Chapter 1

Multivariate Normal Distribution

Normal distribution is one of the most important distributions in Statistics. Multivariate normal distribution is the natural extension of the univariate normal distribution. In this chapter, the multivariate normal distribution will be formally introduced. Some important properties of multivariate normal distribution are also discussed. Methods for generating multivariate normal random numbers are presented with application to simulate correlated stock prices. This is important in calculating the VaR of portfolio of stocks or derivatives using Monte Carlo method.

1.1 Random Vector

ρ_{ij} = correlation btw X_i, X_j $= \frac{Cov(X_i, X_j)}{\sigma_i \sigma_j}$

$Var(X_i)$

Introductory statistics course starts with random variables, statistical distributions, estimation, hypothesis testing, etc. They only deal with one variable at a time. There are few exceptions like covariance and correlation that deal with two random variables at the same time. In reality, the datasets are far more complicated. With the rapid advance in computer technology and data storage, we usually collect huge amount of information with large number of variables and observations. We need to study the relationship among these variables and their resulting joint distribution. We now aim to extend the random variable to random vector; variance to covariance matrix; univariate distribution to **multivariate** distribution, etc. Many results in the univariate analysis can be generalized to the corresponding multivariate setting.

Copulas

nonlinear corr.

Before we discuss the multivariate normal distribution, let us review some matrix

r.v.

algebra. Let X be $p\times 1$ random vector, $X = \begin{bmatrix} x_1 \\ \vdots \\ x_p \end{bmatrix}$ $E(X) = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_p) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_p \end{bmatrix}$ and

Vector t

$Var(X) = E\{[X - E(X)][X - E(X)]'\} = \Sigma = (\sigma_{ij})$ is the $p\times p$ covariance matrix of X.

when $i=j$, $Var(X_i)$; $j \neq i$, $Cov(X_i, X_j)$

The followings are some important results that we shall use them later on.

1. Let $Y = a'X = a_1 x_1 + \cdots + a_p x_p$ be a linear combination of X, then
   $E(Y) = a'\mu$, $Var(Y) = a'\Sigma a$, $Cov(a'X, b'X) = a'\Sigma b$.

σ_{ij}

$= b'\Sigma a = b'\Sigma'(a')'$

2. In general, if A and B are $r\times p$ matrices and $Y=AX$ is $r\times 1$ random vector, then
   $Var(Y) = A\Sigma A'$ $Cov(AX, BX) = A\Sigma B'$.

$E(Y) = AE(X)$

$(X_i - E(X_i))(X_j - E(X_j)) = b'\Sigma a$

Symmetric

$\begin{bmatrix} x_1 - E(x_1) \\ x_2 - E(x_2) \\ \vdots \\ x_p - E(x_p) \end{bmatrix} [x_1 - E(x_1) \ \ldots \ x_p - E(x_p)] = p \times p$

quadratic form

$Var(y) = Var(a'X)$
$= E((y-E(y))(y-E(y))')$
$= E((a'X - a'\mu)(a'X - a'\mu)')$
$= E(a'(X-\mu)(X-\mu)'a)$
$= a' E((X-\mu)(X-\mu)')a$
$= a'\Sigma a$

3. The correlation matrix R of X can be computed as follows:

   $R = D^{-1/2}\Sigma D^{-1/2}$, where $\Sigma = (\sigma_{ij})$, $D = diag(\sigma_{11},\ldots,\sigma_{pp})$ is a diagonal matrix and $D^{-1/2} = diag(1/\sqrt{\sigma_{11}},\ldots,1/\sqrt{\sigma_{pp}})$.

   The (i, j) element of $R = D^{-1/2}\Sigma D^{-1/2}$ is $\dfrac{\sigma_{ij}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}}} = \rho_{ij}$

   $r_{ij} = \dfrac{\sigma_{ij}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}}}$

   Note that $R = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ \rho_{p1} & & \cdots & 1 \end{bmatrix}$ is a $p \times p$ symmetric matrix.

   $-1 \le \rho_{ij} \le 1$

   nonlinear relation

   $\rho_{xy} = 0$

There are some basic concepts in matrix we need to mention. Let $A = (a_{ij})$ be a $n \times p$ matrix. The **transpose** of A is a $p \times n$ matrix, denoted by $A' = (a_{ji})$. Let $B = (b_{ij})$ be a $p \times p$ **square matrix**. B is **symmetric** if $B' = B$. The **trace** of B, denoted by *tr(B)*, is defined as $tr(B) = \sum_{i=1}^{p} b_{ii}$. It can be shown that $tr(ABC) = tr(CAB) = tr(BCA)$, that is the order can be cyclically changed.

determinant

Example: Blue Chips in Hong Kong Stock Exchange

The data file "fin-ratio.csv" contains financial ratios of 680 securities listed in the main board of Hong Kong Stock Exchange in 2002. There are six financial variables, namely, **Earning Yield** (EV), **Cash Flow to Price** (CFTP), **logarithm of Market Value** (ln_MV), **Dividend Yield** (DY), **Book to Market Equity** (BTME), **Debt to Equity Ratio** (DTE). These financial variables are publicly available information. Among these companies, there are 32 Blue Chips which are the Hang Seng Index Constituent Stocks. The last column **HSI** is a binary variable indicating whether the stock is a Blue Chip or not. For the time being, we ignore this HSI variable in the last column and compute the mean vector, sample covariance matrix and correlation matrix using R as follows:

dimensional analysis

take ln to get linear relat

Rcode: m<- apply(x, 1 (or 2), mean): apply function mean to row (or columns)

```
> d<-read.csv("fin-ratio.csv")          # read in dataset
> names(d)                              # display the var in d
[1] "EY"     "CFTP"   "ln_MV"  "DY"     "BTME"   "DTE"    "HSI"
> x<-d[,1:6]                            # extract the first 6 columns in d
> m<-apply(x,2,mean)                    # save sample mean vector to m
> m
        EY       CFTP      ln_MV         DY       BTME        DTE
-0.6502403 -0.2338956  6.2668068  2.4961735  1.9082626  0.7097322

> S<-var(x)                             # save sample covariance matrix
> S
              EY      CFTP      ln_MV         DY       BTME        DTE
EY    18.4979068 2.9089644  1.1601886  1.9203766  1.4781279  0.3379530
CFTP   2.9089644 3.6930613  0.7662995  1.2371466  1.8228390  0.3287908
ln_MV  1.1601886 0.7662995  2.7439362  0.9720714 -0.7734227 -0.0741322
DY     1.9203766 1.2371466  0.9720714 13.8715626 -0.2575337  0.1581528
BTME   1.4781279 1.8228390 -0.7734227 -0.2575337 68.3081966  1.9617652
DTE    0.3379530 0.3287908 -0.0741322  0.1581528  1.9617652 12.9929072

> cor(x)                                # sample correlation matrix
              EY       CFTP       ln_MV          DY        BTME         DTE
EY    1.00000000 0.35195234  0.16284719  0.11988433  0.04158285  0.02179926
CFTP  0.35195234 1.00000000  0.24072338  0.17284835  0.11476743  0.04746497
ln_MV 0.16284719 0.24072338  1.00000000  0.15756091 -0.05649285 -0.01241557
DY    0.11988433 0.17284835  0.15756091  1.00000000 -0.00836633  0.01178043
BTME  0.04158285 0.11476743 -0.05649285 -0.00836633  1.00000000  0.06585025
DTE   0.02179926 0.04746497 -0.01241557  0.01178043  0.06585025  1.00000000
```

TSS~~X~~ = SSW + SSR

Note that the trace of the sample covariance matrix represents the **total variation** (sum of total variances) of all the variables in the dataset. Also, it is straightforward to check $R = D^{-1/2}SD^{-1/2}$ using R's built-in matrix operations. It is useful to write the sample mean vector and the sample covariance matrix in matrix form as follows: ← $tr(\hat{\Sigma})$

$tr(\hat{\Sigma})$ 80%~90%

Matrix theory

We need more matrix theory in the multivariate analysis. Let us recall some basic definitions and concepts.

Determinant

Let A and B be $p \times p$ square matrices and $|A|$ denotes the **determinant** of A.

1. $|A'| = |A|$.
2. $|\alpha A| = \alpha^p |A|$, where α is a scalar.
3. $|AB| = |A| \times |B|$. [Note that $|A+B| \neq |A| + |B|$ in general].

$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = ad - bc$

The general formula for $|A|$ is

$$|A| = \sum_{\sigma \in S_n} \left(\text{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma_i} \right)$$

where the sum is over all permutations σ of the set $\{1,2,\ldots,n\}$, and $\mathrm{sgn}(\sigma)=1$ with even number of swaps over the ordered set $\{1,2,\ldots,n\}$, while $\mathrm{sgn}(\sigma)=-1$ with odd number of swaps.

Inverse

Let A be a $p\times p$ **nonsingular** matrix, i.e. its inverse A^{-1} exist.

$A^{-1}A = AA^{-1} = I$

4. $(A^{-1})'=(A')^{-1}$.
5. $(AB)^{-1}=B^{-1}A^{-1}$.
6. $|A^{-1}|=1/|A|$.
7. If A is an orthogonal matrix, then $A^{-1}=A'$.
8. If $D=diag(d_1,\ldots d_p)$ with $d_i\neq 0$, then $D^{-1}=diag(d_1^{-1},\ldots d_p^{-1})$.

$X^tA = AX^t = I$

Positive definite matrix

A $p\times p$ symmetric matrix A is called **positive definite** (denoted by $A>0$) if for any nonzero $p\times 1$ vector x, $x'Ax>0$.(semi-positive definite if $x'Ax\geq 0$ denoted by $A\geq 0$).

$x'Ax \geq 0 \quad \forall x$.

9. If A>0, then $A^{-1}>0$.
10. For any matrix B, $B'B\geq 0$.

Proof. Let $y=Bx$, $x'B'Bx=y'y=\sum_{i=1}^{p}y_i^2\geq 0$.

inner product.

11. The sample covariance matrix is semi-positive definite.

Proof. It is suffice to show that the SSCP matrix $A\geq 0$. Note that

$$A=\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})'=(X-1_n\bar{x}')'(X-1_n\bar{x}')\geq 0 \quad \text{(from 10).}$$

$\det\left(\begin{pmatrix} a & b \\ c & d\end{pmatrix} - \lambda\begin{pmatrix} 1 & \\ & 1\end{pmatrix}\right) = \det\begin{pmatrix} a-\lambda & b \\ c & d-\lambda\end{pmatrix} = (a-\lambda)(c-\lambda)-b^2 = \lambda^2-(a+c)\lambda+ac-b^2 = \lambda^2-tr(A)\lambda+\det(A)$

Eigenvalue and eigenvector

$\det(A-\lambda I_p)$

$|A-\lambda I_p|$ is a polynomial of degree p in λ. The **eigenvalues** (or **latent roots**) of A, denoted by $\lambda_1,\ldots,\lambda_p$ are the roots of the equation $|A-\lambda I_p|$ = 0. The nonzero vector h_i such that $(A-\lambda_i I_p)h_i=0$ (or $Ah_i=\lambda_i h_i$) is called the **eigenvector** of A corresponding to λ_i. If the vector h_i has unit length (i.e., $h'h=1$), then it is called the **normalized (unit) eigenvector** of A.

$(A-\lambda_i I_p)h_i = 0$

There are some important properties of eigenvalues and eigenvectors.

12. If A is real symmetric matrix, then its eigenvalues are all real.

13. If $A\geq 0$, then all the eigenvalues of A are non-negative.

$\neq 0$

$(u_1, u_2, \ldots, u_p)$

column vector $u_i \sim \lambda_i$

unit eigenvectors

14. The eigenvalues of *A* and *A'* are the same.
15. If *A* and *B* are $p \times p$, *A* is nonsingular then eigenvalues of *AB* and *BA* are equal.
16. If $\lambda_1, \ldots, \lambda_p$ are the eigenvalues of a nonsingular matrix *A*, then $\lambda_1^{-1}, \ldots, \lambda_p^{-1}$ are the eigenvalues of A^{-1}.
17. If *A* is real symmetric matrix and λ_i and λ_j are two distinct eigenvalues of *A*, then the corresponding eigenvectors h_i and h_j are orthogonal.

    Proof. By definition, $Ah_i = \lambda_i h_i$ and $Ah_j = \lambda_j h_j$,

    => $h_j' A h_i = \lambda_i h_j' h_i$ and $h_i' A h_j = \lambda_j h_i' h_j$

    => $(\lambda_i - \lambda_j) h_j' h_i = 0$ => h_i and h_j are orthogonal.
18. Let *A* be a real symmetric matrix and *H* be an $p \times p$ matrix whose column is normalized eigenvectors $h_1, ..., h_p$ of *A*, i.e. $H = (h_1, \ldots h_p)$

    then $H' AH = D = diag(\lambda_1, \ldots, \lambda_p)$ (**Diagonalization** of A)

    or $A = HDH'$ (**Spectral decomposition** of A)

    Proof. Write $AH = (Ah_1, \ldots, Ah_p) = (\lambda_1 h_1, \ldots, \lambda_p h_p) = HD \Rightarrow H' AH = D$.
19. Trace of a matrix *A* is the sum of all its eigenvalues and the determinant of A is the product of all its eigenvalues, i.e., $tr(A) = \sum_{i=1}^{p} \lambda_i$ and $|A| = \prod_{i=1}^{p} \lambda_i$.

    Proof. The proof is left as an exercise.

Symmetric square root of a semi-positive definite matrix

Let $A \geq 0$ with eigenvalues $\lambda_1, \ldots, \lambda_p$ and the corresponding eigenvectors $h_1, \ldots h_p$.
Define the matrix $H = (h_1, \ldots h_p)$, $D = diag(\lambda_1, \ldots \lambda_p)$ and $D^{1/2} = diag(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_p})$.
Note that $A = HDH'$. The **symmetric square root** of *A* is given by $A^{1/2} = HD^{1/2}H'$.
To see this, $A^{1/2}A^{1/2} = HD^{1/2}H' HD^{1/2}H' = HDH' = A$.
What happen if *A* is not semi-positive definite?
The above matrix concepts can be illustrated using R's built-in matrix function. Let us continue with the Blue chips example.

R-code: t(S)=transpose of S, %*% = matrix multiplication

```
> options(digits=4)              # control display to 4 decimals
> det(solve(S))                  # det of inverse of S
[1] 5.706e-07
> 1/det(S)                       # 1/det(S)
[1] 5.706e-07

> eig<-eigen(S)                  # save eigenvalues and vector of S
> names(eig)                     # display items in eig
[1] "values"  "vectors"
> eval<-eig$values               # save eigenvalues
> eval                           # display eigenvalues
[1] 68.487 19.918 13.205 12.899  3.341  2.257

> H<-eig$vectors                 # save matrix of eigenvector
> H                              # display H
          [,1]     [,2]      [,3]      [,4]     [,5]     [,6]
[1,]  0.031107  0.91185  0.364402  0.016136  0.18218 -0.03637
[2,]  0.029477  0.19142 -0.018447  0.005915 -0.81898  0.53980
[3,] -0.010938  0.09104 -0.043829 -0.020125 -0.53213 -0.84030
[4,] -0.003038  0.34621 -0.911976 -0.188393  0.11223  0.01856
[5,]  0.998380 -0.03383 -0.007556 -0.036516  0.01255 -0.02334
[6,]  0.035663  0.05094 -0.182190  0.981058  0.01304 -0.01720

> round(t(H)%*%H,3)              # H' H=I, H is orthogonal, HH' =I as well
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    0    0    0    0    0
[2,]    0    1    0    0    0    0
[3,]    0    0    1    0    0    0
[4,]    0    0    0    1    0    0
[5,]    0    0    0    0    1    0
[6,]    0    0    0    0    0    1

> h1<-H[,1]                      # extract first column of H to h1
> eval[1]*h1                     # compute lambda1*h1
[1]  2.1304  2.0188 -0.7491 -0.2080 68.3765  2.4425

> t(S%*%h1)                      # compute (S*h1)'
        EY  CFTP   ln_MV      DY  BTME   DTE
[1,] 2.130 2.019 -0.7491 -0.2080 68.38 2.442

> round(t(H)%*%S%*%H,3)          # compute H' SH (should = D)
      [,1]  [,2]  [,3] [,4]  [,5]  [,6]
[1,] 68.49  0.00  0.00  0.0 0.000 0.000
[2,]  0.00 19.92  0.00  0.0 0.000 0.000
[3,]  0.00  0.00 13.21  0.0 0.000 0.000
[4,]  0.00  0.00  0.00 12.9 0.000 0.000
[5,]  0.00  0.00  0.00  0.0 3.341 0.000
[6,]  0.00  0.00  0.00  0.0 0.000 2.257

> D<-diag(eval)                  # form diagonal matrix D
> H%*%D%*%t(H)                   # compute HDH' (should = S)
        [,1]   [,2]     [,3]    [,4]    [,5]     [,6]
[1,] 18.4979 2.9090  1.16019  1.9204  1.4781  0.33795
[2,]  2.9090 3.6931  0.76630  1.2371  1.8228  0.32879
[3,]  1.1602 0.7663  2.74394  0.9721 -0.7734 -0.07413
[4,]  1.9204 1.2371  0.97207 13.8716 -0.2575  0.15815
[5,]  1.4781 1.8228 -0.77342 -0.2575 68.3082  1.96177
[6,]  0.3380 0.3288 -0.07413  0.1582  1.9618 12.99291
```

```
> rS<-H%*%sqrt(D)%*%t(H)          # H*sqrt(D)*H'
> rS
        [,1]   [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 4.26492 0.4603  0.17720  0.22591  0.11267  0.03736
[2,] 0.46026 1.8359  0.19270  0.19921  0.17666  0.05180
[3,] 0.17720 0.1927  1.62490  0.16722 -0.08301 -0.01540
[4,] 0.22591 0.1992  0.16722  3.70834 -0.02570  0.02000
[5,] 0.11267 0.1767 -0.08301 -0.02570  8.26013  0.16422
[6,] 0.03736 0.0518 -0.01540  0.02000  0.16422  3.60017

> rS%*%rS                          # rS*rS
        [,1]   [,2]     [,3]    [,4]    [,5]     [,6]
[1,] 18.4979 2.9090  1.16019  1.9204  1.4781  0.33795
[2,]  2.9090 3.6931  0.76630  1.2371  1.8228  0.32879
[3,]  1.1602 0.7663  2.74394  0.9721 -0.7734 -0.07413
[4,]  1.9204 1.2371  0.97207 13.8716 -0.2575  0.15815
[5,]  1.4781 1.8228 -0.77342 -0.2575 68.3082  1.96177
[6,]  0.3380 0.3288 -0.07413  0.1582  1.9618 12.99291
```

1.2 The Multivariate Normal Distribution

First let us review the univariate normal distribution. If the random variable *X* follows a normal distribution with mean μ and variance σ^2, denoted by $X \sim N(\mu,\sigma^2)$, then the density function of *X* is

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{-(x-\mu)^2}{2\sigma^2} \right\} \tag{1.1}$$

Now suppose $X = (X_1, \ldots, X_p)'$ is a $p \times 1$ random vector. If the random vector *X* follows a p-variate Normal distribution with mean vector μ and covariance matrix Σ, denoted by $X \sim N_p(\mu, \Sigma)$. The density of *X* is

$$f(X) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp\left\{ -\frac{1}{2}(X-\mu)'\Sigma^{-1}(X-\mu) \right\} \tag{1.2}$$

In particular when p=2, $X = (X_1, X_2)'$ has a **bivariate normal** distribution with

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \quad \Sigma^{-1} = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix}$$

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left\{ \frac{-1}{2(1-\rho^2)} \left[\left(\frac{x_1-\mu_1}{\sigma_1} \right)^2 - 2\rho \left(\frac{x_1-\mu_1}{\sigma_1} \right)\left(\frac{x_2-\mu_2}{\sigma_2} \right) + \left(\frac{x_2-\mu_2}{\sigma_2} \right)^2 \right] \right\}$$

Here ρ is the correlation between X_1 and X_2.

The followings are important properties of Multivariate Normal distribution.

If $X \sim N_p(\mu, \Sigma)$ and B is $q \times p$, b is $q \times 1$, then $Y = BX + b \sim N_q(B\mu + b, B\Sigma B')$

There are some important results worth mentioning:

1. Let $X_1, ..., X_n$ iid $N_p(\mu, \Sigma)$. Then $\bar{X} \sim N_p[\mu, (1/n)\Sigma]$.

2. Let $X_1, ..., X_n$ iid $N_p(\mu, \Sigma)$. Then $n(\bar{X} - \mu)' \Sigma^{-1}(\bar{X} - \mu) \sim \chi^2_p$.

Proof. Let $Y = \sqrt{n}\, \Sigma^{-1/2}(\bar{X} - \mu)$. From (1), $Y \sim N_p(0, I_p)$.

Note that $n(\bar{X} - \mu)' \Sigma^{-1}(\bar{X} - \mu) = Y'Y = \sum_{i=1}^{p} y_i^2 \sim \chi^2_p$.

3. (The Central Limit Theorem)

Let $X_1, ... X_n$ be iid from any population with finite mean μ and finite covariance matrix Σ. Then $\sqrt{n}(\bar{X} - \mu) \to N_p(0, \Sigma)$ for large *n* (relative to *p*).

4. Let $X_1, ... X_n$ be iid from any population with finite mean μ and finite covariance matrix Σ. Then $n(\bar{X} - \mu)' S^{-1}(\bar{X} - \mu) \sim \chi^2_p$ approx. for large *n* (relative to *p*).

5. Let $X_1, ... X_n$ be iid from $N_p(\mu, \Sigma)$. Then the **squared generalized distance** (also known as the **Mahalanobis distance**) $D_i^2 = (X_i - \bar{X})' S^{-1}(X_i - \bar{X}) \sim \chi^2_p$ approx. for large *n* (relative to *p*). Roughly speaking, D_i^2 measures the distance of X_i to $\bar{X}$ in terms of standardized unit.

1.3 Sample estimate of parameters

If $x_1, ..., x_n$ are **independently and identically distributed (iid)** as Normal with mean μ and variance σ^2, denoted by $N(\mu, \sigma^2)$, then the sample estimate for μ is the sample mean $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and **unbiased** estimate of σ^2 is $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$. The maximum likelihood estimate **(MLE)** of σ^2 is $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$. Note that when n is large, the unbiased estimate and MLE of σ^2 is numerically indifferent. Now we can extent this to multivariate case.

Let $X = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{np} \end{bmatrix} = \begin{bmatrix} x_1' \\ \vdots \\ x_n' \end{bmatrix}$ where $x_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{ip} \end{bmatrix}$ is **iid** as $N_p(\mu, \Sigma)$ for *i=1,…,n*.

The sample mean vector is

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \sim N_p(\mu, \frac{1}{n}\Sigma) . \tag{1.3}$$

Define the **Sum of Squares and Cross product (SSCP)** matrix

$$A = \sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})' . \tag{1.4}$$

Then the **Unbiased estimator** of Σ is *S = A/(n-1)* and the **Maximum Likelihood estimator (MLE)** of μ and Σ is $\bar{x}$ and *A/n* respectively. The sample correlation matrix of *X* is $R = D^{-1/2} S D^{-1/2}$ where $S = (s_{ij})$, $D = diag(s_{11}, \ldots, s_{pp})$ is a diagonal matrix and $D^{-1/2} = diag(1/\sqrt{s_{11}}, \ldots, 1/\sqrt{s_{pp}})$.

1.3.1 Important properties of sample mean vector and covariance matrix

In the univariate case, it is well-known that the sample mean and the sample variance are **unbiased** estimator of population mean μ and population variance σ^2. These results also hold in the multivariate setting.

Let $X_1, \ldots, X_n$ be random vectors which are independently and identically distributed (iid) with mean vector $E(X_i) = \mu$ and covariance matrix $Var(X_i) = \Sigma$.

1. $E(\bar{X}) = \mu$.

Proof. $E(\bar{X}) = \frac{1}{n}\sum_{i=1}^{n} E(X_i) = \mu$.

2. $E(S) = \Sigma$.

Proof. Recall that the SSCP matrix $A = \sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})' = \sum_{i=1}^{n} X_i X_i' - n\bar{X}\bar{X}'$.

$E(A) = \sum_{i=1}^{n} E(X_i X_i') - nE(\bar{X}\bar{X}')$.

Note that $E(X_i X_i')$ is a $p \times p$ matrix whose (*j,k*) element is $E(x_{ij}x_{ik}) = \sigma_{jk} + \mu_j \mu_k$.

Therefore $E(X_i X_i') = \Sigma + \mu\mu'$. Similarly, the (*j,k*) element of $E(\bar{X}\bar{X}')$ is

$E(\bar{x}_j \bar{x}_k) = (1/n)\sigma_{jk} + \mu_j \mu_k$ so that $nE(\bar{X}\bar{X}') = \Sigma + n\mu\mu'$.

Hence, $E(A) = n\Sigma + n\mu\mu' - \Sigma - n\mu\mu' = (n-1)\Sigma$, which implies $E(S) = \Sigma$.

Alternative proof:

$$A = \sum_{i=1}^{n} (X_i - \overline{X})(X_i - \overline{X})'$$
$$= \sum_{i=1}^{n} (X_i - \mu + \mu - \overline{X})(X_i - \mu + \mu - \overline{X})'$$
$$= \sum_{i=1}^{n} \left[(X_i - \mu)(X_i - \mu)' + (\overline{X} - \mu)(\overline{X} - \mu)' - (X_i - \mu)(\overline{X} - \mu)' - (\overline{X} - \mu)(X_i - \mu)' \right]$$

$$\mathrm{E}(A) = \sum_{i=1}^{n} \left\{ \mathrm{E}[(X_i - \mu)(X_i - \mu)'] + \mathrm{E}[(\overline{X} - \mu)(\overline{X} - \mu)'] - \mathrm{E}[(X_i - \mu)(\overline{X} - \mu)'] - \mathrm{E}[(\overline{X} - \mu)(X_i - \mu)'] \right\}$$

$$\mathrm{E}(A) = \sum_{i=1}^{n} \begin{bmatrix} \mathrm{E}[(X_i - \mu)(X_i - \mu)'] + \frac{1}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} \mathrm{E}[(X_j - \mu)(X_k - \mu)'] \\ -\frac{1}{n} \sum_{j=1}^{n} \mathrm{E}[(X_i - \mu)(X_j - \mu)'] - \frac{1}{n} \sum_{j=1}^{n} \mathrm{E}[(X_j - \mu)(X_i - \mu)'] \end{bmatrix}$$
$$= n\mathrm{E}[(X_i - \mu)(X_i - \mu)'] + \frac{n}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} \mathrm{E}[(X_j - \mu)(X_k - \mu)'] - \frac{2}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{E}[(X_i - \mu)(X_j - \mu)']$$
$$= n\mathrm{E}[(X_i - \mu)(X_i - \mu)'] + \frac{1}{n} \sum_{j=1}^{n} \mathrm{E}[(X_j - \mu)(X_j - \mu)'] + \frac{1}{n} \sum_{\substack{j=1, \\ j \neq k}}^{n} \sum_{k=1}^{n} \mathrm{E}[(X_j - \mu)(X_k - \mu)']$$
$$-\frac{2}{n} \sum_{i=1}^{n} \mathrm{E}[(X_i - \mu)(X_i - \mu)'] - \frac{2}{n} \sum_{\substack{j=1, \\ j \neq i}}^{n} \sum_{i=1}^{n} \mathrm{E}[(X_j - \mu)(X_i - \mu)']$$

Since $\mu \equiv E(X_i)$, while X_i and X_j are independent for $i \neq j$, we get:

$$\sum_{\substack{j=1, \\ j \neq i}}^{n} \sum_{i=1}^{n} \mathrm{E}[(X_j - \mu)(X_i - \mu)'] = \sum_{\substack{j=1, \\ j \neq i}}^{n} \sum_{i=1}^{n} \mathrm{E}[(X_j - \mu)\mathrm{E}(X_i - \mu)'] = 0$$

Denote $\Sigma \equiv \mathrm{E}[(X_i - \mu)(X_i - \mu)']$,
We get: $\mathrm{E}(A) = (n-1)\Sigma$

It follows that the expectation value of sample variance matrix equals the covariance matrix:

$$\mathrm{E}(S) = \frac{1}{n-1} \mathrm{E}(A) = \Sigma$$

3. $Var(\overline{X}) = (1/n)\,\Sigma$.

Proof. We know that: $\mu = \mathrm{E}(\overline{X})$

$$Var(\bar{X}) = E[(\bar{X} - E(\bar{X}))(\bar{X} - E(\bar{X}))']$$
$$= E[(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu)(\frac{1}{n}\sum_{i=1}^{n} X_i - \mu)']$$
$$= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} E[(X_i - \mu)(X_j - \mu)']$$
$$= \frac{1}{n^2}\sum_{i=1}^{n} E[(X_i - \mu)(X_i - \mu)'] + \frac{1}{n^2}\sum_{\substack{i=1 \\ i\neq j}}^{n}\sum_{j=1}^{n} E[(X_i - \mu)(X_j - \mu)']$$
$$= \frac{1}{n^2}\sum_{i=1}^{n} E[(X_i - \mu)(X_i - \mu)'] + \frac{1}{n^2}\sum_{\substack{i=1 \\ i\neq j}}^{n}\sum_{j=1}^{n} E(X_i - \mu)E(X_j - \mu)'$$
$$= \frac{1}{n}\Sigma$$

1.3.2 Important results of some univariate distributions

The following distributions are important and serve as building blocks for constructing and deriving null distributions of many test statistics. We review the definitions and related important results here.

1. Chi-square distribution

Definition: If $Z_1, \ldots, Z_\nu$ are iid $N(0,1)$, then $X = Z_1^2 + \cdots + Z_\nu^2 \sim \chi_\nu^2$.

- If $Y_1, \ldots, Y_n$ are iid $N(\mu, \sigma^2)$, then $\dfrac{(n-1)S^2}{\sigma^2} = \sum_{i=1}^{n}\dfrac{(Y_i - \bar{Y})^2}{\sigma^2} \sim \chi_{n-1}^2$.

  Proof: Define $\mathrm{E}(Y_i) = \mu$

$$\sum_{i=1}^{n}\frac{(Y_i - \bar{Y})^2}{\sigma^2} = \sum_{i=1}^{n}\frac{(Y_i - \mu + \mu - \bar{Y})^2}{\sigma^2}$$
$$= \frac{1}{\sigma^2}[\sum_{i=1}^{n}(Y_i - \mu)^2 + 2(\mu - \bar{Y})\sum_{i=1}^{n}(Y_i - \mu) + n(\bar{Y} - \mu)^2]$$
$$= \frac{1}{\sigma^2}[\sum_{i=1}^{n}(Y_i - \mu)^2 - 2n(\bar{Y} - \mu)^2 + n(\bar{Y} - \mu)^2]$$
$$= \sum_{i=1}^{n}(\frac{Y_i - \mu}{\sigma})^2 - (\frac{\bar{Y} - \mu}{\sigma/\sqrt{n}})^2$$

  Terms in the last line are square of standardized random variables distributed in $N(0,1)$. Rearranging the last line,

$\sum_{i=1}^{n}(\dfrac{Y_i - \mu}{\sigma})^2 = \sum_{i=1}^{n}(\dfrac{Y_i - \bar{Y}}{\sigma})^2 + (\dfrac{\bar{Y} - \mu}{\sigma/\sqrt{n}})^2$, note that $\bar{Y}$ is independent of all $Y_i - \bar{Y}$ as $Cov(\bar{Y}, Y_i - \bar{Y}) = 0$ for all i, since they are multivariate normal, $\bar{Y}$ is independent of $Y_i - \bar{Y}$.

Since $\sum_{i=1}^{n}(\frac{Y_i-\mu}{\sigma})^2 \sim \chi_n^2$ and $(\frac{\bar{Y}-\mu}{\sigma/\sqrt{n}})^2 \sim \chi_1^2$, we get $\sum_{i=1}^{n}(\frac{Y_i-\bar{Y}}{\sigma})^2 \sim \chi_{n-1}^2$

2. Student's t distribution

Definition: If $Z \sim N(0,1)$ and independent of $X \sim \chi_v^2$, then $T = \frac{Z}{\sqrt{X/v}} \sim t_v$.

- If $Y_1,\ldots,Y_n$ are iid $N(\mu,\sigma^2)$, then $T = \frac{\sqrt{n}(\bar{Y}-\mu)}{S} \sim t_{n-1}$.

Proof: $\bar{Y} \sim N(\mu,\frac{\sigma^2}{n}) \Rightarrow Z = \frac{\sqrt{n}(\bar{Y}-\mu)}{\sigma} \sim N(0,1)$ and $X = \frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2$.

$\bar{Y}$ is independent of S^2.

$$\therefore \quad T = \frac{Z}{\sqrt{X/(n-1)}} = \frac{\sqrt{n}(\bar{Y}-\mu)}{S} \sim t_{n-1}.$$

3. Fisher's F distribution

Definition: If $X_1 \sim \chi_u^2$ and independent of $X_2 \sim \chi_v^2$, then $F = \frac{X_1/u}{X_2/v} \sim F_{u,v}$.

- If $X_1,\ldots,X_m$ are iid $N(\mu_1,\sigma_1^2)$ and independent of $Y_1,\ldots,Y_n$ iid $N(\mu_2,\sigma_2^2)$,

  then $F = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} = \frac{\sigma_2^2 S_1^2}{\sigma_1^2 S_2^2} \sim F_{m-1,n-1}$

Proof: Since $\frac{(m-1)S_1^2}{\sigma_1^2} \sim \chi_{m-1}^2$ and independent of $\frac{(n-1)S_2^2}{\sigma_2^2} \sim \chi_{n-1}^2$,

$$\therefore \quad F = \frac{S_1^2/\sigma_1^2}{S_2^2/\sigma_2^2} = \frac{\sigma_2^2 S_1^2}{\sigma_1^2 S_2^2} \sim F_{m-1,n-1}.$$

- Particularly, $T \sim t_v$, then $T^2 \sim F_{1,v}$.

4. Sampling distribution of $\bar{X}$ and S^2

- If $X_1,\ldots,X_n$ are iid $N(\mu,\sigma^2)$, then $\bar{X} \sim N(\mu,\sigma^2/n)$ and is independent of $(n-1)S^2/\sigma^2 \sim \chi^2(n-1)$.

Reference: Chapter 2 and 3 of *Applied Multivariate Statistical Analysis*, 6*th* *ed.*, R.A. Johnson and D.W. Wichern.

1.4 The Normal assumption

The basic assumption of **Black-Scholes-Merton** model is that the percentage change in the stock price in a short period of time of length δt is normally distributed. More precisely,

$$\delta S / S \sim N(\mu\,\delta t, \sigma^2\,\delta t),$$

where δS is the change in the stock price S in $(t, t+\delta t)$, $\mu \delta t$ is the mean of percentage change and $\sigma\sqrt{\delta t}$ is the standard deviation of this percentage change. μ is called the drift rate and σ^2 is called the variance rate. Usually the stock price is observed over a fixed time interval (e.g. the closing price is observed daily, so that $\delta t = 1/252$). Now suppose we have *n* observed daily stock price $S_1, \cdots, S_n$. Define

$$u_i = (S_i - S_{i-1})/S_{i-1} \sim N[\mu\tau, \sigma^2\tau] \quad i = 1, \cdots, n \tag{1.5}$$

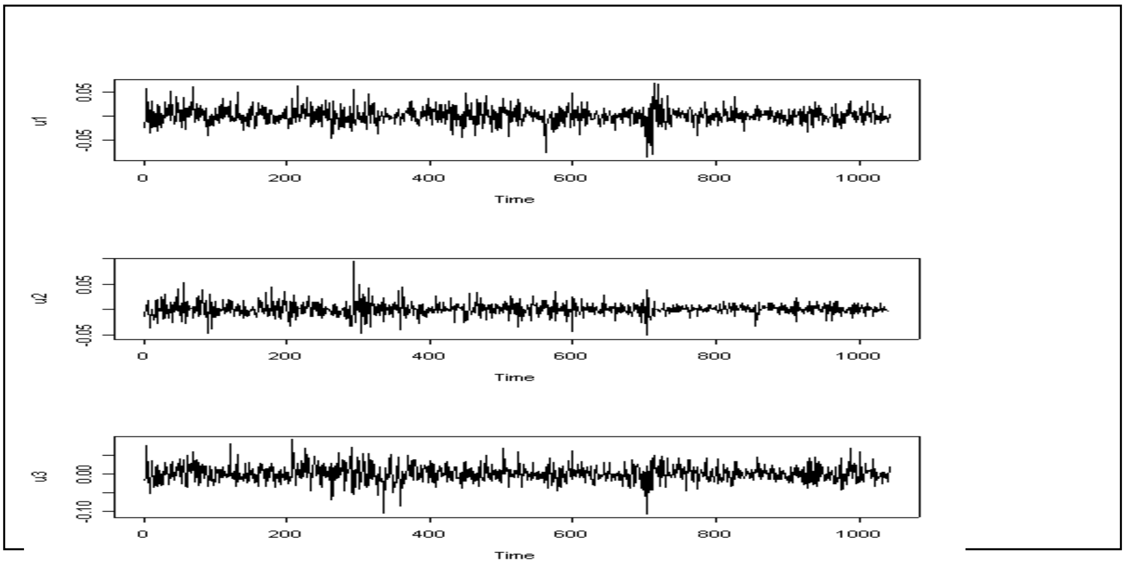
where $\tau = 1/252$. To see if this assumption is correct, we can use some graphical methods to test for normality. Let us illustrate this by the following example.

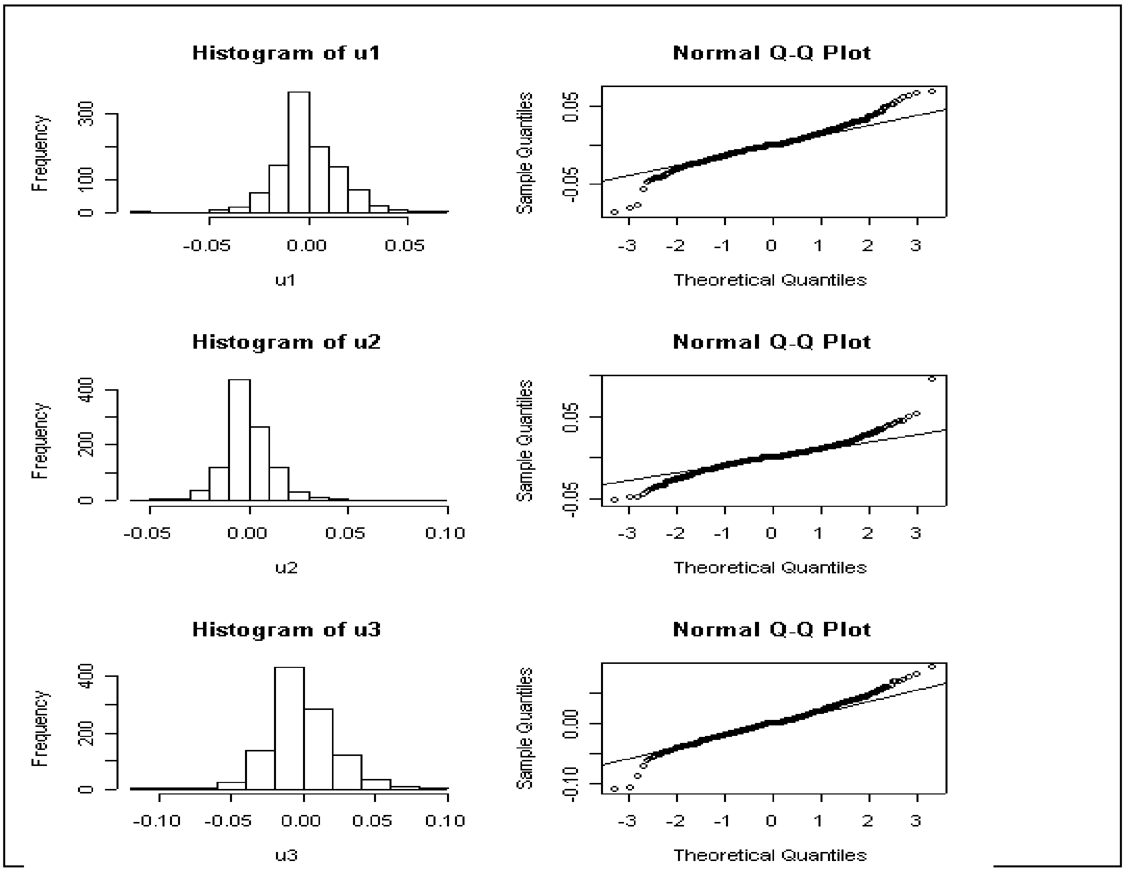
The file "stock.csv" contains adjusted daily closing price for the stock HSBC (0005), CLP (0002) and Cheung Kong (0001) from 1/1/1999 to 31/12/2002. Let us first read in these stock prices and compute the u_i for each stock respectively according to (1.5); also normalizing them by subtracting with $\hat{\mu}\tau = \overline{\mu}$. Then we produce a time series plot, histogram, together with the QQ-normal plot for u_i.

```
> d<-read.csv("stock.csv")# read in data file
> names(d)                    # display names in d
[1] "HSBC" "CLP"  "CK"
> t1<-as.ts(d$HSBC)           # save as time series
> t2<-as.ts(d$CLP)
> t3<-as.ts(d$CK)
> u1<-(lag(t1)-t1)/t1         # compute daily percentage return
> u2<-(lag(t2)-t2)/t2
> u3<-(lag(t3)-t3)/t3

> par(mfrow=c(3,1))           # define multi-frame for ploting
> plot(u1)                    # plot u
> plot(u2)
> plot(u3)

> par(mfrow=c(3,2))
> hist(u1)            # histogram
> qqnorm(u1)          # qq-normal plot
> qqline(u1)          # add a line for reference
> hist(u2)            # if the dist is normal, the plot should
> qqnorm(u2)          # close to this line
> qqline(u2)
> hist(u3)
> qqnorm(u3)
> qqline(u3)
```





These time series plots show that u_i is fluctuating around zero. The volatility is small in certain periods and large in other periods. Obviously the volatility varies with time. We shall discuss how to model this changing volatility in the Chapter 2, but first let us focus on the plausibility of the normality assumption of u_i.

According to (1.5), u_i should be normally distributed. The most commonly used graphical method is histogram. However, the histogram only gives us the general picture of the distribution. A more sophisticated graphical method is the QQ-normal plot. Let us ordered the u_i in ascending order and denoted by $u_{(i)}$, i.e., $u_{(1)} \le u_{(2)} \le \cdots \le u_{(n)}$. The QQ-normal plot is the plot of these ordered $u_{(i)}$ against $q_{(i)}$, the *i*th quantile of the standard normal distribution, i.e., *Pr{Z<q(i)}=(i-0.5)/n*. Conceptually, the right hand side should be *i/n*. The *(i-0.5)/n* is used to avoid difficulty when *i=n*. If u_i is normally distributed, this plot should be close to a straight line. The sample plots illustrate that the proportional changes are usually heavily tailed for large changes, while light-tailed or positively skewed for small changes.

Next, we look at the histogram and the QQ-normal plot of u_i. These plots show that the distribution of u_i have a fatter tail than a normal distribution. This is a reason for the volatility smiles in option pricing. More seriously, the VaR will be under-estimated if the normal model approach is used.

The QQ-normal plot is a graphical method for testing normality. However there are other statistical tests for testing normality as well. **Kolmogorov-Smirnov (KS)** test can be used to test whether a random sample is coming from a specific distribution. The test statistic is $D_n = \sup_x |F_n(x) - F(x)|$, where $F_n(x)$ and $F(x)$ are the empirical and theoretical distribution function. Note that the distribution of $\sqrt{n}D_n$ converges to the absolute maximum of a standardized Brownian bridge.

```
> ks.test(u1,pnorm)                 # KS-test for normality on u1, u2 and u3
D = 0.4773, p-value < 2.2e-16
> ks.test(u2,pnorm)
D = 0.4795, p-value < 2.2e-16
> ks.test(u3,pnorm)
D = 0.4721, p-value < 2.2e-16
```

Another commonly used normality test is Jarque-Bera (JB) test. It based on the fact that the skewness=0 and kurtosis=3 for a normal distribution. The test statistic is

$JB = n[\hat{\varsigma}_1^2/6 + (\hat{\varsigma}_2 - 3)^2/24] \sim \chi_2^2$ asymptotically, where $\hat{\varsigma}_1, \hat{\varsigma}_2$ are the sample skewness ($Skew(X) = E[(X-\mu)/\sigma]^3$) and kurtosis ($Kurt(X) = E[((X-\mu)/\sigma)]^4$. We write the following function `JB.test()` to perform this test.

```
JB.test<-function(u) {                      # function for JB-test
  n<-length(u)                              # sample size
  s<-sd(u)                                  # compute sd
  sk<-sum(u^3)/(n*s^3)                      # compute skewness
  ku<-sum(u^4)/(n*s^4)-3                    # excess kurtosis
  JB<-n*(sk^2/6+ku^2/24)                    # JB test stat
  p<-1-pchisq(JB,2)                         # p-value
  cat("JB-stat:",JB," p-value:",p,"\n")     # output
}
> JB.test(u1)
JB-stat: 317.7214  p-value: 0
> JB.test(u2)
JB-stat: 981.1252  p-value: 0
> JB.test(u3)
JB-stat: 136.4649  p-value: 0
```

The p-value for u1, u2 and u3 from KS test or JB test are small, the normality assumption for u1, u2 and u3 is not valid which is consistent with conclusion from the QQ-normal plots.

1.5 Student's t(v)-distribution

To deal with the non-normal fat tail issue, we use a scaled student's *t* distribution with ν degree of freedom and scale parameter σ, denoted by $t(\nu)$, instead of the normal distribution to model u_i. The density of $t(\nu)$ distribution is given by:

$$f(T,\nu) = \frac{\Gamma[(\nu+1)/2]}{\Gamma(\nu/2)\sqrt{\nu\pi\sigma^2}}\left[1+\frac{T^2}{\nu\sigma^2}\right]^{-(\nu+1)/2} \quad \text{for} \quad \nu > 2.$$

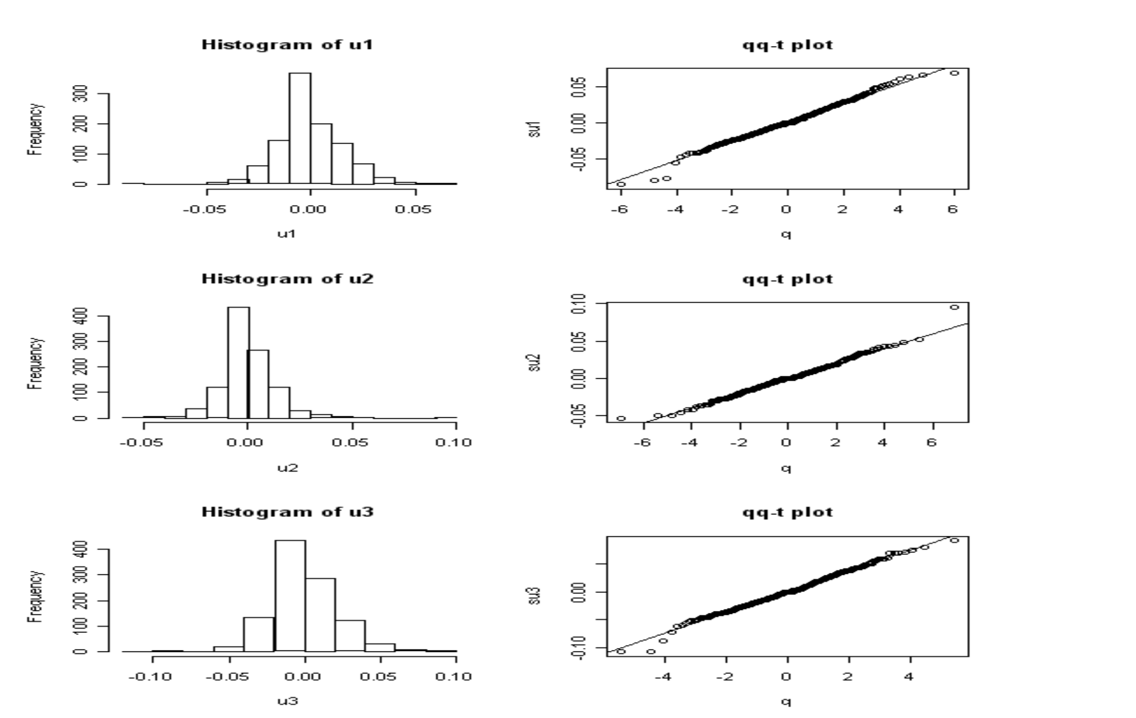
where $\Gamma(\cdot)$ is the gamma function. The first four moments of this distribution are:
$\mu = E(T) = 0$, $\sigma^2 = E(T^2) = V(T) = \sigma^2\nu/(\nu-2)$, Skewness: $\varsigma_1 = E(T^3)/\sigma^3 = 0$, excess kurtosis: $\varsigma_2 = E(T^4)/\sigma^4 - 3 = 6/(\nu-4)$.

The *t* distribution will have fatter tail than normal when ν is small and approach to normal when $\nu \to \infty$ and hence it is useful in modeling data with extreme values. For common securities in usual markets, ν ranges from 3 to 5. We can produce a similar *QQ-t* plot to see if the fitting is better. First, we need to estimate the degree of freedom *v* in the $t(v)$ distribution. A simple method is let the sample excess kurtosis $\hat{\varsigma}_2 = 6/(\nu-4) \Rightarrow \nu = 6/\hat{\varsigma}_2 + 4$. $\hat{\varsigma}_2$ can be computed from u_i and *v* is rounded to the nearest integer. The *p*-th quantile of the $t(v)$ distribution is obtained by $\Pr\{T =< t_p(v)\} = p$ and is denoted by $q_p = t_p(\nu)$.

Again we can the following QQt.plot() function to produce the QQ-t plot:

```
QQt.plot<-function(u) {        # function for QQ-t plot
  su<-sort(u)                  # sort u
  n<-length(u)                 # sample size
  s<-sd(u)                     # sd
  ku<-sum(u^4)/(n*s^4)-3       # excess kurtosis
  v<-round(6/ku+4)             # estimate df, round to the nearest integer
  i<-((1:n)-0.5)/n             # create a vector of percentile
  q<-qt(i,v)                   # percentile point from t(v)

  hist(u)                      # histogram of u
  plot(q,su,main="qq-t plot")  # plot(q,su)
  abline(lsfit(q,su))          # add reference line
  v                            # output degree of freedom
}
v1<-QQt.plot(u1)               # QQ-t plot for u1, u2, u3
v2<-QQt.plot(u2)               # and save degree of freedom to v1, v2, v3
v3<-QQt.plot(u3)
```



Note that *t(v)* distribution fits the tails better than using the normal distribution. Again we can use KS-test to test whether u1, u2 and u3 follow a t distribution as follows:

```
> ks.test(u1,pt,v1)                      # ks-t test for u1, u2 and u3
D = 0.478, p-value < 2.2e-16
> ks.test(u2,pt,v2)
D = 0.4803, p-value < 2.2e-16
> ks.test(u3,pt,v3)
D = 0.4729, p-value < 2.2e-16
```

Again all the p-values are small, u1, u2 and u3 do not completely follow a t-distribution. Significant deviation from the *t*-distribution occurs at the extreme negative values of q. Such pronounced fat-tail, compared to the t-distribution, show extreme return values are more likely at market downturns, and they are all positively skewed, while t-distribution does not.

1.6 Testing Multivariate Normality

The QQ-normal plot in the previous section is only for testing univariate normality. We may want to test whether the vector $u=(u_1,u_2,u_3)'$ jointly follows a multivariate normal distribution, i.e.,

$$u_i = \begin{pmatrix} u_{i1} \\ \vdots \\ u_{ip} \end{pmatrix} = \begin{pmatrix} \delta S_{i1}/S_{i1} \\ \vdots \\ \delta S_{ip}/S_{ip} \end{pmatrix} \sim N_p[\begin{pmatrix} \mu_1 \\ \vdots \\ \mu_p \end{pmatrix}(\delta\, t),\ \Sigma\,(\delta\, t)] \quad \text{for } i=1,\ldots,n \tag{1.6}$$

We can extend the idea of QQ-normal plot to test multivariate normality. First let us define the squared generalized distance (the **Mahalanobis** distance):

$$d_i^2 = (u_i-\bar{u})' S^{-1}(u_i-\bar{u}) \quad \text{for} \quad i=1,\ldots,n \tag{1.7}$$

where $\bar{u}$ and S are the sample unbiased estimate of $\mu(\delta t)$ and $\Sigma(\delta t)$ given in Section 1.3. Note that d_i^2 is a number representing the distance from the observation u_i to the central tendancy and is called the squared generalized distance of u_i. According to the distribution theory, if u_i has a multivariate normal distribution, then for large n, d_i^2 will follow a Chi-square distribution with p degrees of freedom. Therefore, we can produce a QQ-chisquare plot as follow:

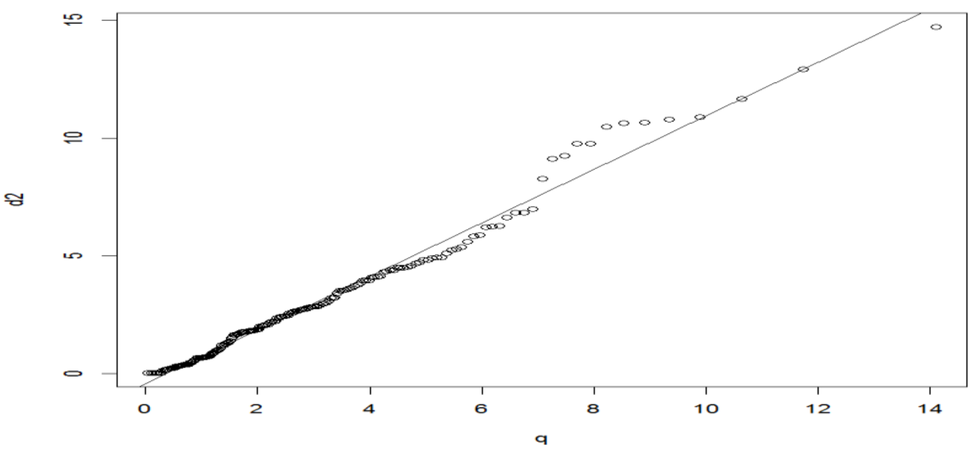
1. Order the d_i^2 in ascending order, denoted by $d_{(i)}^2$.
2. Compute $q_p(i)$, the ith quantile of a Chi-square distribution with p degrees of freedom, i.e., $\Pr\{\chi_p^2 < q_p(i)\} = (i-0.5)/n$.
3. Plot $d_{(i)}^2$ against $q_p(i)$. If the points close to a straight line, the original u_i's follow a multivariate normal distribution.

Let us illustrate this QQ-chisquare plot by testing whether $u=(u_1,u_2,u_3)'$ has a tri-variate normal distribution. First, we compute the sample mean vector, sample variance-covariance matrix of the daily return using the most recent 180 days. (By the way, why don't we use all the data instead of just using the most recent 180 days?)

```
> u<-cbind(u1,u2,u3)           # combine into matrix u
> n2<-nrow(u)                  # no of row in u
> n1<-n2-180+1                 # 180-th obs before n2
> u180<-u[n1:n2,]              # save the most recent 180 days to u180
> (m<-apply(u180,2,mean))      # compute column mean of u180
           u1            u2            u3
-0.0003229707  0.0002479614 -0.0019383056
> (s<-var(u180))
              u1            u2           u3
u1  1.408494e-04 -7.129762e-07 1.303655e-04
u2 -7.129762e-07  6.318444e-05 1.872616e-05
u3  1.303655e-04  1.872616e-05 3.507328e-04
```

Then we compute the inverse of S, the squared generalized distance and the quantiles of Chi-square distribution. Note that R does not have a built-in function for finding inverse of a matrix but we can use the built-in function solve() to do it.

```
> sinv<-solve(s)                              # compute inv(s)
> m<-matrix(m,nr=180,nc=3,byrow=T)          # change m into 180x3 matrix
> d2<-diag((u180-m)%*%sinv%*%t(u180-m))# compute squared gen. dist.
> d2<-sort(d2)                                # sort d2 in ascending order
> i<-((1:180)-0.5)/180                        # create vector of percentile
> q<-qchisq(i,3)                              # compute quantiles of chisq(3)
> qqplot(q,d2)                                # QQ-chisquare plot
> abline(lsfit(q,d2))                         # add the reference line
```



Note that u180 is an 180×3 matrix and hence (u180-m)%*%sinv%*%t(u180-m) is an 180×180 matrix. The diagonal of this matrix is a vector of length 180 equal to the d_i^2 in (1.7). This is not a straight forward but a convenience way to compute d_i^2. From the plot, the distribution of u180 is not significantly different from a multivariate normal distribution. We can use the KS-test for test whether $d_i^2 \sim \chi^2(p)$.

```
> ks.test(d2,pchisq,3)# ks Chisquare test for d2
D = 0.0994, p-value = 0.05702
```

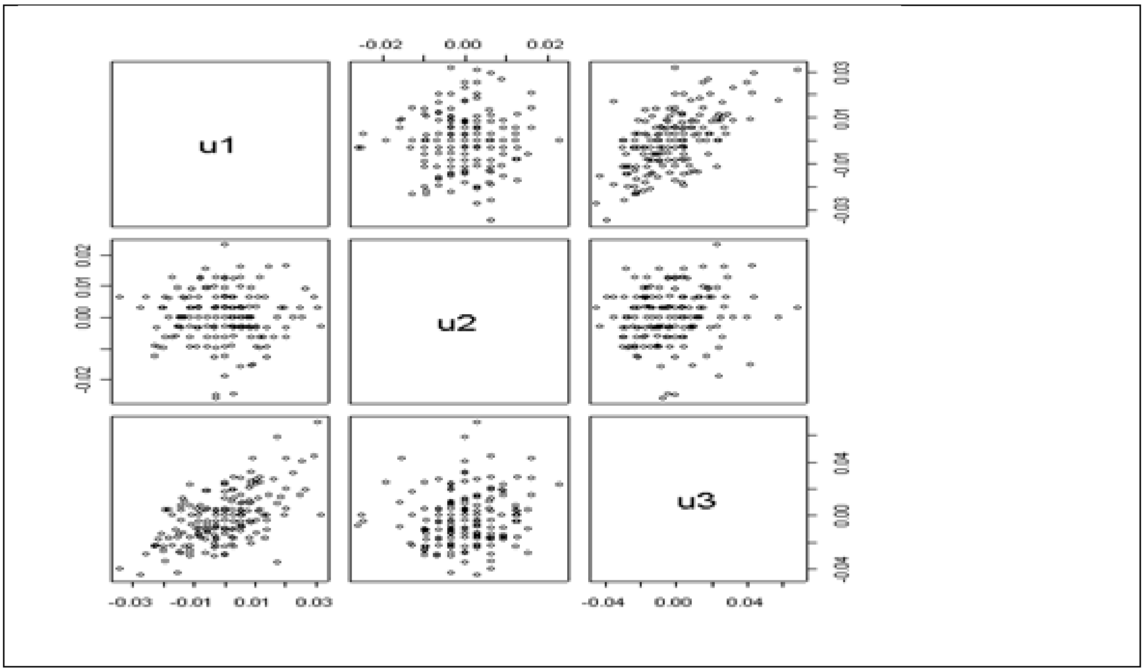
Now, the p-value is large, we may accept Ho that d2 follows a Chi-square distribution, or $u = (u_1, u_2, u_3)'$ have a tri-variate normal distribution.

1.7 Sample Correlation Matrix

When we consider more than one variable, correlation between variables are important parameters. Let us compute the correlation matrix of u180 by

```
> cor(u180)
             u1           u2         u3
u1  1.000000000  -0.007557749 0.5865390
u2  -0.007557749 1.000000000  0.1257927
u3  0.586538960  0.125792732  1.0000000
```

We can produce the scatter plot matrix of these daily returns by using `pairs(u180)`. This can be considered as a graphical display of the correlation matrix.



From the output, u1 and u2 are mostly uncorrelated; u1 and u3 are positively correlated while u2 and u3 have slight positive correlation. The sample correlation matrix can be computed from the sample covariance matrix S by

$$r_{ij} = s_{ij} / \sqrt{s_{ii} s_{jj}} \quad i, j = 1, \dots p$$

1.8 Generating Multivariate Normal Random Numbers

All statistical packages, including R and EXCEL, should have built-in function for generating standard normal random numbers. These random numbers can be transformed into a normal distribution with mean μ and standard deviation σ. For example, if $z \sim N(0,1)$ then $X = \mu + \sigma z \sim N(\mu, \sigma^2)$. This is important in simulating random paths of the stock prices, valuing options and derivatives and calculating VaR using simulation approach, etc. However, simulating multivariate normal random numbers is not that straight forward. Suppose we need to generate $p \times 1$ random vectors from a *p*-variate normal distribution with mean vector μ and covariance matrix Σ.

1. First, we generate p independent standard normal random numbers,

   $z = (z_1, \dots, z_p)'$ where z_i *iid N(0,1).*

2. Transform z to $x = \mu + C' z$ where C is an $p \times p$ matrix such that $C'C = \Sigma$.

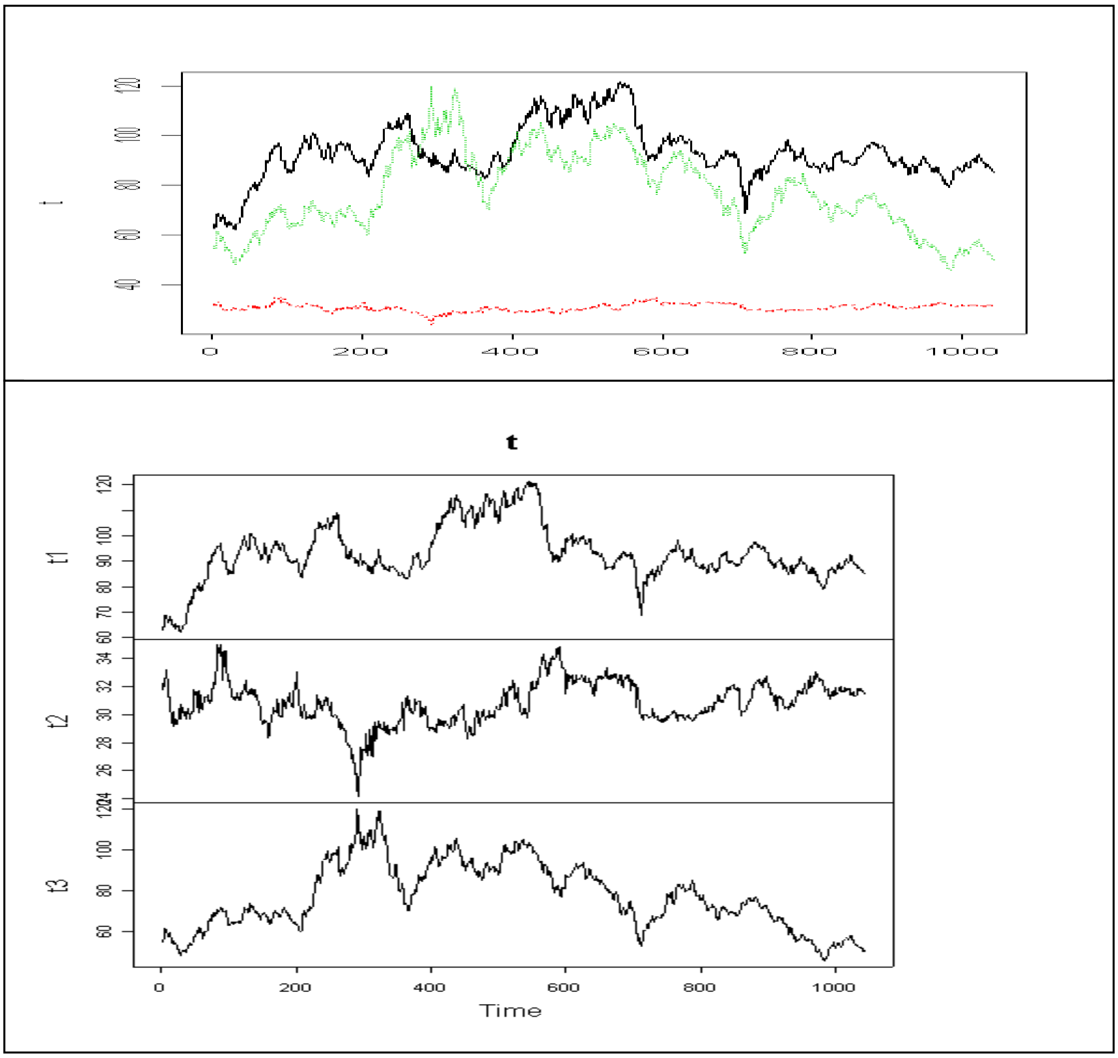
Then x follows a p-variate normal distribution with mean μ and covariance matrix Σ.

The matrix C in step 2 is called the **Cholesky decomposition** of Σ. R has a built-in function for performing Cholesky decomposition. We will illustrate this by simulating random sample path of the stock prices of HSBC, CLP and CK according to the Black-Scholes model. First let us look at the plot of these stock prices.

```
> t<-cbind(t1,t2,t3)       # combine the three series
> matplot(t,type="l")      # plot the series in one plot
```

Matplot plots all the series in one plot, or we can plot these series using separate axes.

```
> plot(t)
```



Recall that the assumption for the Black-Scholes model for the stock price is

$$\frac{\delta S}{S} \sim N(\mu \delta t, \sigma^2 \delta t) \tag{1.8}$$

where δS is the change of stock price in a short time interval $(t, t+\delta t)$, μ is the drift rate and σ is the annual volatility. If δS is the daily change in stock price, δt=1/252=0.004. According to (1.8),

$$\frac{S_{t+1}-S_t}{S_t} = \mu\, \delta t + \sigma \sqrt{\delta t}\, \varepsilon \quad \text{where } \varepsilon \sim N(0,1)$$
$$\Rightarrow S_{t+1} = S_t (1 + \mu\, \delta t + \sigma \sqrt{\delta t}\, \varepsilon) \tag{1.9}$$

However, (1.9) is only for one stock. We need the multivariate extension of (1.9) if we want to simulate more than one stock. The multivariate version of (1.9) is as follow:

$$\begin{pmatrix} \delta S_1 / S_1 \\ \vdots \\ \delta S_p / S_p \end{pmatrix} \sim N_p[\begin{pmatrix} \mu_1 \\ \vdots \\ \mu_p \end{pmatrix}(\delta t), \Sigma\,(\delta t)] \tag{1.10}$$

Therefore we can simulate $p \times 1$ random vectors v from the distribution in (1.10),

$$\begin{pmatrix} v_1 \\ \vdots \\ v_p \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_p \end{pmatrix}(\delta t) + (\sqrt{\delta t})\, C' \begin{pmatrix} z_1 \\ \vdots \\ z_p \end{pmatrix} \quad \text{where } z_i \text{ iid } N(0,1),$$

and compute the simulated sample path of the p stocks using the following recursion:

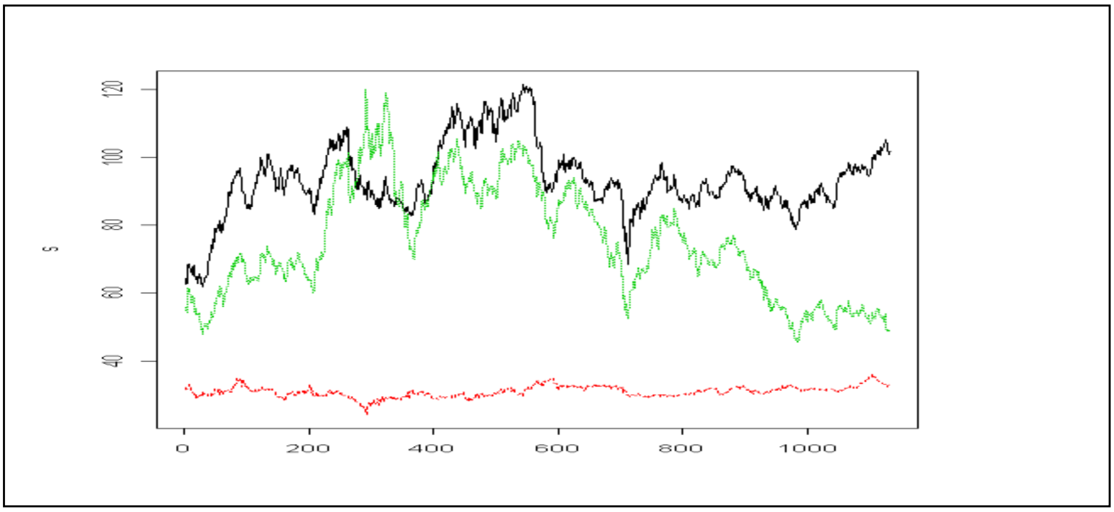
$$\begin{pmatrix} S_{t+1,1} \\ \vdots \\ S_{t+1,p} \end{pmatrix} = \begin{pmatrix} S_{t,1}(1+v_1) \\ \vdots \\ S_{t,p}(1+v_p) \end{pmatrix}.$$

The following R commands implement this Monte Carlo simulation.

```
set.seed(7)                   # set random seed
mu<-apply(u180,2,mean)        # compute daily return rate
sigma<-var(u180)              # compute daily variance rate
C<-chol(sigma)                # Cholesky decomposition of sigma
s<-cbind(t1,t2,t3)            # combine t1,t2,t3 to form s
s0<-s[1043,]                  # set s0 to the most recent price
for (i in 1:90) {             # simulate price for future 90 days
   z<-rnorm(3)                # generate normal random vector
   v<-mu+t(C)%*%z             # transform to multivariate normal
   s1<-s0*(1+v)               # new stock price
   s<-rbind(s,t(s1))          # append s1 to s
   s0<-s1                     # update s0
}
```

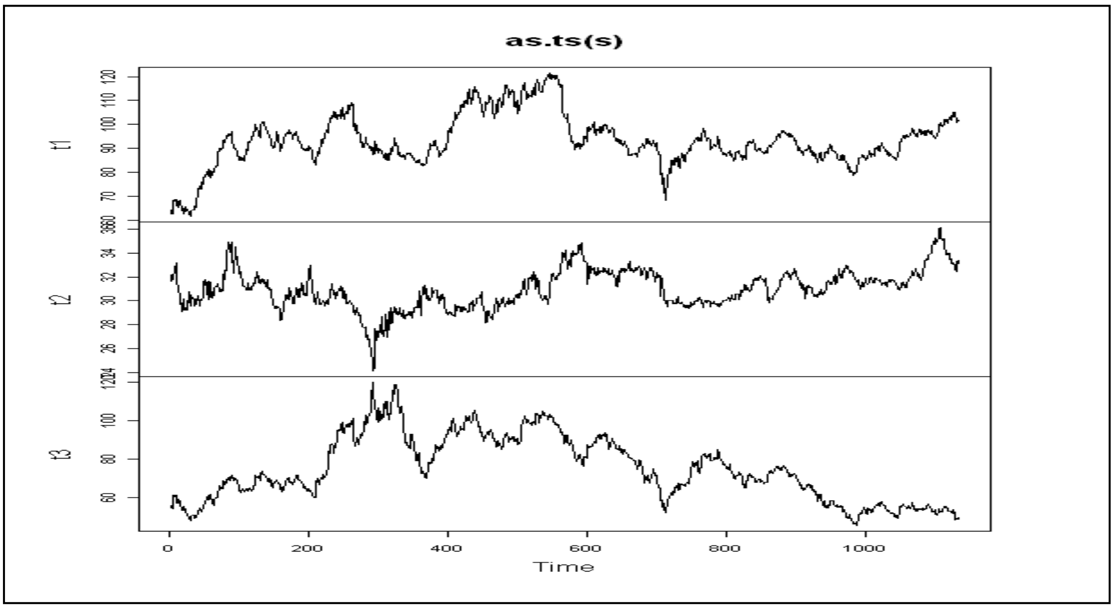
Note that the matrix C such that $C'C=\Sigma$ is not unique. Therefore the Cholesky decomposition of Σ is usually found by restricting C to be an upper triangular matrix. The time unit is daily and therefore the factor $\delta t = 1/252$, (252 trading days in one year) is not needed. Setting the random seed in the simulation ensures that the same set of pseudo random numbers is generated each time. This is useful and allowing us to check and debugging our programs. Finally we can plot the real and simulated stock price by

```
> matplot(s,type="l")
```



or using

```
> plot(as.ts(s))
```



Based on these simulated stock prices, we can calibrate the price of the portfolio or those of its related derivatives. Here we only simulated one sample path of these stock prices. In principle, we can simulate as many paths as we want and build up the corresponding loss distribution from which the VaR can be computed.

1.9 QQ-Chisquare plot using EXCEL (To be elaborated in the tutorial)

EXCEL has no built-in function for QQ-Chisquare plot. However, we can use the built-in matrix multiplication function *mmult()* to compute the generalized distance and produce the QQ-Chisquare plot.

1. We set up u1, u2, and u3 in cells A3:C1044.
2. Compute the mean vector of u180, say m and store it in I12:K12.
3. Compute the $u_i - m$ and store them in E865:G1044.
4. In cell I3, enter the formula:

=COVARIANCE.S(OFFSET($A$865:$A$1044,0,I$2),OFFSET($A$865:$A$1044,0,$H3)) to compute the covariance between u1 and u1. Copy this formula to I3:K5. This will produce the covariance matrix of u180.

5. Highlight the cell I8:K10 and enter the formula:
   =MINVERSE(I3:K5). You need to use **shift-ctrl-enter** to enter. This will produce the inverse of the Covariance matrix S.
6. In cell I865, enter the formula:
   =SUMPRODUCT(MMULT(E865:G865,$I$8:$K$10),E865:G865)
   This formula used to compute the distance $d_1^2 = (u_i - \bar{u})' S^{-1} (u_i - \bar{u})$.
7. Create 1 to 180 in J865:J1044 and the corresponding (i-0.5)/180 in K865:K1044.
8. In cell L865, enter =CHIINV(1-K865,3) to compute the quantile from the Chi-square(3) distribution. Copy the formula to L866:L1044.
9. Sorted the distance in I865:I1044 in ascending order and store them in M865:M1044.
10. Finally, plot the cell L865:N865. This will produce the QQ-Chisquare plot.
11. Add in the least square trend line as the reference line by right-clicking any points in the plot.

Using VBA in EXCEL

There is a built-in function to compute covariance matrix in EXCEL. However, the resulting matrix is in lower triangular form. Step 4 is to use the COVAR() to compute covariance between two vectors and use OFFSET() function to create the covariance matrix in full mode. A better way to do this is to write a function mcov() using VBA.

1. In EXCEL press alt-F11 to invoke VB menu. Choose File -> Import to read in multivariate.bas.
2. Highlight a 3x3 range of cells as in Step 5, enter the formula:
   =mcov(A865:C1044), and use **shift-ctrl-enter** to enter. Then you will have the same covariance as in step 4.

In fact, there are *mcor()* and *cholesky()* function in the file “multivariate.bas” that you can use similarly. The algorithm for cholesky decomposition is given in the appendix.

1.10 Marginal and Conditional distribution

In simple linear regression, we have to study the marginal and the conditional distributions of bivariate normal distribution. These results are readily extended to multivariate normal distribution.

Let $X \sim N_p(\mu, \Sigma)$ and X, μ and Σ are partitioned into:

$$X = \begin{bmatrix} X_1 \\ \cdots \\ X_2 \end{bmatrix}, \mu = \begin{bmatrix} \mu_1 \\ \cdots \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \vdots & \Sigma_{12} \\ \cdots & & \cdots \\ \Sigma_{21} & \vdots & \Sigma_{22} \end{bmatrix}.$$

where X_1, μ_1 are $q \times 1$; X_2, μ_2 are $(p-q)\times 1$ vectors and Σ_{11} is $q \times q$, Σ_{12} is $q \times (p-q)$, Σ_{21} is $(p-q)\times q$ and Σ_{22} is $(p-q)\times(p-q)$ matrices.

1. If $X \sim N_p(\mu, \Sigma)$, then the marginal distribution of any subset of q component of X is q-variate normal.

Proof. Without loss of generality, it suffices to show that $X_1 \sim N_q(\mu_1, \Sigma_{11})$.

For $X_1 = BX \sim N_q(B\mu, B\Sigma B')$, take the $q \times p$ matrix $B = [I_q \mid 0]$, $X_1 \sim N_q(\mu_1, \Sigma_{11})$.

2. X_1 and X_2 are independent if and only if $\Sigma_{12} = 0$

3. $X_1 - \Sigma_{12}\Sigma_{22}^{-1} X_2 \sim N_q(\mu_1 - \Sigma_{12}\Sigma_{22}^{-1}\mu_2, \Sigma_{11\bullet 2})$ and is independent of X_2, where $\Sigma_{11\bullet 2} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$.

Proof. Let $C = \begin{bmatrix} I_q & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & I_{p-q} \end{bmatrix}$. Then $CX = \begin{bmatrix} X_1 - \Sigma_{12}\Sigma_{22}^{-1} X_2 \\ X_2 \end{bmatrix}$ is p-variate

normal with mean $\begin{bmatrix} \mu_1 - \Sigma_{12}\Sigma_{22}^{-1}\mu_2 \\ \mu_2 \end{bmatrix}$ and covariance matrix

$$C\Sigma C' = \begin{bmatrix} I_q & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & I_{p-q} \end{bmatrix}\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\begin{bmatrix} I_q & 0 \\ -\Sigma_{22}^{-1}\Sigma_{21} & I_{p-q} \end{bmatrix} = \begin{bmatrix} \Sigma_{11\bullet 2} & 0 \\ 0 & \Sigma_{22} \end{bmatrix}.$$

4. The conditional distribution of $X_1 \mid X_2 = x_2$ is $N_q[\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11\bullet 2}]$.

Proof. This result followed from (3) by conditioning on $X_2 = x_2$.

Before closing this chapter, there is a multivariate extension of the Chi-square distribution, known as the **Wishart** distribution. Let $Z_1, \dots, Z_m$ iid $N_p(0, \Sigma)$, then $B = \sum_{i=1}^{m} Z_i Z'_i$ has a Wishart distribution with *m* degrees of freedom and scale matrix Σ, denoted by $W_p(m, \Sigma)$. The density of $W_p(m, \Sigma)$ is

$$f_m(B \mid \Sigma) = |B|^{(m-p-1)/2} |\Sigma|^{-m/2} e^{-tr(\Sigma^{-1}B)/2} \Big/ K,$$

where $K = 2^{mp/2}\pi^{p(p-1)/4}\prod_{i=1}^{p}\Gamma(\frac{m+1-i}{2})$.

There is a related important result for sampling distribution of $\overline{X}$ and *S*:

If $X_1, \dots, X_n$ iid $N_p(\mu, \Sigma)$, then $\overline{X} \sim N_p[\mu, (1/n)\Sigma]$ and is independent of the SSCP matrix $A = (n-1)S \sim W_p(n-1, \Sigma)$, which has a Wishart distribution with *n-1* degrees of freedom and scale matrix Σ. [Compare this result with $(n-1)S/\sigma^2 \sim \chi^2_{n-1}$ in Section (1.5).] The density of the SSCP matrix *A* is

$$f_{n-1}(A \mid \Sigma) = |A|^{(n-p-2)/2} |\Sigma|^{-(n-1)/2} e^{-tr(\Sigma^{-1}A)/2} \Big/ K \text{ and } K = 2^{(n-1)p/2}\pi^{p(p-1)/4}\prod_{i=1}^{p}\Gamma(\frac{n-i}{2}).$$

Particularly, when p=1, $B = Z_1^2 + ... + Z_m^2$, where $Z_1, \ldots, Z_m$ iid $N(0, \sigma^2)$, or $B \sim W_1(m, \sigma^2)$.

The density of B is

$$f_m(B \mid \sigma^2) = B^{m/2-1} (\sigma^2)^{-m/2} e^{-B/(2\sigma^2)} \Big/ K \text{, where } K = 2^{m/2}\Gamma(m/2) \text{, or } B/\sigma^2 \sim \chi_m^2 .$$

Reference

Chapter 4 of Applied Multivariate Statistical Analysis, 5^{th} ed., Richard Johnson and Dean Wichern, Prentice Hall.

Chapter 1 of Elements of Financial Risk Management, Peter Christoffersen, Academic Press.

Risk Management and Financial Institutions (4^{th} ed.), John Hull, Wiley.

Appendix:

Empirical properties of Stock price

Modeling the stock price is one of the most active research areas in Quantitative Finance. Stock prices are affected by many factors, for example, interest rate, economic and political status etc. However, there are some empirical findings about the stock prices that are widely accepted by researchers.

1. Instead of modeling the stock price itself, we usually model the (daily) return of the stock. There are two type of returns:
   logarithm return $\tilde{u}_i = \ln(S_i) - \ln(S_{i-1})$ or
   arithmetic return $u_i = (S_i - S_{i-1})/S_{i-1} = S_i / S_{i-1} - 1$
2. These two returns are close if u_i is small. This can be seen easily by the fact that $\tilde{u}_i = \ln(S_i / S_{i-1}) = \ln(1 + u_i) \approx u_i$.
3. Although the logarithm return is additive, we usually use arithmetic return for simplicity.
4. If the time interval is short (say daily, δt =1/252=0.004) then we may assume the mean of u_i is zero, though it is not necessary with high computing performance nowadays.
5. There are little autocorrelation in u_i, i.e., $Corr(u_{i+k}, u_i) \approx 0$ for *k=1,2,…*. In other words, returns are almost impossible to predict from their own history.
6. Although the Black-Scholes-Merton model assumed that the distribution of u_i is normal, we found that the distribution may have a fatter tail than normal in many cases. Fatter tail means a higher probability of large losses (or profit) than the normal distribution. This is particularly important for calculating VaR.
7. The standard deviation of u_i is large compare to the mean of u_i. Empirical finding of this s.d. of daily return is usually lies between 0.95% and 3.15%.
8. This s.d. is of course varies with time and depends on many unknown and uncontrollable factors.
9. Since the mean of u_i is assumed to be zero, the variance of u_i can be

estimated by the squared return u_i^2. For these squared returns, there are some positive autocorrelation exist, i.e., $Corr(u_{i+k}^2, u_i^2) > 0$ for some small *k*.

10. A generic stochastic model for the return is

$$u_i = \mu_i + \sigma_i z_i \quad \text{where } z_i \text{ i.i.d. } D(0,1),$$

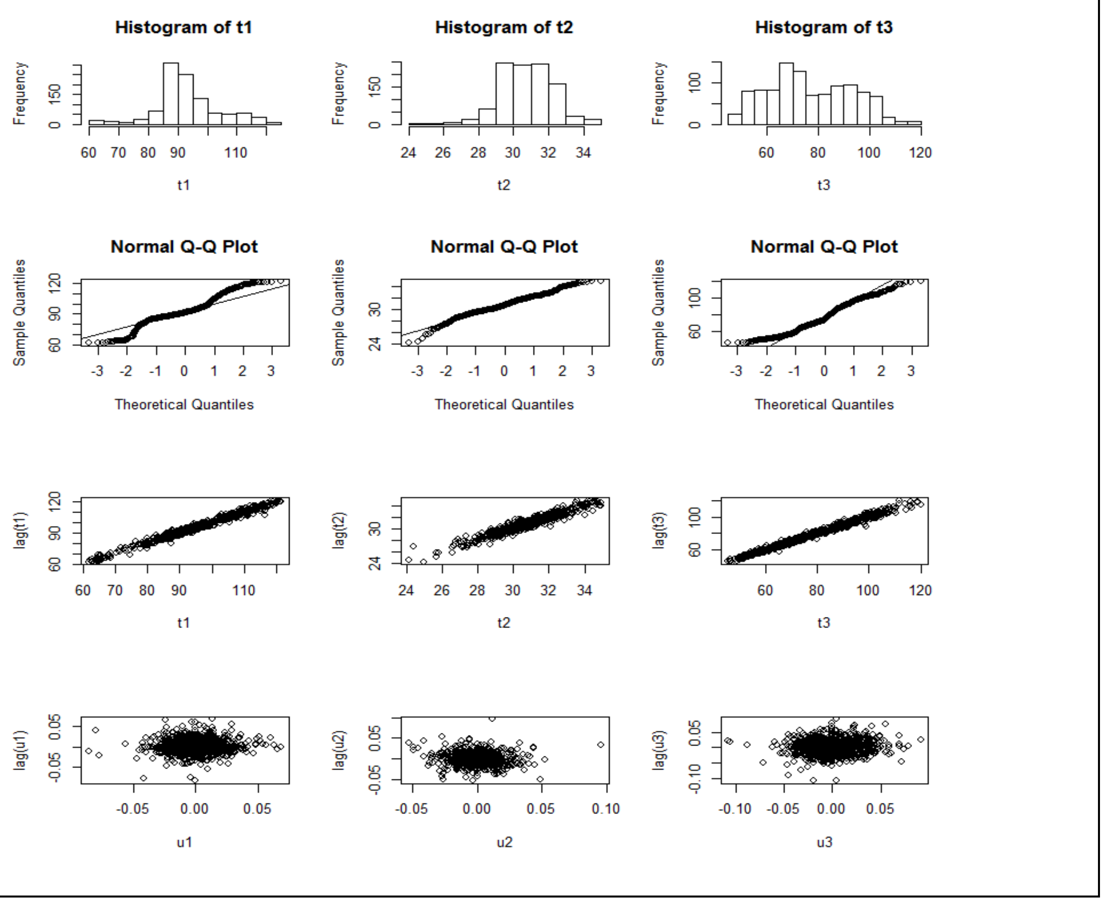
where z_i are independently identically distributed according to a distribution *D* with mean 0 and variance 1.

11. A special case of the model in 10 is

$$u_i = \mu\, \delta t + \sigma \sqrt{\delta t}\, z_i \quad \text{where } z_i \text{ i.i.d. } N(0,1).$$

To understand the difference in statistical properties between the original series S_i and the relative return u_i, we produce the histogram, the qq-normal plot for S_i, S_{i+1} vs S_i and u_{i+1} vs u_i.

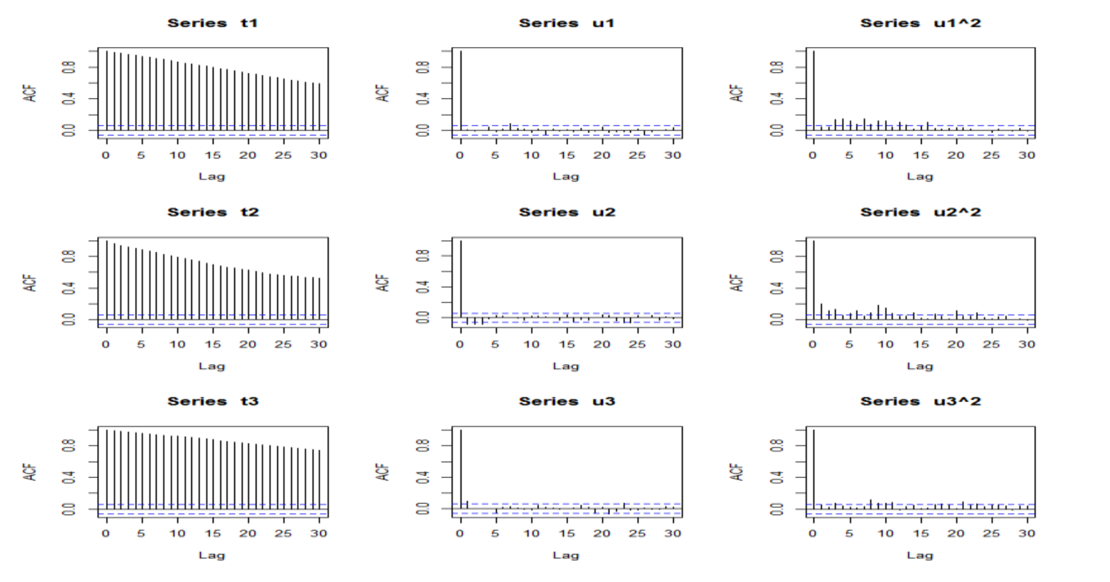
```
par(mfrow=c(4,3))              # set multi-frame graphic
hist(t1); hist(t2); hist(t3)     # histogram for t1, t2, t3
qqnorm(t1); qqline(t1)           # qq-normal plot
qqnorm(t2); qqline(t2)
qqline(t3); qqline(t3)
plot(t1,lag(t1))                 # plot t(i+1) vs t(i)
plot(t2,lag(t2))
plot(t3,lag(t3))
plot(u1,lag(u1))                 # plot u(i+1) vs u(i)
plot(u2,lag(u2))
plot(u3,lag(u3))
```



Compare with the plots of u_i on page 5, it is obvious that the distributions of S_i's are very different from normal distribution. More importantly, strong autocorrelations exist in S_i while almost does not exist in u_i. These plots show that S_i has strong autocorrelation (of lag 1) while u_i has small or no autocorrelation which confirms point 5 on page 15. We can also plot the **auto-correlation function (acf)** of lag *k*:

$$a_k = \sum_{i=1}^{n-k}(a_i - \bar{a})(a_{i+k} - \bar{a}) \Big/ \sum_{i=1}^{n}(a_i - \bar{a})^2 \quad \text{for } k = 1,\ldots,K .$$

```
par(mfrow=c(3,3))
acf(t1); acf(u1); acf(u1^2)      # acf for t1, u1 and u1^2
acf(t2); acf(u2); acf(u2^2)      # acf for t2, u2 and u2^2
acf(t3); acf(u3); acf(u3^2)      # acf for t3, u3 and u3^2
```



These plots also show that strong auto-correlation in S_i; little or no auto-correlation in u_i and some auto-correlation in u_i^2.

Algorithm for Cholesky decomposition

Recall that the Cholesky decomposition of a $p \times p$ positive definite matrix *A* is to find a $p \times p$ upper triangular matrix *C* such that *C'C=A*. Let $A = (a_{ij})$ and $C = (c_{ij})$ where $c_{ij} = 0$ for $i > j$. The following is the algorithm for finding *C* given *A*.

[1] $c_{11} = \sqrt{a_{11}}$, $c_{1j} = a_{1j} / c_{11}$ *for* $j = 2,\ldots,p$

[2] For *i=2,...,p*

$$c_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} c_{ki}^2} \quad \text{and} \quad c_{ij} = (a_{ij} - \sum_{k=1}^{i-1} c_{ki}c_{kj}) \Big/ c_{ii} \quad \textit{for} \quad j = i+1,\ldots,p$$

Proof:

This algorithm can be derived by noting that the *(i,j)* element of $C'C = \sum_{k=1}^{p} c_{ki}c_{kj} = a_{ij}$.

When $i=1$ and $j=1$, $c_{11}^2 = a_{11} \Rightarrow c_{11} = \sqrt{a_{11}}$.

when $i=1$ and $j=2,..p$, $a_{1j} = \sum_{k=1}^{p} c_{k1} c_{kj} = c_{11} c_{1j} \Rightarrow c_{1j} = a_{1j} / c_{11}$ (since $c_{ij} = 0$ for $i>j$)

For $i=2,\dots,p$, $a_{ii} = \sum_{k=1}^{p} c_{ki}^2 = \sum_{k=1}^{i} c_{ki}^2 = \sum_{k=1}^{i-1} c_{ki}^2 + c_{ii}^2$ (again $c_{ij} = 0$ for $i>j$)

Hence $c_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} c_{ki}^2}$. Finally, for $j=i+1,\dots,p$,

$$a_{ij} = \sum_{k=1}^{i} c_{ki} c_{kj} = c_{ii} c_{ij} + \sum_{k=1}^{i-1} c_{ki} c_{kj} \quad \Rightarrow c_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} c_{ki} c_{kj}\right) \Big/ c_{ii} \quad for \quad j = i+1, \dots, p \,.$$

As an example, the cholesky decomposition for 3x3 matrices is:

$$A = \begin{bmatrix} c_{11} & 0 & 0 \\ c_{12} & c_{22} & 0 \\ c_{13} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ 0 & c_{22} & c_{23} \\ 0 & 0 & c_{33} \end{bmatrix}$$
$$= \begin{bmatrix} c_{11}^2 & c_{12} c_{11} & c_{13} c_{11} \\ c_{21} c_{11} & c_{22}^2 + c_{12}{}^2 & c_{13} c_{12} + c_{23} c_{22} \\ c_{31} c_{11} & c_{13} c_{12} + c_{23} c_{22} & c_{33}^2 + c_{23}{}^2 + c_{13}{}^2 \end{bmatrix}$$

For instance:

$$c_{33} = \sqrt{a_{33} - a_{23}^2 - a_{13}^2}$$
$$c_{32} = \frac{1}{c_{22}} (a_{23} - c_{13} c_{12}) \,.$$