

SIMULATION FOR THE MASSES: SPREADSHEET-BASED MONTE CARLO SIMULATION

Thomas J. Schriber

The University of Michigan
Business Information Technology
Ann Arbor, MI 48109-1234, USA

ABSTRACT

Spreadsheet-based Monte Carlo simulation has come into relatively widespread use as a methodology supporting studies in risk analysis for a broad range of problem types in areas such as operations management, finance, and marketing. The basic methodology is quite easily assimilated by those familiar with the fundamentals of probability, statistics, and sampling, and the practical applications of the methodology are compelling. Practitioners of discrete-event simulation can master spreadsheet-based simulation especially easily, and can expand their skill set importantly by doing so. This paper provides suggestive examples for the use of spreadsheet-based simulation and briefly explores the underlying methodology, mentioning the relevant native Excel tools and reviewing elements of one of the commercial Excel add-ins available to support use of the methodology. The presentation itself will include demonstration and discussion of the dynamics of spreadsheet-based Monte Carlo simulation.

1 APPLICATION EXAMPLES

Perhaps the best way to introduce and tease out the nature and spirit of spreadsheet-based Monte Carlo simulation is to present a series of problems appropriate for its use, including a brief review of aspects of the results that can be developed using the methodology. A series of 5 examples is introduced in this section for that purpose, moving from small-scale problems to those of larger scale. With one exception, the problems and results are taken from Winston and Albright 2009, and are presented here in paraphrased form. Winston and Albright 2009 is a useful source from which the methodology can be learned on a self-study basis, especially by discrete-event simulation practitioners.

1.1 A Tax Saver Benefit Plan

This problem (Winston and Albright 2009) is of the “beginner” variety, having only one random input variable. The problem could be investigated by means other than simulation, but the problem itself is easily digested, is relevant for many people, conveys the spirit of spreadsheet simulation, and illustrates a type of simulation output.

A Tax Saver Benefit (TSB) Plan lets you put money deducted from wages into an account during the course of a calendar year and use that money to pay that year’s authorized medical expenses not covered by health insurance. (The money put into the account is not subject to federal income tax, which is why the name “tax saver” comes into play.) Medical expenses not covered by insurance are reimbursed to you from the account until the account balance becomes zero or the calendar year ends (whichever comes first). The catch is that you must decide how much to put into the account *before the year begins*. Medical expenses not covered by insurance that year are an uncertain future outcome (the realized value of a random variable), forcing you to do some educated guessing about how much to commit to the account before the year begins. If your reimbursable expenses exceed the committed amount, you forego potential savings. But if the reimbursable expenses fall short of the committed amount, the unused money in the account is lost to you. (People sometimes buy new prescription eyeglasses at year’s end to deal with this latter situation.)

In terms of a specific situation, suppose your annual salary is \$80,000 and your marginal federal income tax rate is 30%. Assume your otherwise uncovered annual medical expenses are gamma distributed with $\alpha = 16$ and $\beta = 125$ as the distribution’s parameters. These parameter values imply a mean of \$2000 and standard deviation of \$500, with the distribution skewed to the right, which might be realistic. Build a simulation model in which the output is the amount left to you after

putting a portion of your pre-tax wages into a TSB, paying taxes on the remaining wages, and paying medical expenses (if any) in excess of the amount committed to the TSB. Experiment with the amount put into the TSB. What contribution to the TSB do you estimate maximizes the estimated expected amount left to you?

The spreadsheet model called for here is straightforward. For an assumed TSB commitment, sample from the distribution of unreimbursed medical expenses and compute the amount left to you. Repeat this process (“iterate”) numerous times (by carrying out automated and repeated workbook recalculations). Process the resulting sample of amounts left to you to compute the sample mean (and any other sample statistics that might provide useful insights into the decision process).

With the amount committed to the TSB set at \$1000, the histogram for the amount left to you and based on 1000 iterations using an Excel 2007 add-in named @RISK 5.0 (Palisade Corporation 2008) is shown in Figure 1. (@RISK 5.5 has recently been released by Palisade, but @RISK 5.0 is used in this paper.) An alternative add-in such as Crystal Ball (Oracle Crystal Ball 2008; Ragsdale 2007) could be used or, for that matter, the model could be built in Excel without use of an add-in, albeit it more laboriously. The minimum in the sample is about \$52,345, the maximum is \$55,300, and the mean is about \$54,300. (See Figure 1.) Further experimentation shows that the estimated optimal TSB amount is about \$1,750, with the expected amount left to you estimated to be about \$54,450. (You are not much better off by committing \$1750 to the TSB than by committing \$1000 in this example.)

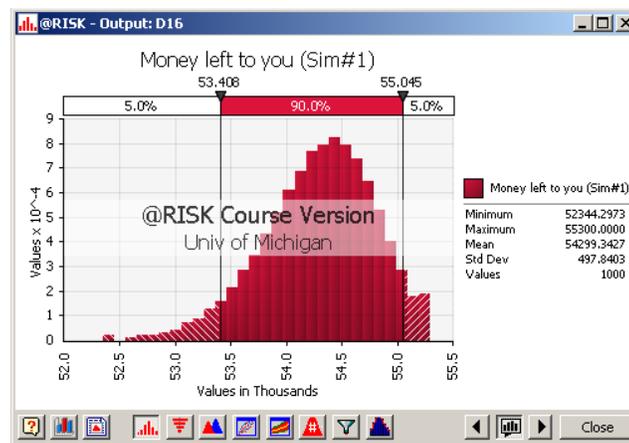


Figure 1: Distribution of the amount of money left to you in the Tax Saver Benefit model

A relevant question in the realm of modeling is: “where do the input numbers come from?” (Winston and Albright 2009 consistently address and answer this question in their examples.) In this problem, “where do the estimated distribution (of otherwise unreimbursed annual medical costs) and the values of the distribution’s parameters come from?” If data exist for the experiences of people (perhaps segmented into demographic groups) in this nook of health care, then distribution fitting can be used to estimate the underlying distribution. (Excel add-ins such as @RISK provide distribution-fitting capabilities.) In addition, sensitivity analysis can be used to estimate how sensitive the results of a risk analysis study are to assumptions made about model inputs. (@RISK provides tools for sensitivity analysis.)

Another relevant question is: “why 1000 iterations? Why not 500, or 2500, or 10000?” Here again, add-ins like @RISK provide tools that can be used to estimate the number of iterations required to achieve a user-specified degree of stability in such sample statistics as sample mean, sample standard deviation, and various percentile values. @RISK’s auto-convergence feature has not been used in this example, however, to estimate the sample size needed to achieve an acceptable degree of convergence in one or more sample statistics.

1.2 Meeting an Order Due Date at Wozac

In this problem (Winston and Albright 2009), there are three random inputs, ratcheting things up from the preceding TSB problem.

Wozac, a manufacturer of drugs, has accepted an order for 8000 ounces of a drug, and wants to plan its production schedule to meet the promised delivery date. There are three sources of uncertainty that make planning difficult. First, the drug must be produced in batches sequentially, and there is uncertainty in the time required to produce a batch, which can range from 5 to 11 days. (The uncertainty is described by a discrete distribution, not shown here.) Second, the yield (usable quantity) from any batch is uncertain. Wozac estimates that the yield can be modeled by a (continuous) triangular distribution with

minimum, most likely, and maximum values equal to 600, 1000, and 1100 ounces, respectively. Third, each batch must be inspected after it has been produced. The probability that a batch passes inspection is estimated to be 0.8. (The various probabilities assumed in this problem are based on previous production data.) If a batch fails inspection, none of it can be used to help fill the order. Wozac wants to use simulation to help decide how many days prior to the due date to begin production.

In the spreadsheet model, the production of consecutive batches is simulated (days to completion; yield; and result of the inspection are recorded, batch by batch) and a running total is kept for the usable ounces obtained and the number of days elapsed “so far.” As many batches are simulated as are required to fulfill the order. This makes it possible to determine when production must begin (for each simulated order completion, i.e., iteration) assuming a specified number of production days per week. Iterations are then carried out to form a sample of counts of the “days to fulfill the order” random variable. This information is then processed in percentile terms and can be used by Wozac to help decide when production should begin, as determined by the choice of a probability that the order will be fulfilled by the order’s due date.

Based on 1000 iterations, a histogram for the “days to complete” random variable is shown in Figure 2.

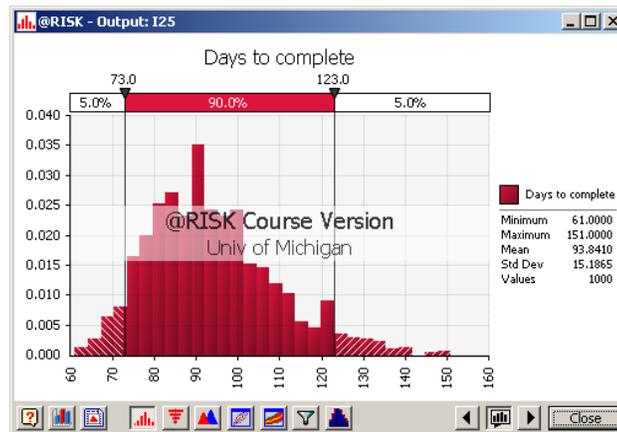


Figure 2: Distribution of the number of days required to fulfill an order in the order due date model

How does the Figure 2 histogram help Wozac decide when to start production? Well, if Wozac wants to start production to be certain with estimated probability 0.95 that it will meet the due date, it should start production 123 “production days” before the due date. (Note that there are two “vertical sliders” in Figure 2. The rightmost slider is positioned to leave 5% of the “days to complete” sample in the right tail of the distribution. The corresponding value of the “days to complete” random variable is 123. (These vertical sliders can be positioned dynamically in the on-screen version of Figure 2.) If Wozac decides to be more risk-averse and lowers the estimated probability of missing the due date to only 0.02, or 2%, then it should start production 131 days before the due date. (This result, not shown here, was produced by adjusting the right vertical slider in Figure 2 so that only 2% of the sample remained in the right tail of the distribution.) More generally, statistics from the simulation (not shown here) indicate how the estimated probability of missing the due date varies as a function of when production starts.

1.3 Estimating the Chances that a Proposed Project Will Make (or Lose) Money: A Net Present Value Model

This small-scale problem, which brings financial considerations into play, has been developed for introductory use in an MBA course at the University of Michigan. The problem is characterized by 7 random inputs.

A proposed project has a projected four-year life. (The projected life of a project is typically not known with certainty, and generally should be treated as the realized value of a random variable, but here we treat it as known.) Construction will take place during year 1. The construction costs are estimated to be normally distributed, with mean \$4,000,000 and standard deviation \$500,000. Revenue begins in year 2, and is estimated to be normally distributed with mean \$14,000,000 and standard deviation \$1,500,000. Revenue in years 3 and 4 is estimated to be normally distributed, with mean equal to the preceding year’s sampled revenue, and standard deviation set at 10% of the preceding year’s sampled revenue. (This year-to-year relationship reflects a dependency among input variables. More generally, correlated input variables can be described in spreadsheet add-ins such as @RISK by providing matrices of correlation coefficients in models.)

Operating expenses in years 2, 3, and 4 are estimated to range from 60% to 80% (uniformly distributed) of the corresponding year’s revenues. Taxes are 40% of the post-expense revenues.

Using a discount rate of 15%, and based on 1000 iterations, estimate the distribution of the project’s net present value (NPV) and internal rate of return (IRR). What is the estimated probability that the project will lose money, i.e., that the estimated NPV will be \$0 or less? What is the estimated probability that the NPV will exceed its estimated expected value by 20% or more? What is the estimated probability that the project’s IRR will exceed the discount rate of 15%?

In this problem, the modeler needs to understand the concepts of net present value and internal rate of return. Broadly speaking, the net present value of a series of projected costs and benefits is the algebraic sum of the costs and benefits after they have been discounted to the time corresponding to the start of the project. The internal rate of return is the annualized interest rate at which a project’s estimated costs are exactly balanced by the project’s estimated benefits. Excel provides built-in functions named NPV and IRR that compute the corresponding values for a time series of costs and benefits. More generally, those not familiar with the fundamentals of the “mathematics of finance” can bring themselves up to speed in broad terms by studying chapter 14 in Gips 2007. (This reference is only one among many that could be mentioned.)

The simulation model is straightforward, and can be built using the tools in native Excel or, with less time-consuming overhead, can be built using the @RISK (or Crystal Ball) add-in. Using @RISK, Figures 3 and 4 show the behavior of the NPV and IRR variables, based on samples of size 1000 iterations.

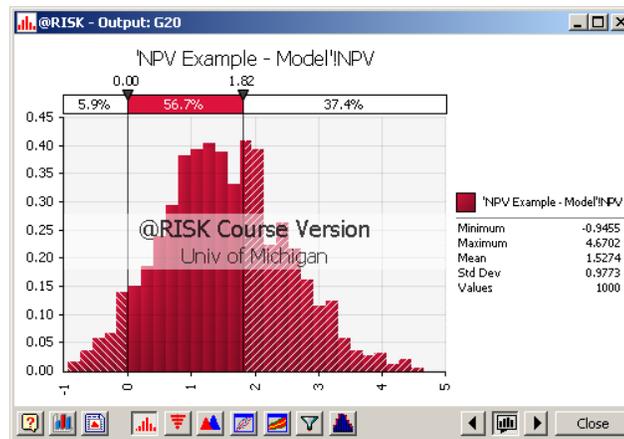


Figure 3: Distribution of net present value in the model of a proposed project

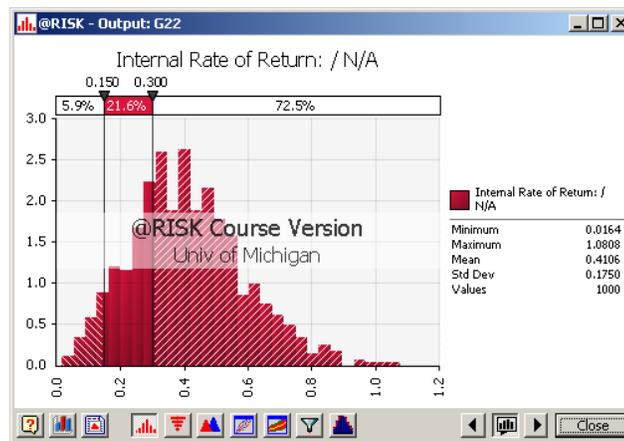


Figure 4: Distribution of the internal rate of return in the model of a proposed project

As shown in Figure 3, with the “left slider” set to \$0.00, the estimated probability of losing money (in the sense that the net present value is \$0.00 or less) is 0.059, or 5.9%. In the same figure, we see from the “right slider” that the estimated probability of achieving an NPV that exceeds its estimated mean by 20% or more is 0.725, or 72.5%. (Figure 3 also shows that the sample mean NPV is about \$1,530,000.)

Figure 4 shows the sample distribution of the internal rate of return. The left slider indicates that the estimated probability of an internal rate of return less than the discount rate of 15% is 0.059, or 5.9%. (The 5.9% is consistent with the Figure 3

indication that the estimated probability of “losing money” is also 5.9%.) The right slider indicates that the estimated probability of an internal rate of return exceeding 0.30, or 30% (twice the discount rate) is 0.725, or 72.5%. The results are strongly suggestive of a favorable project outcome, even though there is some chance that the project, if undertaken, will lose money.

What are the most important influencing inputs promoting the favorable project outcome in this example? If we can answer that question, we can gain valuable insights into the situation. If we then have the opportunity to reduce the uncertainty in one or more input variables by additional study, we can determine where best to spend our time and money when investigating the character of the inputs and trying to reduce their uncertainties. (Some uncertain inputs can be relatively unimportant, and can perhaps even be replaced by their expected values, whereas the decision process can be quite sensitive to assumptions made about some other inputs, which should then continue to be random inputs.) Figure 5 shows a “Tornado Chart” that can be requested of @RISK to visualize the relative strengths of the inputs.

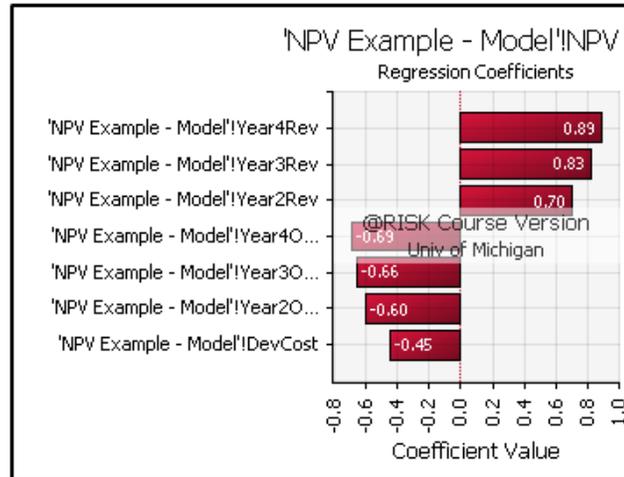


Figure 5: Tornado chart displaying regression coefficients in the model of a proposed project

Figure 5 provides a graphical representation of the relative influences of the model inputs in terms of their regression coefficients. Evidently the most strongly influencing input is the year 4 revenue, which has a positive influence (regression coefficient) of 0.89 on the net present value. On the negative side, the most strongly influencing input is the year 4 operating cost, which has a (negative) influence on the net present value of 0.69. These are among the types of insights that can be gained in spreadsheet-based Monte Carlo simulation.

Sometimes the question is asked, “why not just use expected values in the model?” Add-ins such as @RISK provide a toggle that let the modeler set the otherwise-random inputs to their expected values. When that toggle is used in this model, the net present value is \$1,525,000 and the internal rate of return is 40.1%. Note how close these values happen to be to the corresponding sample means in this model. But beware of the “flaw of averages” (the phrase is attributed to Sam Savage, of Stanford University), which Winston and Albright 2009 describe this way (lightly paraphrased): “When an output is a nonlinear function of one or more uncertain inputs, the expected value of the output is generally *not* equal to the output obtained by using the mean values of the inputs. This latter value is what a naïve analyst would obtain by ignoring the uncertainty and using only “best guesses” for the inputs. It typically provides a misleading estimate, possible too high and possibly too low, of the true mean.”

What’s more, if means are used as inputs, no information is produced by the resulting deterministic model that can be used to answer questions such as “what is the estimated probability that this project will lose money?” or “what is the estimated probability that the realized internal rate of return will exceed the discount rate by a factor of 2 or more?”

As an alternative to Monte Carlo simulation, modelers will sometimes build a series of scenarios, such as a “most likely” scenario (using estimated most likely input values), a “pessimistic scenario” (using input values that are very unfavorable for the project), an “optimistic scenario” (using input values that are very favorable for the project), and perhaps a handful of other scenarios positioned along the spectrum ranging from the pessimistic to the optimistic. Excel provides a Scenario Manager that can be used to build scenarios and provide scenario reports. But such scenarios do not yield information that can be used to respond to questions such as those asked in the preceding paragraph.

Spreadsheet-based Monte Carlo simulation can be thought of as an “automatic scenario generator.” Each recalculation of the worksheet (or workbook, or system of workbooks) produces a randomly generated scenario. Hundreds or thousands of such scenarios can be produced automatically, continuing if desired until a user-specified degree of stability in selected sam-

ple statistics has been achieved. (This is thanks to the “auto-convergence” feature of add-ins such as @RISK that was mentioned earlier.) Monte Carlo simulation is the Grand Daddy of scenario analysis. The value of the resulting ability to analyze and assess the degree of risk in a proposed undertaking is formidable.

1.4 Developing a New Car at GF Auto

This is another example of a financially-oriented problem but, unlike the preceding problem, this one is more realistic in the level of detail provided to derive the numbers used to estimate the distribution of net present value. There are only two random inputs, but uncertainty in various other inputs clearly could be incorporated into those inputs, too.

General Ford (GF) Auto Corporation (Winston and Albright 2009) is developing a new model of compact car, which is assumed to generate sales for the next 5 years. GF has gathered information about the following quantities through focus groups with the marketing and engineering departments (experts within the company) and through focus groups of potential customers, too (relative to estimated sales of the proposed compact car).

- The fixed cost of developing the car is assumed to be \$1.4 billion, incurred at the beginning of year 1.
- The margin per car is the selling price minus the variable cost of producing a car. GF assumes that in year 1, the margin will be \$5000. For each succeeding year, GF assumes the margin will decrease by 4%. (This is because variable costs tend to increase through time, whereas selling prices tend to remain fairly constant through time.)
- The demand for the car is an uncertain quantity. GF assumes that the number of cars sold in the first year will range from 100,000 to 170,000, with a most likely value of 150,000, triangularly distributed. In each succeeding year, GF assumes that the sales will decrease by from 5% to 10%, with a most likely value of 8%, triangularly distributed. The percentage decreases in successive years are assumed to be independent of one another.
- GF depreciates its development cost on a straight-line basis over the assumed 5-year lifetime of car production.
- The corporate tax rate is 40%
- The estimated cost of capital is 15%.

GF wants to use simulation to assess the behavior of the net present value for the proposed compact-car project, based on after-tax cash flows over the assumed 5-year time horizon, and using 1000 iterations.

As Winston and Albright 2009 point out, the model is like most financial multiyear spreadsheet models. The model extends a number of years “to the right” in the spreadsheet, but the bulk of the effort in building the model is centered on constructing formulas that express the governing logic and relationships in the first year or two. After those formulas have been put into place, many of them can simply be copied “to the right” to incorporate the logic for additional years into the model.

Figure 6 shows the histogram for the net present value produced by the model corresponding to the above specifications.

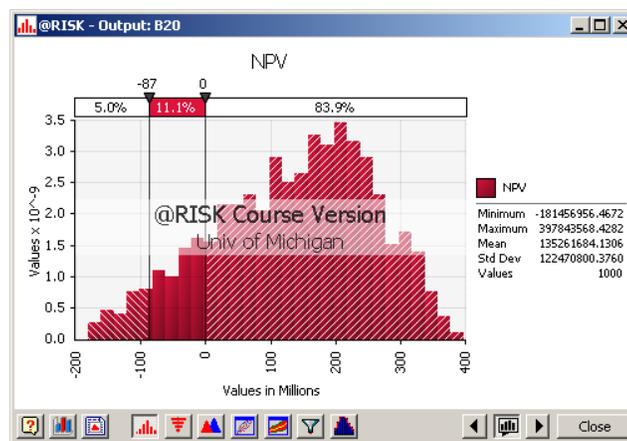


Figure 6: Distribution of the net present value in the model for development of a new car

Figure 6 indicates that there is good news, and there is bad news. The good news is that the estimated mean NPV is about \$135 million, and that the NPV could go well above that figure, potentially reaching almost \$400 million. On the other hand, the right slider in the figure shows there is an estimated probability of about 0.16, or 16%, that the NPV will be \$0.00 or less, potentially reaching almost -\$200 million. The left slider indicates that at the 5th percentile, the NPV is -\$87 million. Financial analysts often call the value at the 5th percentile the value at risk, because it indicates *nearly* how bad things could

get if enough unlucky outcomes occur. The point here is that analysts don't necessarily focus on the estimated mean result. Their interest can focus on the extreme values of a distribution, especially with respect to the potential downside.

The high variability of the net present value in this problem is due to the variability in first-year sales, not to the variability in annual sales decreases. This fact is demonstrated in the Figure 7 Tornado chart.

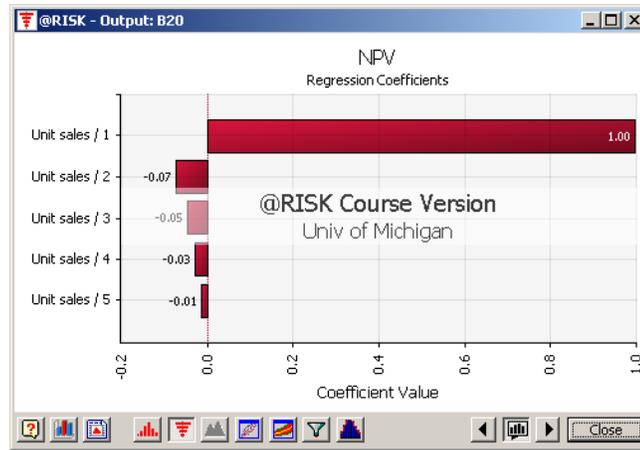


Figure 7: Tornado chart displaying regression coefficients in the model for development of a new car

If the assumptions built into the model are realistic, GF should do whatever it can to influence the distribution of the year 1 sales by trying to decrease its variability and/or to increase its expected value.

Financial models of this type are sometimes enhanced by inclusion of income statements and balance sheets. (These are asked for in Winston and Albright 2009, Case 12.3, "Financials at Carco." When an income statement and balance sheet are modeled, one or more *deliberate* circular references can come about naturally. For example, interest expenses depend on debt, debt depends on the balance sheet total, the balance sheet total depends on equity, equity depends on earnings, and earnings depend on interest expenses (thereby completing the circle). This means that iteration (not in the simulation sense, but in the sense of repeated workbook recalculations that result in a *single* logically acceptable updating of the workbook) must take place if the numbers in the model are to converge to values that are consistent with the workbook's formulas. Excel provides a control that can be used to authorize that a workbook is permitted to recalculate multiple times in this sense, until the workbook's numbers have (potentially) converged to a set of values that simultaneously satisfy each of the workbook's formulas.

1.5 Maintaining a Minimum Cash Balance

Companies typically track their cash balance over time. A minimum cash balance is sometimes maintained as a matter of policy so that expenses can be paid promptly on favorable terms, without incurring a late-payment penalty and/or high-rate interest charges. Short-term loans might have to be taken out from time to time to maintain a minimum cash balance. Furthermore, assumptions must be made about the timing of receipts (income from sales; interest received from cash balances; cash received as a result of taking out short-term loans) and disbursements (payments for fixed costs; taxes; bills from suppliers; repayment of short-term loans and interest). For simplicity, it can be assumed that receipts and disbursements all take place at the beginning of each month, with receipts accounted for first and then disbursements made. For example, receipts for some of the sales made in *preceding* months arrive at the beginning of the next month and can be used (along with the month's beginning cash balance and interest on the preceding month's cash balance) to help cover disbursements. The preceding month's short-term loan (if any) is repaid and another short-term loan is taken out, if necessary, to bring the cash balance up to the specified minimum level. The updated cash balance is then carried forward during the month and accumulates interest in the process, with that interest then received at the beginning of the next month. And so on. (Some situations might call for a more nuanced set of assumptions than the ones just described.) The final example of spreadsheet simulation provided here is based on the minimum cash balance theme, and uses the simplifying assumptions just described.

The Entson Company (Winston and Albright 2009) estimates from previous experience that monthly sales from November through July are normally distributed with given means and standard deviations that depend on the month in question (not shown here). Each month Entson has fixed costs of \$250,000. Taxes of \$150,000 and \$50,000 are due at the beginning of March and June, respectively. Entson estimates that receipts at the beginning of a month are a weighted sum of sales from the

preceding three months, with weights 0.2, 0.6, and 0.2, respectively. To help maintain favorable relations with its customers, Entson does not levy late-payment penalties on them.

The materials and labor needed to produce a month's sales must be arranged and paid for one month in advance, at a cost averaging 80% of sales. (This is "Catch 22," because a month's sales are not known one month in advance, although in the model under discussion it is assumed they are. The insights drawn from use of the model might not be materially affected by this assumption. More realistic workarounds can be envisioned, but they are not considered here.)

At the beginning of January, Entson has \$250,000 in cash. Entson wants to be sure that each month's updated cash balance (which then becomes the beginning cash balance for the following month) is never less than \$250,000. Entson will take out one-month loans, if necessary, to ensure that this condition is met. Such loans are paid back with 1% interest at the beginning of the following month. Each month, Entson earns 0.5% interest on its beginning cash balance. (This interest has been accumulated during the preceding month.) Entson wants to model this situation to come to an understanding of how its short-term loans vary over time, based on 1000 iterations, and to estimate the total interest paid on the loans.

The monthly calculations in the model occur in this sequence:

- Entson receives interest (accumulated during the preceding month) on its beginning cash balance.
- Receipts arrive and expenses are paid (including repayment of the preceding month's short-term loan, if any).
- If necessary to meet the minimum cash balance requirement, another short-term loan is taken out.
- The resulting updated cash balance accrues interest and becomes the next month's beginning cash balance.

Summary results from the simulation indicate that the maximum short-term loan over the relevant time frame ranges from about \$461,000 to about \$1,534,000, averaging about \$952,500. The average interest on the short-term loans over the relevant time frame is about \$20,000.

Figure 8 shows the summary trend chart available in @RISK for the series of loans.

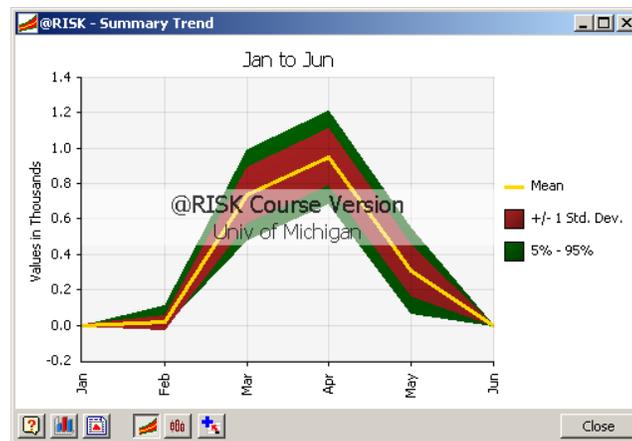


Figure 8: Summary trend chart showing the range of short-term loan amounts in the minimum cash balance problem

The Figure 8 chart shows how the month-by-month short-term loans (if any) vary over time. The middle line (yellow) is the mean short-term loan amount for the month in question. The inner bands (shown in red in the original figure, but unfortunately indistinguishable from the green outer bands in a black-and-white rendition) extend to plus or minus 1 standard deviation of the mean, and the outer bands (green) extend to the 5th and 95th percentiles.

Why do the required short-term loans peak in March and April, the months (not shown here) when Entson's sales are relatively high? There are two reasons. First, Entson has to pay its costs early. Second, most of Entson's receipts arrive late. The timing and amounts of short-term loans are therefore somewhat complex, but nevertheless can be captured in a simulation model.

This completes the series of five spreadsheet-based Monte Carlo simulation examples chosen to pique interest in the application potential for methodology of this type. To repeat an earlier point, practitioners of discrete-event simulation should be able to master this methodology especially easily. The resulting increase in a practitioner's skill set will enhance a practitioner's potential value in an organization.

2 BUILDING AND USING SPREADSHEET-BASED MONTE CARLO SIMULATION MODELS IN GENERAL

This section briefly describes the activities carried out when conducting Monte Carlo simulation projects of the spreadsheet type, and points out the potential utility of an add-in to support many of these activities.

2.1 The Steps Involved in Conducting Spreadsheet-Based Simulations

Building and using a spreadsheet-based Monte Carlo simulation model involves five broad steps:

1. The model capturing the logic and relationships inherent in the problem at hand must first be built and tested. (This must be done, of course, whether or not the eventual use of an add-in is intended.)
2. Model inputs involving uncertain quantities must be expressed with formulas designed to sample from specified probability distributions. For example, if a particular input is assumed to be normally distributed with a given mean and standard deviation, then a corresponding formula must be built to return a randomly selected value from the relevant population each time the workbook is recalculated.
3. The model must be equipped with features to record the observed value of each output variable of interest produced each time the workbook is recalculated. These outputs are realizations of random variables that depend on the values sampled from the input distributions (as well as depending on the logic and relationships that tie variables together in the model).
4. The workbook must be forced to recalculate repeatedly in automated fashion, with the effect of forming a set of observed values for each output variable of interest, iteration by iteration (or, to use alternative language, replication by replication).
5. Each set of sampled values must then be processed in various ways (e.g., sample mean; sample standard deviation; minimum and maximum values; percentiles; the relative frequencies with which sampled values fall into specified frequency classes; the forming of graphical displays, such as histograms; etc.) to produce information that will provide insights into the problem being modeled and will support the relevant decision-making process.

2.2 Building and Using Monte Carlo Simulation Models in Native Excel

The tools in Excel itself (“native Excel”) can be used to support the steps described above, albeit possibly awkwardly in general when it comes to steps 2 through 5. (The native Excel tools are used very conveniently in step 1.)

Step 2 raises the question, “how many built-in functions are there in native Excel that can be used to sample from various continuous and discrete probability distributions?” The answer is “not nearly as many as one would like.” There is a void here that an add-in can fill.

Steps 3 and 4 raise the question, “in native Excel, how can you automate repeated workbook recalculation and at the same time store selected values that are produced at each recalculation?” The answer: “use an Excel Data Table.” It is novel and somewhat unconventional and cumbersome to use a Data Table for this purpose, but it can be done. (Data Tables were not originally intended for such a purpose.) That said, there is a compound shortfall in steps 3 and 4 in native Excel that an add-in can remedy.

As for step 5, samples can be processed in native Excel to extract information from sampled values, but at a potentially high cost in time and effort (unless generalized macros are perhaps created for this purpose). But an add-in can largely eliminate this cost in time and effort by anticipating the types of information likely of interest in the context of spreadsheet-based Monte Carlo simulation and making various types of information available with a couple of mouse clicks.

In conclusion, it soon becomes clear what the motivations can be for simulation practitioners working in this area to take advantage of the existence of add-ins such as @RISK and Crystal Ball.

3 WHAT @RISK PROVIDES TO SUPPORT SPREADSHEET-BASED SIMULATION

This section itemizes selected features of Excel 2007’s @RISK 5.0 add-in to support spreadsheet-based Monte Carlo simulation. Similar features are available in the Crystal Ball add-in. It is not the purpose of this section or paper to tout any particular add-in. Features of the @RISK add-in are described because the Winston Albright 2009 textbook used in a University of Michigan MBA course includes @RISK to support spreadsheet-based Monte Carlo simulation. That makes @RISK a convenient choice here.

The @RISK features itemized below are presented without much elaboration, and without attempting to provide operational details. Many of these features will be demonstrated dynamically in the session that this paper supports. Operational

details are described in a number of sources, including Winston and Albright 2009, Palisade Corporation 2008, and in a series of tutorials at the <http://palisade.com> website.

Items 1, 2, and 3 below correspond to steps 2, 3, 4, and 5 in section 2.1. (More specifically, item 1 corresponds to step 2, item 2 to steps 3 and 4, and item 3 to step 5.)

1. @RISK currently provides 39 functions for sampling from continuous and discrete probability distributions. Typical of such functions are RiskGamma (for sampling from the gamma distribution); RiskPert (for sampling from the Pert distribution, which is a special form of the beta distribution); RiskTNormal (for sampling from a truncated normal distribution); RiskTrigen (for sampling from a triangular distribution based on three points: the most likely; the value at a user-specified bottom percentile; and the value at a user-specified top percentile); and RiskDiscrete (for sampling from a discrete distribution with user-specified outcomes and relative frequencies).
2. @RISK carries out iterations “behind the scenes” automatically (in response to a mouse click), recording the values of user-specified output variables in the process. (The resulting sets of recorded observed values can be accessed by the modeler at will, of course.) No Data Tables in the Excel sense of the word are used.
3. @RISK automatically processes the information in each set of recorded observations and makes numerous reports and graphical displays available either on a pre-specified basis (requested before the iterations are performed), or on an after-the-fact basis (requested after the iterations have been performed), e.g., routine sample statistics; histograms; percentiles; regression and correlation coefficients; Tornado Charts that rank and quantify the relative influence of the probabilistic inputs on the behavior of the outputs.

The remaining items, listed here selectively but not exhaustively, describe additional ways in which @RISK supports the simulation process.

4. @RISK makes a variance reduction technique known as stratified sampling available for optional use. The advantage of using a variance reduction technique is that sample statistics converge more rapidly on their expected values for a given sample size than they would if a variance reduction technique were not used. As a result, fewer iterations are needed than otherwise to achieve a user-specified degree of stability in sample statistics.
5. @RISK provides an “Automatic” option as an alternative to specifying how many iterations are to be performed. With this option, a “Convergence Tolerance” and “Confidence Level” can be set. (These default to 3% and 95%, respectively.) The sample statistics for which convergence testing is conducted can be chosen from one or more of these three: sample mean; sample standard deviation; and a user-specified percentile, such as 1% or 5%, etc. @RISK will then perform as many iterations as are needed to attempt to achieve the specified degree of convergence. The user indicates how often @RISK should “pause” to test for convergence, such as every 50 iterations, or every 100 iterations, etc.
6. An “expected value option” is available. In response to a mouse click, the workbook recalculates one time and displays the results when probabilistic inputs are replaced by their expected values. (This “replacement” is done logically, not by formula deletion. Another mouse click restores the original distribution-sampling formulas.)
7. @RISK provides a distribution-fitting capability, whereby theoretical distributions are fitted to a set of observed values. On request, @RISK automatically carries out a fit for each relevant type of distribution (within its current repertoire of 39 distributions), ranks the results, provides goodness-of-fit statistics, and displays a histogram for the observed values, with the theoretical density function (in the case of continuous data) or discrete-function equivalent superimposed on the histogram. The formula for sampling from the user’s choice of a theoretical fit can be put into a user-specified worksheet cell with a mouse click. (What is the motivation for possibly using a theoretical distribution in lieu of the underlying set of observed values? One motivation, for example, is that a theoretical distribution might provide better insights into the behavior of a random variable in the tails of the distribution.)
8. @RISK provides a function that can be used to support carrying out two or more consecutive simulations automatically (with each simulation consisting of a user-specified number of iterations) for each in a set of alternative settings of a policy variable (or variables). For example, in an inventory control situation the user might want to simulate for each of three alternative settings of the reorder point, performing 1000 iterations for each alternative. The resulting experiment is said to take the form of “three simulations, 1000 iterations per simulation.”
9. @RISK lets the user specify the “starting point” of the 0-1 uniform random number generator used to sample from a model’s input distributions. This is done by supplying a number that @RISK calls an “initial seed.” As an alternative, the user can let @RISK choose the initial seed at random. (@RISK later reports the value it has chosen.)
10. When a sequence of simulations (see item 8 above) is performed, each of the simulations can use the same initial seed, or each can use a different initial seed. (The motivation for use of the same initial seed from simulation to simulation is that the policy alternatives tend to be compared under conditions of “all other things being equal.”)
11. @RISK offers a set of 8 distinct 0-1 uniform random number generators (algorithms) from which to choose. These go by the names Mersenne Twister (the default generator), MRG32k3a, KISS, MWC, SWB, LFIB4, KISS_SWB,

and RAN3I. Only one of these (user's choice) is in effect throughout a simulation. The bottom line is that the user is not put into the inflexible position of being forced to use a random number generator with properties that he or she might prefer to avoid for whatever reason for a given modeling project. Generally speaking, this topic involves a potentially subtle aspect of random number generation.

12. @RISK accommodates the use of Excel macros for automated pre-, and/or interim, and/or post-processing. In particular, at the user's request, @RISK will run macros at one or more of these times: before each simulation; before each iteration; after each iteration; after each simulation.

The preceding list details features of @RISK that can be of broad interest for spreadsheet-based Monte Carlo simulation. There are other features of @RISK not mentioned in the list, some of them major in nature. For example, @RISK can be used with software named RiskOptimizer to combine simulation with optimization. @RISK can also be used with Microsoft Project to support risk analysis modeling in the context of project management. The ultimate source for features of @RISK 5.0 is Palisade Corporation 2008.

4 CONFERENCE PRESENTATION

The presentation and discussion of this material at the 2009 Winter Simulation Conference will be supported with the use of Excel files and the @RISK add-in. The files can be obtained on request after the conference by sending email to <schriber@umich.edu>.

ACKNOWLEDGMENTS

The heavy use of example models and paraphrased material from Winston and Albright 2009 is recognized and gratefully acknowledged with thanks to Professors Wayne L. Winston and S. Christian Albright, both of Indiana University. Also acknowledged with thanks are two reviewers of the material, 2009 Winter Simulation Conference Program Chair Professor Ricki G. Ingalls of Oklahoma State University, and Thompson T. Terry, Jr., Senior Risk Analyst at the Palisade Corporation.

REFERENCES

- Gips, J. 2007. *Mastering Excel 2007*. Boston, Massachusetts: Pearson Custom Publishing.
- Oracle Crystal Ball. 2008. *Oracle Crystal Ball User Manual*. Denver, Colorado: Oracle Crystal Ball.
- Palisade Corporation. 2008. *Guide to Using @ Risk Version 5.0*. Ithaca, New York: Palisade Corporation.
- Ragsdale, C. T. 2007. *Spreadsheet Modeling and Decision Analysis*, Revised 5th ed. Mason, Ohio: South-Western/Cengage Learning.
- Winston, W. L., and S. C. Albright. 2009. *Practical Management Science*, Revised 3rd ed. Mason, Ohio: South-Western/Cengage Learning.

AUTHOR BIOGRAPHY

THOMAS J. SCHRIBER is a Professor of Business Information Technology at The University of Michigan. He is a recipient of the INFORMS Simulation Society's Lifetime Professional Achievement Award, and of the Society's Distinguished Service Award, and has received the James R. Wilson Winter Simulation Conference Board of Directors' Award for Distinguished Service. He is a Charter Member and Fellow of the Decision Sciences Institute and author of *Simulation Using GPSS*, which was in widespread use for many years. He has been a WSC Program Chair and served ten years on the WSC Board of Directors, chairing the board for two years. He is a member of ASIM (the German-language simulation society), the Decision Sciences Institute, the Institute of Industrial Engineers, and INFORMS. His email and web addresses are: <schriber@umich.edu> and <<http://www.bus.umich.edu/FacultyBios>>