5 Monte Carlo methods

5.1 What are Monte Carlo methods?

Monte Carlo methods, also known as 'stochastic simulation', represent the solution of a problem as a parameter of a hypothetical population, and using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained (Halton, 1970).

Another way to put this, is that Monte Carlo methods are a class of computational algorithms that rely on repeated random sampling to generate a set of hypothetical observations (scenarios) from which to reach some statistical conclusion. Monte Carlo methods allow us to solve problems where a deterministic solution is difficult to achieve by using a probabilistic analog. Sufficiently large experiments will converge towards their 'true' deterministic solution.

5.2 Modelling Asset Price Dynamics

A financial time series is a sequence of observations corresponding to the value of a financial instrument over time. A typical example is the price of a stock, which follows what appears to be a random pattern, usually with an upward trend.

Two important time series properties to consider are 'drift' and 'volatility':

- Drift: Characterized as the change of the average value of a random (stochastic) process (i.e., the drift is the long-term trend of the time series).
- Volatility: Measures the variation over time. In other words, this is the standard deviation of the randomness in our process.
- Volatility to drift relationship: Typically, volatility increases when the drift decreases, and vice versa. This can be observed through the bull and bear market cycles in markets.

5.3 Random Walks

The classic example of a stochastic (random) process is the random walk. For simplicity, we will focus on two types of random walks, arithmetic and geometric. Our arithmetic random walk will follow a Gaussian process (i.e., its randomness will be governed by independent normal distributions).

Some useful notation:

- S_t is the price of our asset at time t
- $w_t \sim \mathcal{N}(0, \sigma)$ is a normally distributed random variable with standard deviation σ.
- $\varepsilon_t \sim \mathcal{N}(0, 1)$ is a standard normal random variable

Consider the price of an asset that can move by an amount that follows a normal distribution with mean μ and volatility (i.e., standard deviation) σ,

$$S_{t+1} = S_t + \mu + w_t.$$

This price movement is called an arithmetic random walk with drift. This process can be represented as the sum of two terms: a deterministic term and a sum of noisy terms

$$\begin{aligned} S_t &= S_{t-1} + \mu + w_{t-1} \\ &= (S_{t-2} + \mu + w_{t-2}) + \mu + w_{t-1} \\ &= S_0 + \mu \cdot t + \sum_{i=0}^{t-1} w_i \end{aligned}$$

We can replace our term w_t with a <u>standard</u> normal random variable that will capture all the randomness in our process

$$\begin{aligned} S_t &= S_{t-1} + \mu + \sigma \cdot \varepsilon_{t-1} \\ &= S_0 + \mu \cdot t + \sigma \sum_{i=0}^{t-1} \varepsilon_i \end{aligned}$$

and since ε_0, ..., ε_{t-1} are independent standard normal random variables, their sum is a normal variable with mean zero and standard deviation equal to

$$\sqrt{\sum_{i=0}^{t-1} 1} = \sqrt{t}$$

Thus, if we model our asset price through an arithmetic random walk, the price at time t is

$$S_t = S_0 + \mu \cdot t + \sigma \cdot \sqrt{t} \cdot \varepsilon$$

The resulting price change over t time periods has a normal distribution with mean $\mu \cdot t$ and standard deviation $\sigma \cdot \sqrt{t}$.

Using an arithmetic random walk works well for short time frames (e.g., to model intraday price changes). However, it has two drawbacks

1. An arithmetic random walk could potentially take on negative values, particularly if we have a low initial value S_0. Asset prices cannot have negative values.

2. It has been shown in both academia and industry that asset prices are non-stationary, meaning that their mean and standard deviation are not constant through time.

5.4 Geometric Random Walks

Throughout this course we have assumed that the returns (and not the prices themselves) follow a normal distribution, $r_t \sim \mathcal{N}(\mu, \sigma)$. Therefore, this assumes the returns follow a Gaussian process

$$r_t = \frac{S_t - S_{t-1}}{S_{t-1}} = \mu + \sigma\varepsilon_{t-1}$$

where, as before, $\varepsilon_0, ..., \varepsilon_{t-1}$ are independent normal random variables. It follows that the price from time $t-1$ to t (where we take a time step $\Delta t = 1$) can be computed as

$$\begin{aligned} S_t &= S_{t-1} \cdot (1 + r_t) \\ &= S_{t-1} \cdot (1 + \mu + \sigma\varepsilon_{t-1}) \\ &= S_{t-1} + \mu \cdot S_{t-1} + \sigma \cdot S_{t-1} \cdot \varepsilon_{t-1}. \end{aligned}$$

From this equation we can calculate the discrete-time change in price,

$$\begin{aligned} \Delta S_t &= S_t - S_{t-1} \\ &= \mu S_{t-1} + \sigma S_{t-1} \; \varepsilon_{t-1} \end{aligned}$$

For an arbitrarily sized time step Δt, this equation becomes,

$$\Delta S_t = \mu S_{t-1} \Delta t + \sigma S_{t-1} \sqrt{\Delta t} \; \varepsilon_{t-1}$$

which, for a sufficiently small time step $\Delta t \approx dt$, gives us the following stochastic differential equation (SDE) in continuous-time

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where W_t is a Brownian motion (or a Wiener process). Note that a Wiener process with time step dt can be discretized as $dW_t \sim \sqrt{dt}\mathcal{N}(0, 1)$.

Under this framework, we have that

$$\int_0^t \frac{dS_t}{S_t} = \mu t + \sigma W_t$$

where $\frac{dS_t}{S_t}$ looks similar to the derivative of $\ln S_t$. However, S_t is a stochastic process, meaning we must use Itô calculus to solve for it. Applying Itô's lemma,

$$d(\ln(S_t)) = \frac{dS_t}{S_t} - \frac{1}{2}\frac{1}{S_t^2}\sigma^2 S_t^2 dt$$

which we can use to solve our problem as follows

$$\begin{aligned}\int_0^t d(\ln(S_t)) + \int_0^t \frac{1}{2}\sigma^2 dt &= \mu t + \sigma W_t\\ \ln(S_t) - \ln(S_0) &= \Big(\mu - \frac{1}{2}\sigma^2\Big)t + \sigma W_t\\ S_t &= S_0 \exp\Big[\Big(\mu - \frac{1}{2}\sigma^2\Big)t + \sigma\sqrt{t}\ \varepsilon\Big]\end{aligned}$$

We can also use the equation above for a given time step dt,

$$S_{t+1} = S_t \exp\Big[\Big(\mu - \frac{1}{2}\sigma^2\Big)dt + \sigma\sqrt{dt}\ \varepsilon_t\Big]$$

Note that the time step dt must be in the same time frame as your estimates μ and σ, i.e., if μ is the yearly return, then $dt = 1$ would be a 1-year time step.

5.5 Simulations

If we wanted to simulate the price path of a single stock, we could do the following:

1. Estimate our parameters μ and σ from historical data. Be sure to select an appropriate frequency for your data (e.g., daily, monthly, etc).

2. Given the frequency of your estimates, select the number of time steps you wish to take.

   - If we have yearly estimates μ_y and σ_y, and we wish to simulate one year's worth of daily price changes, we would have $dt = T/N = 1/252$

     $$S_{t+1} = S_t \exp\Big[\frac{1}{252}\Big(\mu_y - \frac{1}{2}\sigma_y^2\Big) + \frac{\sigma_y \varepsilon_t}{\sqrt{252}}\ \Big]$$

     and we would perform this simulation 252 times for $t = 0, ..., 251$. Every time we take a step we update the current value of S_t and simulate a new value value for $\varepsilon_t \sim \mathcal{N}(0,1)$.

   - On the other hand, if we have weekly estimates μ_w and σ_w and we wish to simulate the price path after one year with four time steps, we would have $dt = T/N = 52/4 = 13$

     $$S_{t+1} = S_t \exp\Big[13\Big(\mu_w - \frac{1}{2}\sigma_w^2\Big) + \sigma_w\sqrt{13}\ \varepsilon_t\Big]$$

     for $t = 0, ..., 3$. As before, we simulate a new value value for $\varepsilon_t \sim \mathcal{N}(0,1)$ every time we take a step.

3. In order to generate multiple scenarios, we can perform Step 2 multiple times. For Monte Carlo simulations we may want to generate at least 1,000 paths. Keep in mind that each path may have multiple time steps. For example, 10,000 one-year price paths with a one-month time step would require you to perform $10,000 \times 12 = 120,000$ simulations.

In most cases we do not need to observe the individual steps taken by each path. Instead, we can take a single step to simulate the price change from S_0 to S_t. This can drastically reduce the number of simulations.

5.6 Simulating Correlated Assets

In general, different assets do not behave independently. This can be modelled by introducing correlation between our simulated Brownian motions.

Suppose we have n correlated assets in our portfolio. Let $\varepsilon \in \mathbb{R}^n$ be a vector of independent $\mathcal{N}(0,1)$ variables. Moreover, let $\rho \in \mathbb{R}^{n\times n}$ be our correlation matrix, where $\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$.

Let us define a new vector

$$\xi = L\varepsilon$$

where $L \in \mathbb{R}^{n\times n}$. Each element of $\xi \in \mathbb{R}^n$ is normally distributed with

$$\mathbb{E}[\xi] = L\mathbb{E}[\varepsilon] = 0$$

since the elements of ε are independent and have unit variance $\mathbb{E}[\varepsilon_i^2] = 1$, then $\mathbb{E}[\varepsilon\varepsilon^T] = I_{n\times n}$, where $I_{n\times n}$ is the identity matrix of size n.

Thus, we have that the covariance matrix of ξ is

$$\mathbb{E}[\xi\xi^T] = \mathbb{E}[L\varepsilon\varepsilon^T L^T] = L\mathbb{E}[\varepsilon\varepsilon^T]L^T = LI_{n\times n}L^T = LL^T$$

To enforce the correlation between our assets, we must find $LL^T = \rho$. Since ρ is a symmetric matrix (and any real symmetric matrix is also a Hermitian matrix), the easiest way to find L is to use a Cholesky factorization in which L is the lower-triangular matrix. In MATLAB, the command is

$$L = \text{chol}(\rho, \text{`lower'})$$

Finally, to simulate our correlated price paths, we now have

$$S_{t+1}^i = S_t^i \exp\left[\left(\mu_i - \frac{1}{2}\sigma_i^2\right)dt + \sigma_i\sqrt{dt}\ \xi_t^i\right]$$

for each asset $i = 1, ..., n$.