FE535: Introduction to Financial Risk Management
Session 2

Anthony Diaco

Agenda

- Introduction to Monte Carlo Methods
  - Law of Large Numbers
  - Pricing using Numerical Methods
- Univariate Stochastic Processes
  - Brownian Motion
    - Standard
    - General
    - Geometric
  - Application: Simulating Stock Prices

Introduction to Monte Carlo Methods

Monte Carlo Simulations: Overview I

- Monte Carlo (henceforth MC) Simulations are central to financial engineering and risk management
- In particular, MC methods allow risk managers to
  - avoid complicated analytical solutions
  - price complex instruments, e.g. derivatives
  - derive complex portfolio distribution
- Today, MC methods have become widespread thanks to technological advancement

Monte Carlo Simulations: Overview II

- Nonetheless, there are also drawbacks of MC methods
- MC relies heavily on the model's assumptions, such as
  - Distribution shape
  - Underlying parameters
  - Pricing functions
- Risk managers should be aware of possible errors in such assumptions
  - recall Model Risk?

Game 1

- By flipping a coin, let's consider the following game
  - earn a \$1 for heads (H)
  - lose a \$1 for tails (T)
- **How much would you pay to participate in such a game**?

Game 1

- By flipping a coin, let's consider the following game
  - earn a \$1 for heads (H)
  - lose a \$1 for tails (T)
- **How much would you pay to participate in such a game**?

The answer solely depends on the probability of H, denoted by π

- If it is a fair coin, then $\pi = 0.5$ and $0.5 \times (-1) + 0.5 \times 1 = 0$
- But what if the coin is unfair?
  - If $\pi > 0.5$, would pay a premium to participate?
  - If $\pi < 0.5$, would require a premium to participate?

Game 2

- Let's consider a little more complicated game
- Assume we have a fair dice, such that result i has a probability of $1/6$ $\forall i = 1, .., 6$
- The rules of the game go as follows
  - Roll the dice 3 times, giving three results X_1, X_2, and X_3
  - Let $X_{max} = max(X_1, X_2, X_3)$
  - You earn a \$1 if $X_{max} = 6$
  - Otherwise, you get nothing
  - You need to pay p dollars to play this game
- The question is **how much would you pay for this game**?

Fair Price

- With a loss of generality, let's ignore the time value of money, such that the fair price of a single period game should take into account all payoffs and their respective likelihood, i.e.

$$p = \sum_{s=1}^{S} CF_s \times \mathbb{P}(s) \tag{1}$$

where CF_s denotes the cash-flows from state s and $\mathbb{P}(s)$ is the probability of s[a]

[a] This resembles an Arrow-Debreu security (see Wiki).

Fair Price

- With a loss of generality, let's ignore the time value of money, such that the fair price of a single period game should take into account all payoffs and their respective likelihood, i.e.

$$p = \sum_{s=1}^{S} CF_s \times \mathbb{P}(s) \tag{1}$$

where CF_s denotes the cash-flows from state s and $\mathbb{P}(s)$ is the probability of s[a]

[a] This resembles an Arrow-Debreu security (see Wiki).

- In game 2, there are multiple states in which one gets paid, i.e. the different permutations of $\{X_1, X_2, X_3\}$ in which $X_{max} = 6$ takes place
- In other words,

$$p = \mathbb{P}(X_{max} = 6) \tag{2}$$

- To know p, we need to know the probability $\mathbb{P}(X_{max} = 6)$
- To know $\mathbb{P}(X_{max} = 6)$, we need to know the distribution of X_{max}

- We can find the distribution of X_{max} (hence the price p) in two different ways
  - Analytically
  - Numerically (MC)

- We can find the distribution of X_{max} (hence the price p) in two different ways
  - Analytically
  - Numerically (MC)

Analytically

- We know that $X_j \sim U(1, 6)$, for $j = 1, 2, 3$, such that $\mathbb{P}(X_j \leq i) = i/6$
- It also can be shown that

$$\mathbb{P}(X_{max} \leq i) = (i/6)^3 \tag{3}$$

- Hence,

$$\mathbb{P}(X_{max} = 6) = 1 - \mathbb{P}(X_{max} \leq 5) = 1 - (5/6)^3 = 0.4213 \tag{4}$$

Numerically

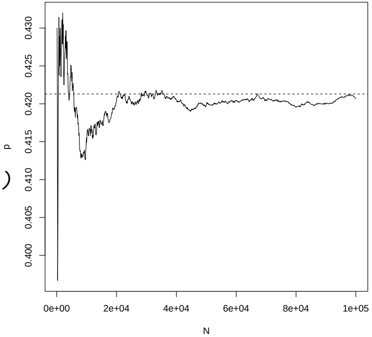
- Start with $n = 1$
- Generate three random variables from $X_{j,n} \sim U(1,6)$ for $j = 1,2,3$
- Compute $X_{max,n} = max(X_{1,n}, X_{2,n}, X_{3,n})$, if $X_{max,n} = 6$ return 1 and zero otherwise
- Repeat the above $N = 10^5$ times
- Finally, compute how many times out of N, we have $X_{max,n} = 6$

Numerically

- Start with $n = 1$
- Generate three random variables from $X_{j,n} \sim U(1,6)$ for $j = 1,2,3$
- Compute $X_{max,n} = max(X_{1,n}, X_{2,n}, X_{3,n})$, if $X_{max,n} = 6$ return 1 and zero otherwise
- Repeat the above $N = 10^5$ times
- Finally, compute how many times out of N, we have $X_{max,n} = 6$

- In other words...

```
> N <- 10^5
> I_seq <- numeric()
> for (n in 1:N) {
+   X_1 <- sample(1:6,1)
+   X_2 <- sample(1:6,1)
+   X_3 <- sample(1:6,1)
+   X_max_n <- max(c(X_1,X_2,X_3))
+   I_n <- (X_max_n == 6)*1
+   I_seq <- c(I_seq,I_n)
+ }
> round(mean(I_seq),4)
[1] 0.4227
```



Law of Large Numbers

- To put formally, if we have a sequence of independent identically distributed (iid) X_n for $i = 1, .., N$ with mean μ and variance $\sigma^2 < \infty$, then

$$\lim_{N \to \infty} \frac{\sum_{n=1}^{N} X_n}{N} \to \mu \tag{5}$$

Law of Large Numbers

- To put formally, if we have a sequence of independent identically distributed (iid) X_n for $i = 1, .., N$ with mean μ and variance $\sigma^2 < \infty$, then

$$\lim_{N\to\infty} \frac{\sum_{n=1}^{N} X_n}{N} \to \mu \tag{5}$$

- In game 2, the price of the game is equivalent to the expected payoff
- At each time the payoff of the game is either 1 or zero depending on the result of $X_{max,n}$
- If we denote

$$\mathbf{I}_n = \begin{cases} 1 & \mathbb{P}(X_{max} = 6) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

- As a result, it follows that

$$\lim_{N\to\infty} \frac{\sum_{n=1}^{N} \mathbf{I}_n}{N} \to \mathbb{E}[\mathbf{I}_n] = \mathbb{P}(X_{max,n} = 6) = p \tag{7}$$

Game 3: MC Application to Asset Prices

- Game 2 resembles what is known as a binary option
- A binary option returns \$1 in case some event takes place and zero otherwise
- Let's consider the following game now

Game 3

- The current stock price is \$100
- If the stock price goes beyond \$110 next month, you get paid \$1
- If not, you are paid zero and you lose the down payment of p
- **What is the fair price of p?**

- To answer the above, we need to know the behavior (distribution) of the stock price
  - What is the potential growth of the stock?
  - What is the volatility of the stock?

- Let P_t denote the price in month t
- Game 3 pays a \$1, if
  - the price goes up $P_1 > 110$, where $P_0 = 100$

    or
  - the return on stock goes up by 9.53%, i.e.

$$R_1 > log(110/100) = 9.53\% \tag{8}$$

- Hence, if we know the distribution of the monthly return R_t, then similar to (7), the option price would be

$$\mathbb{P}(R_1 > 9.53\%) \tag{9}$$

- Let P_t denote the price in month t
- Game 3 pays a \$1, if
  - the price goes up $P_1 > 110$, where $P_0 = 100$
    or
  - the return on stock goes up by 9.53%, i.e.

$$R_1 > log(110/100) = 9.53\% \tag{8}$$

- Hence, if we know the distribution of the monthly return R_t, then similar to (7), the option price would be

$$\mathbb{P}(R_1 > 9.53\%) \tag{9}$$

- For instance, if $R_1 \sim N(0.02, 0.04^2)$, then

$$\mathbb{P}(R_1 > 0.0953) = 1 - \Phi\left(\frac{0.0953 - 0.02}{0.04}\right) = 1 - 0.9701 \approx 3\% \tag{10}$$

- If we assume a discount rate of zero over one month period, then the price of the binary option today is \$0.03
  - An increase of 9.53% over one month seems very unlikely
  - For this reason the option is cheap

If we were to find the price of the binary option numerically we should do the following steps

1. Start with $n = 1$
2. Generate one random variable from $R_{1,n} \sim N(0.02, 0.04^2)$
3. Check whether $R_{1,n} > 9.53\%$ and assign $\mathbf{I}_n = 1$ if true and zero otherwise
4. Repeat the above $N = 10^5$ times
5. Finally, the average $\mathbf{I}_n = 1$ over the N iterations should converge to the true probability, i.e.

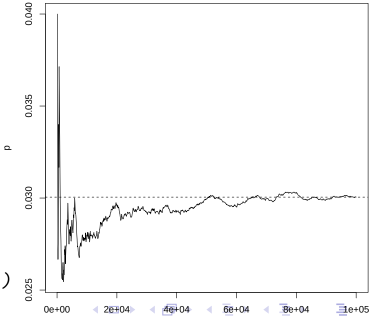
$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}_n \to \mathbb{P}(R_{1,n} > 9.53\%) = 0.03 \tag{11}$$

If we were to find the price of the binary option numerically we should do the following steps

1. Start with $n = 1$
2. Generate one random variable from $R_{1,n} \sim N(0.02, 0.04^2)$
3. Check whether $R_{1,n} > 9.53\%$ and assign $\mathbf{I}_n = 1$ if true and zero otherwise
4. Repeat the above $N = 10^5$ times
5. Finally, the average $\mathbf{I}_n = 1$ over the N iterations should converge to the true probability, i.e.

$$\frac{1}{N}\sum_{n=1}^{N} \mathbf{I}_n \to \mathbb{P}(R_{1,n} > 9.53\%) = 0.03 \tag{11}$$

```
> N <- 10^5
> I_seq <- numeric()
> for (n in 1:N) {
+   R_1 <- rnorm(1,0.02,0.04)
+   I_n <- (R_1 > log(110/100))*1
+   I_seq <- c(I_seq,I_n)
+ }
> mean(I_seq)
[1] 0.02998
> # alternative fast solution
> mean(rnorm(N,0.02,0.04) > log(110/100))
[1] 0.03045
```



Complicated Games

- So far, the problems we have talked about all have analytical solutions
- However, in practice, things can be too complex to price analytically
  - analytical solution may not exist at all
- Relying on numerical solution is inevitable

Examples of More Complicated Options

- European Option - has a closed form solution
- American Option - no analytical solution
- Asian Option - no analytical solution

- what makes the American and Asian complicated is the fact that both are path-dependent

Univariate Stochastic Processes

Simulating Price Path

- Under market efficiency, financial prices should exhibit a random walk
- Prices are assumed to follow what it is known as a **Markov Process**
  - the next period price depends on today's alone
- The future prices are stochastic and obey certain motion
- It is common to represent the price over time using a number of components:
  - Growth or expected return - μ
  - Volatility - σ
  - Time - t
  - Stochastic Component - Z_t

Standard Brownian Motion

- The stochastic component Z_t is a specific Markov Process known as a standard **Brownian Motion** (BM) or **Weiner** process
- Z_t has a number of main properties
  1. $Z_0 = 0$
  2. It has a normal (Gaussian) distribution
  $$Z_t \sim N(0, t) \tag{12}$$
  3. Its increments are independent and also follow a normal distribution
  $$Z_t - Z_s \sim N(0, t - s) \tag{13}$$
  which is independent of any past values Z_u for $u < s$
- It is common to represent the change in the process over small time increment as ΔZ_t, i.e. that is the change in the value of Z_t over Δt period
- Given the above properties it follows that
$$Z_{t+\Delta t} - Z_t = \Delta Z_t \sim N(0, \Delta t) \tag{14}$$

Simulating Standard Brownian Motion

- Assume we have t units of time

Simulating Standard Brownian Motion

- Assume we have t units of time
- We like to simulate the BM between $t = 0$ and $t = 1$

Simulating Standard Brownian Motion

- Assume we have t units of time
- We like to simulate the BM between $t = 0$ and $t = 1$
- The simulated process is observed over $\Delta t = 0.01$ time increments

Simulating Standard Brownian Motion

- Assume we have t units of time
- We like to simulate the BM between $t = 0$ and $t = 1$
- The simulated process is observed over $\Delta t = 0.01$ time increments
- At time $t = 0$, the process is $Z_0 = 0$

Simulating Standard Brownian Motion

- Assume we have t units of time
- We like to simulate the BM between $t = 0$ and $t = 1$
- The simulated process is observed over $\Delta t = 0.01$ time increments
- At time $t = 0$, the process is $Z_0 = 0$
- The process moves with ΔZ_t increments over $t = 0.01, 0.02, ..., 1$

$$Z_{0.01} = Z_0 + \Delta Z_{0.01}, \text{where } \Delta Z_{0.01} \sim N(0, 0.01) \tag{15}$$

Simulating Standard Brownian Motion

- Assume we have t units of time
- We like to simulate the BM between $t = 0$ and $t = 1$
- The simulated process is observed over $\Delta t = 0.01$ time increments
- At time $t = 0$, the process is $Z_0 = 0$
- The process moves with ΔZ_t increments over $t = 0.01, 0.02, ..., 1$

$$Z_{0.01} = Z_0 + \Delta Z_{0.01}, \text{where } \Delta Z_{0.01} \sim N(0, 0.01) \tag{15}$$

- For 0.02, the same applies

$$Z_{0.02} = Z_{0.01} + \Delta Z_{0.02}, \text{where } \Delta Z_{0.02} \sim N(0, 0.01) \tag{16}$$

Simulating Standard Brownian Motion

- Assume we have t units of time
- We like to simulate the BM between $t = 0$ and $t = 1$
- The simulated process is observed over $\Delta t = 0.01$ time increments
- At time $t = 0$, the process is $Z_0 = 0$
- The process moves with ΔZ_t increments over $t = 0.01, 0.02, ..., 1$

$$Z_{0.01} = Z_0 + \Delta Z_{0.01}, \text{where } \Delta Z_{0.01} \sim N(0, 0.01) \tag{15}$$

- For 0.02, the same applies

$$Z_{0.02} = Z_{0.01} + \Delta Z_{0.02}, \text{where } \Delta Z_{0.02} \sim N(0, 0.01) \tag{16}$$

- Generalizing, in d steps from $t = 0$, the process value is given by

$$Z_{\frac{d}{100}} = Z_0 + \sum_{i=1}^{d} \Delta Z_{\frac{i}{100}} \tag{17}$$

Simulating Standard Brownian Motion

- Assume we have t units of time
- We like to simulate the BM between $t = 0$ and $t = 1$
- The simulated process is observed over $\Delta t = 0.01$ time increments
- At time $t = 0$, the process is $Z_0 = 0$
- The process moves with ΔZ_t increments over $t = 0.01, 0.02, ..., 1$

$$Z_{0.01} = Z_0 + \Delta Z_{0.01}, \text{where } \Delta Z_{0.01} \sim N(0, 0.01) \tag{15}$$

- For 0.02, the same applies

$$Z_{0.02} = Z_{0.01} + \Delta Z_{0.02}, \text{where } \Delta Z_{0.02} \sim N(0, 0.01) \tag{16}$$

- Generalizing, in d steps from $t = 0$, the process value is given by

$$Z_{\frac{d}{100}} = Z_0 + \sum_{i=1}^{d} \Delta Z_{\frac{i}{100}} \tag{17}$$

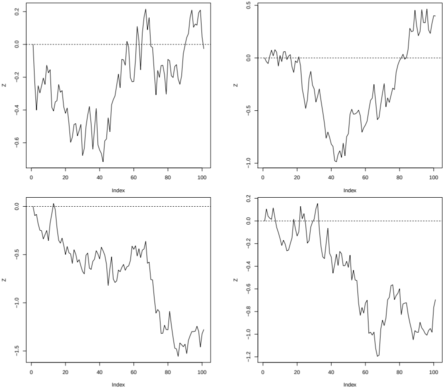
- Since $Z_0 = 0$ and $\Delta Z_{\frac{i}{100}} \sim N(0, 0.01)$ is iid, it follows that

$$Z_{\frac{d}{100}} \sim N(0, \frac{d}{100}) \tag{18}$$

```
> BM_path <- function(n) {
+   d <- 100
+   t <- 1
+   dt <- t/d
+   Z <- 0
+   for (i in 1:d) {
+     dZ <- rnorm(1,0,sqrt(dt))
+     Z_i <- Z[i] + dZ
+     Z <- c(Z,Z_i)
+     }
+   return(Z)
+   }
> Z <- BM_path()
> plot(Z, type = "l")
> abline(h = 0,lty = 2)
```

```
> Z_mat <- sapply(1:10^4,BM_path)
> dim(Z_mat)
```

```
[1]   101 10000
```



```
> Z_1 <- Z_mat[nrow(Z_mat),]
> mean(Z_1);sd(Z_1)
```

```
[1] -0.01323055
[1] 1.006404
```

General Brownian Motion

- Similar to standard BM, the general BM has the following properties

$$\Delta X_t = \mu \Delta t + \sigma \Delta Z_t \tag{19}$$

where

$$\Delta X_t \sim N(\mu \Delta t, \sigma^2 \Delta t) \tag{20}$$

General Brownian Motion

- Similar to standard BM, the general BM has the following properties

$$\Delta X_t = \mu \Delta t + \sigma \Delta Z_t \tag{19}$$

where

$$\Delta X_t \sim N(\mu \Delta t, \sigma^2 \Delta t) \tag{20}$$

Illustration

- Let t refer to annual frequency with 252 trading days a year
- Let Δt denote the change in time over one day, i.e. $\Delta t = 1/252$
- Assume the price at time 0 is $X_0 = 100$, with $\mu = 0.10$ and $\sigma = 0.2$
- We can simulate the price in the next day as

$$X_{\frac{1}{252}} = X_0 + \Delta X_{\frac{1}{252}} \tag{21}$$

where

$$\Delta X_{\frac{1}{252}} = 0.1 \times \frac{1}{252} + 0.2 \times \Delta Z_{\frac{1}{252}} \sim N\left(\frac{0.1}{252}, \frac{0.2^2}{252}\right) \tag{22}$$

with $\Delta Z_1 \sim N(0, 1/252)$

- If we repeat the previous procedure multiple times, we can simulate the process over number of periods using today's price
- To illustrate this, the price in d periods ahead is given by

$$X_{\frac{d}{252}} = X_0 + \sum_{i=1}^{d} \Delta X_{\frac{i}{252}} \tag{23}$$

- Since the increments ΔX_i are iid, then we can simulate d random numbers from $N(\frac{0.1}{252}, \frac{0.2^2}{252})$

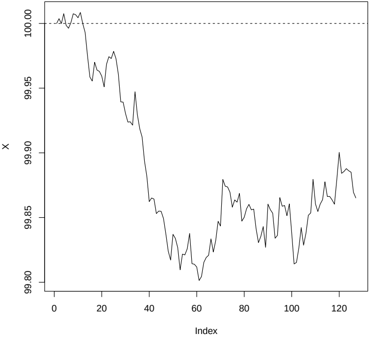
- If we repeat the previous procedure multiple times, we can simulate the process over number of periods using today's price
- To illustrate this, the price in d periods ahead is given by

$$X_{\frac{d}{252}} = X_0 + \sum_{i=1}^{d} \Delta X_{\frac{i}{252}} \tag{23}$$

- Since the increments ΔX_i are iid, then we can simulate d random numbers from $N(\frac{0.1}{252}, \frac{0.2^2}{252})$

- To implement,

```
> X <- 100
> mu <- 0.1; sig <- 0.2
> dt <- 1/252
> X <- 100
> for(i in 1:126) {
+   dX <- rnorm(1,mu*dt,sig*sqrt(dt))
+   X_1 <- X[i] + dX
+   X <- c(X,X_1)
+ }
> plot(X,type = "l")
> abline(h = 100, lty = 2)
```



- The problem with the previous BM is that prices could take negative values

- The problem with the previous BM is that prices could take negative values
- For instance, if the current price were \$1 while $\sigma = 0.5$, by simulating the price 10^4 times, we get the following

```
> prices <- numeric()
> N <- 10^4
> X <- 1
> sig <- 0.5
> for(n in 1:N) {
+   X_end <-  1 + sum(rnorm(126,mu*dt,sig*sqrt(dt)))
+   prices <- c(prices,X_end)
+   }
> summary(prices)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.3484  0.8218  1.0572  1.0566  1.2945  2.5832
```

- The problem with the previous BM is that prices could take negative values
- For instance, if the current price were \$1 while $\sigma = 0.5$, by simulating the price 10^4 times, we get the following

```
> prices <- numeric()
> N <- 10^4
> X <- 1
> sig <- 0.5
> for(n in 1:N) {
+   X_end <-  1 + sum(rnorm(126,mu*dt,sig*sqrt(dt)))
+   prices <- c(prices,X_end)
+   }
> summary(prices)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.3484  0.8218  1.0572  1.0566  1.2945  2.5832
```

- Additional problems with the this process is that fails to mimic other aspects of prices, e.g.
  1. Prices change relative to the previous levels
  2. Prices exhibit positive skewness - which is not the case for normal

Geometric Brownian Motion

- The most common process to simulate stock prices is the Geometric Brownian Motion (GBM)
- In this case,

$$\Delta S_t = S_t \mu \Delta t + S_t \sigma \Delta Z_t \tag{24}$$

  alternatively,

$$\frac{\Delta S_t}{S_t} = \mu \Delta t + \sigma \Delta Z_t \tag{25}$$

- Note that $\frac{\Delta S_t}{S_t}$ resembles the stock return between t and $t + \Delta t$.
- To see this,

$$\frac{\Delta S_t}{S_t} = \frac{S_{t+\Delta t} - S_t}{S_t} \approx log\left(\frac{S_{t+\Delta t}}{S_t}\right) = \Delta log(S_t) \tag{26}$$

- In fact, the solution to (24) or (25), requires the solution to the stochastic differential equation (SDE) $\Delta log(S_t)$

- This class doesn't require knowledge about SDEs, but it follows that the solution for the GBM is

$$log\left(\frac{S_{t+\Delta t}}{S_t}\right) = \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma \Delta Z_t \tag{27}$$

which is the same as Equation (4.6) from the Textbook

- To simplify the notation, let ΔR_t denote the return of the stock over Δt

$$\Delta R_t = log\left(\frac{S_{t+\Delta t}}{S_t}\right) \tag{28}$$

- In fact, it follows that the ΔR_t is a general BM, such that

$$\Delta R_t \sim N\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t, \sigma^2 \Delta t\right) \tag{29}$$

is an iid process

- This class doesn't require knowledge about SDEs, but it follows that the solution for the GBM is

$$log\left(\frac{S_{t+\Delta t}}{S_t}\right) = \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma \Delta Z_t \tag{27}$$

which is the same as Equation (4.6) from the Textbook

- To simplify the notation, let ΔR_t denote the return of the stock over Δt

$$\Delta R_t = log\left(\frac{S_{t+\Delta t}}{S_t}\right) \tag{28}$$

- In fact, it follows that the ΔR_t is a general BM, such that

$$\Delta R_t \sim N\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t, \sigma^2 \Delta t\right) \tag{29}$$

is an iid process

- Therefore, to simulate the price at time $t + \Delta t$, one needs
  1. the price at time t, S_t
  2. estimate μ and σ
  3. Finally, simulate ΔR_t, i.e. draw a random number from the normal distribution described in (29)
- In other words

$$S_{t+\Delta t} = S_t \times \exp(\Delta R_t) \tag{30}$$

Let's consider again the same example as before

Implementation of GBM Simulation

- Let $\Delta t = 1/252$
- The current price at time $t = 0$ is $S_0 = 100$
- Given μ and σ, draw a random number from (29) denoted by $\Delta R_{\frac{1}{252}}$
- The price next day is

$$S_{\frac{1}{252}} = S_0 \times \exp(\Delta R_{\frac{1}{252}}) \tag{31}$$

- To simulate the second day price, draw another random number from (29) denoted by $\Delta R_{\frac{2}{252}}$, such that

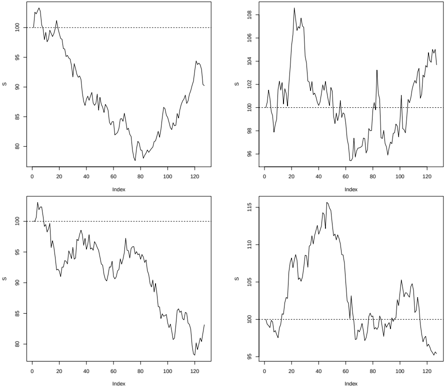
$$S_{\frac{2}{252}} = S_{\frac{1}{252}} \times \exp(\Delta R_{\frac{2}{252}}) \tag{32}$$

- To generalize it follows that the next d days price is given by

$$S_{\frac{d}{252}} = S_0 \prod_{i=1}^{d} \exp(\Delta R_{\frac{i}{252}}) = S_0 \times \exp\left(\sum_{i=1}^{d} \Delta R_{\frac{i}{252}}\right) \tag{33}$$

Let's demonstrate how to implement GBM

```
> S <- 100
> dt <- 1/252
> mu <- 0.1
> sig <- 0.2
> for(i in 1:126) {
+   dR <- rnorm(1,
+               dt*(mu - 0.5*sig^2),
+               sig*sqrt(dt) )
+   S_dt <- S[i]*exp(dR)
+   S <- c(S,S_dt)
+ }
> plot(S,type = "l")
> abline(h = S[1], lty = 2)
```



Concluding Remarks

- This session covers the basic idea behind Law of Large Numbers and MC simulations
- The next session will cover the application of GBP into asset prices
  - Calibration using real-data
  - Simulating future prices
  - Application to portfolio