The Monte Carlo Framework, Examples from Finance and Generating Correlated Random Variables

1 The Monte Carlo Framework

Suppose we wish to estimate some quantity, $\theta = \mathrm{E}[h(\mathbf{X})]$, where $\mathbf{X} = \{X_1, \ldots, X_n\}$ is a random vector in $\mathbf{R}^n$, $h(\cdot)$ is a function from $\mathbf{R}^n$ to $\mathbf{R}$, and $\mathrm{E}[|h(\mathbf{X})|] < \infty$.

Note that $\mathbf{X}$ could represent the values of a stochastic process at different points in time. For example, X_i might be the price of a particular stock at time i and $h(\cdot)$ might be given by

$$h(\mathbf{X}) = \frac{X_1 + \ldots + X_n}{n}$$

so then θ is the expected average value of the stock price. To estimate θ we use the following algorithm:

Monte Carlo Algorithm

> **for** $i = 1$ **to** n
>
>   **generate** $\mathbf{X}_i$
>
>   **set** $h_i = h(\mathbf{X}_i)$
>
> **set** $\widehat{\theta}_n = \frac{h_1+h_2+\ldots+h_n}{n}$

Note: If n is large, it may be necessary to keep track of $\sum_i h_i$ within the **for** loop, so that we don't have to store each value of h_i.

Question: Why is $\widehat{\theta}$ a good estimator?

Answer: As we saw previously, there are two reasons:

1. $\widehat{\theta}_n$ is *unbiased*. That is
$$\mathrm{E}[\widehat{\theta}_n] = \frac{\mathrm{E}[\sum_i^n h_i]}{n} = \frac{\mathrm{E}[\sum_i^n h(\mathbf{X_i})]}{n} = \frac{n\theta}{n} = \theta$$

2. $\widehat{\theta}_n$ is *consistent*. That is
$$\widehat{\theta}_n \to \theta \quad \text{wp } 1 \text{ as } n \to \infty.$$
This follows from the Strong Law of Large Numbers (SLLN).

Remark 1 *We can also estimate probabilities this way by representing them as expectations. In particular, if* $\theta = \mathbf{P}(\mathbf{X} \in A)$*, then* $\theta = \mathrm{E}[I_A(\mathbf{X})]$ *where*

$$I_A(\mathbf{X}) = \begin{cases} 1 & \textit{if } \mathbf{X} \in A \\ 0 & \textit{otherwise} \end{cases}$$

2 Examples from Finance

Example 1 (Portfolio Evaluation)

Consider two stocks, A and B, and let $S_a(t)$ and $S_b(t)$ be the time t prices of A and B, respectively. At time $t = 0$, I buy n_a units of A and n_b units of B so my initial wealth is $W_0 = n_a S_a(0) + n_b S_b(0)$. Suppose my investment horizon is T years after which my *terminal* wealth, W_T, is given by

$$W_T = n_a S_a(T) + n_b S_b(T).$$

Note that this means that I do *not* trade in $[0, T]$. Assume $S_a \sim GBM(\mu_a, \sigma_a)$, $S_b \sim GBM(\mu_b, \sigma_b)$, and that S_a and S_b are independent. We would now like to estimate

$$\mathbf{P}\left(\frac{W_T}{W_0} \leq .9\right),$$

i.e., the probability that the value of my portfolio drops by more than 10%. Note that we may write

$$\begin{aligned} S_a(T) &= S_a(0)\ \exp\left((\mu_a - \sigma_a^2/2)T + \sigma_a B_a(T)\right) \\ S_b(T) &= S_b(0)\ \exp\left((\mu_b - \sigma_b^2/2)T + \sigma_b B_b(T)\right) \end{aligned}$$

where B_a and B_b are independent SBM's.

Let L be the event that $W_T/W_0 \leq .9$ so that that the quantity of interest is $\theta := \mathbf{P}(L) = \mathrm{E}[I_L]$. The problem of estimating θ therefore falls into our Monte Carlo framework. In this example, $\mathbf{X} = (S_a(T), S_b(T))$ and

$$I_L(\mathbf{X}) = \begin{cases} 1 & \text{if } \frac{n_a S_a(T) + n_b S_b(T)}{n_a S_a(0) + n_b S_b(0)} \leq 0.9 \\ 0 & \text{otherwise} \end{cases}$$

Let's assume the following parameter values:

$T = .5$ years, $\mu_a = .15$, $\mu_b = .12$, $\sigma_a = .2$, $\sigma_b = .18$, $S_a(0) = \$100$, $S_b(0) = \$75$ and $n_a = n_b = 100$.

This then implies $W_0 = \$17,500$.

We then have the following algorithm for estimating θ:

Monte Carlo Estimation of θ

for $i = 1$ **to** n
  generate $\mathbf{X}^i = (S_a^i(T), S_b^i(T))$
  compute $I_L(\mathbf{X}^i)$
set $\widehat{\theta}_n = \frac{I_L(\mathbf{X}^1) + \ldots + I_L(\mathbf{X}^n)}{n}$

Sample Matlab Code

```
> n=1000; T=0.5; na =100; nb=100;
> S0a=100; S0b=75; mua=.15; mub=.12; siga=.2; sigb=.18;
> W0 = na*S0a + nb*S0b;

> BT = sqrt(T)*randn(2,n);
> STa = S0a * exp((mua - (siga^2)/2)*T + siga* BT(1,:));
> STb = S0b * exp((mub - (sigb^2)/2)*T + sigb* BT(2,:));

> WT = na*STa + nb*STb;

> theta_n = sum(WT./W0 < .9) / n
```

2.1 Introduction to Security Pricing

We assume again that $S_t \sim GBM(\mu, \sigma)$ so that

$$S_T = S_0 e^{(\mu - \sigma^2/2)T + \sigma B_T}.$$

In addition, we will *always* assume that there exists a risk-free cash account so that if W_0 is invested in it at $t = 0$, then it will be worth $W_0 \exp(rt)$ at time t. We therefore interpret r as the continuously compounded interest rate. Suppose now that we would like to estimate the price of a security that pays $h(\mathbf{X})$ at time T, where $\mathbf{X}$ is a random quantity (variable, vector, etc.) possibly representing the stock price at different times in $[0, T]$. The theory of asset pricing then implies that the time 0 value of this security is

$$h_0 = \mathrm{E}^Q[e^{-rT} h(\mathbf{X})]$$

where $\mathrm{E}^Q[.]$ refers to expectation under the **risk-neutral** probability[1] measure.

Risk-Neutral Asset Pricing

In the GBM model for stock prices, using the risk-neutral probability measure is equivalent to assuming that

$$S_t \sim GBM(r, \sigma).$$

Note that we are *not* saying that the *true* stock price process is a $GBM(r, \sigma)$. Instead, we are saying that for the purposes of *pricing* securities, we *pretend* that the stock price process is a $GBM(r, \sigma)$.

Example 2 (Pricing a European Call Option)

Suppose we would like to estimate[2] C_0, the price of a European call option with strike K and expiration T, using simulation. Our risk-neutral pricing framework tells us that

$$C_0 = e^{-rT} \mathrm{E}\left[\max\left(0,\ S_0 e^{(r-\sigma^2/2)T + \sigma B_T} - K\right)\right]$$

and again, this falls into our simulation framework. We have the following algorithm.

[1] The risk-neutral probability measure is often called the *equivalent martingale measure* or EMM.

[2] C_0 can be calculated exactly using the Black-Scholes formula but we ignore this fact for now!

Estimating the Black-Scholes Option Price

> **set** $sum = 0$
> **for** $i = 1$ **to** n
>
>   **generate** S_T
>   **set** $sum = sum + \max(0,\ S_T - K)$
>
> **set** $\widehat{C_0} = e^{-rT} sum/n$

Example 3 (**Pricing Asian Options)**

Let the time T payoff of the Asian option be

$$h(\mathbf{X}) = \max\left(0,\ \frac{\sum_{i=1}^{m} S_{\frac{iT}{m}}}{m} - K\right)$$

so that $\mathbf{X} = \left(S_{\frac{T}{m}}, S_{\frac{2T}{m}}, \ldots, S_T\right)$.
We can then use the following Monte Carlo algorithm to estimate $C_0 = \mathrm{E}^Q[e^{-rT} h(\mathbf{X})]$.

Estimating the Asian Option Price

> **set** $sum = 0$
> **for** $i = 1$ **to** n
>
>   **generate** $S_{\frac{T}{m}}, S_{\frac{2T}{m}}, \ldots, S_T$
>   **set** $sum = sum + \max\left(0,\ \frac{\sum_{i=1}^{m} S_{\frac{iT}{m}}}{m} - K\right)$
>
> **set** $\widehat{C_0} = e^{-rT} sum/n$

---

3 Generating Correlated Normal Random Variables and Brownian Motions

In the portfolio evaluation example of Lecture 4 we assumed that the two stock price returns were independent. Of course this assumption is too strong since in practice, stock returns often exhibit a high degree of correlation. We therefore need to be able to simulate correlated random returns. In the case of geometric Brownian motion, (and other models based on Brownian motions) simulating correlated returns means simulating correlated normal random variables. And instead of posing the problem in terms of only two stocks, we will pose the problem more generally in terms of n stocks. First, we will briefly review correlation and related concepts.

3.1 Review of Covariance and Correlation

Let X_1 and X_2 be two random variables. Then the covariance of X_1 and X_2 is defined to be

$$\mathrm{Cov}(X_1, X_2) := \mathrm{E}[X_1 X_2] - \mathrm{E}[X_1]\mathrm{E}[X_2]$$

and the correlation of X_1 and X_2 is then defined to be

$$\text{Corr}(X_1, X_2) = \rho(X_1, X_2) = \frac{\text{Cov}(X_1, X_2)}{\sqrt{\text{Var}(X_1)\text{Var}(X_2)}}.$$

If X_1 and X_2 are independent, then $\rho = 0$, though the converse is *not* true in general. It can be shown that $-1 \leq \rho \leq 1$.

Suppose now that $\mathbf{X} = (X_1, \ldots, X_n)$ is a random vector. Then $\mathbf{\Sigma}$, the *covariance matrix* of $\mathbf{X}$, is the $(n \times n)$ matrix that has $(i,j)^{th}$ element given by $\mathbf{\Sigma}_{i,j} := \text{Cov}(X_i, X_j)$.

Properties of the Covariance Matrix $\mathbf{\Sigma}$

1. It is symmetric so that $\mathbf{\Sigma}^T = \mathbf{\Sigma}$
2. The diagonal elements satisfy $\mathbf{\Sigma}_{i,i} \geq 0$
3. It is *positive semi-definite* so that $x^T \mathbf{\Sigma} x \geq 0$ for all $x \in \mathbf{R}^n$.

We will now see how to simulate correlated normal random variables.

3.2 Generating Correlated Normal Random Variables

The problem then is to generate $\mathbf{X} = (X_1, \ldots, X_n)$ where $\mathbf{X} \sim \text{MN}(\mathbf{0}, \mathbf{\Sigma})$. Note that it is then easy to handle the case where $\text{E}[\mathbf{X}] \neq \mathbf{0}$. By way of motivation, suppose $Z_i \sim \text{N}(0,1)$ and IID for $i = 1, \ldots, n$. Then

$$c_1 Z_1 + \ldots c_n Z_n \sim \text{N}(0, \sigma^2)$$

where $\sigma^2 = c_1^2 + \ldots + c_n^2$. That is, a linear combination of normal random variables is again normal.

More generally, let $\mathbf{C}$ be a $(n \times m)$ matrix and let $\mathbf{Z} = (Z_1 \; Z_2 \; \ldots \; Z_n)^T$. Then

$$\mathbf{C}^T Z \sim \text{MN}(0, \mathbf{C}^T \mathbf{C})$$

so our problem clearly reduces to finding $\mathbf{C}$ such that

$$\mathbf{C}^T \mathbf{C} = \mathbf{\Sigma}.$$

Question: Why is this true?

Finding such a matrix, $\mathbf{C}$, requires us to compute the *Cholesky decomposition* of $\mathbf{\Sigma}$.

3.3 The Cholesky Decomposition of a Symmetric Positive-Definite Matrix

A well known fact from linear algebra is that any symmetric positive-definite matrix, $\mathbf{M}$, may be written as

$$\mathbf{M} = \mathbf{U}^T \mathbf{D} \mathbf{U}$$

where $\mathbf{U}$ is an upper triangular matrix and $\mathbf{D}$ is a diagonal matrix with positive diagonal elements.

Since our variance-covariance matrix, $\mathbf{\Sigma}$, is symmetric positive-definite, we can therefore write

$$\begin{aligned} \Sigma &= \mathbf{U}^T \mathbf{D} \mathbf{U} \\ &= (\mathbf{U}^T \sqrt{\mathbf{D}})(\sqrt{\mathbf{D}} \mathbf{U}) \\ &= (\sqrt{\mathbf{D}} \mathbf{U})^T (\sqrt{\mathbf{D}} \mathbf{U}). \end{aligned}$$

The matrix $\mathbf{C} = \sqrt{\mathbf{D}} \mathbf{U}$ therefore satisfies $\mathbf{C}^T \mathbf{C} = \mathbf{\Sigma}$. It is called the Cholesky Decomposition of $\mathbf{\Sigma}$.

3.3.1 Cholesky Decomposition in Matlab

It is easy to compute the Cholesky decomposition of a symmetric positive-definite matrix in Matlab using the *chol* command. This means it is also easy to simulate multivariate normal random vectors as well.

As before, let $\mathbf{\Sigma}$ be an $(n \times n)$ variance-covariance matrix and let $\mathbf{C}$ be its Cholesky decomposition. If $\mathbf{X} \sim \mathsf{MN}(\mathbf{0}, \mathbf{\Sigma})$ then we can generate random samples of $\mathbf{X}$ in Matlab as follows:

Sample Matlab Code

```
>> Sigma = [1.0 0.5 0.5;
            0.5 2.0 0.3;
            0.5 0.3 1.5];

>> C=chol(Sigma);
>> Z=randn(3,1000000);
>> X=C'*Z;
>>  cov(X')

ans =
    0.9972    0.4969    0.4988
    0.4969    1.9999    0.2998
    0.4988    0.2998    1.4971
```

Remark 2 *We must be very careful to premultiply* Z *by* $\mathbf{C}^T$ *and not* $\mathbf{C}$.

Example 4 **(A Faulty Σ)**

We must ensure that $\mathbf{\Sigma}$ is a genuine variance-covariance matrix. Consider the following Matlab code.

Matlab Code

```
>> Sigma=[0.5 0.9 0.4;
          0.9 0.7 0.9;
          0.4 0.9 0.9];

>> C=chol(Sigma)
```

Question: What is the problem here? ∎

More formally, we have the following algorithm for generating multivariate random vectors, $\mathbf{X}$.

Generating Correlated Normal Random Variables

generate $\mathbf{Z} \sim \mathsf{MN}(\mathbf{0}, \mathbf{I})$
/$*$ Now compute the Cholesky Decomposition $*$/
compute $\mathbf{C}$ such that $\mathbf{C}^T\mathbf{C} = \mathbf{\Sigma}$
set $\mathbf{X} = \mathbf{C}^T\mathbf{Z}$

3.4 Generating Correlated Brownian Motions

Generating correlated Brownian motions is, of course, simply a matter of generating correlated normal random variables.

Definition 1 *We say B_t^a and B_t^b are correlated SBM's with correlation coefficient ρ if* $\mathrm{E}[B_t^a B_t^b] = \rho t$.

For two such SBM's, we then have

$$\begin{aligned}\mathsf{Corr}(B_t^a, B_t^b) &= \frac{\mathrm{Cov}(B_t^a, B_t^b)}{\sqrt{\mathrm{Var}(B_t^a)\mathrm{Var}(B_t^b)}} \\ &= \frac{\mathrm{E}[B_t^a B_t^b] - \mathrm{E}[B_t^a]\mathrm{E}[B_t^b]}{t} \quad = \quad \rho.\end{aligned}$$

Now suppose S_t^a and S_t^b are stock prices that follow GBM's such that

$$\mathsf{Corr}(B_t^a, B_t^b) = \rho$$

where B_t^a and B_t^b are the standard SBM's driving S_t^a and S_t^b, respectively. Let r_a and r_b be the continuously compounded returns of S_t^a and S_t^b, respectively, between times t and $t+s$. Then it is easy to see that

$$\mathsf{Corr}(r_a, r_b) = \rho.$$

This means that when we refer to the correlation of stock returns, we are at the same time referring to the correlation of the SBM's that are driving the stock prices. We will now see by example how to generate correlated SBM's and, by extension, GBM's.

Example 5 (Portfolio Evaluation Revisited)

Recalling the notation of Example 1, we assume again that $S^a \sim GBM(\mu_a, \sigma_a)$ and $S^b \sim GBM(\mu_b, \sigma_b)$. However, we no longer assume that S_t^a and S_t^b are independent. In particular, we assume that

$$\mathrm{Corr}(B_t^a, B_t^b) = \rho$$

where B_t^a and B_t^b are the SBM's driving A and B, respectively. As mentioned earlier, this implies that the correlation between the return on A and the return on B is equal to ρ. We would like to estimate

$$\mathbf{P}\left(\frac{W_T}{W_0} \leq .9\right),$$

i.e., the probability that the value of the portfolio drops by more than 10%. Again, let L be the event that $W_T/W_0 \leq .9$ so that that the quantity of interest is $\theta := \mathbf{P}(L) = \mathrm{E}[I_L]$, where $\mathbf{X} = (S_T^a, S_T^b)$ and

$$I_L(\mathbf{X}) = \begin{cases} 1 & \text{if } \frac{n_a S_T^a + n_b S_T^b}{n_a S_0^a + n_b S_0^b} \leq 0.9 \\ \\ 0 & \text{otherwise.} \end{cases}$$

Note that we can write

$$\begin{aligned}S_{t+s}^a &= S_t^a \ \exp\left((\mu_a - \sigma_a^2/2)s + \sqrt{s}\ V_a\right) \\ S_{t+s}^b &= S_t^b \ \exp\left((\mu_b - \sigma_b^2/2)s + \sqrt{s}\ V_b\right)\end{aligned}$$

where $(V_a, V_b) \sim \mathsf{MN}(\mathbf{0}, \mathbf{\Sigma})$ with

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_a^2 & \sigma_a\sigma_b\rho \\ \sigma_a\sigma_b\rho & \sigma_b^2 \end{pmatrix}.$$

So to generate one sample value of (S_{t+s}^a, S_{t+s}^b), we need to generate one sample value of $\mathbf{V} = (V_a, V_b)$. We do this by first generating $\mathbf{Z} \sim \mathsf{MN}(\mathbf{0}, \mathbf{I})$, and then setting $\mathbf{V} = \mathbf{C}^T\mathbf{Z}$, where $\mathbf{C}$ is the Cholesky decomposition of $\mathbf{\Sigma}$. To estimate θ we then set $t = 0$, $s = T$ and generate n samples of $I_L(\mathbf{X})$. The following Matlab function accomplishes this.

A Matlab Function

```
function[theta] = portfolio_evaluation(mua,mub,siga,sigb,n,T,rho,S0a,S0b,na,nb);

% This function estimates the probability that wealth of the portfolio falls
%by more than 10% n is the number of simulated values of W_T that we use

W0 = na*S0a + nb*S0b;

Sigma = [siga^2 siga*sigb*rho;
         siga*sigb*rho  sigb^2];

B = randn(2,n);
C=chol(Sigma);
V = C' * B;

STa = S0a * exp((mua - (siga^2)/2)*T + sqrt(T)*V(1,:));
STb = S0b * exp((mub - (sigb^2)/2)*T + sqrt(T)*V(2,:));

WT = na*STa + nb*STb;

theta = mean(WT/W0 < .9);
```

The function *portfolio_evaluation.m* can now be executed by typing *portfolio_evaluation* at the Matlab prompt.
■

3.4.1 Generating Correlated Log-Normal Random Variables

Let X be a multivariate lognormal random vector with mean μ' and variance -covariance matrix[3] $\mathbf{\Sigma}'$. Then we can write $\mathbf{X} = (e^{Y_1}, \ldots, e^{Y_n})$ where

$$\mathbf{Y} := (Y_1, \ldots, Y_n) \sim \mathsf{MN}(\mu, \mathbf{\Sigma}).$$

Suppose now that we want to generate a value of the vector $\mathbf{X}$. We can do this as follows:

1. Solve for μ and $\mathbf{\Sigma}$ in terms of μ' and $\mathbf{\Sigma}'$.
2. Generate a value of $\mathbf{Y}$
3. Take $\mathbf{X} = \exp(\mathbf{Y})$

Step 1 is straightforward and only involves a few lines of algebra. (See Law and Kelton for more details.) In particular, we now also know how to generate multivariate log-normal random vectors.

[3]For a given positive semi-definite matrix, $\mathbf{\Sigma}'$, it should be noted that it is not necessarily the case that a multivariate lognormal random vector exists with variance -covariance matrix, $\mathbf{\Sigma}'$.

4 Simulating Correlated Random Variables in General

In general, there are two distinct approaches to modelling with correlated random variables. The first approach is one where the joint distribution of the random variables is fully specified. In the second approach, the joint distribution is not fully specified. Instead, only the marginal distributions and correlations between the variables are specified.

4.1 When the Joint Distribution is Fully Specified

Suppose we wish to generate a random vector $\mathbf{X} = (X_1, \ldots, X_n)$ with joint CDF

$$F_{x_1,\ldots,x_n}(x_1, \ldots, x_n) = \mathbf{P}(X_1 \leq x_1, \ldots, X_n \leq x_n)$$

Sometimes we can use the *method of conditional distributions* to generate $\mathbf{X}$. For example, suppose $n = 2$. Then

$$\begin{aligned} F_{x_1,x_2}(x_1, x_2) &= \mathbf{P}(X_1 \leq x_1, X_2 \leq x_2) \\ &= \mathbf{P}(X_1 \leq x_1)\,\mathbf{P}(X_2 \leq x_2 | X_1 \leq x_1) \\ &= F_{x_1}(x_1) F_{x_2|x_1}(x_2|x_1) \end{aligned}$$

So to generate (X_1, X_2), first generate X_1 from $F_{x_1}(\cdot)$ and then generate X_2 independently from $F_{x_2|x_1}(\cdot)$. This of course will work in general for any value of n, assuming we can compute and simulate from all the necessary conditional distributions. In practice the method is somewhat limited because we often find that we cannot compute and / or simulate from the distributions.

We do, however, have flexibility with regards to the *order* in which we simulate the variables. Regardless, for reasonable values of n this is often impractical. However, one very common situation where this method is feasible is when we wish to simulate stochastic processes. In fact we use precisely this method when we simulate Brownian motions and Poisson processes. More generally, we can use it to simulate other stochastic processes including Markov processes and time series models among others.

Another very important method for generating correlated random variables is the *Markov Chain Monte Carlo* (MCMC) method[4].

4.2 When the Joint Distribution is not Fully Specified

Sometimes we do not wish to specify the full joint distribution of the random vector that we wish to simulate. Instead, we may only specify the marginal distributions and the correlations between the variables. Of course such a problem is then not fully specified since in general, there will be many different joint probability distributions that have the same set of marginal distributions and correlations. Nevertheless, this situation often arises in modelling situations when there is only enough data to estimate marginal distributions and correlations between variables. In this section we will briefly mention some of the issues and methods that are used to solve such problems.

Again let $\mathbf{X} = (X_1, \ldots, X_n)$ be a random vector that we wish to simulate.

- Now, however, we do *not* specify the joint distribution $F(x_1, \ldots, x_n)$
- Instead, we specify the marginal distributions $F_{x_i}(x)$ and the covariance matrix $\mathbf{R}$

[4] See Ross for an introduction to MCMC.

  - Note again that in general, the marginal distributions and $\mathbf{R}$ are not enough to uniquely specify the joint distribution

Even now we should mention that potential difficulties already exist. In particular, we might have a *consistency* problem. That is, the particular $\mathbf{R}$ we have specified may not be consistent with the specified marginal distributions. In this case, there is *no* joint CDF with the desired marginals and correlation structure.

Assuming that we do not have such a consistency problem, then one possible approach, based on the multivariate normal distribution, is as follows.

1. Let $(Z_1, \ldots, Z_n) \sim \mathsf{MN}(\mathbf{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma}$ is a covariance matrix with 1's on the diagonal. Therefore $Z_i \sim \mathsf{N}(0, 1)$ for $i = 1, \ldots, n$.
2. Let $\phi(\cdot)$ and $\Phi(\cdot)$ be the PDF and CDF, respectively, of a standard normal random variable. It can then be seen that $(\Phi(Z_1), \ldots, \Phi(Z_n))$ is a random vector where the marginal distribution of each $\Phi(Z_i)$ is uniform. How would you show this?
3. Since the Z_i's are correlated, the uniformly distributed $\Phi(Z_i)$'s will also be correlated. Let $\mathbf{\Sigma}'$ denote the variance-covariance matrix of the $\Phi(Z_i)$'s.
4. We now set $X_i = F_{x_i}^{-1}(\Phi(Z_i))$. Then X_i has the desired marginal distribution, $F_{x_i}(\cdot)$. Why?
5. Now since the $\Phi(Z_i)$'s are correlated, the X_i's will also be correlated. Let $\mathbf{\Sigma}''$ denote the variance-covariance matrix of the X_i's.
6. Recall that we want $\mathbf{X}$ to have marginal distributions $F_{x_i}(\cdot)$ and covariance matrix $\mathbf{R}$. We have satisfied the first condition and so to satisfy the second condition, we must have $\mathbf{\Sigma}'' = \mathbf{R}$. So we must choose the original $\mathbf{\Sigma}$ in such a way that
$$\mathbf{\Sigma}'' = \mathbf{R}. \tag{1}$$
7. In general, choosing $\mathbf{\Sigma}$ appropriately is not trivial and requires numerical work. It is also true that there does not always exist a $\mathbf{\Sigma}$ such that (1) holds.