

Features of SIMTOOLS 3.4:

SIMTOOLS.XLAM adds to Excel the following 32 statistical functions, listed in six categories. Optional parameters are shown in *italics*.

Inverse cumulative-probability functions. These functions, along with Excel's NORMINV or NORM.INV function, can be used for generating random variables when the first parameter (called "probability" or "randprob") is generated by a RAND() or CORAND function:

- BETINV(probability, mean, stdevn, *lowerbound, upperbound*) returns inverse cumulative values for a beta random variable, parameterized by its mean and standard deviation. When the first parameter is a RAND, BETINV yields a bounded random variable. (Default lower and upper bounds are 0 and 1.)
- BINOMINV(probability, n, p) returns inverse cumulative values for a binomial random variable. When the first parameter is a RAND, BINOMINV yields a bounded integer random variable between 0 and n, with mean $n \cdot p$.
- DISCRINV(randprob, values, probabilities) returns inverse cumulative values for a discrete random variable. When the first parameter is a RAND, DISCRINV returns a discrete random variable with possible values and corresponding probabilities in the given ranges.
- EXPOINV(probability, mean) returns inverse cumulative values for an exponential random variable. When the first parameter is a RAND, EXPOINV yields a nonnegative random variable (often used for random waiting times).
- GAMINV(probability, mean, stdevn) returns inverse cumulative values for a gamma random variable, parameterized by its mean and standard deviation. When the first parameter is a RAND, GAMINV yields a nonnegative random variable.
- GENLINV(probability, quart1, quart2, quart3, *lowest, highest*) returns inverse cumulative values for a generalized-lognormal random variable that has 25% probability below the quart1 value (the top of the first quartile), 50% probability below quart2, and 75% probability below quart3. A generalized-lognormal random variable is a constant plus or minus a lognormal random variable. When the first parameter is a RAND(), GENLINV yields a random variable which could be positive or negative, but is bounded on the side of the narrower quartile range. If optional lowest and highest values are specified (satisfying $\text{lowest} < \text{quart1} < \text{quart2} < \text{quart3} < \text{highest}$), then values of the generalized-lognormal random variable are adjusted as necessary to keep GENLINV within these bounds (increasing to the lowest value from below it, decreasing to the highest value from above it).
- LAPLAINV(probability, mean, stdevn) returns inverse cumulative values for a Laplace (or double-exponential) random variable, parameterized by its mean and standard deviation. When the first parameter is a RAND, LAPLAINV yields a Laplace random variable which has more probability in the tails than a Normal with the same parameters.
- LNORMINV(probability, mean, stdevn) returns inverse cumulative values for a lognormal random variable, parameterized by its mean and standard deviation. When the first parameter is a RAND, LNORMINV yields a nonnegative random variable.

- POISINV(probability, mean) returns inverse cumulative values for a Poisson random variable. When the first parameter is a RAND, POISINV yields a nonnegative integer random variable.
- TRIANINV(probability, lowerbound, mostlikely, upperbound) returns inverse cumulative values for a random variable with a triangular probability density. When the first parameter is a RAND, TRIANINV yields a bounded random variable.
- XTREMINV(probability, mean, stdevn) returns inverse cumulative values for an extreme-value (or Gumbel) random variable, parameterized by its mean and standard deviation. When the first parameter is a RAND, XTREMINV yields a random variable that may be positive or negative. (If W is a Weibull random variable then $-\text{LN}(W)$ has this extreme-value distribution.)

Functions for working with correlations among random variables:

- CORAND(CorrelArray, RandSource), entered as an array formula in a range of cells in a row, returns RANDom values for making random variables that have correlations as in the given CorrelArray. (See also NORMIZE.) Each value in an array returned by CORAND is like a RAND in that it is generated uniformly between 0 and 1, but values in a CORAND array are not independent. For any integer n greater than 1, the CorrelArray parameter may be a square n-by-n array of correlations for n random variables (as returned by MCOLRELS), and then CORAND returns an array of n correspondingly correlated values. Such an n-by-n correlation array should be symmetric and must have ones on the diagonal. The CorrelArray parameter can also be a single number, in which case CORAND functions as if the CorrelArray parameter were a 2-by-2 array, returning two random values with the given correlation. When CorrelArray is a number and the optional RandSource parameter is a reference to another cell that contains a RAND or CORAND formula, the CORAND function in one cell returns a uniform random value that is correlated with the RandSource cell according to the CorrelArray number. (For more about CORAND, see technical notes below.)
- MCOLRELS(dataRange) returns the matrix of correlation coefficients among the columns of the data range. If the data range has n columns, then MCOLRELS should be entered as an array formula in a square n-by-n range.
- MIDRAND(correlation, givenCoValue) returns the conditional median of a CORAND given another CORAND's value and the correlation between them. Used for subjective assessment of correlations.
- MSQRT(squarearray) returns a lower-triangular matrix square root (or Cholesky factor) of a given square array. An array function.
- NORMIZE(datacolumn) returns an array of normalized rank values, sampled from the standard normal distribution (at fractile medians), and rank-ordered as the data column. When CORANDs are used to make continuous random variables that are not normal, the correlation parameters of CORAND should be normalized rank correlations, which can be estimated from data by applying NORMIZE to each data series and then computing the correlations among these normalized arrays.
(Note: NORMIZE may return an error message when it is entered into a range that is not the same size as the data column.)
- PRODS(values) multiplies each pair of values in the given range and returns the products as a square array. The values must be given in one row or one column. To illustrate the

use of this function, suppose that a range named "correls" contains a square symmetric array that lists the correlations of the random returns-per-share of various stocks, a range named "stdevns" lists the standard deviations of these stock returns, and a range named "shares" lists the numbers of shares of these stocks in some investment portfolio; then the standard deviation of the total returns of the portfolio is

$$\text{SUMPRODUCT}(\text{PRODS}(\text{shares}), \text{PRODS}(\text{stdevns}), \text{correls})^{0.5}$$

Functions for decision analysis:

- ARGMAX(labels, values, *testCells*, *criterion*) returns a label corresponding to a maximal value, where the corresponding test cells (if any) match the criterion.
- CE(incomes, RiskTolConst, *RiskTolSlope*) returns the certainty equivalent of a random draw from the incomes range, for a decision-maker with constant risk tolerance (or linear risk tolerance when the optional RiskTolSlope parameter is used). When the risk tolerance is positive number, the CE value is between the minimum and the average of the income values, and the CE becomes closer to the average as the risk tolerance becomes larger. If the risk tolerance is a negative constant (denoting risk-seeking behavior), then the CE value is between the average and the maximum of the income values. If the RiskTolConst parameter is exactly equal to 0 and the RiskTolSlope is omitted, then the CE function returns the average of the income values. (The CE is computed by converting the incomes to utility values by UTIL, averaging these utility values, and converting this average utility back into monetary units by UINV. Nonnumerical entries in the incomes range are ignored. The CE's dependence on the risk tolerance is discontinuous when the risk tolerance is zero, because a slightly lower risk tolerance yields a CE near the maximal income, and a slightly higher risk tolerance yields a CE near the minimal income. If the RiskTolSlope parameter is used then a zero or negative risk tolerance generates an error.)
- RISKTOL(HighIncome, LowIncome, CertainEquiv) returns the constant risk tolerance such that a lottery paying either the high or low income, each with probability 1/2, has the assessed certainty equivalent value.
- UTIL(income, RiskTolConst, *RiskTolSlope*) returns a utility value of monetary income, for a decision-maker with constant or linear risk tolerance.
- UINV(utility, RiskTolConst, *RiskTolSlope*) returns the monetary certainty equivalent of an expected utility from the UTIL function with the same risk-tolerance parameters.

Functions for analyzing discrete probability distributions:

- CEPR(values, probabilities, RiskTolConst, *testCells*, *criterion*) returns the certainty equivalent, for a decision-maker with constant risk tolerance, of a random income drawn from the specified values according to the corresponding probabilities, conditional on the event where the corresponding test cells (if any) match the criterion. When the RiskTolConst parameter is 0, the CEPR function returns the conditional expected value.
- CORRELPR(values1, values2, probabilities) returns the correlation coefficient for a discrete probability distribution with corresponding values of two random variables.
- COVARPR(values1, values2, probabilities) returns the covariance for a discrete probability distribution with corresponding values of two random variables.

- STDEVPR(values, probabilities) returns the standard deviation for a discrete probability distribution with corresponding values of a random variable.

Functions for regression analysis:

- REGRESSN(XDataRange, YDataRange) actively returns multiple regression output, for predicting Y as a random variable that depends linearly on the explanatory X variables. REGRESSN should be entered as an array formula in a range with 7 rows and as many columns as the X data range.
- YHATSTE(XDataRange, NewXRow, RegressnStdErr) returns the standard error of the estimated conditional mean of Y (often called Y-hat) at the given new X row in a multiple regression.

Functions for randomly generating discrete distributions:

- DIRICH(alphaArray, RandSource), entered as an array formula in a range that has the same size as the alpha array, returns Dirichlet random fractions that sum to 1. These random fractions have means that are proportional to the corresponding values in the alpha array. Higher alpha-array values yield less variance. (RANDOM inputs to drive the random DIRICH values can be provided externally in an optional RandSource range that is the same size as the alpha-array. Otherwise, DIRICH gets random inputs automatically by implicit calls to Excel-VBA's random number generator.)
- DIRALPHA(dataRange), entered as an array formula in a range of cells in one row, returns estimated alpha parameters for a Dirichlet distribution. Each row of the data range should contain nonnegative numbers that sum to 1.
- LGT(x) is the logistic distribution. It transforms random variables from a logit model to fractions between 0 and 1 or (as an array formula) to discrete probability distributions proportional to the EXP(x(i)) values. In the array-formula usage, x must be an array of numbers in a row, and LGT must be entered into a similar array.
- LGTINV(p) is the inverse of the logistic or logit function. Applied to a fraction or (as an array formula) to a discrete probability distribution, LGTINV returns log-odds ratios for a logit model. In the array-formula usage, p must be an array of probability values in a row, and LGTINV must be entered into a similar array.
- SHUFFLE(n, RandSource), entered as an array formula in a range of n cells in one row, returns a random ordering of the numbers from 1 to n. When entered into a row range of fewer than n cells, this function generates random samples from {1,...,n} without replacement. The values in a given range of n cells in a row can be shuffled by entering the array formula =INDEX(givenrange,1,SHUFFLE(n)) into another n cells in a row. (An optional RandSource cell containing a RAND value can be used to determine the outcome of SHUFFLE. When RandSource is omitted, SHUFFLE uses instead an implicit call to Excel-VBA's random number generator.)

SIMTOOLS.XLAM also adds **two Lookup-Reference functions**:

- FORMULAS(auditRange) actively displays the current formulas in the audit range, as an array in a column. To display all formulas, the FORMULAS function should be entered

as an array formula into a range in one column that includes more cells than there are formulas in the audit range. The words "FORMULAS FROM RANGE..." should appear below the last displayed formula.

- FORMRC(auditCell) returns the formula of the designated cell in R1C1-style notation (with braces on array formulas). If the cell does not contain a formula, then FORMRC returns the cell's value (or a zero-length string for a blank cell)

SIMTOOLS.XLAM also adds **four macro procedures** on the Simtools tab in the Ribbon:

- SIMULATION TABLE, in a selected range, tabulates outputs from repeated recalculations of a Monte Carlo simulation model. The outputs to be tabulated should be in the top row of the selected range, but the top-left cell of this selected range should be unused. Recalculated values of the simulation outputs will fill the lower rows of the selected range, with each row containing the output values from an independent recalculation of the simulation model. The left column of the selected range is used for a percentile index, which can be useful for making cumulative-distribution charts after the output data is sorted (but the Simulation Table procedure itself does not sort the output data).
- ITERATIVE PROCESS iteratively copies values to a state range from an update range, while tabulating output.
- COMBINE ROWS makes all combinations of rows from selected ranges. It can be used to make a table of possible event-sequences that is equivalent to a decision tree.
- FORMULA LIST is a procedure for auditing the formulas that are used in a selected range. Under this procedure, the formulas from all cells in the audit range can be listed as text in any selected column. If a cell is part of an array formula then the range of this array is listed with the formula. The list can also include the names that are defined in the workbook.

If the selected output range is the same as the audit range, then FORMULA LIST instead does a formula/text toggle, inserting a single quote ['] before the initial equals sign [=] in each formula, to display formulas in place as text. A second application of this formula/text toggle removes these quotes, to restore the formulas. Array formulas are not affected by a formula/text toggle.

More technical notes about CORAND: With an n-by-n CorrelArray parameter, CORAND's optional RandSource parameter can be a reference either to a single cell or to n cells in a row. RandSource cells should contain independent uniform random values, as generated by RAND or CORAND. When RandSource is a single cell, this cell's value is returned in CORAND's output array as the last value on the right. When RandSource is a range of n cells in a row, CORAND looks in these cells for all the random inputs that it needs to generate its n correlated outputs, preserving the rightmost value but transforming the others to generate appropriately correlated random outputs. This usage (along with a similar RandSource usage for the DIRICH and SHUFFLE function) can be applied when you want to work with another simulation add-in's random number generator. When RandSource is omitted, CORAND automatically generates its n random values by transforming the results of n implicit calls to Excel-VBA's random number generator. The seed for this VBA random-number generator is randomly reset (with a reference to the computer's clock) whenever the SimulationTable or IterativeProcess macro procedure is run.

Installation:

Simtools is available online at <http://home.uchicago.edu/~rmyerson/simtools.xlam>

To install Simtools, download the *Simtools.xlam* file from this site, and save this file in a directory of your choice on your hard drive. (You could choose to save it wherever Excel Add-Ins are stored on your computer, but this is not necessary.) When saving, make sure that the file's name has the correct *xlam* extension that signifies an Excel add-in.

Then to install Simtools in Excel, use the File:Options:Add-Ins:Manage ExcelAddIns:Go command sequence on a Windows machine, or the the Tools:Add-ins command sequence on a Mac, and select the "Simtools" option in the "Add-Ins available" dialogue box or browse to find Simtools.xlam where you saved it. After installation, "SimTools" should appear as tab on the Ribbon in Excel. If it does not appear, you may need to add the folder where you put simtools.xlam as a trusted location for Excel by File: Options: TrustCenter: TrustCenterSettings: TrustedLocations: AddNewLocation.

This documentation is available at <https://home.uchicago.edu/~rmyerson/addins.htm> .