

Monte Carlo Simulation

The modern version of the Markov Chain Monte Carlo method was invented in the late 1940s by Stanislaw Ulam, while he was working on nuclear weapons projects at the Los Alamos National Laboratory. Immediately after Ulam's breakthrough, John von Neumann understood its importance and programmed the ENIAC computer to carry out Monte Carlo calculations. In 1946, physicists at Los Alamos Scientific Laboratory were investigating radiation shielding and the distance that neutrons would likely travel through various materials. Despite having most of the necessary data, such as the average distance a neutron would travel in a substance before it collided with an atomic nucleus, and how much energy the neutron was likely to give off following a collision, the Los Alamos physicists were unable to solve the problem using conventional, deterministic mathematical methods. This can be applied to business if we have probability of how much will be purchased if a certain number of items are run.

From Microsoft:

Many companies use Monte Carlo simulation as an important part of their decision-making process. Here are some examples.

- General Motors, Proctor and Gamble, Pfizer, Bristol-Myers Squibb, and Eli Lilly use simulation to estimate both the average return and the risk factor of new products. At GM, this information is used by the CEO to determine which products come to market.
- GM uses simulation for activities such as forecasting net income for the corporation, predicting structural and purchasing costs, and determining its susceptibility to different kinds of risk (such as interest rate changes and exchange rate fluctuations).
- Lilly uses simulation to determine the optimal plant capacity for each drug.
- Proctor and Gamble uses simulation to model and optimally hedge foreign exchange risk.
- Sears uses simulation to determine how many units of each product line should be ordered from suppliers—for example, the number of pairs of Dockers trousers that should be ordered this year.
- Oil and drug companies use simulation to value "real options," such as the value of an option to expand, contract, or postpone a project.
- Financial planners use Monte Carlo simulation to determine optimal investment strategies for their clients' retirement.

The =Rand() function

When you type the formula =*RAND()* in a cell, you get a number that is equally likely to assume any value between 0 and 1. Thus, around 25 percent of the time, you should get a number less than or equal to 0.25; around 10 percent of the time you should get a number that is at least 0.90, and so on.

Using a Random Number

I want to make a gambling situation where the number goes up or down by 1 each time a new item is selected. The formula is this:

`=IF(RAND()>0.5,A1+1,A1-1)`

An, I'm going to do this 1000 times using the split screen to get down as far as we need.

View->Split

Copy and paste the formula down 1000 Rows. Now, let's figure out some of the statistical values of this run:

| Average | Min | Max | SD | Final |
|---------------------------------|-----------------------------|-----------------------------|---------------------------------|---------------------|
| <code>=AVERAGE(A1:A1000)</code> | <code>=MIN(A1:A1000)</code> | <code>=MAX(A1:A1000)</code> | <code>=STDEV((A1:A1000))</code> | <code>=A1000</code> |

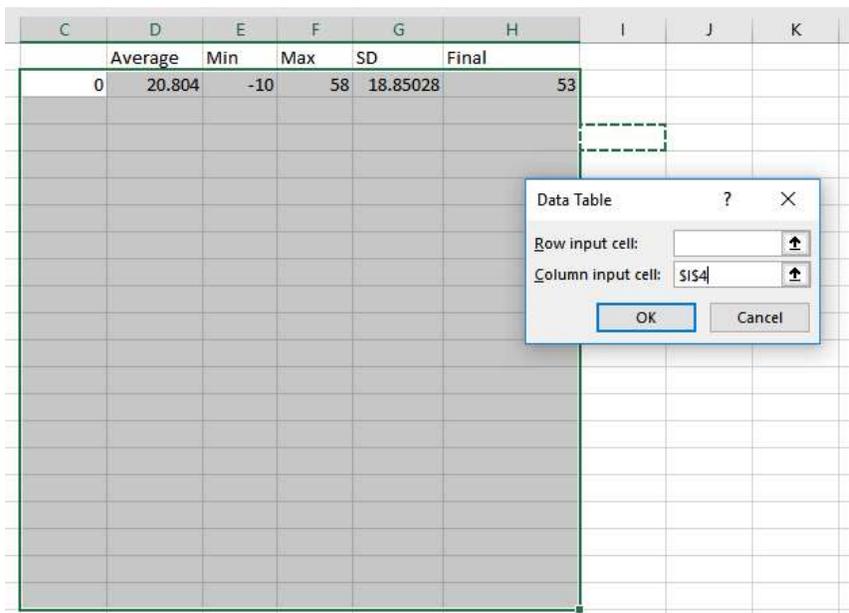
Now, I want to try the simulation again. To recalculate the spreadsheet – use the F9 key.

I can do this lots of times, or I can have the computer run lots of simulations using a Data Table.... I think I will have the computer run 20 simulations at a time.

Data->What if Analysis->Data Table...

Select the formulas and select 19 rows below it. Then, for the column input cell, select one to the right of the selection.

NOTE: I don't know why the syntax is this way. It seems wonky to me!



And then, a new set of tests are run...

| Average | Min | Max | SD | Final |
|---------|-----|-----|----------|-------|
| 23.634 | -19 | 57 | 22.81172 | 37 |
| 31.408 | 0 | 57 | 16.01881 | 47 |
| -13.876 | -40 | 10 | 12.91447 | -35 |
| -9.91 | -24 | 6 | 5.362897 | -7 |
| -21.942 | -46 | 2 | 8.732061 | -23 |
| 5.09 | -29 | 29 | 16.46302 | -23 |
| -15.324 | -33 | 2 | 7.683362 | -15 |
| -6.756 | -21 | 12 | 6.425087 | -7 |
| 3.158 | -17 | 30 | 9.903591 | 29 |
| -6.974 | -25 | 10 | 6.586555 | 5 |
| 10.128 | -18 | 35 | 14.1889 | 29 |
| -28.372 | -71 | 7 | 20.65769 | -69 |
| 36.2 | 0 | 72 | 19.76205 | 67 |
| 2.778 | -16 | 30 | 11.24888 | 27 |
| 8.57 | -9 | 27 | 6.69985 | 1 |
| -16.374 | -33 | 1 | 5.587069 | -13 |
| 14.274 | -11 | 35 | 11.71196 | 7 |
| -28.526 | -79 | 22 | 29.17078 | -75 |
| 18.422 | -5 | 31 | 6.774054 | 31 |
| -14.192 | -48 | 6 | 11.8541 | -47 |

You can't do this by just copying and pasting – because of the TABLE command:

| D | E | F | G | H |
|--------------------|----------------|----------------|--------------------|-------------|
| Average | Min | Max | SD | Final |
| =AVERAGE(A1:A1000) | =MIN(A1:A1000) | =MAX(A1:A1000) | =STDEV((A1:A1000)) | =A1000 |
| =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) |
| =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) |
| =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) |
| =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) | =TABLE(.I4) |

Simulation of values of a discrete random variable

Suppose the demand for a widget is governed by the following discrete random variable:

| Demand | Probability |
|--------|-------------|
| 10,000 | .1 |
| 20,000 | .35 |
| 40,000 | .3 |
| 60,000 | .25 |

How can we have Excel play out, or simulate, this demand for widgets many times? The trick is to associate each possible value of the RAND function with a possible demand for widgets. The following assignment ensures that a demand of 10,000 will occur 10 percent of the time, and so on.

| Demand | Random number assigned |
|--------|------------------------|
| 10,000 | Less than .1 |
| 20,000 | From .1 to .45 |
| 40,000 | From .45 to .75 |
| 60,000 | Greater than .75 |

We can create a run by getting a random number and then using the lookup value to find out what the demand was:

| Trial | Rand | Demand | | | | |
|-------|----------|--------|--|-------|--------|--|
| 1 | 0.177817 | 20000 | | Value | Demand | |
| 2 | 0.946908 | 60000 | | 0 | 10000 | |
| 3 | 0.361813 | 20000 | | 0.1 | 20000 | |
| 4 | 0.030018 | 10000 | | 0.45 | 40000 | |
| 5 | 0.059338 | 10000 | | 0.75 | 60000 | |
| 6 | 0.734428 | 40000 | | | | |

How can a widget company determine how many widgets to produce?

The widget sells for \$4 and the cost of producing each widget is \$1.50. Leftover widgets must be disposed of a \$.20 per widget. How many widgets should be created for maximum profit?

Basically, we simulate each possible production quantity (10,000, 20,000, 40,000, or 60,000) many times (for example, 1000 iterations). Then we determine which order quantity yields the maximum average profit over the 1000 iterations.

Look at the widget.xlsx book to see how that may be!

| Trial | Rand | Demand | Created | Sold |
|-------|---------|--|---------|---------------|
| 1 | =RAND() | =LOOKUP(B3,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | 50000 | =MIN(C3,D3) |
| 2 | =RAND() | =LOOKUP(B4,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D3 | =MIN(C4,D4) |
| 3 | =RAND() | =LOOKUP(B5,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D4 | =MIN(C5,D5) |
| 4 | =RAND() | =LOOKUP(B6,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D5 | =MIN(C6,D6) |
| 5 | =RAND() | =LOOKUP(B7,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D6 | =MIN(C7,D7) |
| 6 | =RAND() | =LOOKUP(B8,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D7 | =MIN(C8,D8) |
| 7 | =RAND() | =LOOKUP(B9,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D8 | =MIN(C9,D9) |
| 8 | =RAND() | =LOOKUP(B10,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D9 | =MIN(C10,D10) |
| 9 | =RAND() | =LOOKUP(B11,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D10 | =MIN(C11,D11) |
| 10 | =RAND() | =LOOKUP(B12,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D11 | =MIN(C12,D12) |
| 11 | =RAND() | =LOOKUP(B13,\$K\$4:\$K\$7,\$L\$4:\$L\$7) | =D12 | =MIN(C13,D13) |

And

| Revenue | Cost | Destruction | Profit |
|---------|----------|--------------------------------|--------------|
| =4*E3 | =D3*1.5 | =IF(D3-C3>0,0.2*(D3-C3),0) | =F3-G3-H3 |
| =4*E4 | =D4*1.5 | =IF(D4-C4>0,0.2*(D4-C4),0) | =F4-G4-H4 |
| =4*E5 | =D5*1.5 | =IF(D5-C5>0,0.2*(D5-C5),0) | =F5-G5-H5 |
| =4*E6 | =D6*1.5 | =IF(D6-C6>0,0.2*(D6-C6),0) | =F6-G6-H6 |
| =4*E7 | =D7*1.5 | =IF(D7-C7>0,0.2*(D7-C7),0) | =F7-G7-H7 |
| =4*E8 | =D8*1.5 | =IF(D8-C8>0,0.2*(D8-C8),0) | =F8-G8-H8 |
| =4*E9 | =D9*1.5 | =IF(D9-C9>0,0.2*(D9-C9),0) | =F9-G9-H9 |
| =4*E10 | =D10*1.5 | =IF(D10-C10>0,0.2*(D10-C10),0) | =F10-G10-H10 |
| =4*E11 | =D11*1.5 | =IF(D11-C11>0,0.2*(D11-C11),0) | =F11-G11-H11 |
| =4*E12 | =D12*1.5 | =IF(D12-C12>0,0.2*(D12-C12),0) | =F12-G12-H12 |
| =4*E13 | =D13*1.5 | =IF(D13-C13>0,0.2*(D13-C13),0) | =F13-G13-H13 |