

ADVANCED TUTORIAL: NETWORKING SIMULATIONS ACROSS PLATFORMS AND ENTERPRISES WITH VIRTUAL SIPS

Sam L Savage
Shaun Doheney
Colin Smith

ProbabilityManagement.org
3507 Ross Rd.
Palo Alto, CA 94303, USA

ABSTRACT

SIPmath represents uncertainties as coherent arrays of realizations called SIPS, which may be shared between diverse simulation applications across the enterprise. This allows simulations to be linked together to form networks in which the output distributions of one simulation become the input distributions of another simulation. Furthermore, the outputs of stochastic models in packages such as R, Python or discrete event simulations may be shared with managers using interactive dashboards in native Microsoft Excel. Two recent open source advances in simulation modeling and analysis have yielded great efficiencies in this approach. Metalog distributions can fit virtually any continuous distribution of data with an analytical F-Inverse function much like a Taylor's series. The HDR Portable Uniform Random Number Generator produces identical results on all platforms including a single cell formula in Excel. Participants are encouraged to bring their laptops for a hands-on learning experience.

1 BACKGROUND

The discipline of probability management represents uncertainties as standardized vectors of coherent simulation trials. The motivation for standardization is twofold. One, it allows simulations running on diverse platforms to be networked together to create auditable, enterprise wide stochastic models (Savage and Thibault 2015). And two, it allows those with statistical expertise to create stochastic inputs for dashboards for decision makers without such expertise.

Vectors of realizations have been used in the fields of financial engineering, insurance and stochastic optimization for decades. Nonprofit ProbabilityManagement.org created the open SIPmath 2.0 standard for conveying such data and metadata across platforms as SIPS (stochastic information packets) (Kirmse and Savage 2014; Thibault 2016). Furthermore, improvements to Excel's data table function enable interactive stochastic models in the common spreadsheet to run thousands of trials per keystroke based on SIP libraries created in R, Python or discrete event simulations.

2 TUTORIAL OUTLINE

This tutorial will:

1. Introduce the metalog distributions and show how to fit them to data.
2. Demonstrate the HDR portable random number generator, and how to use it in distributed random number management of both idiosyncratic and global variables.
3. Demonstrate how to create libraries of virtual SIPS that combine the use of F-Inverse functions such as the metalog and auditable random number generators such as the HDR.

4. Show how these libraries may be generated in analytical applications such as R, Python, or discrete event simulations, and then used in interactive stochastic dashboards in native Excel.

3 TWO RECENT TECHNICAL ADVANCES: THE HDR AND METALOG

Two recent advances in the theory of simulation modeling and analysis are fundamental in their own right, but when combined in the context of probability management represent a breakthrough in both storage requirements and random number management.

1. The HDR portable pseudorandom number generator, to be presented by Douglas W. Hubbard at the 2019 Winter Simulation Conference, is an open source counter (random access) generator. It does well in the dieharder tests (Brown et al. 2019) and gives identical results across platforms. It easily fits into a single cell in Excel and runs interactively with the data table, performing 1,000 trials in roughly one hundredth of a second, or a little over ten seconds per million trials. It is orders of magnitude faster in compiled languages. It has a multi-dimensional seed, which may be assigned across users on different platforms, to 1) prevent inadvertent correlation of random variables, and 2) impose correlation where desired; for example, to model externalities such as labor rates in a particular region, earthquakes, or commodity prices.
2. The metalog system (named for the logistic distribution from which it is derived) fits F-Inverse functions to quantiles allowing simulations to be easily based on historical data or the output of other simulations (Keelin 2016; Keelin 2019). Metalogs may be used to fit nearly all possible continuous data with a single family of formulas, making them akin to Taylor’s series. Virtual SIPs may be created by driving metalog F-Inverse functions with a portable generator like the HDR, This produces auditably identical streams of random variates across platforms.

3.1 Use Case Example

Consider a construction firm that performs various sorts of jobs related to roadwork. A project requires a specified number of units of each of three types of jobs to be performed as shown in Table 1.

Table 1: Units of three types of jobs to be performed.

Job Number	Job Type	Units
76468	Job 1	20
76469	Job 1	30
76470	Job 1	100
76471	Job 2	25
76472	Job 2	50
76473	Job 2	23
76474	Job 3	14
76475	Job 3	53

However, both the labor hours per unit of work, and the labor costs per hour are uncertain. What are the chances that the cost of this project will exceed \$20,000,000?

In most firms today, this would be considered an imponderable question. But those same firms typically have large quantities of data from which cost distributions can be readily estimated, and a stochastic library created. Once such a library is in place, it takes only a few keystrokes to create an interactive dashboard in native Excel that can perform 1,000 trials per keystroke to estimate the distribution of total cost as shown in Figure 2. The SIPs required to express the uncertainties in this model would comprise 1,000 numbers for each of the eight jobs plus one for the labor rates for a total of 9,000 numbers in addition to metadata. With virtual SIPs, these are replaced by four cells in Excel containing less than 500 characters each, as we will show below.

This example displays both idiosyncratic uncertainties, the number of labor hours required per job, and a global uncertainty, the cost per hour of labor in the region. These will be treated differently as demonstrated below.

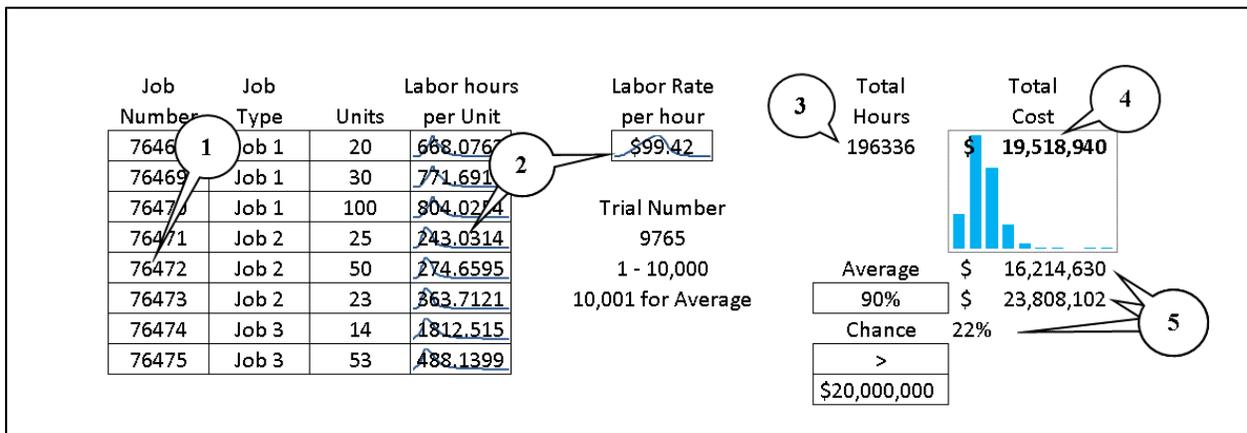


Figure 1: An interactive stochastic dashboard in native Excel based on virtual SIPs.

The steps in creating this demo model were as follows:

1. Specify the unique Job Numbers as seeds in the HDR Generator. This assures that the idiosyncratic uncertain labor hours per unit of all jobs will be independent. Note that Labor Rate per hour is a global variable and has its seed specified within the library.
2. Load the distributions of labor hours per unit for the corresponding Job Type and the global uncertainty, labor cost per hour from the virtual SIP library. The numbers appearing in the cells containing the sparkline graphs of the distributions may be set to display individual trials or metadata such as the average.
3. Specify total labor hours as the Sumproduct of Units and Labor Hours per Unit.
4. Specify Total Cost as Total Labor Hours times Labor Rate and specify this as an output cell for the data table simulation. A sparkline column graph displays the histogram of the results.
5. Average, Percentile and Chance of Exceedance are all calculated from the trials in the data table, whereupon we see that the chance of exceeding \$20,000,000 is 22%. If we were to change any of the input numbers, a model of this size would re-calculate the 1,000 trials nearly instantaneously.

3.2 Stochastic Roll-up

Now imagine ten projects each consisting of some combination of the three jobs in the example above. The manager of each project, with or without statistical experience, can fill in their individual details, whereupon the output SIPs of all ten models will be coherent. That is, because all templates use the same

seed for the global variable of labor cost, the output SIPs will be correctly correlated, and are therefore additive. Thus they may be summed element by element to yield the SIP of total cost across all ten projects.

4 SIPMATH REVIEW

Before addressing the technical advances of the metalog distributions and the HDR generator, we will review the basic concept of SIPmath. The open SIPmath Standard represents uncertainties as vectors (of dimension N number of trials) of specific realizations of a probability distribution along with metadata. The data structure is known as a SIP (Stochastic Information Packet). A set of SIPs that preserve the statistical relationships between its constituents is a Stochastic Library Unit with Relationships Preserved (SLURP). Virtually any sort of relationship, linear or non-linear, may be preserved in this manner, as opposed to simple correlation as produced by Cholesky factorization.

4.1 Advantages of SIPs

There are four primary advantages of representing uncertainties as vectors of realizations.

1. SIPs are **actionable** in that they may be used as direct input to computer simulations.
2. SIPs are **additive**. If the statistical relationships or lack thereof between vectors is preserved and they have the same dimension (number of trials), then they have group theoretic properties that are similar to those of numbers. That is, they may be added, multiplied, or divided element by element to perform stochastic calculations. For example if x and y are coherent SIPs of two random variables, that is, they preserve statistical dependence, then

$$\text{SIP}(x+y)=\text{SIP}(x)+\text{SIP}(y)$$

$$\text{SIP}(x*y)=\text{SIP}(x)*\text{SIP}(y)$$

and, if no element of y equals zero,

$$\text{SIP}(x/y)=\text{SIP}(x)/\text{SIP}(y)$$

These properties extend to virtually any other formulas which are defined for all the values of x and y , for example $\text{SIN}(x)+\text{COS}(y)$. Furthermore the property holds even with highly non-linear relationships between the variables, such as those displayed in Figure 3 by Alberto Cairo's Datasaurus data (Cairo 2016; Savage 2019).

This allows diverse simulations running on various platforms to be networked through their input and output SIPs.

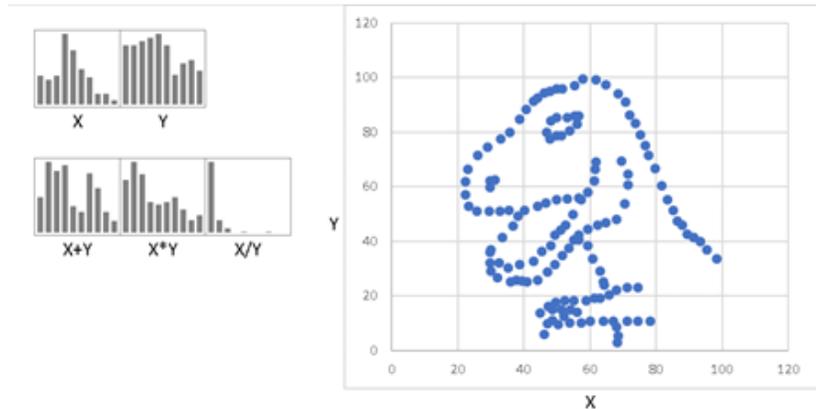


Figure 2: Datasaurus arithmetic.

3. SIPs are **auditable**. Since they consist of numerical data, they may be audited at any level of a simulation or network of simulations and contain metadata that includes provenance.
4. SIPs are **agnostic**, in that as pure data, they may be communicated between any software platforms that process arrays.

4.2 Disadvantages of SIPs

1. Large quantities of data must be stored. Even multiple instantiations of independent, identically distributed (IID) variables require a separate permuted vector of realizations for each, to ensure that no two variables are correlated.
2. The number of trials may not be usefully increased beyond the dimension of the original sample.

5 VIRTUAL SIPS

The 3.0 Open SIPmath Standard combines uniform random number generators, such as the HDR, Mersenne Twister, or others, and F-Inverse functions such as those found in R, Python, or the metalogs to create virtual SIPs taking up just a tiny fraction of the storage of an actual data base of trials.

Virtual SIPs are a way to maintain the four advantages listed above while overcoming the two disadvantages. Instead of actual vectors of data, virtual SIPs may be represented by the seeds of a portable random number generator along with F-Inverse functions and a correlation matrix to be used in Cholesky factorization or other methods of generating statistical dependence on the host machine. The only disadvantage of virtual SIPs is that currently they cannot preserve non-linear statistical relationships between variables. However, virtual SIPs may be supplemented by actual data SIPs where required in such instances.

5.1 Extending the 2.0 Standard for Virtual SIPs

The 2.0 SIPmath standard has a “Type” property, which refers to the encoding format to implement vectors of actual data (for example, CSV). To implement virtual SIPs, we introduce a new Type, F-Inverse.

The F-Inverse type requires both a Subtype and a Uniform random number Generator. The Subtype may include metalog, Excel .INV (native F Inverse function in Excel), R (Inverse), etc. The Uniform Generator may include HDR, Mersenne Twister, or others.

After these parameters are specified, the properties of the chosen Generator and F-Inverse are relied upon for further specification.

5.2 Metalog Virtual SIPs

Figure 4 displays a metalog virtual SIP for an idiosyncratic uncertainty such as the labor hours per unit for a given job in the use case above. This one is a simple lower bounded metalog, which is analogous to a lognormal distribution but with much more shape flexibility. The string “_r_” is replaced by the output of the specified uniform random number generator when the string is instantiated in the host environment. Figure 5 displays a metalog virtual SIP for a global uncertainty such as hourly labor rate in the above use case. Note that example has fewer terms than the one in Figure 4, and _r_ has been replaced by _r,s4321,123,456_ denoting that a multidimensional seed of 4321, 123, 456 has been specified for use with the random number generator as part of the virtual SIP itself. This will ensure that any model referring to uncertain labor rates will not only use the same distribution, but the same outcome on each trial. This will preserve the additive property of the outputs of any models using this variable as we will describe below.

$$=0 + \text{EXP}(6.4693716402351 + -0.151397454836427 * \text{LN}(_r_ / (1 - _r_)) + -0.494136242104418 * (_r_ - 0.5) * \text{LN}(_r_ / (1 - _r_)) + 1.41905625002059 * (_r_ - 0.5) + 2.53239503239294 * (_r_ - 0.5)^2 + 1.50475296232577 * (_r_ - 0.5)^2 * \text{LN}(_r_ / (1 - _r_)) + -4.21909049140061 * (_r_ - 0.5)^3 + 2.15043956616134 * (_r_ - 0.5)^3 * \text{LN}(_r_ / (1 - _r_)) + -10.9472654560604 * (_r_ - 0.5)^4 + 0.733439781442058 * (_r_ - 0.5)^4 * \text{LN}(_r_ / (1 - _r_)))$$

Figure 3: An idiosyncratic metalog virtual SIP.

$$=15 + \text{EXP}(\text{LN}(85 - 15) + (1/2) * (\text{LN}((1 - 0.05)/0.05))^2 - 1 * \text{LN}((120 - 15)/(45 - 15)) * \text{LN}(_r,s4321,123,456_ / (1 - _r,s4321,123,456_)) + ((1 - 2 * 0.05) * (\text{LN}((1 - 0.05)/0.05)))^2 - 1 * \text{LN}(((120 - 15) * (45 - 15)) / (85 - 15)^2) * (_r,s4321,123,456_ - 0.5) * \text{LN}(_r,s4321,123,456_ / (1 - _r,s4321,123,456_)))$$

Figure 4: A global metalog virtual SIP.

Figure 5 displays the full virtual SIP library and metadata for the use case example. It includes the average, the count of data elements upon which the metalog was based, and the initials of the analyst who created the metalog. Additional metadata includes 100 histogram bins, which are available analytically for metalogs, and may be displayed as graphs in the host environment as shown in the simulation input cells labeled 2 in Figure 2 and in Figure 6.

PM_Type	F_Inverse		Metadata	Index
PM_Subtype	Metalog		avg	1
PM_RNG	HDR		count	2
			analyst	3
Name	Job1	Job2	Job3	LaborRate
F-inverse	=IF(PM_Ir	=IF(PM_Ir	=IF(PM_Ir	=IF(PM_Ir
avg	703.8564	392.4618	745.2625	84
count	302	563	198	NA
analyst	ASBM	ASBM	ASBM	SLS
Hist Bin Count	100	100	100	100
Histogram data	0	0	0	0
	6.21E-06	1.92E-05	8.79E-05	7.08E-05
	1.24E-05	3.83E-05	0.000151	0.000142
	1.86E-05	7.09E-05	0.000208	0.000212
	:	:	:	:
	3.1E-05	0.000222	0.000333	0.000354
	4.23E-05	0.000479	0.000414	0.00053
	8.98E-05	0.001014	0.000517	0.000739

Figure 2: Metalogs and associated metadata.

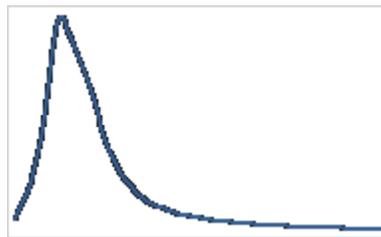


Figure 3: A graph of the metalog distribution generated from metadata.

5.3 The HDR Generator

The discipline of probability management allows simulations to be networked together through stochastic libraries of inputs and outputs. In this environment it may be possible to aggregate the SIPs of simulations developed by modelers who are not even aware of each other’s existence. Improper handling of statistical dependence would invalidate such aggregated results. Because the HDR generator is open source, and is quite new as of the writing of this paper, it is expected that evolution will continue. That being said, there were four necessary design criteria for generator in order to create virtual SIPs with F-Inverse functions as described below.

1. Random Number Management
2. Auditability
3. Accessibility and Speed
4. Quality

$$=(\text{mod}((\text{mod}(\text{mod}(999999999999989,\text{mod}(\text{Trial}*2499997+\text{Var}*1800451+\text{Ent}*2000371,7450589)*4658+7450581)*383,99991)*7440893+\text{mod}(\text{mod}(999999999999989,\text{mod}(\text{Trial}*2246527+\text{Var}*2399993+\text{Ent}*2100869,7450987)*7580+7560584)*17669,7440893))*1343,4294967296)+.5)/4294967296$$

Figure 4: The HDR generator. Trial = trial number, Var = variable ID and Ent = Entity ID.

5.3.1 Random Number Management

There are two considerations in the networked simulation environment that do not occur in conventional simulation within a single computer program.

1. The idiosyncratic uncertainties of local variables on all platforms must be guaranteed to be independent from one another regardless of who created them.
2. Global variables representing such things as earthquakes, commodity prices or other externalities must use the same sequence of trials in all platforms to maintain coherence.

The HDR generator has a multidimensional seed; a variable ID, analogous to the seed in a typical generator; and an Entity ID, which defaults to zero, but which can be assigned to a particular organization or group a bit like IP addresses. There are two additional dimensions, which might be used for such things as departments within an organization, agents, time series elements, etc.

5.3.2 Auditability

To maintain auditability, it is important to be able to view any trial of the simulation instantly. This requires what is known as a counter-based or random access generator. It behaves like a hash function in which you enter the seed(s) and iteration counter, and the result appears as an output without recursion. This was also a required feature in order to make it interactive.

5.3.3 Accessibility and Speed

Because probability management allows managers without statistical experience to utilize SIPs generated by experts on specialized analytical platforms, it was important that the HDR generator work interactively as a single cell formula in Excel, which has upwards of one billion users. It is compatible with Data Table simulation, in which the column input cell, say running from 1 to 1,000 becomes the Trial input in Figure 8. It produces roughly 1,000 variates per one hundredth of a second. The HDR has thus provided a much needed seed-able random number generator for Excel to replace =RAND(), and an early version has been in use by the free SIPmath Modeler Tools available at ProbabilityManagement.org since 2016 (Savage 2016). Of course it works in other environments as well and is an order of magnitude faster in some environments.

5.3.4 Quality

Given the problems with widely adopted random number generators in the past, today there are good test suites. The HDR generator compares favorably with Amazon Web Services and Python on the dieharder tests (Brown 2019), while beating the generators in R and C (Hubbard 2019)

5.4 Fitting Metalogs to Data

Metalogs are somewhat analogous to the Pearson family of probability distributions (Pearson 1895) but have the following desirable properties.

1. They specified by either raw data, or to quantiles of the data, not to abstract parameters such a mean and variance.
2. They have far greater shape flexibility than the Pearson distributions, and can even model multi modal distributions.
3. Like Taylor's series they are governed by a power series which may be arbitrarily extended.
4. Unlike many other distributions they do not require a non-linear optimization to specify parameters, but use a least squares method involving a single matrix operation.

A well-documented Excel template available online (Keelin 2019) may be used directly to fit data and learn more about Metalogs.

6 ENVIRONMENTS

6.1 Current SIPmath 2.0 Standard

The current data formats supported by the SIPmath standard include XML, CSV, JSON, and XLSX. The standard specification document is available for download from ProbabilityManagement.org (Thibault 2016).

6.1.1 SIPmath Applications

SIPmath applications merely run SIPs through a series of calculations by means of vectorization.

6.1.2 Excel

In Excel, as discussed, this may be achieved through the data table function, which can process thousands of SIP trials per keystroke (Savage 2012). The data table functionality may be invoked by hand, but ProbabilityManagement.org has developed the SIPmath Modeler Tools add-in to facilitate the process (Savage 2016). Note that models created with the add-in do not require the add-in to run, and can be shared with any Excel user on Mac or PC as an interactive simulation dashboard.

6.1.3 Other Environments

Environments such as Matlab, R, and Python use vectorized calculations as a matter of course, so SIPs stored in XML may be easily imported, used in calculations, and exported. Colin Smith has developed a SIPmath Package in Python which features the full functionality of the SIPmath tools for Excel and also includes full metalog support (Smith 2019).

6.2 Metals

Metalogs are available online in templates and tools in Excel format (Keelin 2019; ProbabilityManagement.org 2019). They are also supported by Frontline Systems as of the V2017-R2 release and in Analytica Version 5.0 (Analytica 2019).

6.2.1 Fitting Metalogs to Data

There are several ways to fit metalogs to data, and user discretion against over fitting is advised. Keelin (2016) is strongly recommended as well as Excel templates available online (Keelin 2019; ProbabilityManagement.org 2019).

A Python implementation was described above, and an R implementation upon which it was based is available (Faber 2018).

A comparison of metalogs fitted to the same data sets in Excel, R and Python appear in Figures 8, 9 and 10 and demonstrates the effectiveness of a common standard.

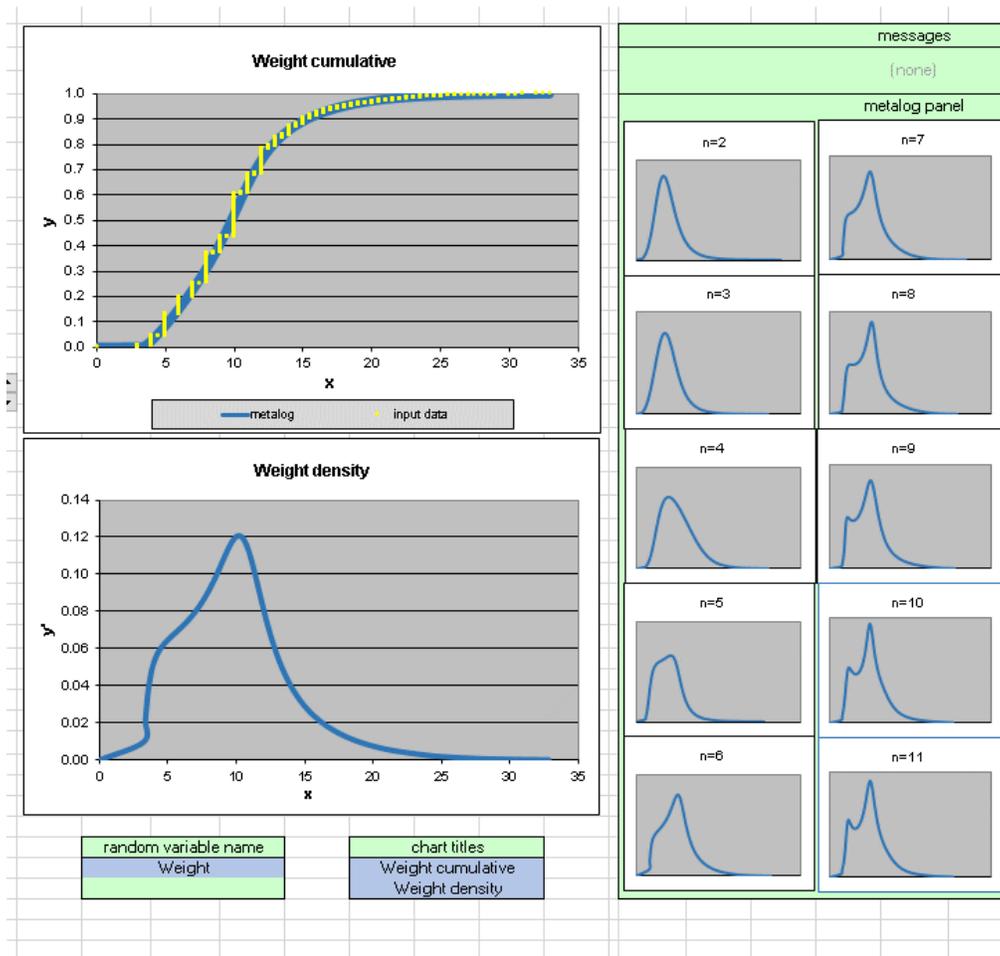


Figure 5: Metalogs in Excel.

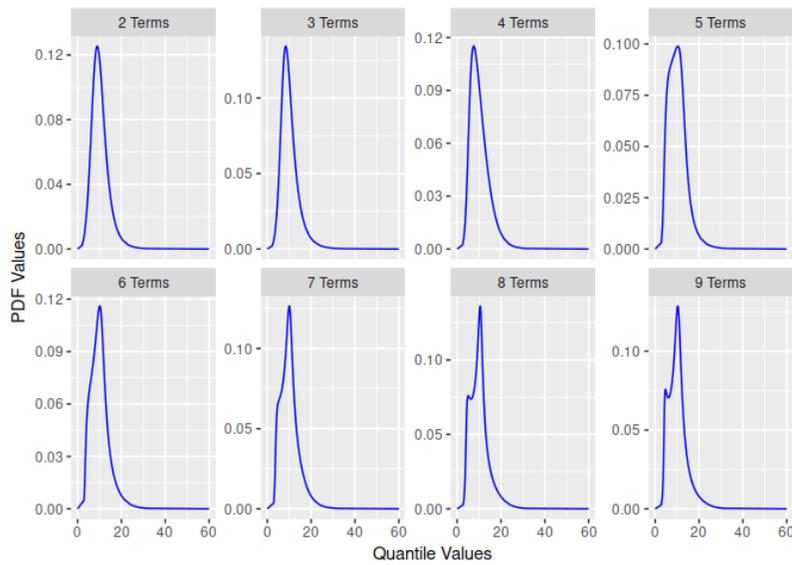


Figure 6: Metalogs in R.

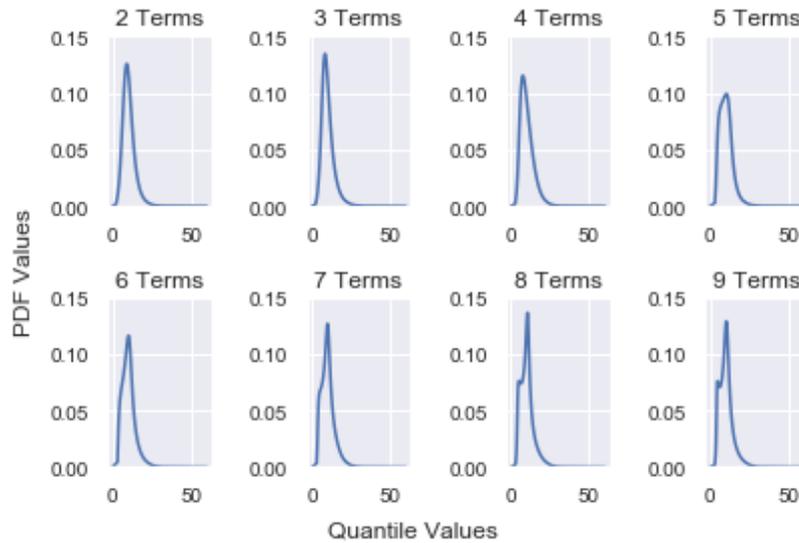


Figure 7: Metalogs in Python.

6.2.2 Metalogs and Machine Learning

Machine learning exposes many probability distributions, which heretofore had no analytical expressions. Because of their shape flexibility metalogs are a good solution to quantifying the results of such techniques as classification trees, as shown in Figure 11.

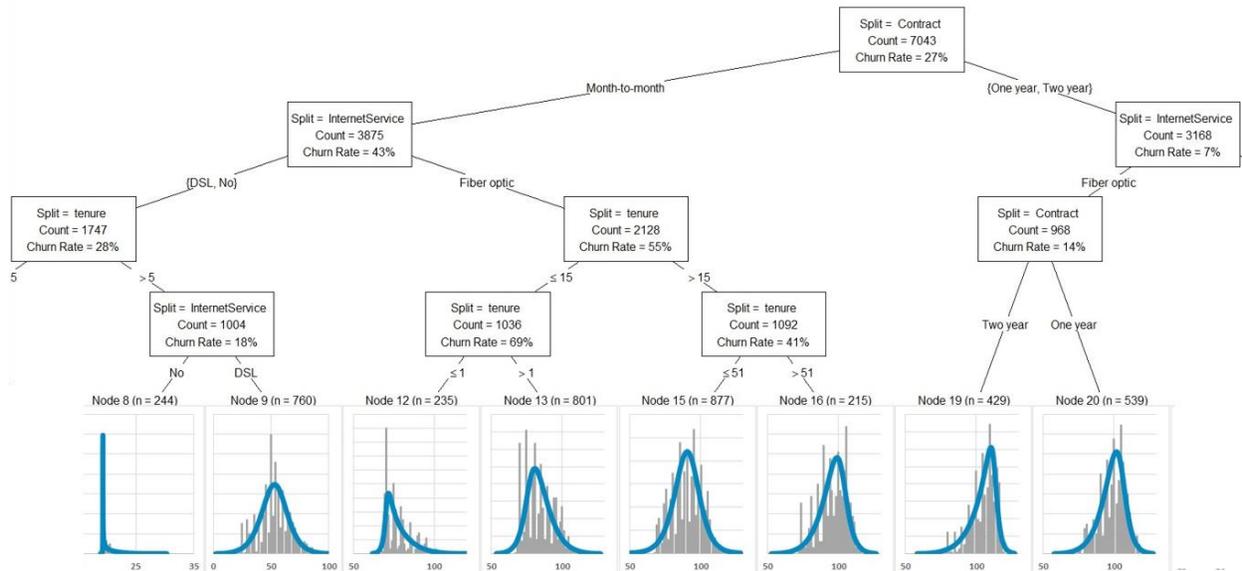


Figure 8: Metalog distributions derived from a classification tree.

7 CONCLUSION

For decades, libraries of realizations of stochastic processes have been used in financial engineering and insurance to aggregate risks. Recent open standards, which store realizations in arrays called SIPs, have allowed such libraries to be shared between applications running on diverse platforms, including Microsoft Excel. Virtual SIPs make use of new forms of random variate generation to vastly reduce the size of such libraries. Attendees of this tutorial are encouraged to bring their laptop computers with Excel to gain hands-on experience.

REFERENCES

- Analytica. 2019. Keelin (MetaLog) Distribution. [https://wiki.analytica.com/index.php?title=Keelin_\(MetaLog\)_distribution](https://wiki.analytica.com/index.php?title=Keelin_(MetaLog)_distribution), accessed 28th April.
- Brown, R., D. Eddebuettel, and D. Bauer. 2019. Dieharder: A Random Number Test Suite, Version 3.31.1. <https://webhome.phy.duke.edu/~rgb/General/dieharder.php>, accessed 28th April.
- Cairo, A. 2016. Download the Datasaurus: Never Trust Summary Statistics Alone; Always Visualize Your Data. <http://www.thefunctionalart.com/2016/08/download-datasaurus-never-trust-summary.html>, accessed 28th April 2019.
- Faber, I. 2018. rMetalog: Implementation of the Metalog Distribution. <https://cran.r-project.org/web/packages/rmetalog/index.html>, accessed 28th April.
- Hubbard, D. 2019. A Simple Pseudo Random Number Generator. <https://probabilitymanagement.org/s/HDR-PRNG-2019>, accessed 2nd May.
- Keelin, T. 2016. "The Metalog Distributions." *Decision Analysis* 13(4).
- Keelin, T. 2019. The Metalog Distributions. <http://metalogs.org/>, accessed 28th April.
- Kirmse, M. and S. Savage. 2014. "Probability Management 2.0." *ORMS Today* 41(5):30-33.
- Pearson, K. 1895. "Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material." *Philosophical Transactions of the Royal Society of London. A*, 186:343-414.
- Savage, S. 2016. "Monte Carlo for the Masses." *Analytics Magazine* September/October 2016:64-68.
- Savage, S. L. 2012. "Distribution Processing and the Arithmetic of Uncertainty." *Analytics Magazine* November/December 2012:28-32.

- Savage, S. L. and J. M. Thibault. 2015. “Towards a Simulation Network, or, The Medium is the Monte Carlo (With Apologies to Marshall McLuhan).” In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W.K.V. Chan, I. Moon, T.M.K. Roeder, C. Macal, and M.D. Rossetti. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Savage, S. L. 2019. Datasaurus Arithmetic, <https://probabilitymanagement.org/s/Datasaurus-Arithmetic.xlsx>, accessed 2nd May.
- Smith, C. 2019. The SIPmath package in Python. <https://github.com/colsmith/sipmath>, accessed 28th April.
- Thibault, M. 2016. “SIP Standard Specification Version 2.1.1.” <https://www.probabilitymanagement.org/s/SIP-Standard-Version-211.pdf>, accessed 28th April 2019.

AUTHOR BIOGRAPHIES

SAM L SAVAGE is Executive Director of ProbabilityManagement.org, a 501(c)(3) nonprofit devoted to the communication and calculation of uncertainty. The organization has received funding from Chevron, Lockheed Martin, PG&E, and others, and he is joined on the board by Harry Markowitz, Nobel Laureate in Economics. Dr. Savage is author of *The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty* (John Wiley & Sons, 2009, 2012), and is an Adjunct Professor in Civil and Environmental Engineering at Stanford University. He is the inventor of the Stochastic Information Packet (SIP), an auditable data array for conveying uncertainty. He received his Ph.D. in computational complexity from Yale University. He can be reached at sam@probabilitymanagement.org.

SHAUN DOHENEY is the Chair of Resources and Readiness Applications at ProbabilityManagement.org, a nonprofit devoted to the communication and calculation of uncertainty. He holds a B.S. in Mathematics from California State University, Long Beach, an M.S. in Operations Analysis from the Naval Postgraduate School, a Graduate Certificate in Data Analytics from George Mason University, is a Certified Data Analytics Professional (CAP®), and is a Project Management Professional (PMP®). As a Marine Corps Lieutenant Colonel (Retired) and Marine Operations Research Analyst, he performed quantitative and qualitative analyses and evaluations across major DoD decision support processes. His past projects featured optimization, multiple-objective decision analysis, quantitative risk analysis, discrete event simulation, and survey design and analysis. His more recent work has focused on guiding adoption of analytic methods associated with risk and readiness, and optimizing allocation of resources across operational scenarios to inform portfolio funding decisions over a multi-year horizon. He can be reached at shaun@probabilitymanagement.org.

COLIN SMITH is a Data Scientist at ProbabilityManagement.org, a nonprofit devoted to the communication and calculation of uncertainty. Colin has a BS in Finance from New York University’s Stern School of Business and a MSc in Data Science from The University of Edinburgh. Colin’s current work involves creating open-source software solutions for probabilistic modeling and simulation, as well as fitting distributions to data. He can be reached at colin.smith.a@gmail.com.