Monte Carlo Simulations

Yvan Lengwiler
Faculty of Business and Economics
University of Basel

yvan.lengwiler@unibas.ch

File: prep-02.tex

Contents

This is a very short handout, but it introduces an extremely powerful and widely used idea in finance and in many other fields: Monte Carlo simulations. The coding required can be a little challenging, but you will find many sources online that can get you started. Also, it can be easier if you work in a group of two.

1 What is a Monte Carlo simulation

The Monte Carlo simulation — often abbreviated with MC — is a versatile tool for numerical integration, forecasting, etc. It is rarely the most efficient method, but it is universal and comparatively simple. The abstract idea is this: Use a large number of randomly drawn values generated by the computer and send it through a function you want to know more about. Observe how the function changes the random inputs to reveal the properties of the function.

2 A warm-up example

Before doing the Monte-Carlo of the stock price, we will first simulate the trowing of dices. This will also demonstrate the mechanics of the central limit theorem.

1 Task. Start from the file `'dices.r'`. This script already contains the code for the simulation of one dice. Your task is to extend this to multiple dices. Note that if you trow `d` dices `n` times, this is equivalent to throwing one dice `d` times `n` times. The details of what you need to do are in the script. The script also already contains the code to produce a kernel estimation.

Your answer here:

---

3 Putting it into code

In our setting, we can try to generate random returns for an asset. These random returns should have the same statistical properties as the empirical returns. If we can reliably do that, we can generate a large number of these random returns and string them into paths. This provides a way of generating random future paths for an asset price, assuming that the random properties remain unaltered.

Once you have completed the next task, maybe you can understand that this method is particularly interesting for risk management.

2 Task. Start from the file `'prog-02.Rmd'`. Please complete the code at the indicated places so that simulations of the stochastic returns are generated. The simulated returns should be normally distributed and have the same mean and variance as the empirical data. A simulated path should cover two calendar years and you should generate 1000 paths:

```
sim_years <- 2 # length of simulation in years
n <- 1000 # number of simulations
```

From the random returns, you can also compute the resulting logarithmic level simply by cumulating the returns ($x_t = x_1 + r_2 + \cdots + r_t$). The function `cumsum` is quite helpful here. Also make charts showing the simulated logarithmic and non-logarithmic paths.

Your answer here:

---

3 Question. Remember the discussion about the central limit theorem. Elaborate about the importance of the distribution you use to underlie your simulation. How important is it that the distribution function you use in the Monte Carlo simulation fits the empirical distribution?

Your answer here:

---

4 Ready for sharing in Zoom

Have this ready for the Zoom meeting. Have your notes, charts, programs etc. ready for the Zoom lecture, so that you can share them with us and we can discuss them. If you failed to complete a task, prepare an explanation that makes clear where exactly you struggled.

References