Monte Carlo Simulation

Nick DeRobertis

December 15, 2020

1 Introduction to Monte Carlo Simulations

- Monte Carlo simulation is the external counterpart to internal randomness

- The core model is still (probably) deterministic, but then we add randomness into the model by randomizing the inputs to the model and running it many times

- Adding this randomness allows us to answer deeper questions about the problem, such as what is the chance of some outcome occurring

- The process is the same as sensitivity or external scenario analysis, just run the model multiple times with different inputs. Only here we are randomly drawing the inputs from distributions

- We will also do some additional analysis on the results from the MC simulation

2 Monte Carlo Investment Returns

- Running Monte Carlo simulations in Excel without the use of an add-in is complex

- Running Monte Carlo simulations in Python just a few lines of code

- If you want to add Monte Carlo simulation to an Excel model, it is easiest to use xlwings to connect Python to run the simulations on your Excel model

- After running the simulations, you must analyze and visualize the output

- A histogram is a good choice for showing the output distribution

- A table of the percentiles of the distribution and values corresponding to those percentiles is a more quantitative way to show the output distribution

- If we have some specific objective or loss in mind, we can determine the probability of achieving the objective/loss

3 Monte Carlo Dividend Discount Model (DDM) Lab Exercise

- This is an example of applying Monte Carlo simulations to a typical model just to better understand the probability distribution of the results

- Be careful that if the growth exceeds the discount rate in the model, it becomes invalid, so some conditions in the model may be needed to address this

4 Formal Introduction to Monte Carlo Simulations

- The process described here to run Monte Carlo simulations may sound very similar to that to run sensitivity analysis, and that's because it is. The only difference is that you randomly pick the input values from distributions with each run of the model rather than having fixed input ranges

- Running the Monte Carlo simulation is not enough. You will have a bunch of outputs, but you must analyze them and visualize them to extract meaning

- The main insights we can draw from analyzing a Monte Carlo simulation relate to the probabilities of certain outcomes in the model. We can also get a deeper picture of the relationships between inputs and outputs in a more complex model where that may not be clear

- The probability table is the quantitative version of plotting the data on a histogram. I would generally recommend including both as the histogram allows quick understanding of the shape of the entire distribution whereas the probability table helps in quantifying the distribution

- The Value at Risk (VaR) represents losing at least some amount with a degree of confidence, e.g. in 95% of periods the portfolio should not lose more than $1,000. The probability table can be interpreted in the same way if the outcome you are analyzing is the gain/loss

- The probability of a certain outcome makes sense when you have some kind of goal in mind, then you can evaluate the probability of achieving that goal. If there is no specific goal in mind, there is no need to carry out this analysis

5 Analyzing Relationships with Monte Carlo Simulations

- The results from the Monte Carlo simulation can be run through multivariate regression or another empirical method to better understand the relationship between inputs and outputs

- Sensitivity analysis gets at the same goal, but sensitivity analysis is a bit more narrow because at most one other input is changing at the same time. With Monte Carlo simulation, all inputs are changing with each run and so if inputs have complex interactions in the model they will be better understood through MC simulation

- The multivariate regression results give the quantitative interpretation of the relationship while scatter plots can help visualize the relationship

6 Applying Monte Carlo Simulation to a Python Model

- It can make sense to set up a separate dataclass for your simulation-specific inputs, or you may add them to the existing dataclass

- Once you start running large numbers of simulations, some unexpected situations may occur in your model such as inputs going negative that were supposed to only be positive, or one input being greater than another when it is supposed to be less. To solve this, we can build functions which produce the random inputs according to the necessary conditions in our model

- Create a function which runs a single simulation, then call that function in a loop over the number of iterations to run all the simulations

- Because we typically have multiple changing inputs and may even have multiple outputs, it is useful to store data as a list of tuples and then create a DataFrame at the end

- It doesn't hurt to take the quantile of the entire DataFrame to see the distributions of the inputs as well. It can be a nice check to make sure your random inputs are working appropriately

- After running a multivariate regression, be sure to add some text interpreting the results

- We can check the standardized coefficients (coef * std) to understand which inputs have the greatest impact on the outputs. Be careful that these results are influenced by your choice of the input distributions. If your input distributions are not reasonable, neither will be the results

7 Applying Monte Carlo Simulation to an Excel Model

- The process for running Monte Carlo simulations in Excel is nearly the same as that in Python when we use Python to run the simulations on the Excel model using xlwings

- The main difference is that we write the inputs into Excel and extract the results using xlwings rather than running Python logic for the core model

- Excel recalculates whenever an input is changed. So writing the inputs in is enough to get the result calculated

- For the analysis, you can either keep the results in Python and follow the process for analyzing the results in Python, or you can output them back to Excel and analyze the outputs there

- Keep in mind that if you visualize the outputs in Excel, next time you run the simulation it will go slow due to the visualizations. Because of this it may be a better idea in general to do the analysis in Python if you have a choice

8 Relationship of Inputs and Outputs in Excel Monte Carlo Simulation

- This continues off the prior lecture to keep the inputs associated with the outputs in the Excel output, and then to do the analysis of how the inputs relate to the outputs

- It is easier to go to DataFrame output into Excel to keep everything together

- We create scatter plots and run a multivariate regression, just as in Python

- You may need to enable the Data Analysis Toolpack add-in in Excel to get access to multivariate regression