Monte Carlo Simulation in Stock

Dr. Jinghan Meng

FINA3351 – Spreadsheet Modelling in Finance

Roadmap for Today

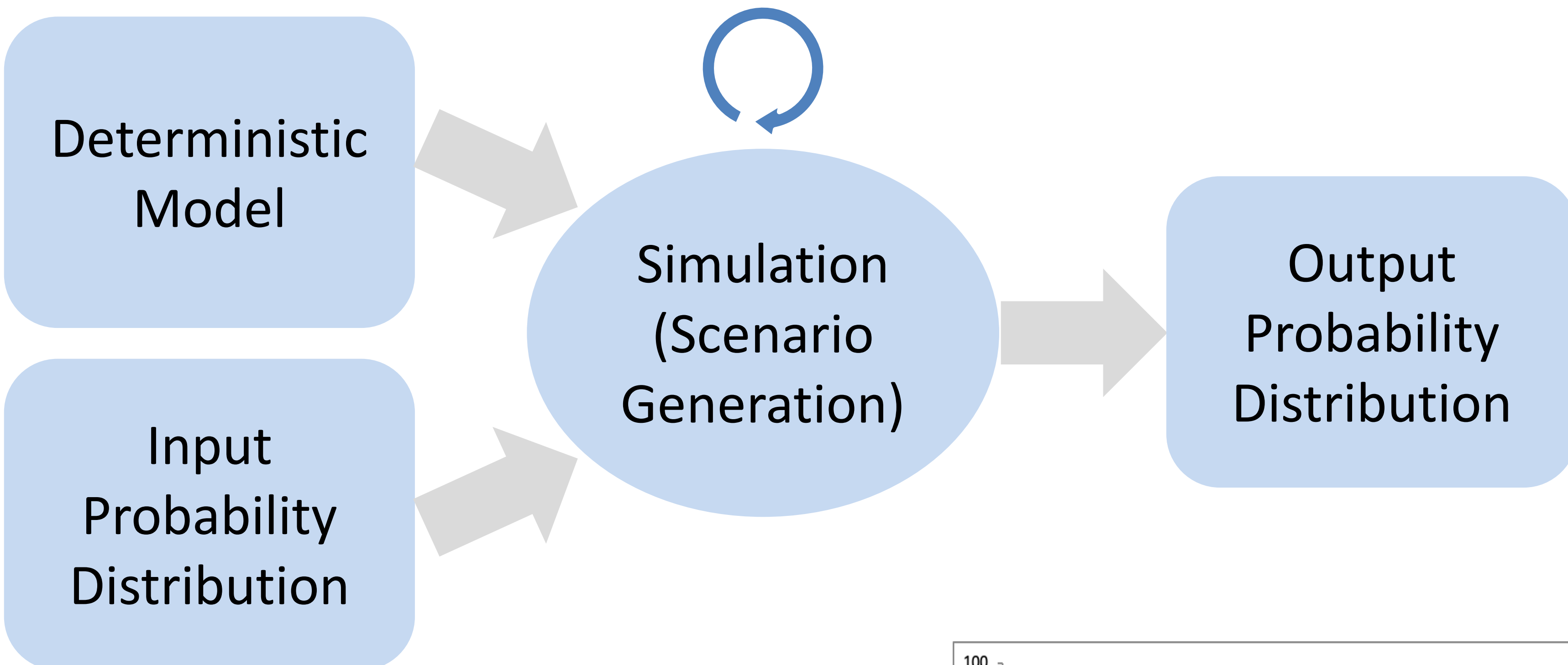
1. Monte Carlo Simulation
2. Random Number Generation
3. Simulate Stock Prices – GBM Model
4. Simulate Portfolio Values – GBM Model

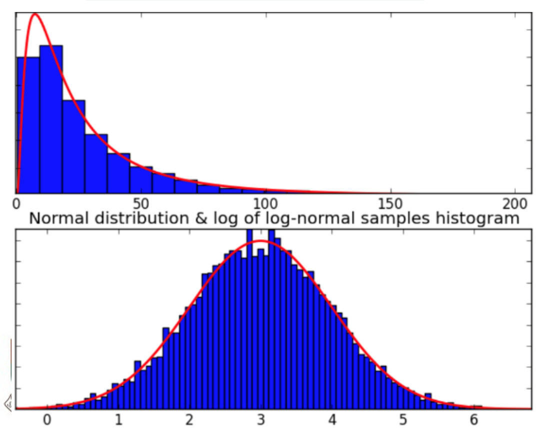
Monte Carlo Simulation

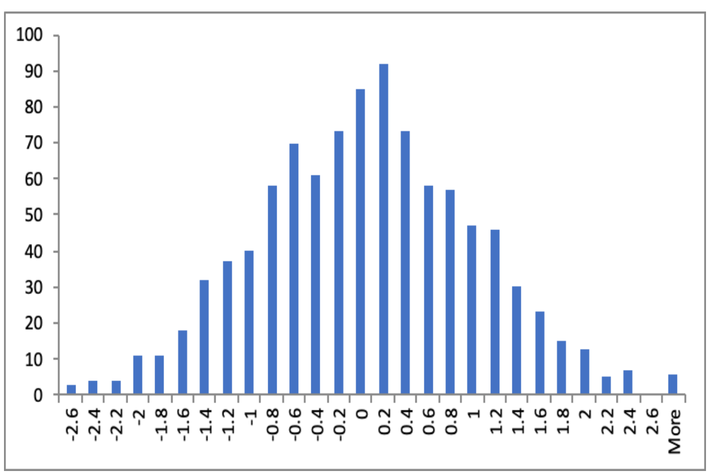
What is Simulation?

- A critical and essential concept in finance is the **risk**.
- **Simulation** is a technique for replicating *uncertain* processes and evaluating decisions under uncertain conditions.
- Simulation models
  - take probability distribution assumptions on the uncertainty as *inputs*,
  - then generate *scenarios* that happen with probabilities,
  - then record *output variables* values over these scenarios,
  - then let us analyze the characteristics of the output probability distributions.
  - For example, draw a *histogram* of outcomes to visualize the *approximate* distribution of outputs, or conduct *statistical inference* by computing mean, variance, skewness and other statistical measures.

What is Simulation?







Selecting Distribution of Random Inputs

- Broadly speaking, two types of distributions:

1. **Historical distribution** of random inputs:
   - Assume that future will behave the same as past.
   - When creating scenarios for future realizations, you can draw randomly from historical scenarios.
   - This approach is based on *bootstrapping* simulation method.
   - This is a *nonparametric* simulation!
2. **Probability distribution function** (CDF or PDF) of random inputs:
   - Use historical data to *estimate* the parameters of this distribution, such as expectation and variance of a normal distribution.
   - This is *Monte Carlo simulation*.
   - A *parametric* simulation!

Monte Carlo Simulation

- Monte Carlo methods (MCM) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results.
- Each simulation should be conducted randomly and independently.
- Simulation is powerful but is only as good as the inputs we provide to our models.
- **Key ingredients**:
  - Specify a stochastic model (e.g., geometric Brownian motion)
    - ✓ value of all parameters (constant inputs)
    - ✓ probability distribution of random variables
  - Generate random draws (scenarios) of each random variable
  - Process the output
- In finance, MCM is used to simulate the various sources of uncertainty that affect the value of the instrument, portfolio or investment in question.

MCM's Application in Finance

1. **Corporate finance**: if a financial analyst wants to construct stochastic financial models as opposed to the traditional static and deterministic models.

❖ Example: To evaluate a project's NPV, she may consider a distribution of inputs (such as variable costs, product price, WACC). These random inputs can be independent or correlated.

❖ Then simulate these inputs for multiple times and generate resulting NPV in each simulation.

❖ She can estimate the following:

- probability distribution of NPV (visualized by the histogram of all simulation results);
- probability that the project has a NPV > 0;
- correlation between NPV and stochastic inputs;
- … …

MCM's Application in Finance

2. **Stock price/return simulation**:

❖ An investor believes that price of a stock (or other assets) follows a specific model (e.g., geometric Brownian motion, a factor model, etc.).

❖ She can estimate all the parameters (constant inputs), and simulate the uncertain components (random variables) to generate the following results:

- Time-series simulation: simulate stock prices/returns from time 0 to *T*, and generate the entire path over the period;
- Cross-sectional simulation: simulate stock prices/returns after a certain time (at time *T*) for multiple times, to obtain the simulated distribution of stock prices/returns.

MCM's Application in Finance

3. **Portfolio value**: simulate prices/returns of correlated component assets over time, the resultant value of each asset is calculated, and the portfolio value is then observed.

4. Valuing **an option on equity:** simulate multiple possible (but random) price paths for the underlying stock, with the associated exercise value (i.e., "payoff") of the option for each path. These payoffs are then averaged and discounted to today, and this result is the value of the option today.

   Simulation is especially useful in path-dependent options such as Asian options, barrier options.

Why Use Simulation?

- An advantage of simulation modeling over pure math modeling is that
  1. simulation enables us to evaluate (approximately) a function of a random variable.
     It is especially powerful when the distribution of random variable is complicated, or the output variable is a complex function of random variables. It may be impossible to derive the distribution of output directly.
  2. Simulation can also visualize the distribution of output.
  3. Simulation can incorporate correlations between input variables.

Random Number Generation

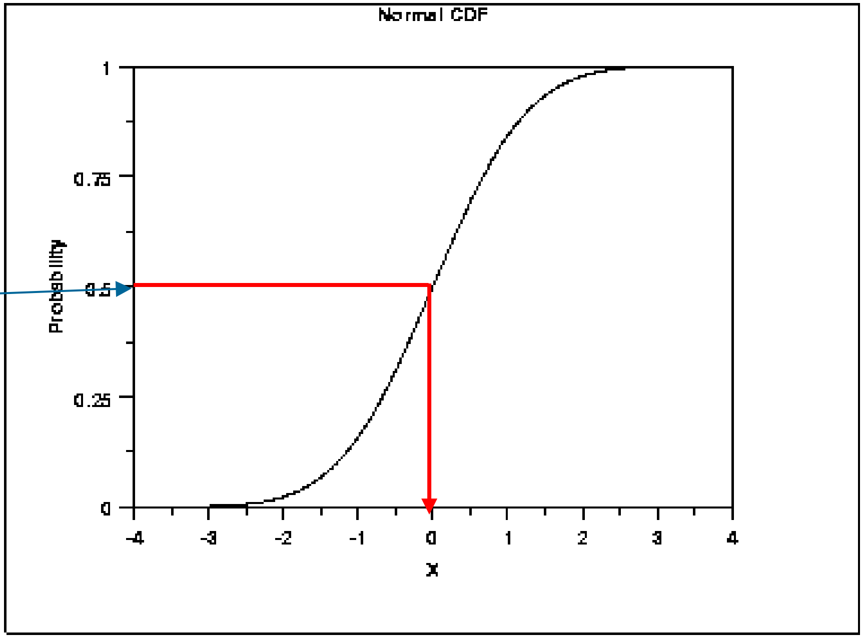
Inverse Transform Sampling

- **Inverse transform sampling** is a basic method for pseudo-random number sampling
- It generates sample random numbers from any probability distribution given its cumulative distribution function (CDF).
- Basic idea:
  - u is a random sample from uniform distribution $U(0,1)$.
  - A random variable X has CDF $F(x) = \Pr(X \leq x)$.
  - For any random draw u, interpreted as a probability, and the number x returned such that $F(x) = u$. That is $x = F^{-1}(u)$, where F^{-1} denotes the inverse function of F.
- Example: if $X \sim N(0,1)$, and the random draw $u = 0.5$. Then $x = F^{-1}(u) = 0$, because for a standard normal distribution, $\Pr(X \leq 0) = 0.5$.

Inverse Transform Sampling

- Basic idea:
  - u is a random sample from $U(0,1)$.
  - A random variable X has CDF $F(x) = \text{Prob}(X \leq x)$.
  - Compute $x = F^{-1}(u)$, where F^{-1} denotes the inverse function of F.

In a simulation, $u = 0.5$ as probability



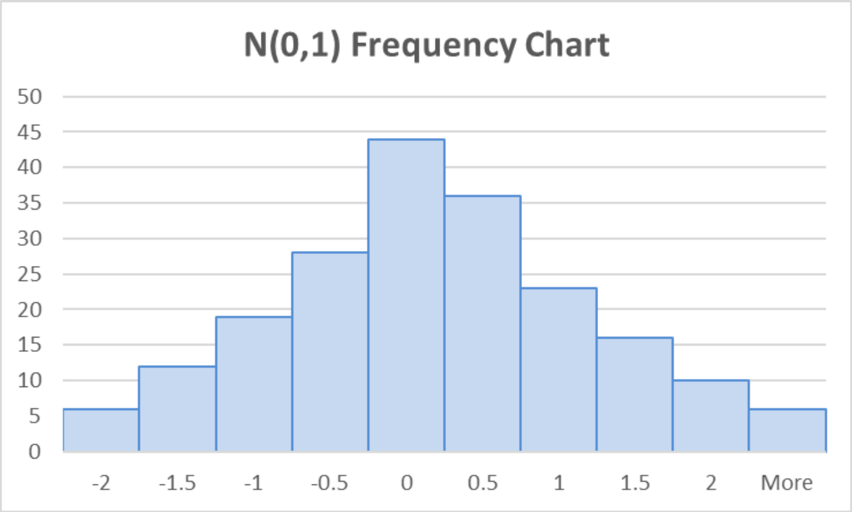
$x = F^{-1}(0.5)$ is the random number from $N(0,1)$

Two Important Functions In Excel For Simulations

- RAND: Inputting this in each cell will deliver a random number between [0,1]. The random draw follows a Uniform(0,1) distribution.
- `NORM.S.INV`: This function will convert a value between [0,1] into a random draw from a standard normal distribution.
- Go to workbook "Lec7_StockSimulation-GBM.xlsm", sheet "Normal".
- Excel functions `NORM.DIST`, `NORM.S.DIST` are used to return PDF and CDF of a Normal distribution.

Simulate Standard Normal Random Number

- Go to sheet "SimNormal".
- We generate 200 random numbers between 0 and 1, and then generate the $N(0,1)$ random numbers. Here is one realization:



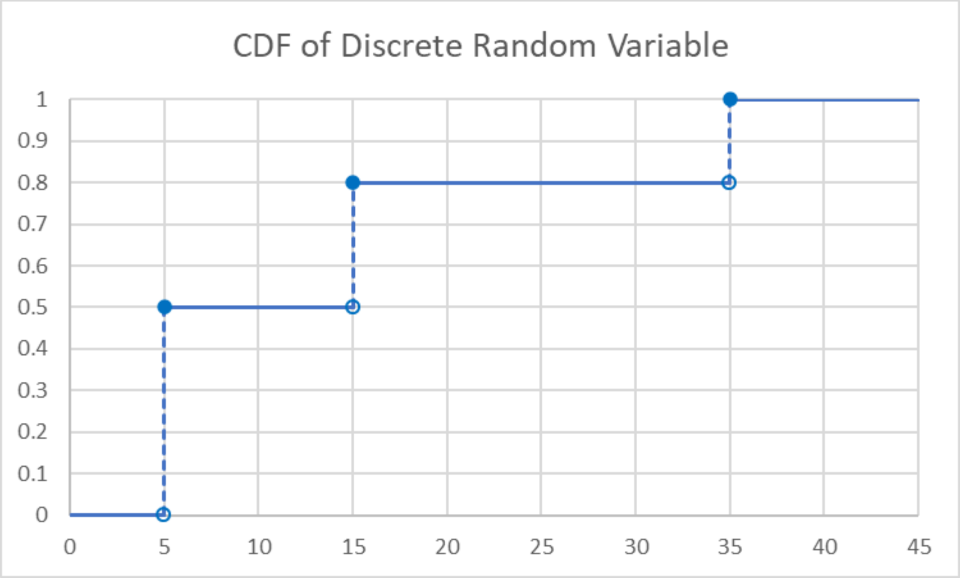
- Note: RAND in Excel cannot set seed, so every time you update the worksheet, RAND regenerates a series of random draws.
- If you want to fix the series, you can use VBA function Rnd to generate random number.

Discrete Distributions

- Example: Random variable X follows a discrete distribution

$$X = \begin{cases} 5, & \text{with probability } 0.5 \\ 15, & \text{with probability } 0.3 \\ 35, & \text{with probability } 0.2 \end{cases}$$

Then the cumulative distribution function of X is



Discrete Distributions

- Example: Random variable X follows a discrete distribution

$$X = \begin{cases} 5, & \text{with probability } 0.5 \\ 15, & \text{with probability } 0.3 \\ 35, & \text{with probability } 0.2 \end{cases}$$

- We generate a random sample of u from $U(0,1)$, then

$$X = \begin{cases} 5, & \text{if } u \le 0.5 \\ 15, & \text{if } 0.5 < u \le 0.8 \\ 35, & \text{if } 0.8 < u \le 1 \end{cases}$$

- Go to sheet "SimDiscrete".

Simulate Stock Prices – GBM

Geometric Brownian Motion for Stock Prices

- Assumptions of geometric Brownian motion (GBM) for stock prices:

$$\frac{dS}{S} = r\,dt + \sigma dB = r\,dt + \sigma\varepsilon\sqrt{dt} \quad \text{(Eqn−1)}$$

  - $dB \sim N(0, dt)$ is a **Wiener process**, also called **Brownian motion**.
  - We rewrite dB as $\varepsilon\sqrt{dt}$ such that $\varepsilon \sim N(0,1)$. We will simulate ε.
  - In GBM, r and σ are constants.

- Eqn-1 is called **stochastic differential equation** (SDE) in the continuous-time model.

- It describes how stock returns (dS/S) over a very short time (dt) evolve. It is normally distributed.

- But SDE does not tell us the distribution of stock price S.

- We first solve SDE to find its **stochastic integral** form, given an initial condition, e.g., S_t is known at time t.

Solving GBM: Log Return over Δt Period

- In FINA3350 course, you have derived the stock price in GBM

$$dS = rSdt + \sigma SdB$$

- Applying Ito's Lemma to a new stochastic process $F(S_t) = \ln S_t$, we can show that SDE of F-process is

$$dF = d\ln S_t = (r - 0.5\sigma^2)dt + \sigma dB = \mu dt + \sigma dB$$

where $\mu = r - 0.5\sigma^2$.

- Take integral on both sides of equation over time interval $[t, t + \Delta t]$

$$\begin{aligned}\ln S_{t+\Delta t} - \ln S_t &= \int_t^{t+\Delta t} d\ln S_t \\ &= \int_t^{t+\Delta t} \mu dt + \int_t^{t+\Delta t} \sigma dB \\ &= \mu\Delta t + \sigma(B_{t+\Delta t} - B_t)\end{aligned}$$

Solving GBM: Log Return over Δt Period

- We derived the log return over the interval $[t, t+\Delta t]$

$$r_{t\to t+\Delta t} = \ln S_{t+\Delta t} - \ln S_t = \mu\Delta t + \sigma(B_{t+\Delta t} - B_t)$$

- Brownian motion B_t satisfies that

$$B_{t+\Delta t} - B_t \sim N(0, \Delta t)$$

- We rewrite the previous equation as

$$r_{t\to t+\Delta t} = \mu\Delta t + \sigma\varepsilon\sqrt{\Delta t} \quad \text{(Eqn-2)}$$

where $\varepsilon \sim N(0,1)$.

- Log return is normally distributed.

Solving GBM: Stock Price over Δt Period

- If S_t is known at time t, stock price at time $t + \Delta t$ is

$$S_{t+\Delta t} = S_t e^{\mu \Delta t + \sigma \varepsilon \sqrt{\Delta t}} \quad \text{(Eqn-3)}$$

where $\varepsilon \sim N(0,1)$.

- Stock price $S_{t+\Delta t}$ follows **lognormal distribution**.
- In the stochastic processes of $r_{t \to t+\Delta t}$ and $S_{t+\Delta t}$,
  - there is one random input $\varepsilon \sim N(0,1)$, and
  - three parameters: μ, σ, and Δt
  - μ is called drift term, it is expectation of log return <u>per unit of time</u>
  - σ is called volatility term, it is standard deviation of log return <u>per unit of time</u>
- How should we estimate μ, σ, and Δt?

Estimate Parameters – μ, σ, and Δt

- Note that the mean and variance of the log returns over interval Δt are

$$E(r_{t\to t+\Delta t}) = E\left(\mu\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right) = \mu\Delta t$$

$$\mathrm{Var}(r_{t\to t+\Delta t}) = \mathrm{Var}\left(\mu\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right) = \sigma^2\Delta t$$

- Both expectation and variance of log return are linear in time.
- We can estimate μ and σ from data on historical stock prices. If we choose adjusted close price as the measure of S_t, no need to consider dividends.
- μ is arithmetic average of log returns divided by Δt

$$\mu = \frac{\mathrm{mean}(\ln S_{t+\Delta t} - \ln S_t)}{\Delta t}$$

- σ is sample variance of log returns divided by Δt

$$\sigma^2 = \frac{\mathrm{variance}(\ln S_{t+\Delta t} - \ln S_t)}{\Delta t}$$

- μ, σ, and Δt should use the same time unit.

Using Excel and VBA to Simulate Stock Prices

Time-Series Simulation of Stock Prices

- We try to answer the following question:
  - The price of a stock today is \$100. The price of the stock is distributed lognormally, with a mean log return of 10% per annum and standard deviation of 40% per annum.
  - How might the price of the stock behave on daily basis over 1 year?
- We conduct a **time-series simulation** to generate a price path based on

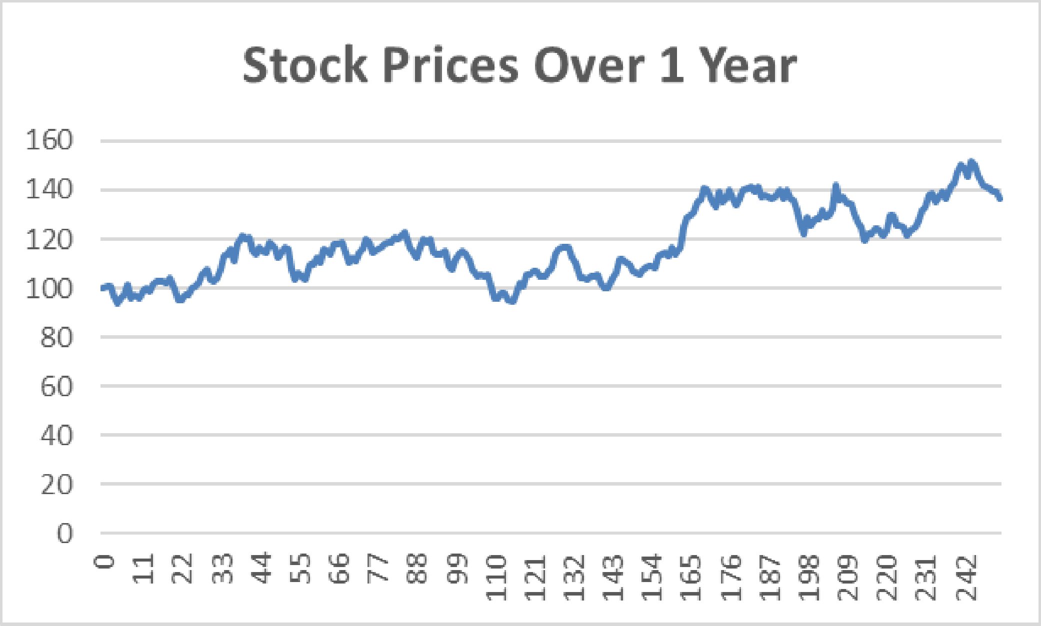
$$S_{t+\Delta t} = S_t e^{\mu \Delta t + \sigma \varepsilon \sqrt{\Delta t}}$$

where $\mu = 10\%$, $\sigma = 40\%$, $\Delta t = \frac{1}{252}$ (year).

- There are an infinite number of price paths for the stock. What we will do is to simulate (randomly) one of these paths. If we want another price path, we can merely rerun the simulation.

Time-Series Simulation: Simulate the Path of Stock Prices

- Go to sheet "Time-series".
- We try to simulate daily returns and stock prices for one year (252 days in 1 year, so $\Delta t = 1/252$).
- Here is one realization:



VBA Function `SimGBM`

- VBA function `SimGBM` in Module "Functions"
- Function `Norm.S.Inv` is a function borrowed from Excel.
- `Rnd` is a VBA function to draw a random number from *U*(0,1).
- `Sqr` is a VBA function for taking the square root.
- `Exp` is a VBA function for taking the exponential.
- VBA also has limited number of built-in numeric functions:

`Abs`, `Atn`, `Cos`, `Exp`, `Int`, `Log`, `Rnd`, `Round`, `Sgn`, `Sin`, `Sqr`, `Tan`.

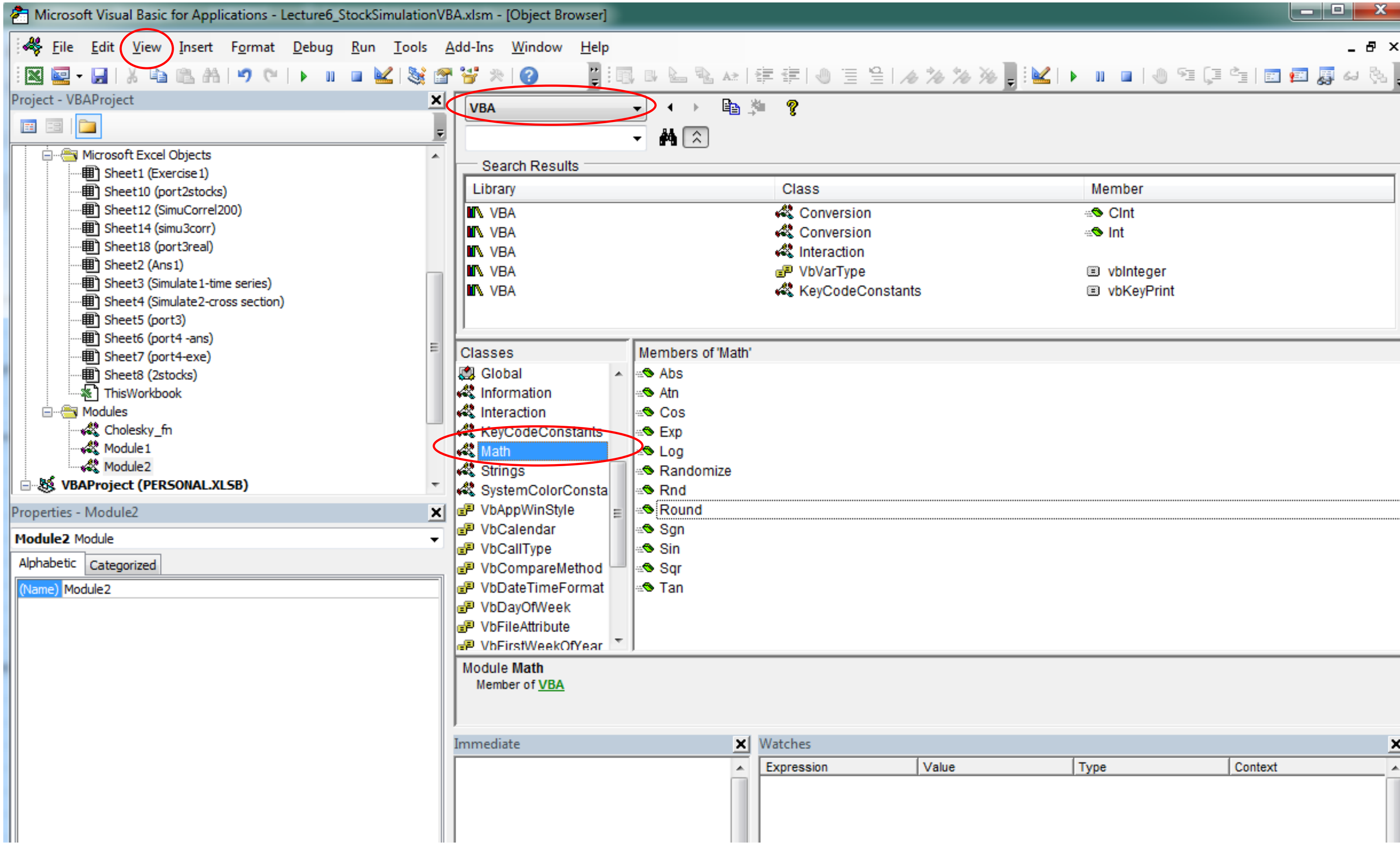
- Some VBA functions are spelt differently from the parallel Excel functions: `Log` in Excel is LN, `Rnd` in Excel is RAND, `Sgn` in Excel is `SIGN`, `Sqr` in Excel is SQRT.

- If both VBA and Excel function exist for the same calculation, VBA form must be used. For example, we must use `Rnd` rather than `WorksheetFunction.Rand`.

How to Find Functions in VBA

- Click "View" in VBA platform => "Object browser". On the left of top panel, you will be able to select libraries.
- When you select "VBA" library, and then "Math" in the classes, you can see all math functions available in VBA, such as `Sqr` and `Rnd`.
- When you select "Excel" library, and then "WorksheetFunction" in the "classes", you will see available Excel functions, such as `NORM.S.INV`.

How to Find Functions in VBA



Cross-Sectional Simulation: Simulate Stock Prices Multiple Times

❖ If we are only interested in the stock price 1-year from now, we must repeat the time-series simulation multiple times and record the end-of-year stock prices.

❖ Based on the simulated prices 1-year from now, we can answer the questions like:

1. What is the sample distribution (mean, volatility, skewness, etc.) of end-of-year stock prices?
2. What is the probability that end-of-year price exceeds current price?

Cross-Sectional Simulation: Simulate Stock Prices Multiple Times

- Go to sheet "Cross-sectional".
- We simulate stock prices 1-year from now (S_T), given $S_0 = 100$ for 200 times:

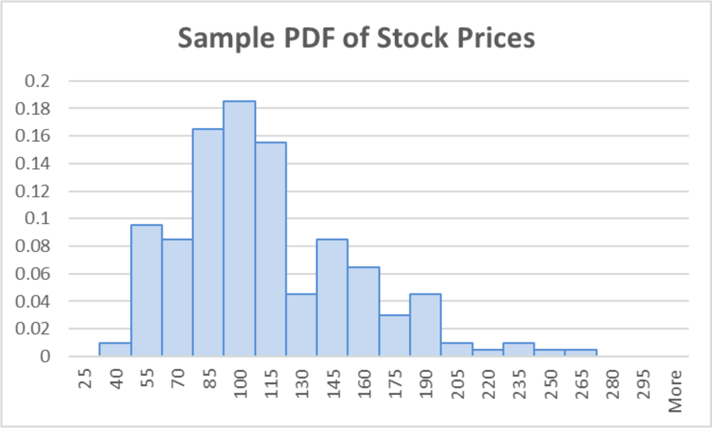
$$S_T = S_0 e^{\mu T + \sigma \varepsilon \sqrt{T}}$$

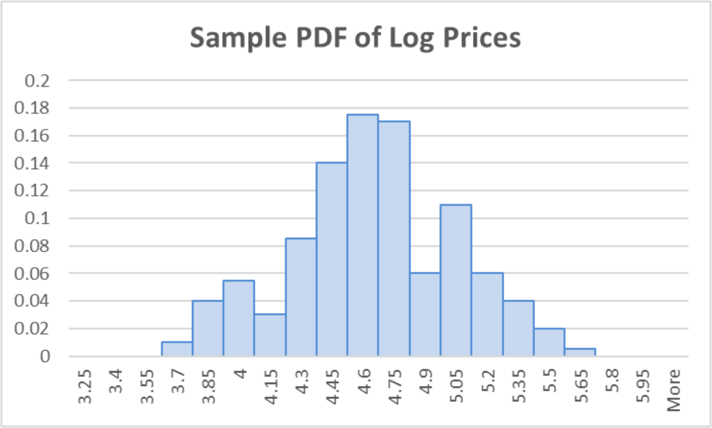
where $\varepsilon \sim N(0,1)$, $T = 1$, $r = 10\%$, $\sigma = 40\%$.

- One attractiveness of GBM is that aggregated return over a long time period (S_T) can be expressed in one explicit and simple equation!
- You don't have to simulate the entire time-series path from S_0 to S_T.
- This doesn't hold for most of other stock pricing models, for example, the factor model that we will discuss later.

Prices vs. Log Prices

❖ Sample distribution of 200 simulated stock prices and log stock prices





❖ Cross-sectional simulations

(1) provide us the probabilities of possible outcomes;

(2) visualize all possible results; and

(3) will help us price options

Simulate Coca-Cola Stock Prices

- Let’s try real data. Assume today is January 2, 2025.
- Historical daily data of Coca-Cola (KO) stock is 02/01/2024 – 31/12/2024.
- We also have realized “future” prices from 02/01/2025 – 30/1/2025.
- Data is saved in worksheet “KO”.
- We will conduct two types of cross-sectional simulations:
  - In-sample simulation
  - Out-of-sample forecasts

In-Sample Simulation

- In-sample simulation:
  - We estimate model inputs (r, σ, T) using historical sample (02/01/2024 – 31/12/2024).
  - The initial stock price S_0 is the price on the **first** day in sample (02/01/2024).
  - We simulate the price S_T on the **last** day in sample (31/12/2024).
  - This is for the model back testing.
  - If simulated stock price S_T is significantly close to realized price on 31/12/2024, GBM fits Coca-Cola stock prices well for this period.
- Go to tab "KO-inSample".
- We measure daily stock returns as $r_t = \ln(S_t/S_{t-1})$.
- Simulate stock price on 31/12/2024 using $S_T = S_0 e^{\mu T + \sigma \varepsilon \sqrt{T}}$ for 10000 times.
- Inputs:
  - $S_0 = \$57.62$ (adj. close price on 02/01/2024)
  - $T = 251$ days (time unit is 1 day).

In-Sample Simulation

- Go to tab “KO-inSample”.
- We generate cross-sectional simulated stock prices S_T for 10000 times using VBA sub procedure `KO_inSample` and VBA function `simGBM`.
- Mean of simulated stock price S_T on 31/12/2024 is $62.50 with 95% confidence interval [47.97, 79.31]. The realized stock price is $61.80. It is within the 95% CI.
- Geometric Brownian motion is a valid model for KO stock in period 02/01/2024 – 31/12/2024.
- Now we use thes model to forecast the stock prices one month later (on 30/01/2025).

Simulate Coca-Cola Stock Prices

- Let’s try real data. Assume today is January 2, 2025.
- Historical daily data of Coca-Cola (KO) stock is 02/01/2024 – 31/12/2024.
- We also have realized “future” prices from 02/01/2025 – 30/1/2025.
- Data is saved in worksheet “KO”.
- We will conduct two types of cross-sectional simulations:

Out-of-Sample Simulation: Forecasting

- Out-of-sample forecasts:
  - The initial stock price S_0 is the price on the **last** day in sample (31/12/2024).
  - We simulate the price S_T in the future (30/1/2025 in this example) to forecast the performance of the stock.
- Go to tab “KO-forecast”.
- We generate cross-sectional simulated stock prices S_T for 10000 times using VBA sub procedure `KO_forecast` and VBA function `simGBM`.
- Inputs:
  - $S_0 = \$61.80$ (adj. close price on 31/12/2024)
  - $T = 19$ days (time unit is 1 day).

Out-of-Sample Simulation: Forecasting

- Mean of simulated stock price S_T on 30/01/2025 is \$62.15 with 95% confidence interval [58.07, 66.51]. The realized stock price is \$63.58. It is within the 95% CI.
- But you may find that out-of-sample simulation result is not as close to realized stock price as in the in-sample simulation.
- Indeed, forecasting stock prices is very challenging because the market should be efficient.

Portfolio Value: Simulating Correlated Stock Returns

Correlated Stock Returns (Two Stocks)

- GBM for stock 1:

$$S_{t+\Delta t}^{1}/S_{t}^{1} = e^{\mu_1 \Delta t + \sigma_1 \varepsilon_1 \sqrt{\Delta t}}$$

- GBM for stock 2:

$$S_{t+\Delta t}^{2}/S_{t}^{2} = e^{\mu_2 \Delta t + \sigma_2 \varepsilon_2 \sqrt{\Delta t}}$$

where $\varepsilon_1, \varepsilon_2 \sim N(0,1)$.

- If log returns of stock 1 and 2 are correlated with coefficient ρ, the correlation must come from the correlation between ε_1 and ε_2.

$$\begin{aligned}
r_{t\to t+\Delta t}^{1} &= \mu_1 \Delta t + \sigma_1 \varepsilon_1 \sqrt{\Delta t} \\
r_{t\to t+\Delta t}^{2} &= \mu_2 \Delta t + \sigma_2 \varepsilon_2 \sqrt{\Delta t} \\
\mathrm{Cov}\left(r_{t\to t+\Delta t}^{1}, r_{t\to t+\Delta t}^{2}\right) &= \sigma_1 \sigma_2 \Delta t \, \mathrm{Cov}(\varepsilon_1, \varepsilon_2) \\
\mathrm{Corr}\left(r_{t\to t+\Delta t}^{1}, r_{t\to t+\Delta t}^{2}\right) &= \mathrm{Corr}(\varepsilon_1, \varepsilon_2) = \rho
\end{aligned}$$

- The question is how to generate correlated ε_1 and ε_2?

Correlated Stock Returns (Two Stocks)

- How to generate correlated ε_1 and ε_2?
- Two steps:
  - Generate two independent random numbers x_1 and x_2, both follow $N(0,1)$
  - Given the correlation between two stock returns $\rho \in [-1,1]$, let

$$\varepsilon_1 = x_1$$
$$\varepsilon_2 = \rho x_1 + \sqrt{1-\rho^2}\, x_2$$

- $\varepsilon_1 \sim N(0,1)$ and it is easy to prove the following:

$$\varepsilon_2 \sim N(0,1)$$
$$\mathrm{Corr}(\varepsilon_1, \varepsilon_2) = \rho$$

- For more than 2 stocks, we will adopt Cholesky's decomposition.
- VBA function `sim2Corr(rho)` in Module "Functions" is to generate two standard normally distributed random variables $(\varepsilon_1, \varepsilon_2)$ with correlation `rho`.

Correlated Stock Returns (Two Stocks)

- Prove the following:

$$\varepsilon_2 \sim N(0,1)$$
$$\mathrm{Corr}(\varepsilon_1, \varepsilon_2) = \rho$$

Proof.

1. Normal distribution is stable under linear transformation. Since x_1 and x_2 are both normally distributed, $\varepsilon_2 = \rho x_1 + \sqrt{1-\rho^2}x_2$ should also follow normal distribution.
2. Expectation:

$$E(\varepsilon_2) = E\left(\rho x_1 + \sqrt{(1-\rho^2)}x_2\right) = \rho E(x_1) + \sqrt{1-\rho^2}E(x_2) = 0$$

3. Variance:

$$\begin{aligned}\mathrm{Var}(\varepsilon_2) &= \mathrm{Var}\left(\rho x_1 + \sqrt{1-\rho^2}x_2\right) \\ &= \rho^2\mathrm{Var}(x_1) + (1-\rho^2)\mathrm{Var}(x_2) + 2\rho\sqrt{1-\rho^2}\mathrm{Cov}(x_1, x_2) \\ &= \rho^2 \times 1 + (1-\rho^2) \times 1 + 2\rho\sqrt{1-\rho^2} \times 0 \\ &= 1\end{aligned}$$

Correlated Stock Returns (Two Stocks)

- Prove the following:

$$\varepsilon_2 \sim N(0,1)$$
$$\mathrm{Corr}(\varepsilon_1, \varepsilon_2) = \rho$$

Proof.

4. Correlation:

$$\begin{aligned}\mathrm{Corr}(\varepsilon_1, \varepsilon_2) &= \frac{\mathrm{Cov}(\varepsilon_1, \varepsilon_2)}{\sqrt{\mathrm{Var}(\varepsilon_1)\mathrm{Var}(\varepsilon_2)}} = \mathrm{Cov}(\varepsilon_1, \varepsilon_2) \\ &= \mathrm{Cov}\left(x_1, \rho x_1 + \sqrt{1-\rho^2}x_2\right) \\ &= \rho \mathrm{Var}(x_1) + \sqrt{1-\rho^2}\mathrm{Cov}(x_1, x_2) \\ &= \rho\end{aligned}$$

Example

- Go to Excel tab “Portfolio_2stocks” for the simulated daily stock prices over one year.
- We estimate the drift and volatility terms for each stock and estimate the correlation (ρ) between two stock returns is 0.3.
  - We first generate x_1 and x_2.
  - We then generate ε_1 and ε_2.
  - We then generate $S_{t+\Delta t}^{1}$ and $S_{t+\Delta t}^{2}$.
  - We also construct a portfolio that have 50% in stock 1 and 50% in stock 2 in the beginning and examine its price movement.

Correlated Stock Returns: Multiple Stocks

Matrix Functions

- In the portfolio models, we make intensive use of Excel's matrix calculations and array functions.
- There are n risky assets, variance σ_i^2. Covariance between asset i and j is σ_{ij}, the correlation between asset i and j is $\rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$.
- In matrix notation,
  - Σ is $n \times n$ variance-covariance matrix.

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_n^2 \end{pmatrix}$$

  - P is $n \times n$ correlation matrix.

$$\mathrm{P} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & 1 \end{pmatrix}$$

VBA Function `CorrMatrix`

- In matrix notation, P is $n \times n$ correlation matrix.

$$\mathrm{P} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \dots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \dots & 1 \end{pmatrix}$$

- There is not Excel built-in function that returns the correlation matrix of multiple data series.
- We write our own VBA function procedure `CorrMat(dataRng)` in Module "Functions".

Correlated Stock Returns (3 Stocks)

- We need to use the Cholesky decomposition to help us construct correlated errors for more than two stocks.
- Cholesky decomposition is like the "square root" of correlation matrix.
- Note that we have to use a user-designed function `Cholesky` (in Module "Functions"). We will just use this function, you do not need to know how to code this function.
- See "Portfolio_3stocks" tab for more details about how to simulate the returns of three stocks.
- Correlation table:

	Stock 1	Stock 2	Stock 3
Stock 1	1.00	-0.50	-0.50
Stock 2	-0.50	1.00	0.90
Stock 3	-0.50	0.90	1.00

Correlated Stock Returns (3 Stocks)

- Cholesky decomposition: To conduct Cholesky decomposition for the correlation matrix (B14:D16), we insert the function "=Cholesky(B14:D16)" in B19 to get:

Cholesky Decomposition (C)			
	Stock 1	Stock 2	Stock 3
Stock 1	1.00	0.00	0.00
Stock 2	-0.50	0.87	0.00
Stock 3	-0.50	0.75	0.43

- Moreover, the transpose of the above mentioned Cholesky matrix is:

Transpose Matrix (C^T)			
	Stock 1	Stock 2	Stock 3
Stock 1	1.00	-0.50	-0.50
Stock 2	0.00	0.87	0.75
Stock 3	0.00	0.00	0.43

Correlated Stock Returns (3 Stocks)

- 3 steps to generate 3 correlated errors each time:

(1) Generate three random numbers x_1, x_2, x_3 following *N*(0,1).

(2) We multiply the Cholesky decomposition of the correlation matrix with the column vector of x_1, x_2, x_3 and get 3 by 1 matrix

$$C \cdot x_{\text{col}} = \begin{pmatrix} 1 & 0 & 0 \\ -0.50 & 0.87 & 0 \\ -0.50 & 0.75 & 0.43 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{pmatrix}$$

or if you generate 3 correlated errors in a row vector:

$$x_{\text{row}} \cdot C^T = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \times \begin{pmatrix} 1 & -0.50 & -0.50 \\ 0 & 0.87 & 0.75 \\ 0 & 0 & 0.43 \end{pmatrix} = \begin{pmatrix} \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{pmatrix}$$

Proof

- $\varepsilon = (\varepsilon_1, \varepsilon_2, \varepsilon_3)$ are standard normally distributed, with correlation matrix $\Sigma = CC^T$. Here, C is Cholesky decomposition of correlation matrix.

- $E(\varepsilon) = E(C \cdot x) = C \cdot E(x) = C \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

- $\mathrm{Var}(\varepsilon) = E(\varepsilon\varepsilon^T) - E(\varepsilon)E(\varepsilon)^T = E(\varepsilon\varepsilon^T) - \mathbf{0}_{3\times 3}$

$$= E(Cxx^TC^T) = CE(xx^T)C^T = C\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}C^T = CC^T = \Sigma$$

- For the standard normal random variable,

$$\text{correlation}=\frac{\text{covariance}}{\text{sd}_1\text{sd}_2}=\text{covariance}$$

Correlated Stock Returns (3 Stocks)

- We then generate the daily stock prices for three correlated stocks for 252 days.
- To check the outcome, we compute the pairwise correlation coefficients among $\varepsilon_1, \varepsilon_2, \varepsilon_3$ to see how accurate our simulations are.
- We write a VBA function `simCorr` with one argument `CorrMat`. It returns a row vector of standard normally distributed errors. `CorrMat` is the correlation matrix of the errors. If argument `CorrMat` is a n-by-n matrix, then the returned array of `simCorr` function is a 1-by-n row vector.