Appendix A: Sampling Methods

Sampling is used in an @RISK simulation to generate possible values from probability distribution functions. These sets of possible values are then used to evaluate your Excel worksheet. Because of this, sampling is the basis for the hundreds or thousands of "what-if" scenarios @RISK calculates for your worksheet. Each set of samples represents a possible combination of input values which could occur. Choosing a sampling method affects both the quality of your results and the length of time necessary to simulate your worksheet.

What is Sampling?

Sampling is the process by which values are randomly drawn from input probability distributions. Probability distributions are represented in @RISK by probability distribution functions and sampling is performed by the @RISK program. Sampling in a simulation is done repetitively, with one sample drawn every iteration from each input probability distribution. With enough iterations, the sampled values for a probability distribution become distributed in a manner which approximates the known input probability distribution. The statistics of the sampled distribution (mean, standard deviation and higher moments) approximate the true statistics input for the distribution. The graph of the sampled distribution will even look like a graph of the true input distribution.

Statisticians and practitioners have developed several techniques for drawing random samples. The important factor to examine when evaluating sampling techniques is the number of iterations required to accurately recreate an input distribution through sampling. Accurate results for output distributions depend on a complete sampling of input distributions. If one sampling method requires more iterations and longer simulation runtimes than another to approximate input distributions, it is the less "efficient" method.

The two methods of sampling used in @RISK — Monte Carlo sampling and Latin Hypercube sampling — differ in the number of iterations required until sampled values approximate input distributions. Monte Carlo sampling often requires a large number of samples to approximate an input distribution, especially if the input distribution is highly skewed or has some outcomes of low probability. Latin Hypercube sampling, a new sampling technique used in @RISK, forces the samples drawn to
correspond more closely with the input distribution and thus converges faster on the true statistics of the input distribution.

**Cumulative Distribution**

It is often helpful, when reviewing different sampling methods, to first understand the concept of a cumulative distribution. Any probability distribution may be expressed in cumulative form. A cumulative curve is typically scaled from 0 to 1 on the Y-axis, with Y-axis values representing the cumulative probability up to the corresponding X-axis value.

In the cumulative curve above, the .5 cumulative value is the point of 50% cumulative probability (.5 = 50%). Fifty percent of the values in the distribution fall below this median value and 50% are above. The 0 cumulative value is the minimum value (0% of the values will fall below this point) and the 1.0 cumulative value is the maximum value (100% of the values will fall below this point).

Why is this cumulative curve so important to understanding sampling? The 0 to 1.0 scale of the cumulative curve is the range of the possible random numbers generated during sampling. In a typical Monte Carlo sampling sequence, the computer will generate a random number between 0 and 1 — with any number in the range equally likely to occur. This random number is then used to select a value from the cumulative curve. For the example above, if a random number of .5 was generated during sampling, the value sampled for the distribution shown would be X1. As the shape of the cumulative curve is based on the shape of the input probability distribution, more likely outcomes will be more likely to be sampled. The more likely outcomes are in the range where the cumulative curve is the "steepest".

What is Sampling?
Monte Carlo Sampling

Monte Carlo sampling refers to the traditional technique for using random or pseudo-random numbers to sample from a probability distribution. The term Monte Carlo was introduced during World War II as a code name for simulation of problems associated with development of the atomic bomb. Today, Monte Carlo techniques are applied to a wide variety of complex problems involving random behavior. A wide variety of algorithms are available for generating random samples from different types of probability distributions.

Monte Carlo sampling techniques are entirely random — that is, any given sample may fall anywhere within the range of the input distribution. Samples, of course, are more likely to be drawn in areas of the distribution which have higher probabilities of occurrence. In the cumulative distribution shown earlier, each Monte Carlo sample uses a new random number between 0 and 1. With enough iterations, Monte Carlo sampling “recreates” the input distributions through sampling. A problem of clustering, however, arises when a small number of iterations are performed.

In the illustration shown here, each of the 5 samples drawn falls in the middle of the distribution. The values in the outer ranges of the distribution are not represented in the samples and thus their impact on your results is not included in your simulation output.

Clustering becomes especially pronounced when a distribution includes low probability outcomes which could have a major impact on your results. It is important to include the effects of these low probability outcomes. To do this, these outcomes must be sampled. But, if their probability is low enough, a small number of Monte Carlo iterations may not sample sufficient quantities of these outcomes to accurately represent their probability. This problem has led to the
development of stratified sampling techniques such as the Latin Hypercube sampling used in @RISK.

**Latin Hypercube Sampling**

Latin Hypercube sampling is a recent development in sampling technology designed to accurately recreate the input distribution through sampling in fewer iterations when compared with the Monte Carlo method. The key to Latin Hypercube sampling is stratification of the input probability distributions. Stratification divides the cumulative curve into equal intervals on the cumulative probability scale (0 to 1.0). A sample is then randomly taken from each interval or "stratification" of the input distribution. Sampling is forced to represent values in each interval, and thus, is forced to recreate the input probability distribution.

![Five Iterations of Latin Hypercube Sampling](image)

In the illustration above, the cumulative curve has been divided into 5 intervals. During sampling, a sample is drawn from each interval. Compare this to the 5 clustered samples drawn using the Monte Carlo method. With Latin Hypercube, the samples more accurately reflect the distribution of values in the input probability distribution.

The technique being used during Latin Hypercube sampling is "sampling without replacement". The number of stratifications of the cumulative distribution is equal to the number of iterations performed. In the example above there were 5 iterations and thus 5 stratifications were made to the cumulative distribution. A sample is taken from each stratification. However, once a sample is taken from a stratification, this stratification is not sampled from again — its value is already represented in the sampled set.

How does sampling within a given stratification occur? In effect, @RISK chooses a stratification for sampling, then randomly chooses value from within the selected stratification.
When using the Latin Hypercube technique to sample from multiple variables, it is important to maintain independence between variables. The values sampled for one variable need to be independent of those sampled for another (unless, of course, you explicitly want them correlated). This independence is maintained by randomly selecting the interval to draw a sample from for each variable. In a given iteration, Variable #1 may be sampled from stratification #4, Variable #2 may be sampled from stratification #22, and so on. This preserves randomness and independence and avoids unwanted correlation between variables.

As a more efficient sampling method, Latin Hypercube offers great benefits in terms of increased sampling efficiency and faster runtimes (due to fewer iterations). These gains are especially noticeable in a PC based simulation environment such as @RISK. Latin Hypercube also aids the analysis of situations where low probability outcomes are represented in input probability distributions. By forcing the sampling of the simulation to include the outlying events, Latin Hypercube sampling assures they are accurately represented in your simulation outputs.

When low probability outcomes are very important it often helps to run an analysis which just simulates the contribution to the output distribution from the low probability events. In this case the model simulates only the occurrence of low probability outcomes — they are set to 100% probability. Through this you will isolate those outcomes and directly study the results they generate.

The concept of convergence is used to test a sampling method. At the point of convergence, the output distributions are stable (additional iterations do not markedly change the shape or statistics of the sampled distribution). The sample mean versus the true mean is typically a measure of convergence, but skewness, percentile probabilities and other statistics are often used as well.

@RISK provides a good environment for testing the speed at which the two available sampling techniques converge on an input distribution. Run an equal number of iterations with each of the sampling techniques while selecting an input distribution @function as a simulation output. Using the built-in Convergence Monitoring capability in @RISK, see how many iterations it takes the percentiles, mean and standard deviation to stabilize. It should be evident that Latin Hypercube sampling converges faster on the true distributions when compared with Monte Carlo sampling.
More About Sampling Techniques

The academic and technical literature has addressed both Monte Carlo and Latin Hypercube sampling. Any of the references to simulation in the Recommended Readings give an introduction to Monte Carlo sampling. References which specifically address Latin Hypercube sampling are included in a separate section.