thus, it would be wrong to independently resample individual transactions because customers do not chose their transactions independently. In this example, you would need to resample the entire customer session instead. Unfortunately, your simulation will not see any behavior patterns that were not in the sample, a particular problem if the number of observed sessions is small.
- Incorporate changes in the input process: Suppose you are not only interested in getting a good model for an input process, but also in seeing how the system will react to changes in that input. For instance, suppose that an input to your simulation will be the time a worker requires to assemble a component. You believe that a new piece of equipment will reduce the variability in this time, although it will not speed it up. With a parametric input model (a probability distribution) you can change its parameters, or even select a new distribution, to reflect the changes. If you are reusing data then somehow you must change the data.

4 WHY ARE THERE SO MANY CHOICES?

Even a low-budget input modeling tool will have ten to twelve different distributions from which to choose. Some tools have twenty or more. One reason that there are so many choices is that distributions arise naturally when considering certain physical processes. The normal distribution is a well-known example. If the time to do some task—assemble the components of a computer, for example—is the result of adding together the times to do a large number of individual tasks (each having some variability), then the total time to complete the task may, according to the Central Limit Theorem, be approximately normally distributed. Thus, the physical nature of the process (sums of random times) leads naturally to a particular type of distribution. To take a less well-known example, consider the Weibull distribution. The Weibull can be derived by considering the minimum (think first event to occur) of a number of random variables. Because time to failure is often the time when the first of a number of possible breakdowns occurs, the Weibull arises as a natural choice in reliability modeling. The number of input model choices is large because the number of physical processes of interest is large. For descriptions of the physical basis of a number of standard distributions see Banks et al. (2001, Chapter 9).

Although the number of choices is often large, there may be fewer distinct choices than it first appears. For instance, input-modeling packages often include the gamma, Erlang and exponential distributions. However, the Erlang and exponential are special cases of the gamma (arising from restrictions on the gamma's parameters), so there is really only one choice.

A practical consequence of this nesting of distributions is that algorithms for automatically selecting input models typically select the most flexible member of a family, and not the others (e.g, gamma instead of Erlang or exponential). This makes sense because a more flexible distribution can more easily accommodate the hills and valleys present in a sample of data. To see this for yourself, try doing the following exercise: Use your simulation software to generate data sets of various sizes (100, 500, 1000, 5000) from an exponential distribution, then ask your software to find the "best fit." Frequently, the exponential will not be selected until the sample size is very large, if it is selected at all.

5 WHAT IS A "GOOD FIT?"

The direct answer is that a good fit occurs when an input model represents the key features of the real process that have a significant impact on the simulation output measures of interest. As a practical matter this definition of "good fit" is very difficult to quantify, so others have been derived.

5.1 Goodness-of-Fit Tests

Undoubtedly the most popular approach to evaluating input model fit is statistical goodness-of-fit (gof) testing. Understanding gof tests is important because they can be both useful and misleading.

The gof test starts with the premise that *there is a true input model to discover*; it then proceeds to determine whether there is substantial evidence that the model you have chosen is *not* the truth. In gof tests, the null or status quo hypothesis is that you are correct (you have found the true distribution and its parameters), and the alternative is that you are wrong. *The test will reject your choice only if there is overwhelming evidence that you are wrong.* The more data that are available, the easier it is for the test to deduce that you are wrong. This only makes sense: if you had a single data point, for instance, then who could say that any choice was incorrect?

One problem with gof tests is that you know, before you run the test, that your model choice is wrong! *You know this because real data come from real processes, not probability distributions*. Probability distributions are mathematical entities that *approximate* real processes, they are not real processes. *So if there are enough data, the test will definitely reject your distribution choice, whatever it is*. Thus, having lots of data—usually considered to be a good thing—is bad if your goal is to get your input model endorsed by a gof test. The statistical term for this is *power*: the more data there are, the more powerful the test is for detecting differences between your distribution choice and the process data. On the other hand, if you do not have much data then almost any choice will be accepted by the test.

So how should gof tests be used, if at all? We suggest they should be advisory only. If you are happy with the fit based on other factors (physical basis, graphical analysis), and the gof test fails to reject your choice, then take that as additional evidence in favor of your selection. If the gof test rejects, then you may want to more carefully examine your choice, but not necessarily give up on it. This is especially true if you have a large data set so that rejection is likely. See Law and Kelton (2000, Chapter 6) for an excellent treatment of gof testing.

We have been describing gof tests as if they provide a go/no-go decision. More typical is that the input-modeling software will present a p-value for the test. The p-value can be confusing, especially in this context, so keep this simple rule in mind: *A large p-value supports your choice of input model, and p-values greater than* 0.10 *are typically considered to be "large."*

5.2 Graphical Comparisons

A feature of all modern input-modeling software is the facility to compare a fitted distribution to data. The most intuitive graphs are based on comparing a fitted density function to a histogram of the data. Unfortunately, your perception of the fit is highly dependent on the width of the histogram cells. The fit may look good when the histogram is formed with a few, wide cells, but poor with a large number of small cells. In fact, if the number of cells is too large (imagine one cell for each data point), then no distribution will appear to fit. Thus, if you use histogram-based graphical comparisons, try different cell divisions to see how they change your perception of fit.

Although less intuitive, graphs based on the cumulative distribution function (cdf) do not require data grouping and are sensitive to lack of fit and to where the lack of fit occurs. The $q-q$ plot is a typical example of this type of graphical assessment tool and is highly recommended. See Vincent (1998) for a thorough discussion of graphical comparisons.

5.3 A Note on Parameter Estimates

Input models nearly always come with parameters that can be tuned to the data set at hand. For instance, the Poisson distribution has one parameter, its mean, while the lognormal distribution has two parameters, its mean and standard deviation (or variance). For some distributions estimating the values of their parameters is a messy numerical analysis problem. One of the nice things that input-modeling software does is parameter estimation.

When statisticians attack the parameter estimation problem they look for criteria that lead to estimators with good statistical properties. The methods of maximum likelihood, least squares and moment matching are three standard approaches. Should you be worried about what parameter-estimation methods your software implements? The answer, typically, is no. All of the standard methods have plusses and minuses. What is more important is that the software implements them correctly, using numerically stable algorithms, and provides diagnostics like gof tests and graphical comparisons. If you are interested in parameter estimation, see Banks et al. (2001, Chapter 9) and Law and Kelton (2000, Chapter 6).

6 WHY NOT JUST USE THE "BEST FIT?"

Commercial input-modeling software invariably includes a feature that will automatically select or recommend a distri-

bution that best fits the sample of data. To our knowledge these automated features only apply to models of independent and identically distributed (i.i.d.) data (see Question 7 below for what to do with dependence, and Question 8 for what to do with distributions that change over time). The following is a generic description of how these features work (details will differ from package to package):

1. Obtain information from the user that could eliminate certain candidate distributions. Examples include whether the data are discrete or continuous valued; whether there are known, unknown, or no bounds on the range of possible values; and specific candidate distributions to try.
2. Fit all feasible candidate distributions to the sample of data by estimating values for any parameters.
3. Rank all the fitted distributions by some summary measure of fit, such as the p-value of a goodness-of-fit test.
4. Recommend the distribution with the best summary measure of fit.

There is nothing inherently wrong with this approach, and it never hurts to see what the software recommends. But it is a mistake to slavishly take the recommendation for the following reasons:

- The selection is based on a summary measure of fit, and different summary measures lead to different recommendations. Which summary measure is the right one? The answer depends on characteristics of the data and on what sort of lack of fit bothers you most. Do you want to get the tails or the center of the distribution right? Are you interested in minimizing the largest discrepancy between the data and the fitted distribution or the average of all the discrepancies? Do you believe that there is indeed a "true distribution" or are you only trying to find a close approximation to the given data?
- Some measures of fit are sensitive to how your data are grouped. In particular, the popular chi-squared statistic depends on the number and size of the cells in your histogram, as described in Question 5. If you change the grouping of your data you may end up with a different recommendation.
- The software usually does not account for the physical basis of the data (see Question 4), and the physical basis may provide the best indication of the right family of distributions to choose.
- You are smarter than the software.

Our recommendation is to use every graphical tool available in the software to examine the fit, and if it is a histogram-based tool to be sure to play with different widths of the cells. If there is a strong physical basis for a particular distribution choice, then use it even if it is not the "best fit." And avoid histogram-based summary measures, if possible, when asking the software for its recommendation.

7 WHAT IF THERE IS DEPENDENCE IN THE PROCESS?

First and foremost, don't ignore it!

Here are some examples of input processes that might exhibit dependence:

1. A distributor places monthly orders for your product. Because the distributor may hold inventory (which is outside the scope of your model), a large order from the distributor one month is likely to be followed by a smaller order the following month, followed by a larger order the next month, etc. Modeling the monthly orders as independent random variables misses this month to month dependence.
2. Customers who log on to your web site have characteristics that influence their behavior, including age, sex, income level and where they live. To treat these customer characteristics as independent random variables misses the obvious relationship between age and income, for instance.
3. In the first example, suppose that the distributor has several warehouses and each places monthly orders for your product. The month-to-month dependence still exists, but there may also be dependence between the orders from different warehouses in the same month if they are able to share inventory or supply the same customers.

The first example calls for a *time series* input model, a sequence of random variables that all have the same probability distribution, but exhibit dependence. The dependence is often measured by the *autocorrelation*, which is the correlation between observations within the series.

The second example calls for a *random vector* input model, where each component of the vector—age, sex, income level and location—may be described by a different probability distribution, but the components depend on the other. This dependence is often characterized by a *correlation matrix* whose elements are the pairwise correlations between the components.

The third example calls for a *vector time series* input model that has dependence in sequence (month to month) and across components (the orders from different distributors).

All simulation software includes input models for i.i.d. processes, and all input modeling packages fit distributions to i.i.d. data. Few of the products include facilities for modeling dependent input processes. Thus, there is an almost overwhelming temptation to use i.i.d. models. Un-

fortunately, many studies have shown that ignoring dependence can greatly distort the simulation output performance measures. For instance, if there is actually positive autocorrelation between the interarrival times of customers to a queue but you ignore it, then the simulation of the queue can grossly underestimate the congestion that will actually occur. Vincent (1998) describes techniques for assessing whether or not there is dependence in a data set.

Multivariate input models based on the normal distribution—including time series, random vectors, and vector time series—are well known to statisticians and easy to fit and simulate. Recently, researchers have developed tools that transform input models with normal distributions into input models with (any) other distributions. See Nelson and Yamnitsky (1998) for an overview, and <www.iems.northwestern.edu/~nelsonb> for software.

8 WHAT IF THE PROCESS CHANGES OVER TIME?

Again, don't ignore it!

Input processes that change over time are said to be *nonstationary*. A typical example is an arrival process in which the arrival rate varies by the time of day, day of the week, etc. For instance, nonstationarity occurs in the arrival of customers to a restaurant (rate is greater around meal times), arrival of e-mail messages to a mail server (lower rate at night), and the times of discovery of bugs in a software product (rate tends to decrease over time).

The Poisson arrival process—where times between arrivals of customers are independent, exponential random variables—is a standard input model used when arrivals occur "at random" (as opposed to, say, on a schedule). The Poisson arrival process has a constant or stationary arrival rate. A generalization of the Poisson arrival process allows the arrival rate to vary with time. Such a process is called a *nonstationary* or *nonhomogeneous* Poisson arrival process. Good references are Law and Kelton (2000, Chapter 6) and Nelson and Yamnitsky (1998).

9 HOW CAN I REUSE THE DATA I HAVE?

As mentioned in the answer to Question 3, there are reasons not to reuse input data that you have collected. However, when an adequate sample is available, the data are thought to be representative and there is no compelling reason to use a probability model (including the case that nothing appears to fit well), then using the data themselves is clearly an option. The idea is to resample the data to produce inputs for the simulation.

When the data are believed to be approximately i.i.d., then they should be sampled, with replacement, in such a way all the data points are equally likely. This is known as using the *empirical cdf* and it has good statistical properties. However, if you believe that values between the observed data points are possible, then there are various interpolation schemes that can be used to smooth the empirical cdf, and even add tails. We highly recommend these. Banks et al. (2001, Chapter 8) shows one way to do it.

As mentioned in the answer to Question 3, simple resampling is not appropriate when the process exhibits dependence or nonstationary. Dependence can occur in one of two ways, or both: (1) There is dependence in sequence (a time series), such as the values of a stock index recorded every 10 minutes; or (2) there is dependence across different input processes, such as the dependence between sales of new cars and the sales of car tires. In case (1) you should resample an entire series of values, while in case (2) you should resample pairs (or in general vectors) of values that were observed together.

Nonstationarity means that the input process changes over time. For instance, consider the number of users connected to an Internet Service Provider (ISP) by time of day. There are clearly peaks and valleys in the user load. Similar to the case of dependence, when there is nonstationary behavior then entire cycles must be resampled (entire days of user load profiles in the example).

10 WHAT IF I HAVE NO DATA?

The short answer is, be resourceful and be creative. When no data are available you have to use anything you can find as a basis for your input models: engineering standards and ratings; expert opinion; physical or conventional limits or bounds; and the physics of the process itself. Here are a few examples:

- To model the time it takes to do computer data entry you could research the world record for typing speed to provide an upper bound, and spend a few minutes doing some one-finger typing to find a lower bound. You probably would not use either of these numbers, but any input model you selected should clearly take values between these extremes.
- In designing a new work cell containing a number of machining processes you might use the manufacturers' ratings for cycle time as a basis for input models on actual cycle times.
- If your model requires the number of defective items found in a shipment of parts, and each item is independently good or bad, then the physics of the situation implies that a binomial distribution is appropriate. You then have to supply a size for the shipment and a probability that an item is defective.

By far the most common approach when data are not available is to use "expert opinion," meaning that you draw

on the knowledge and experience of people who are familiar with the process you want to model. Experts are often able to estimate the center and the extremes. However, even though people may feel comfortable provide an average value, what they may mean by "average" is "most likely," which is not necessarily the same thing. Thus, it is better to ask for the most likely value directly and interpret what you get that way.

The triangular distribution is a an easy-to-use input model that is specified by minimum, most likely and maximum possible values, things experts often can supply. Avoid the temptation to use the uniform distribution, which only requires minimum and maximum values. There are very few real processes in which the extremes are as likely as the center, but that is what the uniform distribution implies.

If there are a small number of discrete outcomes, then you want to ask the expert for the percentage chance of each. For instance, if the event is whether or not you win the contract, then elicit the expert's subjective chance of each outcome. Even when there are a large number of outcomes—far too many to specify the chance of each one individually—an expert might be able to provide a probability of meeting or exceeding several targets. As an example, sales people are sometimes comfortable making statements such as the following: "We will definitely do $300,000 in sales because we have those orders locked up. I think we have a 50% chance of exceeding $600,000, and a 10% chance of beating $700,000. The absolute limit in sales for next year is $850,000 if we get the entire market." These *breakpoints*—numerical values and the chance of exceeding (or, equivalently, not exceeding) them—can be used to specify the piecewise continuous distributions incorporated into nearly all simulation languages. See Banks et al. (2001, Chapter 9) for a detailed example.

In some contexts an expert may be willing to supply a central value and a percentage variation around it. For instance, "the average time to pick an order is 20 minutes, plus or minus 10%." This might suggest a normal distribution for picking time with mean 20 minutes and standard deviation $20 \times 0.10 = 2$ minutes. This could be fine, but there are some cautions to keep in mind. As mentioned above, the mean and the most likely value do not always have to be the same. More critically, people do not naturally think in terms of "standard deviations." In this context you would need to insure that "plus or minus 10%" means the *average* deviation from 20 minutes, not the most extreme deviation. For a normal distribution with mean μ and standard deviation σ, roughly 33% of the values will be outside the range $[\mu - \sigma, \mu + \sigma]$. If the expert meant that *virtually all* orders take between 18 and 22 minutes, then 2 minutes might better correspond to 3 standard deviations, not one. Finally, be careful with models like the normal that have an infinite range. If 0 is within 3 standard deviations of the mean then there is a nontrivial chance that a negative value will be generated, which makes no sense in this example. When an input model with infinite range is used, then be sure to check for values that are generated outside the feasible range for that process.

Assessing the sensitivity of simulation output results to the input models chosen is always important, and this is especially true when the input models are determined without data. Sensitivity to both the center of the distribution and its variability should be checked. For instance, if you were using a triangular distribution, then you could shift the most likely value and move the minimum and maximum closer together and farther apart. Those distributions that have a substantial impact on the simulation output should be reexamined with more care.

REFERENCES

Banks, J., J. S. Carson, B. L. Nelson and D. Nicol. 2001. *Discrete-Event System Simulation*. 3d ed. Upper Saddle River, New Jersey: Prentice Hall.

Law, A. M. and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. 3d ed. New York: McGraw-Hill.

Nelson, B. L. and M. Yamnitsky. 1998. Input modeling tools for complex problems. In *Proceedings of the 1998 Winter Simulation Conference*, ed. D. J. Medeiros, E. F. Watson, J. S. Carson and M. S. Manivannan, 105–112. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Vincent, S. 1998. Input data analysis. In *The Handbook of Simulation*, ed. J. Banks, 55–91. New York: John Wiley & Sons.

AUTHOR BIOGRAPHIES

BAHAR BILLER is an assistant professor of Operations Management and Manufacturing at Carnegie Mellon University. She received her Ph.D. from Northwestern University. Her research interests are in computer simulation of stochastic systems and stochastic input modeling.

BARRY L. NELSON is the Krebs Professor of Industrial Engineering and Management Sciences at Northwestern University, and is Director of the Master of Engineering Management Program there. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems. He has published numerous papers and two books. Nelson has served the profession as the Simulation Area Editor of *Operations Research* and President of the INFORMS (then TIMS) College on Simulation. He has held many positions for the Winter Simulation Conference, including Program Chair in 1997 and current membership on the Board of Directors. His e-mail and web addresses are `<nelsonb@northwestern.edu>` and `<www.iems.northwestern.edu/~nelsonb/>`.