Financial Risk Forecasting
Chapter 7
Simulation methods for VaR for options and bonds

Jon Danielsson ©2019
London School of Economics

To accompany
Financial Risk Forecasting
www.financialriskforecasting.com
Published by Wiley 2011

Version 4.0, November 2019
The focus of this chapter

- Chapter 6 demonstrates limitations of analytical VaR methods for options and bonds
- The focus in this chapter is on simulation methods, sometimes called *Monte Carlo (MC) simulation*
  1. Pseudo random number generators (RNG)
  2. Simulation pricing of options and bonds
  3. Simulation VaR for one asset and portfolios
  4. Issues in Monte Carlo estimation
Idea

- Replicate a part of the world in computer software
- For example market outcomes, based on some model of market evolution
- Sufficient number of simulations (replications) ideally yield a large and representative sample of market outcomes
- Use that to calculate quantities of interest (e.g. VaR)
Calendar time and trading time

- We use two different measures of time
  
  **Calendar time** (365/6 days) used for interest rate calculations

  **Trading time** (≈ 250 days) used for risk calculations

- This is because we earn interest every day

- But calculate volatilities only from days (trading days) when stock exchanges open (Mondays to Fridays, excluding holidays)

- R will allow precise date calculations and has a database of dates when various exchanges are open
Notation

$F$ Futures price
$g$ Derivative price
$S$ Number of simulations
$x^b$ Portfolio holdings (basic assets)
$x^o$ Portfolio holdings (derivatives)
Random numbers and Monte Carlo simulations
Obtaining random numbers

- The fundamental input in MC analysis is a long sequence of random numbers (RNs)
- Creating a large sample of *high-quality* RNs is difficult
- It is impossible to obtain pure random numbers
  - there is no natural phenomena that is purely random
  - computers are deterministic by definition
- Computer algorithm known as *pseudo random number generator* (RNG), which creates outcomes that appear to be random even if they are deterministic
Pseudo random number generators

- A particular of RN is generated by a function of a previous RN
  \[ u_{i+1} = h(u_i) \]
  where \( u_i \) is the \( i^{th} \) RN and \( h(\cdot) \) is the RNG

- If RNs are truly random, it is essential that their unconditional distribution is IID *uniform*
Period of a random number generator

**Definition** Random number generators can only provide a fixed number of different random numbers, after which they repeat themselves. This fixed number is called a *period*.

- Symptoms of low-quality RNGs include:
  - Low period (RNG repeats itself quickly)
  - Serial dependence
  - Deviations from uniform distribution
Linear congruential generators

- No numerical algorithm generates truly random numbers
- The best known RNGs are so-called linear congruential generators (LCGs), which link \( i^{th} \) and \( i + 1^{th} \) integer in the sequence of RNs by

\[
u_{i+1} = (a \times u_i + c) \mod m
\]

where \( a \) is multiplier; \( c \) is increment; \( m \) is integer modulus and \( \mod \) is the modulus function (remainder after division)
- The first RN in a sequence is called seed and is usually chosen by the user
**Illustration**

**RNGs, seed, size and period**

Think of the *RNG* as an ellipse where each point represents a RN and the number of RNs is finite.
The *seed* determines the starting point of the sequence of RNs and is set by the user.
Illustration

RNGs, seed, size and period

The *seed* determines the starting point of the sequence of RNs and is set by the user.
Illustration

RNGs, seed, size and period

The *seed* determines the starting point of the sequence of RNs and is set by the user.
Illustration

RNGs, seed, size and period

The *seed* determines the starting point of the sequence of RNs and is set by the user.
The *seed* determines the starting point of the sequence of RNs and is set by the user.
Illustration

RNGs, seed, size and period

Think of the sequence generated as a specific arc of the ellipse which depends on the chosen seed.
Illustration

RNGs, seed, size and period

The size of the simulation determines the length of the sequence.
The size of the simulation determines the length of the sequence.
Illustration

RNGs, seed, size and period

The *period* of the simulation is the number of RNs that the RNG is able to generate without repeating itself.
Linear congruential generators

- Main flaw of LCGs is serial correlation, which cannot be easily eliminated
- More complicated RNGs which introduce nonlinearities are generally preferred
- The default RNG in R and Matlab is the Mersenne twister, which has a period of $2^{19,937} - 1$ (a Mersenne prime number)
Nonuniform RNGs

- Most RNGs generate uniform random numbers, usually scaled so that \((u) \in [0, 1]\)
- However, most practical applications require RNs from a different distribution
- To obtain such RNs we use *transformation methods* which convert uniform numbers into RNs from the distribution of interest
  - The inverse distribution is obvious candidate, but often slow and inaccurate
Normal inverse distribution

![Normal inverse distribution graph](image-url)
The Box-Muller method

- The most common method for generating normal RNs is the **Box-Muller method**, which is more computationally efficient than using the inverse distribution.

- If we generate two uniforms \((u_1, u_2)\), we can transform the pair into a pair of IID normals \((n_1, n_2)\) by:
  
  \[
  n_1 = \sqrt{-2 \log u_1 \sin(2\pi u_2)} \\
  n_2 = \sqrt{-2 \log u_1 \cos(2\pi u_2)}
  \]

- The Box-Muller method is fine for casual computations.
  - May not be the best method as in some circumstances the two normals are not fully independent.

- Range of transformation methods in R and Matlab.
Random numbers in R

\begin{verbatim}
runif(1) 0.704862
runif(1) 0.2267493
runif(1) 0.9921351

set.seed(999); runif(1) 0.3890714
set.seed(999); runif(1) 0.3890714

x=rnorm(n=5)
-0.2817402 -1.3125596 0.7951840 0.2700705 -0.2773064

x=rt(n=5, df=3)
-0.34910572 -0.34198996 0.39319432 0.07968276 0.46555150

plot(rnorm(n=1000), type='l')
plot(rt(n=1000, df=2), type='l')
\end{verbatim}
Random numbers in Matlab

```
rand(1,1) 0.0639
rand(1,1) 0.8074
rand(1,1) 0.6784

rng(999); rand(1,1) 0.8034
rng(999); rand(1,1) 0.8034

x=randn(1,5)
 0.2011  -0.7825  0.2060  -0.7940  -0.2121

x=trnd(3,1,5)
-0.3981   0.1021   0.4853   0.5690   1.5536

plot(randn(1,1000))
plot(trnd(2,1,1000))
```
Simulation pricing of bonds
Bond pricing

• Price and risk of fixed income assets (e.g. bonds) is based on market interest rates
• Using a model of the distribution of interest rates, we can simulate random yield curves and obtain the distribution of bond prices
• We map distribution of interest rates to the distribution of bond prices
Analytical bond pricing

- Denote $r_t$ the annual interest rate at time $t$
- The present value of a bond is the present discounted value of its cash flows:

\[
P = \sum_{t=1}^{T} \frac{\tau_t}{(1 + r_t)^t}
\]

- Where $P$ is bond price and $\tau_t$ is cash flow at time $t$
Analytical bond pricing

- Suppose we have a bond with 10 years to expiration, 7% annual interest, $10 par value, and current market rates:

\[
\{r_t\}_{t=1}^{10} = (5.00, 5.69, 6.09, 6.38, 6.61, 6.79, 7.07, 7.19, 7.30) \times 0.01
\]

- The bond has a current value of $9.91
Simulated bond pricing

- Assume here that the yield curve can only shift up and down, not change shape
- Shocks to yields, $\epsilon_i$

$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

- For the sake of demonstration we set the number of simulations as $S = 8$, but note that accurate estimates require much more simulations
Eight yield curve simulations
Eight yield curve simulations
Eight yield curve simulations

![Graph showing eight yield curve simulations](image-url)
Eight yield curve simulations

Time

Yield

4 % 5 % 6 % 7 % 8 % 9 % 10 %

2 4 6 8 10

True Simulated

Financial Risk Forecasting © 2011,2019 Jon Danielsson, page 36 of 123
Eight yield curve simulations

![Yield curve simulations graph](image-url)
Simulated bond pricing

- The equation for the $i^{th}$ simulated price, $P_i$, now becomes

$$P_i = \sum_{t=1}^{T} \frac{T_t}{(1 + r_t + \epsilon_i)^t}$$

where $r_t + \epsilon_i$ is the $i^{th}$ simulated interest rate at time $t$

- Compare the eight bond prices obtained with $S = 8$ yield curve simulations with the distribution of bond prices when $S = 50,000$
Eight bond price simulations

<table>
<thead>
<tr>
<th>Simulation</th>
<th>True price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$10</td>
</tr>
<tr>
<td>2</td>
<td>$10</td>
</tr>
<tr>
<td>3</td>
<td>$8</td>
</tr>
<tr>
<td>4</td>
<td>$10</td>
</tr>
<tr>
<td>5</td>
<td>$10</td>
</tr>
<tr>
<td>6</td>
<td>$10</td>
</tr>
<tr>
<td>7</td>
<td>$8</td>
</tr>
<tr>
<td>8</td>
<td>$10</td>
</tr>
</tbody>
</table>
Density of simulated bond prices

Normal distribution superimposed, $S = 50,000$

![Graph showing the density of simulated bond prices with normal distribution superimposed.](Image)
Allowing yield curve to change shape

- Key assumptions:
  - Yield curve only shifts up and down
  - Distribution of interest rate changes is normal
- The assumptions may be unrealistic in practice but it is relatively straightforward to relax them
  - in practice it rotates and twists
  - can use principal components (PCA)
Simulation pricing of options
Simulation approach

• The price of an asset be the expectation of its final payoff under *risk neutrality*
• Depends on the price movements of its underlying asset
• If sufficient number of price paths are simulated
• Obtain an estimate of the true price
Option pricing

- Get price of European options on non-dividend-paying stock where all Black-Scholes (BS) assumptions hold
- Two primitive assets in BS pricing model:
  - risk-free asset with instantaneous rate \( r \)
  - Underlying stock, follows normally distributed random walk with drift \( r \) (geometric Brownian motion in continuous time)
- The no-arbitrage futures price of stock for delivery at time \( T \) is given by:
  \[
  F = P e^{rT}
  \]
Analytical option pricing

• Suppose we have a European call option with
  1. current stock price $50
  2. 20% annual volatility
  3. 5% annual risk-free rate
  4. 6 months to expiration
  5. $40 strike price

• The price is $11.0873
Simulated option pricing

- We simulate returns until expiration and use these values to calculate simulated futures prices.
- With sufficient sample of futures prices we can compute the set of payoffs of the option.
- The MC price is then given by the mean of these payoffs.
Simulated option pricing

- The only complexity is due to expectation of a log-normal RN, i.e. if

\[ O \sim \mathcal{N}(\mu, \sigma^2) \]

then:

\[ \mathbb{E}[\exp(O)] = \exp\left(\mu + \frac{1}{2}\sigma^2\right) \]

- We apply a *log-normal correction* (subtract \(\frac{1}{2}\sigma^2\) from simulated stock return) to ensure that expectation of simulated returns is the same as theoretical value

- See density of \(S = 10^6\) futures prices and option payoffs using same values as in the example before
Density of simulated futures prices

$S = 10^6$, normal distribution superimposed
Density of simulated option prices

Based on simulated futures prices with $S = 10^6$
Simulated option pricing

Numerical example

• Mean of simulated option prices is the MC price
  • In this case R gives $11.08709, close enough to the Black-Scholes price of $11.0873
• Note the asymmetry in the density of simulated futures prices (result of log-normal distribution of prices)
• The VaR can be read off the previous graph, e.g. 1% smallest value of distribution gives 99% VaR
Black-Scholes in R

```r
bs <- function(K, S0, r, sigma, t0, T){
  d1 <- (log(S0/K) + (r + 0.5*sigma^2)*(T-t0)) / (sigma*sqrt(T-t0))
  d2 <- d1 - sigma*sqrt(T-t0)
  Call <- S0*pnorm(d1) - K*exp(-r*(T-t0))*pnorm(d2)
  Put <- K*exp(-r*(T-t0))*pnorm(-d2) - S0*pnorm(-d1)
  Delta.call <- pnorm(d1)
  Delta.call <- Delta.call - 1
  Gamma <- dnorm(d1)/(S0*sigma*sqrt(T-t0))
  Vega <- S0*dnorm(d1)*sqrt(T-t0)
  Theta.call <- -S0*dnorm(d1)*sigma/(2*sqrt(T-t0)) - r*K*exp(-r*(T-t0))
  Theta.put <- -S0*dnorm(d1)*sigma/(2*sqrt(T-t0)) + r*K*exp(-r*(T-t0))
  Rho.call <- K*T*exp(-r*(T-t0))*pnorm(d2)
  Rho.put <- -K*T*exp(-r*(T-t0))*pnorm(-d2)
  return(list(Call=Call, Put=Put, Delta.call=Delta.call))
}
```
source('bs.r')
S0 = 50
sigma = 0.2
r = 0.05
Maturity = 0.5
X = 40
f = bs(X, S0, r, sigma, Maturity)
f = bs(X, S0=S0, r=r, sigma=sigma, Maturity=Maturity)

Call       Put
11.08728   0.09967718
R option simulation

\( S = 1e6 \)
\[
\text{set.seed}(12)
\]
\[
F = S0 * \exp(r * \text{Maturity})
\]
\[
y_{\text{sim}} = \text{rnorm}(S, -0.5 * \sigma * \sigma * \text{Maturity}, \sigma * \sqrt{\text{Maturity}})
\]
\[
F_{\text{sim}} = F * \exp(y_{\text{sim}})
\]
\[
P_{\text{sim}} = F_{\text{sim}} - X
\]
\[
O_{\text{Psim}}[P_{\text{sim}} < 0] = 0
\]
\[
O_{\text{Psim}} = O_{\text{Psim}} * \exp(-r * \text{Maturity})
\]
\[
\text{hist}(F_{\text{sim}}, \text{probability=TRUE, ylim=c(0, 0.06))}
\]
\[
x = \text{seq}(	ext{min}(F_{\text{sim}}), \text{max}(F_{\text{sim}}), \text{length}=100)
\]
\[
\text{lines}(x, \text{dnorm}(x, \text{mean}=\text{mean}(F_{\text{sim}}), \text{sd}=\text{sd}(F_{\text{sim}})))
\]
\[
\text{hist}(O_{\text{Psim}}, \text{nclass}=100, \text{probability=TRUE})
\]
Black-Scholes in Matlab

\[
\begin{align*}
S_0 &= 50; \\
\sigma &= 0.2; \\
r &= 0.05; \\
Maturity &= 0.5; \\
X &= 40; \\
f &= bs(X, S_0, r, \sigma, \text{Maturity}); \\
\end{align*}
\]

\[
\begin{align*}
f &= \\
\text{Call} &: 11.0873 \\
\text{Put} &: 0.0997 \\
\end{align*}
\]
Matlab option simulation

```matlab
randn ('state',0);
S = 1e6;
Fsim = S0*exp(r*Maturity);

ysim = randn(S,1)*sigma*sqrt(Maturity)−0.5*Maturity*sigma^2;
Fsim=Fsim*exp(ysim);
Psim = Fsim−X;
OPsim(find(OPsim < 0)) = 0;
OPsim =OPsim*exp(−r*Maturity) ;

histfit(Psim)
hist(OPsim,100)
```
Simulation of VaR
Simulation of VaR for one asset

- Simulate one-day return of an asset
- Apply analytical pricing formulas to simulated future price
- Obtain simulated profits/losses (P/L) as difference between tomorrow’s simulated future values and today’s known value
- Calculate MC VaR from simulated P/L
Setup

- Consider asset with price $P_t$ and IID normal returns, with one-day volatility $\sigma$ and risk-free rate $r$ (in continuous time)
- Number of units of basic asset held in a portfolio is denoted by $x^b$, while $x^o$ indicates number of options held
- Note that it is the $t + 1$ price that will be simulated
We will go through a series of ever more complicated examples

1. Simulation of VaR for one asset (no option)
2. Simulation of VaR for one option
3. Simulation of VaR for a portfolio of one option and one stock
1. VaR with one basic asset

Six-step procedure for obtaining MC VaR

1. Compute initial portfolio value:

\[ \vartheta_t = x^b P_t \]
1. VaR with one basic asset

Six-step procedure for obtaining MC VaR

1. Compute initial portfolio value:

$$\vartheta_t = x^b P_t$$

2. Simulate $S$ one-day returns

$$y_{t+1,i} \sim \mathcal{N} \left(0, \sigma^2\right), \quad i = 1, \ldots, S$$
1. VaR with one basic asset

Six-step procedure for obtaining MC VaR

3. Calculate one-day future price:

\[ P_{t+1,i} = P_t e^{r(1/365)} \times e^{y_{t+1,i}} \times e^{-0.5\sigma^2} \]

1. multiplying the future price with the exponential of the simulated return
2. and the log-normal correction
1. VaR with one basic asset

Six-step procedure for obtaining MC VaR

4. Calculate the simulated futures value of the portfolio:

\[ \vartheta_{t+1,i} = x^b P_{t+1,i} \]
1. VaR with one basic asset

Six-step procedure for obtaining MC VaR

4. Calculate the simulated futures value of the portfolio:

\[ \vartheta_{t+1,i} = x^b P_{t+1,i} \]

5. The \( i^{th} \) simulated P/L is then:

\[ q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t \]
1. VaR with one basic asset

Six-step procedure for obtaining MC VaR

4. Calculate the simulated futures value of the portfolio:

\[ \vartheta_{t+1,i} = x^b P_{t+1,i} \]

5. The \( i^{\text{th}} \) simulated P/L is then:

\[ q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t \]

6. VaR can be obtained directly from the vector of simulated P/L, \( \{q_{t+1,i}\}_{i=1}^S \), e.g. VaR(0.01) is the 1% smallest value
1. MC VaR with one basic asset

Numerical example

- One stock with price $P_t = 100$ and daily volatility $\sigma = 0.01$
- Annual risk-free rate is $r = 5\%$
- Use $S = 10^7$ simulations to calculate VaR(0.01)
- R and Matlab give $2.285$ and $2.291$, respectively
  - More simulations should give more equal answers
2. VaR with an option

Modified six-step procedure

• For options we need to modify the procedure
• Let $g(\cdot)$ denote the Black-Scholes equation and suppose we have $x^o$ options
• We replace steps 1 and 4 and come up with the following procedure
2. VaR with an option

Modified six-step procedure

1' Initial portfolio is

\[ \varphi_t = x^0 g \left( P_t, X, T, \sqrt{250}\sigma, r \right) \]
2. VaR with an option

Modified six-step procedure

1′ Initial portfolio is

\[ \vartheta_t = x^o g \left( P_t, X, T, \sqrt{250}\sigma, r \right) \]

2. Simulate \( S \) one-day returns

\[ y_{t+1,i} \sim \mathcal{N} \left( 0, \sigma^2 \right), \quad i = 1, \ldots, S \]
2. VaR with an option

Modified six-step procedure

3. Calculate one-day future price:

\[ P_{t+1,i} = P_t e^{r(1/365)} \times e^{y_{t+1,i}} \times e^{-0.5\sigma^2} \]

- Future price
- Exp. of sim. return
- Log-normal correction
2. VaR with an option

Modified six-step procedure

3 Calculate one-day future price:

\[ P_{t+1,i} = P_t e^{r(1/365)} \times e^{\gamma_{t+1,i}} \times e^{-0.5\sigma^2} \]

4’ The \( i^{th} \) simulated future value of the portfolio is

\[ \vartheta_{t+1,i} = x^0 g \left( P_{t+1,i}, X, T - \frac{1}{365}, \sqrt{250\sigma}, r \right) \]
2. VaR with an option

Modified six-step procedure

5 The $i^{th}$ simulated P/L is then:

$$q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t$$
2. VaR with an option

Modified six-step procedure

5 The $i^{th}$ simulated P/L is then:

$$q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t$$

6 VaR can be obtained directly from vector of simulated P/L, $\{q_{t+1,i}\}_{i=1}^S$, e.g. VaR(0.01) is 1% smallest value
2. MC VaR of option

- One call option with strike price $X = 100$ and 3 months to expiry
- R and Matlab both give VaR(0.01) of $1.21$
2. Density of simulated P/L

Normal distribution superimposed
3. VaR with an options and a stock

Modified six-step procedure

- Now consider the case of a portfolio with both a stock and option(s) on the same stock
- Suppose we only have one type of option
- As in the case where we only had one option on a basic asset, we replace steps 1 and 4
3. VaR with an options and a stock

Modified six-step procedure

1” Initial portfolio is

\[ \vartheta_t = x^b P_t + x^o g \left( P_t, X, T, \sqrt{250\sigma}, r \right) \]
3. VaR with an options and a stock

Modified six-step procedure

1” Initial portfolio is

\[ v_t = x^b P_t + x^o g \left( P_t, X, T, \sqrt{250} \sigma, r \right) \]

2 Simulate \( S \) one-day returns

\[ y_{t+1,i} \sim \mathcal{N} \left( 0, \sigma^2 \right), \quad i = 1, \ldots, S \]
3. VaR with an options and a stock

Modified six-step procedure

3. Calculate one-day future price:

\[ P_{t+1,i} = P_t e^{r(1/365)} \times e^{y_{t+1,i}} \times e^{-0.5\sigma^2} \]

- future price
- exp. of sim. return
- log-normal correction
3. VaR with an options and a stock

Modified six-step procedure

3 Calculate one-day future price:

\[ P_{t+1,i} = P_t e^{r \left( \frac{1}{365} \right)} \times e^{y_{t+1,i}} \times e^{-0.5 \sigma^2} \]

\[ \text{future price} \quad \text{exp. of sim. return} \quad \text{log-normal correction} \]

4 The \( i^{th} \) simulated future value of the portfolio is

\[ \vartheta_{t+1,i} = x^b P_{t+1,i} + x^o g \left( P_{t+1,i}, X, T - \frac{1}{365}, \sqrt{250 \sigma}, r \right) \]
3. **VaR with an options and a stock**

**Modified six-step procedure**

5 The \( i^{th} \) simulated P/L is then:

\[
q_{t+1,i} = \varrho_{t+1,i} - \varrho_t
\]
3. VaR with an options and a stock

Modified six-step procedure

5 The $i^{th}$ simulated P/L is then:

$$q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t$$

6 VaR can be obtained directly from vector of simulated P/L, $\{q_{t+1,i}\}_{i=1}^S$, e.g. VaR(0.01) is 1% smallest value
3. MC VaR of options and a stock

Numerical example

- One call option with strike price $X_1 = 100$ and one put option with strike $X_2 = 110$ along with underlying stock
- Assume that the options expire in 3 months
- R and Matlab both give a VaR(0.01) of $1.50$
R and Matlab code
R VaR with option

```
set.seed(2)
sigma2 = 0.01^2
probability = 0.01
r = 0.05
Price = 100

Maturity = 0.25;
X = 100;
f = bs(X, Price, r, sqrt(sigma2*250, Maturity)
       Call       Put
3.793687  2.551467

S = 1e6
ysim = rnorm(S, mean=r/365-0.5* sigma2, sd=sqrt(sigma2))
Psim = Price*exp(ysim)
q = sort(Psim-Price)
VaR1 = -q[probability*S]
     2.294032
f sim = bs(X, Psim, r, sqrt(sigma2*250, Maturity -(1/365))
q = sort(fsim[,1]-f[,1])
VaR2 = -q[probability*S]
     1.216569
```
Matlab VaR with option

```matlab
randn('state',0);
sigma2 = 0.01^2;
probability = 0.01;
r = 0.05;
Price = 100;
Maturity = 0.25;
X = 100;

f = bs(X, Price, r, sqrt(sigma2*250), Maturity)
   Call: 3.7937
   Put:  2.5515

S=1e6
ysim = randn(S,1)*sqrt(sigma2)+r/365-0.5*sigma2;
Psim = Price*exp(ysim);
q = sort(Psim-Price);
VaR1 = -q(S*probability)
   2.2930
fsim=bs(X, Psim, r, sqrt(sigma2*250), Maturity-(1/365));
q = sort(fsim.Call-f.Call);
VaR2 = -q(probability*S)
   1.2161
```
<table>
<thead>
<tr>
<th>$j$</th>
<th>$S$</th>
<th>MA $\text{VaR}_1$</th>
<th>MA $\text{VaR}_2$</th>
<th>R $\text{VaR}_1$</th>
<th>R $\text{VaR}_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1 \times 10^3$</td>
<td>2.2707</td>
<td>1.2058</td>
<td>2.240173</td>
<td>1.19168</td>
</tr>
<tr>
<td>2</td>
<td>$1 \times 10^3$</td>
<td>2.4058</td>
<td>1.2677</td>
<td>2.228544</td>
<td>1.186287</td>
</tr>
<tr>
<td>3</td>
<td>$1 \times 10^3$</td>
<td>2.5767</td>
<td>1.3447</td>
<td>2.350268</td>
<td>1.242397</td>
</tr>
<tr>
<td>4</td>
<td>$1 \times 10^3$</td>
<td>2.4762</td>
<td>1.2997</td>
<td>2.129596</td>
<td>1.140111</td>
</tr>
<tr>
<td>1</td>
<td>$1 \times 10^4$</td>
<td>2.2592</td>
<td>1.2005</td>
<td>2.397793</td>
<td>1.264096</td>
</tr>
<tr>
<td>2</td>
<td>$1 \times 10^4$</td>
<td>2.3507</td>
<td>1.2426</td>
<td>2.347318</td>
<td>1.241046</td>
</tr>
<tr>
<td>3</td>
<td>$1 \times 10^4$</td>
<td>2.2930</td>
<td>1.2161</td>
<td>2.324661</td>
<td>1.230656</td>
</tr>
<tr>
<td>4</td>
<td>$1 \times 10^4$</td>
<td>2.2902</td>
<td>1.2148</td>
<td>2.208264</td>
<td>1.176864</td>
</tr>
<tr>
<td>1</td>
<td>$1 \times 10^5$</td>
<td>2.3097</td>
<td>1.2238</td>
<td>2.289294</td>
<td>1.214386</td>
</tr>
<tr>
<td>2</td>
<td>$1 \times 10^5$</td>
<td>2.2667</td>
<td>1.2039</td>
<td>2.280243</td>
<td>1.210211</td>
</tr>
<tr>
<td>3</td>
<td>$1 \times 10^5$</td>
<td>2.2845</td>
<td>1.2122</td>
<td>2.292542</td>
<td>1.215883</td>
</tr>
<tr>
<td>4</td>
<td>$1 \times 10^5$</td>
<td>2.2932</td>
<td>1.2162</td>
<td>2.289084</td>
<td>1.214289</td>
</tr>
<tr>
<td>1</td>
<td>$1 \times 10^6$</td>
<td>2.2932</td>
<td>1.2161</td>
<td>2.2903091</td>
<td>1.216135</td>
</tr>
<tr>
<td>2</td>
<td>$1 \times 10^6$</td>
<td>2.2934</td>
<td>1.2163</td>
<td>2.289913</td>
<td>1.214671</td>
</tr>
<tr>
<td>3</td>
<td>$1 \times 10^6$</td>
<td>2.2902</td>
<td>1.2148</td>
<td>2.289533</td>
<td>1.214496</td>
</tr>
<tr>
<td>1</td>
<td>$1 \times 10^7$</td>
<td>2.2911</td>
<td>1.2152</td>
<td>2.291071</td>
<td>1.215205</td>
</tr>
<tr>
<td>2</td>
<td>$1 \times 10^7$</td>
<td>2.2920</td>
<td>1.2156</td>
<td>2.292068</td>
<td>1.215664</td>
</tr>
<tr>
<td>3</td>
<td>$1 \times 10^7$</td>
<td>2.2934</td>
<td>1.2163</td>
<td>2.292396</td>
<td>1.215816</td>
</tr>
<tr>
<td>4</td>
<td>$1 \times 10^7$</td>
<td>2.2912</td>
<td>1.2153</td>
<td>2.291006</td>
<td>1.215175</td>
</tr>
<tr>
<td>1</td>
<td>$1 \times 10^8$</td>
<td>2.2912</td>
<td>1.2153</td>
<td>2.290599</td>
<td>1.214987</td>
</tr>
<tr>
<td>2</td>
<td>$1 \times 10^8$</td>
<td>2.2912</td>
<td>1.2153</td>
<td>2.291051</td>
<td>1.215196</td>
</tr>
<tr>
<td>3</td>
<td>$1 \times 10^8$</td>
<td>2.2908</td>
<td>1.2151</td>
<td>2.291521</td>
<td>1.215412</td>
</tr>
<tr>
<td>4</td>
<td>$1 \times 10^8$</td>
<td>2.2913</td>
<td>1.2153</td>
<td>2.290906</td>
<td>1.215129</td>
</tr>
</tbody>
</table>
Simulation pricing of a portfolio
Simulation of portfolio VaR

- Consider the multivariate case, i.e. the case of more than one underlying assets
  - Main difference: We need to simulate *correlated returns* for all assets
  - Simulated future prices calculated as before and portfolio value obtained by summing up individual simulated asset holdings
Simulation of portfolio VaR

• Suppose we have two non-derivative assets with daily return distribution

\[ \mathcal{N} \left( \mu = \begin{pmatrix} 0.05/365 \\ 0.05/365 \end{pmatrix}, \Sigma = \begin{pmatrix} 0.01 & 0.0005 \\ 0.0005 & 0.02 \end{pmatrix} \right) \]

• Let \( x^b \) be a vector of holdings
Notation

• The notation becomes cluttered for the multivariate case
• Now we have to denote variables by time period, asset and simulation
• We let $P_{t,k,i}$ denote the $i^{th}$ simulated price of asset $k$ at time $t$, that is:

$$P_{\text{time},\text{asset},\text{simulation}} = P_{t,k,i}$$
Portfolio VaR for basic assets

Six-step procedure for obtaining MC portfolio VaR

1. Compute initial portfolio value:

\[ \vartheta_t = \sum_{k=1}^{K} x_k^b P_{t,k} \]
Portfolio VaR for basic assets
Six-step procedure for obtaining MC portfolio VaR

1. Compute initial portfolio value:

\[ \vartheta_t = \sum_{k=1}^{K} x_k^b P_{t,k} \]

2. Simulate a vector of one-day returns from today to tomorrow

\[ y_{t+1,i} \sim \mathcal{N} \left( \mu - \frac{1}{2} \text{Diag}\Sigma, \Sigma \right) \]

Diag\Sigma extracts the diagonal elements of \( \Sigma \) (because of log-normal correction)
**Portfolio VaR for basic assets**

Six-step procedure for obtaining MC portfolio VaR

3 The $i^{th}$ simulated future price of asset $k$ is:

$$P_{t+1,k,i} = P_{t,k} \exp (y_{t+1,k,i})$$
Portfolio VaR for basic assets
Six-step procedure for obtaining MC portfolio VaR

3 The $i^{\text{th}}$ simulated future price of asset $k$ is:

$$P_{t+1,k,i} = P_{t,k} \exp(y_{t+1,k,i})$$

4 The $i^{\text{th}}$ simulated futures value of the portfolio is:

$$\varrho_{t+1,i} = \sum_{k=1}^{K} x_k^b P_{t+1,k,i}$$
Portfolio VaR for basic assets

Six-step procedure for obtaining MC portfolio VaR

5 The $i^{th}$ simulated P/L is then:

$$q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t$$
Portfolio VaR for basic assets

Six-step procedure for obtaining MC portfolio VaR

5 The $i^{th}$ simulated P/L is then:

$$q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t$$

6 VaR can be obtained directly from vector of simulated P/L, $\{q_{t+1,i}\}_{i=1}^S$, as before
Portfolio VaR for options

Modified six-step procedure

- For options we need to modify steps 1 and 4 from the procedure outlined above
  - Similar to modifications for the univariate case before
- For simplicity suppose the portfolio has only one type of option type per stock
Portfolio VaR for options

Modified six-step procedure

1' Initial portfolio is

$$\vartheta_t = \sum_{k=1}^{K} \left( x_k^b P_{t,k} + x_k^o g \left( P_{t,k}, X_k, T, \sqrt{250} \sigma_k, r \right) \right)$$
Portfolio VaR for options

Modified six-step procedure

1' Initial portfolio is

\[ \vartheta_t = \sum_{k=1}^{K} \left( x_k^b P_{t,k} + x_k^o g \left( P_{t,k}, X_k, T, \sqrt{250} \sigma_k, r \right) \right) \]

2 Simulate a vector of one-day returns from today to tomorrow

\[ y_{t+1,i} \sim N \left( \mu - \frac{1}{2} \text{Diag} \Sigma, \Sigma \right) \]

\text{Diag} \Sigma \text{ extracts the diagonal elements of } \Sigma \text{ (because of the log-normal correction)
3 The $i^{th}$ simulated future price of asset $k$ is:

$$P_{t+1,k,i} = P_{t,k} \exp(y_{t+1,k,i})$$
Portfolio VaR for options

Modified six-step procedure

3. The $i^{th}$ simulated future price of asset $k$ is:

$$P_{t+1,k,i} = P_{t,k} \exp(y_{t+1,k,i})$$

4'. The $i^{th}$ simulated future value of the portfolio is

$$\vartheta_{t+1,i} = \sum_{k=1}^{K} (x^b P_{t+1,i} + x^o g \left(P_{t+1,k,i}, X_k, T - \frac{1}{365}, \sqrt{250} \sigma_k, r \right)$$
5 The $i^{th}$ simulated P/L is then:

$$q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t$$
Portfolio VaR for options

Modified six-step procedure

5 The $i^{th}$ simulated P/L is then:

$$q_{t+1,i} = \vartheta_{t+1,i} - \vartheta_t$$

6 VaR can be obtained directly from vector of simulated P/L, $\{q_{t+1,i}\}_{i=1}^S$, as before
Richer versions

- We used simple examples to avoid cluttered notation, straightforward to allow for more complicated portfolios
  - Number of stocks and multiple options on each stock
  - American (or more exotic) options
  - Combination of fixed income assets with stocks and options
- Also, we could use other distributions (e.g. Student-t or even historical simulation)
Issues in simulation estimation
Simulation issues

- Several issues need to be addressed in all MC exercises, of which two are most important:
  1. Quality of RNG and transformation method
  2. Number of simulations
Quality of RNG

- MC simulation is not only dependent on quality of the underlying stochastic model, also depends on quality of the RNG used
- Low-quality generators give biased or inaccurate results
  - E.g. a simulation size of 100 with period of 10 will repeat same calculation 10 times
- Complicated portfolios may demand large number of RNs and therefore high-quality RNGs
Quality of RNG

- Many transformation methods are only optimally tuned for the center of the distribution
- This becomes particularly problematic when simulating extreme events
- Some transformation methods use linear approximations for extreme tails, which leads to extreme uniforms being incorrectly transformed
Choosing number of simulations

- Choosing appropriate number of simulations is important
  - Too few give inaccurate answers
  - Too many waste time and computer resources
- In special cases formal statistical tests provide guidance, but usually informal methods have to be relied upon
- It is sometimes stated that accuracy of simulations is related to inverse simulation size
  - This is based on assumption of linearity, which is not correct for the problems in this chapter
Choosing number of simulations

- Best way is to simply increase number of simulations and see how MC estimate converges
- Rule of thumb: Sufficient simulation size when numbers have stopped changing up to three significant digits
- We can also compare *convergence of MC estimate* to the true (analytical) price
Convergence of MC estimate

Comparison with analytical Black-Scholes price

- In a example on slide 31 we computed analytical call price of $11.0873 for a European option
- Now calculate MC estimates for different simulation sizes and compare the results with the true (analytical) price
Cumulative MC estimates

Comparison with analytical Black-Scholes price
Cumulative MC estimates

Comparison with analytical Black-Scholes price

Black-Scholes price
Cumulative MC estimates

Comparison with analytical Black-Scholes price

![Graph showing cumulative MC estimates compared to Black-Scholes price across simulations]
Cumulative MC estimates

Comparison with analytical Black-Scholes price

![Graph showing cumulative MC estimates and comparison with analytical Black-Scholes price]
Cumulative MC estimates

Comparison with analytical Black-Scholes price
Convergence of MC estimate

Comparison with analytical Black-Scholes price

- Based on graph on previous slide, it seems to take about 5000 simulations to get three significant digits correct
- However, there are still fluctuations in the estimate for 5 million simulations
Convergence of MC VaR estimate

• Look at the convergence of MC VaR estimates as the simulation size increases
• Graph MC VaR for a stock with daily volatility 1% along with ±99% confidence intervals
Convergence of MC VaR estimates

With ±99% confidence intervals

![Graph showing convergence of Monte Carlo VaR estimates with confidence intervals.](image-url)
Convergence of MC VaR estimates

With ±99% confidence intervals
Convergence of MC VaR estimates

With ±99% confidence intervals
Convergence of MC VaR estimates

With ±99% confidence intervals