NLA Homework1

Group O

Dafu Zhu, Jiawei Ni

Problem 1

Find the value of x such that the matrix below has rank 3:

$$\begin{pmatrix} x & 1 & 1 & 1 \\ 1 & x & 1 & 1 \\ 1 & 1 & x & 1 \\ 1 & 1 & 1 & x \end{pmatrix}.$$

Solution:

Given matrix:

$$A = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} x & 1 & 1 & 1 \\ 1 & x & 1 & 1 \\ 1 & 1 & x & 1 \\ 1 & 1 & 1 & x \end{pmatrix}$$

is row equivalent to:

$$\begin{pmatrix} r_4 \\ r_2 \\ r_3 \\ r_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & x \\ x & 1 & 1 & 1 \\ 1 & x & 1 & 1 \\ 1 & 1 & x & 1 \end{pmatrix}$$

Performing row operations:

$$\begin{pmatrix} r_4 \\ r_2 - r_4 \\ r_3 - r_4 \\ r_1 - xr_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & x \\ 0 & x-1 & 0 & 1-x \\ 0 & 0 & x-1 & 1-x \\ 0 & 1-x & 1-x & 1-x^2 \end{pmatrix}$$

$$\begin{pmatrix} r_4 \\ r_2 - r_4 \\ r_3 - r_4 \\ r_1 + r_2 - (x+1)r_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & x \\ 0 & x-1 & 0 & 1-x \\ 0 & 0 & x-1 & 1-x \\ 0 & 0 & 1-x & (1-x^2)+(1-x) \end{pmatrix}$$

$$\begin{pmatrix} r_4 \\ r_2 - r_4 \\ r_3 - r_4 \\ r_1 + r_2 + r_3 - (x+2)r_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & x \\ 0 & x-1 & 0 & 1-x \\ 0 & 0 & x-1 & 1-x \\ 0 & 0 & 0 & (1-x^2)+2(1-x) \end{pmatrix}$$

where

$$(1-x^2)+2(1-x) = -x^2-2x+3 = -(x+3)(x-1)$$

The determinant of A gives

$$\det A = -(x+3)(x-1)^3 = 0$$

Solving for x:

$$x_1 = 1, \quad x_2 = -3$$

If $x = 1$, rank of A is 1.

If $x = -3$, rank of A is 3.

Thus, $x = -3$.

Problem 2

Show that the product of two symmetric matrices is not necessarily a symmetric matrix.

Hint: Find 2×2 matrices A and B such that $A = A^t, B = B^t$, and $(AB)^t \neq AB$.

Solution:

Let:

$$A = \begin{bmatrix} 1 & 3 \\ 3 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

$$AB = \begin{bmatrix} 5 & 10 \\ 8 & 9 \end{bmatrix} \neq (AB)^T$$

To generalize:

$$A = \begin{bmatrix} a & c \\ c & b \end{bmatrix}, \quad B = \begin{bmatrix} d & f \\ f & e \end{bmatrix}$$

$$AB = \begin{bmatrix} ad + cf & af + ce \\ cd + bf & cf + be \end{bmatrix}$$

AB is symmetric if and only if:

$$cd + bf = af + ce$$

which simplifies to:

$$c(d - e) = f(a - b)$$

This condition does not always hold. Thus, the product of two symmetric matrices is not necessarily a symmetric matrix.

Problem 3

Let A be a square matrix of size n. Show that $AD = DA$ for any diagonal matrix D of size n if and only if A is a diagonal matrix.

Solution:

If A is diagonal, given $D = \mathrm{diag}(d_k)_{k=1:n}$:

$$A = \mathrm{diag}(a_k)_{k=1:n} = \mathrm{col}(a_k e_k)_{k=1:n}$$

$$AD = \mathrm{col}(d_k \cdot a_k e_k)_{k=1:n} = \mathrm{diag}(d_k a_k)_{k=1:n}$$

$$DA = \mathrm{col}(a_k \cdot d_k e_k)_{k=1:n} = \mathrm{diag}(a_k d_k)_{k=1:n}$$

Thus, $AD = DA$.

If $AD = DA$,

Denote $A = \mathrm{col}(a_k)_{k=1:n} = \mathrm{row}(r_j)_{j=1:n}$

$D = \mathrm{col}(d_k e_k)_{k=1:n} = \mathrm{row}(d_j e_j^t)_{j=1:n}$, where e_i is the i-th column of the identity matrix

$$AD(j,k) = r_j \cdot d_k e_k = d_k r_j(k)$$

$$DA(j,k) = d_j e_j \cdot a_k = d_j a_k(j)$$

Since $r_j(k) = a_k(j) = A(j,k)$, we have:

$$\begin{aligned} AD = DA &\Leftrightarrow AD(j,k) = DA(j,k) \\ &\Leftrightarrow d_k \cdot A(j,k) = d_j \cdot A(j,k) \\ &\Leftrightarrow (d_k - d_j)A(j,k) = 0 \end{aligned}$$

For $\forall j \neq k$, $d_k \neq d_j \therefore A(j,k) = 0$, A is diagonal matrix

Problem 4

Let M_1, M_2, M_3, and M_4 be square matrices of the same size such that $M_1 M_2 M_3 M_4 = I$. Show that

$$M_1 M_2 M_3 M_4 = M_2 M_3 M_4 M_1 = M_3 M_4 M_1 M_2 = M_4 M_1 M_2 M_3 = I.$$

Solution:

By taking inverses from both sides, we get

$$M_2M_3M_4 = M_1^{-1}$$

$$M_3M_4 = M_2^{-1}M_1^{-1}$$

$$M_4 = M_3^{-1}M_2^{-1}M_1^{-1}$$

Since for any nonsingular matrix A, $A^{-1}A = AA^{-1} = I$

$$\therefore M_2M_3M_4M_1 = M_1^{-1}M_1 = I$$

$$M_3M_4M_1M_2 = M_2^{-1}M_1^{-1}(M_1M_2) = M_2^{-1}M_2 = I$$

$$M_4M_1M_2M_3 = M_3^{-1}M_2^{-1}M_1^{-1}(M_1M_2M_3) = I$$

Problem 5

Let A be a nonsingular matrix. If $I + A$ and $I + A^{-1}$ are also nonsingular matrices, show that

$$(I + A)^{-1} = I - \left(I + A^{-1}\right)^{-1}.$$

Solution:

$I + A^{-1} = (I + A)A^{-1}$

$(I + A)^{-1}(I + A^{-1}) = A^{-1}$ since $I + A$ is nonsingular

$(I + A)^{-1}(I + A^{-1}) + I = A^{-1} + I$ (add I on both sides)

$I + A^{-1} - (I + A)^{-1}(I + A^{-1}) = I$

$(I - (I + A)^{-1})(I + A^{-1}) = I$

$I - (I + A)^{-1} = (I + A^{-1})^{-1}$ since $(I + A^{-1})$ is nonsingular

$I - (I + A^{-1})^{-1} = (I + A)^{-1}$, Q.E.D.

Problem 6

Show that an $n \times n$ matrix A has rank 1 if and only if there exist an $n \times 1$ nonzero column vector v and an $1 \times n$ nonzero row vector w^t such that $A = vw^t$

Solution:

Denote vectors $v = (u_i)_{i=1:n}$, $w = (w_i)_{i=1:n}$, not all zero.

If $A = vw^t = col((w_k \cdot v))_{k=1:n}$

All columns are multiples of vector v, which means all column vectors are linearly dependent, the column space of A is spanned by the single vector v, $\Rightarrow rank(A) = 1$

If $rank(A) = 1$, then the column space of A is spanned by a single nonzero vector v. $\Rightarrow$ every column of A is a multiple of vector v

$\exists w_i \in \mathbb{R}, i = 1 : n, s.t.\ A = (w_1v|\cdots|w_nv) = v(w_1|\cdots|w_n) = vw^t$

Since A is nonzero (or else the rank would be 0), then v and w are both nonzero.

Problem 7

Let x and y be column vectors of size n, and let I be the identity matrix of size n. Show that the matrix $I - xy^t$ is nonsingular if and only if $y^tx \neq 1$.

Solution:

If $y^tx \neq 1$, we show that $I - xy^t$ is nonsingular

cf. derivation of Sherman-Morrison-Woodbury formula:

Solve $(I - xy^t)v = b$

$v - xy^tv = b$, denote $\xi = y^tv$

Consider the bordered matrix

$$\begin{bmatrix} I & -x \\ y^t & -1 \end{bmatrix} \begin{bmatrix} v \\ \xi \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

Factorizing

$$\begin{bmatrix} I & -x \\ y^t & -1 \end{bmatrix} = \begin{bmatrix} I & 0 \\ y^t & L_{22} \end{bmatrix} \begin{bmatrix} I & -x \\ 0 & U_{22} \end{bmatrix}$$

$L_{22}U_{22} = y^t x - 1$

Consider intermediate variable $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$

s.t. $\begin{bmatrix} I & 0 \\ y^t & L_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$

$$y_1 = b \tag{1}$$

$$y^t y_1 + L_{22} y_2 = 0 \Rightarrow y_2 = -L_{22}^{-1} y^t y_1 \tag{2}$$

$$\begin{bmatrix} I & -x \\ 0 & U_{22} \end{bmatrix} \begin{bmatrix} v \\ \xi \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

$$v = y_1 + x\xi \tag{3}$$

$$\xi = U_{22}^{-1} y_2 \tag{4}$$

By (1)(2)(3)(4),

$$\begin{aligned} v &= b + xU_{22}^{-1} y_2 \\ &= b - xU_{22}^{-1} L_{22}^{-1} y^t y_1 \\ &= b - \frac{1}{y^t x - 1} x y^t b \\ &= \left(I + \frac{1}{1 - y^t x} x y^t \right) b \end{aligned}$$

Since $y^t x \neq 1$,

$$(I - xy^t)^{-1} = I + \frac{1}{1 - y^t x} x y^t$$

Now we only need to show that if $y^t x = 1$ then the matrix $I - xy^t$ is singular.

Assume that $y^t x = 1$

$(I - xy^t)x = x - xy^t x = x - x = 0$

If $I - xy^t$ is nonsingular, multiply its inverse from both sides,

$(I - xy^t)^{-1}(I - xy^t)x = 0 \Rightarrow x = 0$

This is contradictory to $y^t x = 1$

$\therefore I - xy^t$ is singular

Problem 8

The covariance matrix of five random variables is

$$\Sigma = \begin{pmatrix} 1 & -0.525 & 1.375 & -0.075 & -0.75 \\ -0.525 & 4 & 0.1875 & 0.1875 & -0.675 \\ 1.375 & 0.1875 & 12.25 & 0.4375 & -1.875 \\ -0.075 & 0.1875 & 0.4375 & 6.25 & 0.3 \\ -0.75 & -0.675 & -1.875 & 0.3 & 4.41 \end{pmatrix}$$

Find the correlation matrix of these random variables.

Solution:

$\Omega_x = D_\sigma^{-1} \Sigma D_\sigma^{-1}$

$\sigma_x = \mathrm{diag}(\sqrt{\sum(i,i)})_{i=1:5}$

$$\sigma_x^{-1} = \mathrm{diag}\left(\frac{1}{\sqrt{\Sigma(i,i)}}\right)_{i=1:5}$$

$$\sigma_x^{-1} = \begin{pmatrix} 1 & 0 & 0 & -0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3.5 & 0 & 0 \\ 0 & 0 & 0 & 1/2.5 & 0 \\ 0 & 0 & 0 & 0 & 1/2.1 \end{pmatrix}$$

Through a bit of coding

$$\Omega_x = \begin{pmatrix} 1 & -0.2625 & 0.3929 & -0.03 & -0.3571 \\ -0.2625 & 1 & 0.0268 & 0.0375 & -0.1607 \\ 0.3929 & 0.0268 & 1 & 0.05 & -0.2551 \\ -0.03 & 0.0375 & 0.05 & 1 & 0.0571 \\ -0.3571 & -0.1607 & -0.2551 & 0.0571 & 1 \end{pmatrix}$$

In [10]:

```
import numpy as np
import pandas as pd
np.set_printoptions(suppress=True, precision=4)
```

In [11]:

```
# (8)
sig_inv = np.diag([1, 1/2, 1/3.5, 1/2.5, 1/2.1])
l = np.array([
    [1, 0, 0, 0, 0],
    [-0.525, 4, 0, 0, 0],
    [1.375, 0.1875, 12.25, 0, 0],
    [-0.075, 0.1875, 0.4375, 6.25, 0],
    [-0.75, -0.675, -1.875, 0.3, 4.41]
])
Sig = l + np.tril(l, -1).T
Ome = sig_inv @ Sig @ sig_inv
Ome
```

Out[11]:

```
array([[ 1.    , -0.2625,  0.3929, -0.03  , -0.3571],
       [-0.2625,  1.    ,  0.0268,  0.0375, -0.1607],
       [ 0.3929,  0.0268,  1.    ,  0.05  , -0.2551],
       [-0.03  ,  0.0375,  0.05  ,  1.    ,  0.0571],
       [-0.3571, -0.1607, -0.2551,  0.0571,  1.    ]])
```

Problem 9

The correlation matrix of five random variables is

$$\Omega = \begin{pmatrix} 1 & -0.25 & 0.15 & -0.05 & -0.30 \\ -0.25 & 1 & -0.10 & -0.25 & 0.10 \\ 0.15 & -0.10 & 1 & 0.20 & 0.05 \\ -0.05 & -0.25 & 0.20 & 1 & 0.10 \\ -0.30 & 0.10 & 0.05 & 0.10 & 1 \end{pmatrix}$$

(i) Compute the covariance matrix of these random variables if their standard deviations are $0.1, 0.2, 0.5, 1$, and 2 , in this order.

(ii) Compute the covariance matrix of these random variables if their standard deviations are $2, 1, 0.5, 0.2$, and 0.1 , in this order.

Solution:

(i) $\Sigma = D_\sigma \Omega D_\sigma$, $D_\sigma = \mathrm{diag}(0.1, 0.2, 0.5, 1, 2)$

$$\Sigma = \begin{bmatrix} 0.01 & -0.005 & 0.0075 & -0.005 & -0.06 \\ -0.005 & 0.04 & -0.01 & -0.05 & 0.04 \\ 0.0075 & -0.01 & 0.25 & 0.1 & 0.05 \\ -0.005 & -0.05 & 0.1 & 1. & 0.2 \\ -0.06 & 0.04 & 0.05 & 0.2 & 4. \end{bmatrix}$$

(ii)

$$\Sigma_2 = \begin{bmatrix} 4. & -0.5 & 0.15 & -0.02 & -0.06 \\ -0.5 & 1. & -0.05 & -0.05 & 0.01 \\ 0.15 & -0.05 & 0.25 & 0.02 & 0.0025 \\ -0.02 & -0.05 & 0.02 & 0.04 & 0.002 \\ -0.06 & 0.01 & 0.0025 & 0.002 & 0.01 \end{bmatrix}$$

In [12]:

```
# (9)
## (i)
D_sig = np.diag([0.1, 0.2, 0.5, 1, 2])
l = np.array([
    [1, 0, 0, 0, 0],
    [-0.25, 1, 0, 0, 0],
    [0.15, -0.1, 1, 0, 0],
    [-0.05, -0.25, 0.2, 1, 0],
    [-0.3, 0.1, 0.05, 0.1, 1]
])
Ome = l + np.tril(l, -1).T
a1 = D_sig @ Ome @ D_sig
print("(i)")
print(a1)

## (ii)
D_sig2 = np.diag([2, 1, 0.5, 0.2, 0.1])
a2 = D_sig2 @ Ome @ D_sig2
print("(ii)")
print(a2)
```

```
(i)
[[ 0.01   -0.005   0.0075 -0.005   -0.06   ]
 [-0.005   0.04   -0.01   -0.05    0.04   ]
 [ 0.0075 -0.01    0.25    0.1     0.05   ]
 [-0.005  -0.05    0.1     1.      0.2    ]
 [-0.06    0.04    0.05    0.2     4.     ]]
(ii)
[[ 4.     -0.5     0.15   -0.02   -0.06   ]
 [-0.5     1.     -0.05   -0.05    0.01   ]
 [ 0.15   -0.05    0.25    0.02    0.0025]
 [-0.02   -0.05    0.02    0.04    0.002 ]
 [-0.06    0.01    0.0025  0.002   0.01   ]]
```

Problem 10

The file indeces-close-jan3-jan31-2017.xlsx contains the January 3, 2017 - January 31, 2017 end of day values of Dow Jones, Nasdaq, and S&P 500.

(i) Compute the log daily returns of the three indices over the given time period.

(ii) Compute the sample covariance matrix of the log daily returns of the three indices over the given time period.

(iii) Compute the percentage daily returns of the three indices over the given time period.

(iv) Compute the sample covariance matrix of the percentage daily returns of the three indices over the given time period.

In [16]:

```
# adjust precision
np.set_printoptions(precision=8)

# import data
close = pd.read_excel("data/indeces-close-jan3-jan31-2017.xlsx", index_col=0)
close.head()
```

Out[16]:

Date	DOW	NASDAQ	SP500
2017-01-03	19881.75977	5429.080078	2257.830078
2017-01-04	19942.16016	5477.000000	2270.750000
2017-01-05	19899.28906	5487.939941	2269.000000
2017-01-06	19963.80078	5521.060059	2276.979980
2017-01-09	19887.38086	5531.819824	2268.899902

Solutions:

(i) Compute the log daily returns of the three indices over the given time period.

$$r = P_2/P_1 - 1$$

$$\log\_\text{return} = \log(P_2/P_1) = \log(r+1)$$

In [17]:
```python
def get_lr(df):
    """
    get log return
    """
    log_return = np.log((df.pct_change().dropna() + 1).values)
    return log_return

log_return = get_lr(close)
log_return
```

Out[17]:
```
array([[ 0.00303337,  0.0087878 ,  0.00570596],
       [-0.00215209,  0.00199544, -0.00077097],
       [ 0.00323667,  0.00601693,  0.00351079],
       [-0.00383527,  0.00194696, -0.00355491],
       [-0.00160288,  0.00360893,  0.        ],
       [ 0.0049611 ,  0.00212858,  0.00282564],
       [-0.00317625, -0.00290873, -0.00214711],
       [-0.00026496,  0.00478886,  0.00184813],
       [-0.00296939, -0.00636925, -0.00297191],
       [-0.00111269,  0.00305018,  0.0017622 ],
       [-0.00365835, -0.00280646, -0.00361584],
       [ 0.00479528,  0.00274889,  0.00336058],
       [-0.00138291, -0.00043033, -0.00269375],
       [ 0.00568393,  0.0086105 ,  0.00654314],
       [ 0.00779364,  0.00983901,  0.00799405],
       [ 0.00161319, -0.00020504, -0.00073565],
       [-0.00035482,  0.00098968, -0.00086684],
       [-0.00612251, -0.00834984, -0.00602767],
       [-0.0053742 ,  0.00019238, -0.0008903 ]])
```

(ii) Compute the sample covariance matrix of the log daily returns of the three indices over the given time period.

Denote the matrix `log_return` as T_X. Recall that

$$\bar{T}_{X_k} = T_{X_k} - \mu_{X_k}$$

$$\bar{T}_X = \text{col}(\bar{T}_{X_k})_{k=1:n}$$

By (1.37) in the textbook, the sample covariance matrix

$$\hat{\Sigma}_X = \frac{1}{N-1}\bar{T}_X^t \bar{T}_X$$

In [18]:
```python
def get_Sig(T):
    """
    get the sample cov Sigma given matrix T
    """
    T_bar = T - T.mean(axis=0)
    Sig = 1 / (T_bar.shape[0]) * T_bar.T @ T_bar
    return Sig

T = log_return
Sig = get_Sig(T)
Sig
```

Out[18]:
```
array([[0.0000151 , 0.00001386, 0.00001289],
       [0.00001386, 0.00002199, 0.00001562],
       [0.00001289, 0.00001562, 0.00001359]])
```

(iii) Compute the percentage daily returns of the three indices over the given time period.

In [19]:
```python
def get_pctchg(df):
    """
    get percentage return
    """
    pct_chg = df.pct_change().dropna().values
    return pct_chg

pct_chg = get_pctchg(close)
pct_chg
```

Out[19]:
```
array([[ 0.00303798,  0.00882653,  0.00572227],
       [-0.00214977,  0.00199743, -0.00077067],
       [ 0.00324191,  0.00603507,  0.00351696],
       [-0.00382792,  0.00194886, -0.00354859],
       [-0.0016016 ,  0.00361545,  0.        ],
       [ 0.00497343,  0.00213085,  0.00282964],
       [-0.00317121, -0.00290451, -0.00214481],
       [-0.00026492,  0.00480035,  0.00184984],
       [-0.00296499, -0.00634901, -0.0029675 ],
       [-0.00111207,  0.00305484,  0.00176375],
       [-0.00365167, -0.00280252, -0.00360931],
       [ 0.0048068 ,  0.00275267,  0.00336624],
       [-0.00138196, -0.00043024, -0.00269013],
       [ 0.00570011,  0.00864768,  0.00656459],
```

```
       [ 0.00782409,  0.00988757,  0.00802609],
       [ 0.00161449, -0.00020502, -0.00073538],
       [-0.00035475,  0.00099017, -0.00086646],
       [-0.0061038 , -0.00831508, -0.00600954],
       [-0.00535979,  0.0001924 , -0.00088991]])
```

(iv) Compute the sample covariance matrix of the percentage daily returns of the three indices over the given time period.

In [20]:
```python
T2 = pct_chg
Sig2 = get_Sig(T2)
Sig2
```

Out[20]:
```
array([[0.00001512, 0.00001389, 0.00001292],
       [0.00001389, 0.00002204, 0.00001566],
       [0.00001292, 0.00001566, 0.00001363]])
```

We observed an inflation in covariance values if we use percentage returns instead of log returns

In [21]:
```python
Sig2 - Sig > 0
```

Out[21]:
```
array([[ True,  True,  True],
       [ True,  True,  True],
       [ True,  True,  True]])
```

Problem 11

The file indices-july2016.csv contains the January 2016 - July 2016 end of day values of nine major US indeces.

(i) Compute the sample covariance matrix of the daily percentage returns of the indeces, and the corresponding sample corelation matrix.

Compute the sample covariance and correlation matrices for daily log returns, and compare them with the corresponding matrices for daily percentage returns.

(ii) Compute the sample covariance matrix of the weekly percentage returns of the indeces, and the corresponding sample corelation matrix.

Compute the sample covariance and correlation matrices for weekly log returns, and compare them with the corresponding matrices for weekly percentage returns.

(iii) Compute the sample covariance matrix of the monthly percentage returns of the indeces, and the corresponding sample corelation matrix.

Compute the sample covariance and correlation matrices for monthly log returns, and compare them with the corresponding matrices for monthly percentage returns.

(iv) Comment on the differences between the sample covariance and correlation matrices for daily, weekly, and monthly returns.

Solutions:

In [22]:
```python
# import data
value = pd.read_csv("data/indices-july2016.csv", index_col=0)
value.index = pd.to_datetime(value.index)
value.head()
```

Out[22]:

Date	BAC	BNS	C	GS	JPM	MS	TD	USB	WFC
2016-01-04	16.176800	38.183614	50.516574	174.428759	61.870034	30.587742	36.790199	40.463358	50.916643
2016-01-05	16.176800	38.451837	50.249813	171.425438	61.977008	30.393412	36.503965	40.658457	50.897396
2016-01-06	15.832194	37.522631	49.518689	167.240487	61.082316	29.635520	35.689043	40.043897	49.925449
2016-01-07	15.261132	36.909548	46.989405	162.100381	58.612182	28.158602	34.889533	39.283015	48.501207
2016-01-08	14.965755	36.928707	45.576561	161.430796	57.299313	27.575607	34.735410	38.726986	47.692854

Compact the covariance matrix and correlation matrix of log returns and percentage returns into functions.

In [23]:
```python
def cov_log(df):
    """
    covariance matrix of log returns
    """
    T = get_lr(df)
    cov = get_Sig(T)
    return cov

def corr_log(df):
    """
```

```
    correlation matrix of log returns
    """
    Sig = cov_log(df)
    D_sig = np.diag(np.sqrt(np.diag(Sig)))
    D_sig_inv = np.linalg.inv(D_sig)
    corr = D_sig_inv @ Sig @ D_sig_inv
    return corr

def cov_pct(df):
    """
    covariance matrix of percentage returns
    """
    T = get_pctchg(df)
    cov = get_Sig(T)
    return cov

def corr_pct(df):
    """
    correlation matrix of percentage returns
    """
    Sig = cov_pct(df)
    D_sig = np.diag(np.sqrt(np.diag(Sig)))
    D_sig_inv = np.linalg.inv(D_sig)
    corr = D_sig_inv @ Sig @ D_sig_inv
    return corr
```

(i) Compute the sample covariance matrix of the daily percentage returns of the indeces, and the corresponding sample corelation matrix.

Compute the sample covariance and correlation matrices for daily log returns, and compare them with the corresponding matrices for daily percentage returns.

In [24]:
```
daily = value
print("Sample covariance matrix of daily percentage returns:")
pct_day_cov = pd.DataFrame(cov_pct(daily))
pct_day_cov
```

```
Sample covariance matrix of daily percentage returns:
```

Out[24]:

	0	1	2	3	4	5	6	7	8
0	0.000527	0.000244	0.000517	0.000372	0.000374	0.000493	0.000195	0.000291	0.000293
1	0.000244	0.000301	0.000272	0.000210	0.000209	0.000263	0.000215	0.000168	0.000162
2	0.000517	0.000272	0.000570	0.000397	0.000399	0.000529	0.000222	0.000307	0.000310
3	0.000372	0.000210	0.000397	0.000347	0.000292	0.000408	0.000173	0.000237	0.000230
4	0.000374	0.000209	0.000399	0.000292	0.000324	0.000384	0.000165	0.000243	0.000233
5	0.000493	0.000263	0.000529	0.000408	0.000384	0.000578	0.000221	0.000307	0.000303
6	0.000195	0.000215	0.000222	0.000173	0.000165	0.000221	0.000192	0.000138	0.000134
7	0.000291	0.000168	0.000307	0.000237	0.000243	0.000307	0.000138	0.000221	0.000196
8	0.000293	0.000162	0.000310	0.000230	0.000233	0.000303	0.000134	0.000196	0.000224

In [25]:
```
print("Sample correlation matrix of daily percentage returns:")
pct_day_corr = pd.DataFrame(corr_pct(daily))
pct_day_corr
```

```
Sample correlation matrix of daily percentage returns:
```

Out[25]:

	0	1	2	3	4	5	6	7	8
0	1.000000	0.614084	0.943582	0.869843	0.905068	0.893289	0.612314	0.853221	0.852008
1	0.614084	1.000000	0.656837	0.649994	0.669019	0.631424	0.895600	0.651481	0.625134
2	0.943582	0.656837	1.000000	0.891605	0.928207	0.921500	0.672348	0.865484	0.868332
3	0.869843	0.649994	0.891605	1.000000	0.869316	0.911971	0.669761	0.853871	0.824126
4	0.905068	0.669019	0.928207	0.869316	1.000000	0.888550	0.662420	0.909170	0.865720
5	0.893289	0.631424	0.921500	0.911971	0.888550	1.000000	0.663019	0.860713	0.842499
6	0.612314	0.895600	0.672348	0.669761	0.662420	0.663019	1.000000	0.670164	0.647531
7	0.853221	0.651481	0.865484	0.853871	0.909170	0.860713	0.670164	1.000000	0.879214
8	0.852008	0.625134	0.868332	0.824126	0.865720	0.842499	0.647531	0.879214	1.000000

In [26]:
```
print("Sample covariance matrix of daily log returns:")
log_day_cov = pd.DataFrame(cov_log(daily))
log_day_cov
```

```
Sample covariance matrix of daily log returns:
```

Out[26]:

	0	1	2	3	4	5	6	7	8

0	0.000532	0.000245	0.000522	0.000376	0.000375	0.000498	0.000196	0.000294	0.000294
1	0.000245	0.000297	0.000272	0.000211	0.000208	0.000264	0.000214	0.000168	0.000162
2	0.000522	0.000272	0.000576	0.000402	0.000400	0.000536	0.000225	0.000310	0.000312
3	0.000376	0.000211	0.000402	0.000351	0.000293	0.000414	0.000174	0.000239	0.000231
4	0.000375	0.000208	0.000400	0.000293	0.000323	0.000387	0.000165	0.000244	0.000233
5	0.000498	0.000264	0.000536	0.000414	0.000387	0.000586	0.000223	0.000311	0.000305
6	0.000196	0.000214	0.000225	0.000174	0.000165	0.000223	0.000192	0.000139	0.000135
7	0.000294	0.000168	0.000310	0.000239	0.000244	0.000311	0.000139	0.000222	0.000196
8	0.000294	0.000162	0.000312	0.000231	0.000233	0.000305	0.000135	0.000196	0.000224

In [27]:
```python
print("Sample correlation matrix of daily log returns:")
log_day_corr = pd.DataFrame(corr_log(daily))
log_day_corr
```

```
Sample correlation matrix of daily log returns:
```

Out[27]:

	0	1	2	3	4	5	6	7	8
0	1.000000	0.615775	0.943308	0.870803	0.905344	0.893048	0.614656	0.853932	0.850373
1	0.615775	1.000000	0.658394	0.652060	0.670548	0.632548	0.896778	0.652902	0.626568
2	0.943308	0.658394	1.000000	0.893366	0.928466	0.922524	0.676455	0.866574	0.867723
3	0.870803	0.652060	0.893366	1.000000	0.871169	0.912862	0.671991	0.854523	0.824505
4	0.905344	0.670548	0.928466	0.871169	1.000000	0.889834	0.664224	0.911052	0.865464
5	0.893048	0.632548	0.922524	0.912862	0.889834	1.000000	0.665056	0.860777	0.842024
6	0.614656	0.896778	0.676455	0.671991	0.664224	0.665056	1.000000	0.671095	0.649924
7	0.853932	0.652902	0.866574	0.854523	0.911052	0.860777	0.671095	1.000000	0.878271
8	0.850373	0.626568	0.867723	0.824505	0.865464	0.842024	0.649924	0.878271	1.000000

Comparisons

- Overall, the differences between covariance matrices and correlation matrices of daily percentage returns and log returns are very close to each other

(ii) Compute the sample covariance matrix of the weekly percentage returns of the indeces, and the corresponding sample corelation matrix.

Compute the sample covariance and correlation matrices for weekly log returns, and compare them with the corresponding matrices for weekly percentage returns.

In [28]:
```python
weekly = value.resample("W").last()
print("Sample covariance matrix of weekly percentage returns:")
pct_week_cov = pd.DataFrame(cov_pct(weekly))
pct_week_cov
```

```
Sample covariance matrix of weekly percentage returns:
```

Out[28]:

	0	1	2	3	4	5	6	7	8
0	0.002042	0.000884	0.001804	0.001245	0.000961	0.001709	0.000552	0.000767	0.000823
1	0.000884	0.001714	0.001267	0.001016	0.000701	0.000983	0.001021	0.000647	0.000555
2	0.001804	0.001267	0.002060	0.001453	0.001156	0.001821	0.000811	0.000873	0.000862
3	0.001245	0.001016	0.001453	0.001358	0.000838	0.001440	0.000724	0.000745	0.000659
4	0.000961	0.000701	0.001156	0.000838	0.000762	0.000998	0.000456	0.000548	0.000531
5	0.001709	0.000983	0.001821	0.001440	0.000998	0.001923	0.000739	0.000804	0.000814
6	0.000552	0.001021	0.000811	0.000724	0.000456	0.000739	0.000747	0.000415	0.000358
7	0.000767	0.000647	0.000873	0.000745	0.000548	0.000804	0.000415	0.000524	0.000426
8	0.000823	0.000555	0.000862	0.000659	0.000531	0.000814	0.000358	0.000426	0.000577

In [29]:
```python
print("Sample correlation matrix of weekly percentage returns:")
pct_week_corr = pd.DataFrame(corr_pct(weekly))
pct_week_corr
```

```
Sample correlation matrix of weekly percentage returns:
```

Out[29]:

	0	1	2	3	4	5	6	7	8
0	1.000000	0.472693	0.879529	0.747518	0.770150	0.862662	0.446917	0.742043	0.757836
1	0.472693	1.000000	0.674238	0.666234	0.613691	0.541635	0.902576	0.683105	0.558006

2	0.879529	0.674238	1.000000	0.868409	0.922650	0.914616	0.653940	0.840487	0.790964
3	0.747518	0.666234	0.868409	1.000000	0.823210	0.891044	0.718928	0.883104	0.744359
4	0.770150	0.613691	0.922650	0.823210	1.000000	0.823958	0.604450	0.867464	0.799954
5	0.862662	0.541635	0.914616	0.891044	0.823958	1.000000	0.616434	0.801593	0.772431
6	0.446917	0.902576	0.653940	0.718928	0.604450	0.616434	1.000000	0.663532	0.545015
7	0.742043	0.683105	0.840487	0.883104	0.867464	0.801593	0.663532	1.000000	0.775686
8	0.757836	0.558006	0.790964	0.744359	0.799954	0.772431	0.545015	0.775686	1.000000

In [30]:
```
print("Sample covariance matrix of weekly log returns:")
log_week_cov = pd.DataFrame(cov_log(weekly))
log_week_cov
```

```
Sample covariance matrix of weekly log returns:
```

Out[30]:

	0	1	2	3	4	5	6	7	8
0	0.002040	0.000869	0.001791	0.001244	0.000944	0.001704	0.000553	0.000752	0.000817
1	0.000869	0.001682	0.001250	0.001008	0.000689	0.000990	0.001009	0.000634	0.000541
2	0.001791	0.001250	0.002030	0.001443	0.001133	0.001814	0.000814	0.000856	0.000855
3	0.001244	0.001008	0.001443	0.001352	0.000826	0.001439	0.000723	0.000735	0.000657
4	0.000944	0.000689	0.001133	0.000826	0.000747	0.000986	0.000454	0.000538	0.000525
5	0.001704	0.000990	0.001814	0.001439	0.000986	0.001926	0.000747	0.000793	0.000812
6	0.000553	0.001009	0.000814	0.000723	0.000454	0.000747	0.000738	0.000411	0.000356
7	0.000752	0.000634	0.000856	0.000735	0.000538	0.000793	0.000411	0.000514	0.000420
8	0.000817	0.000541	0.000855	0.000657	0.000525	0.000812	0.000356	0.000420	0.000574

In [31]:
```
print("Sample correlation matrix of weekly log returns:")
log_week_corr = pd.DataFrame(corr_log(weekly))
log_week_corr
```

```
Sample correlation matrix of weekly log returns:
```

Out[31]:

	0	1	2	3	4	5	6	7	8
0	1.000000	0.469213	0.879804	0.749428	0.764970	0.859807	0.450752	0.734511	0.755260
1	0.469213	1.000000	0.676502	0.668763	0.615029	0.550044	0.905788	0.681816	0.550721
2	0.879804	0.676502	1.000000	0.870898	0.919974	0.917388	0.664763	0.838013	0.792214
3	0.749428	0.668763	0.870898	1.000000	0.821866	0.891789	0.724464	0.882196	0.745531
4	0.764970	0.615029	0.919974	0.821866	1.000000	0.822116	0.611227	0.868739	0.801646
5	0.859807	0.550044	0.917388	0.891789	0.822116	1.000000	0.626950	0.797574	0.771984
6	0.450752	0.905788	0.664763	0.724464	0.611227	0.626950	1.000000	0.667419	0.546301
7	0.734511	0.681816	0.838013	0.882196	0.868739	0.797574	0.667419	1.000000	0.773278
8	0.755260	0.550721	0.792214	0.745531	0.801646	0.771984	0.546301	0.773278	1.000000

Comparisons

- The covariance matrix's values of weekly log returns are smaller than those of percentage returns, which means the variation of log returns is smaller than that of percentage returns
- The first statement does not hold for sample correlation matrices, since correlation is scaled using the standard deviation of each variable

(iii) Compute the sample covariance matrix of the monthly percentage returns of the indeces, and the corresponding sample corelation matrix.

Compute the sample covariance and correlation matrices for monthly log returns, and compare them with the corresponding matrices for monthly percentage returns.

In [32]:
```
monthly = value.resample("ME").last()
print("Sample covariance matrix of monthly percentage returns:")
pct_month_cov = pd.DataFrame(cov_pct(monthly))
pct_month_cov
```

```
Sample covariance matrix of monthly percentage returns:
```

Out[32]:

	0	1	2	3	4	5	6	7	8
0	0.007375	0.004346	0.006127	0.004750	0.004029	0.004591	0.002074	0.003706	0.003395
1	0.004346	0.008465	0.004206	0.003257	0.002215	0.001355	0.003879	0.002702	0.001685
2	0.006127	0.004206	0.005722	0.003902	0.003654	0.003724	0.002018	0.003222	0.002993

3	0.004750	0.003257	0.003902	0.003251	0.002472	0.003135	0.001563	0.002445	0.002023
4	0.004029	0.002215	0.003654	0.002472	0.002414	0.002452	0.001053	0.002044	0.002019
5	0.004591	0.001355	0.003724	0.003135	0.002452	0.003796	0.000653	0.002232	0.001956
6	0.002074	0.003879	0.002018	0.001563	0.001053	0.000653	0.002022	0.001370	0.000884
7	0.003706	0.002702	0.003222	0.002445	0.002044	0.002232	0.001370	0.001960	0.001715
8	0.003395	0.001685	0.002993	0.002023	0.002019	0.001956	0.000884	0.001715	0.001760

In [33]:
```
print("Sample correlation matrix of monthly percentage returns:")
pct_month_corr = pd.DataFrame(corr_pct(monthly))
pct_month_corr
```

```
Sample correlation matrix of monthly percentage returns:
```

Out[33]:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	1.000000	0.549961	0.943076	0.970124	0.954980	0.867778	0.537142	0.974807	0.942299
1	0.549961	1.000000	0.604378	0.620937	0.490081	0.239122	0.937724	0.663404	0.436542
2	0.943076	0.604378	1.000000	0.904589	0.983171	0.798992	0.593265	0.962246	0.943275
3	0.970124	0.620937	0.904589	1.000000	0.882392	0.892545	0.609491	0.968568	0.845865
4	0.954980	0.490081	0.983171	0.882392	1.000000	0.810234	0.476495	0.939730	0.979739
5	0.867778	0.239122	0.798992	0.892545	0.810234	1.000000	0.235619	0.818322	0.756643
6	0.537142	0.937724	0.593265	0.609491	0.476495	0.235619	1.000000	0.688214	0.468561
7	0.974807	0.663404	0.962246	0.968568	0.939730	0.818322	0.688214	1.000000	0.923670
8	0.942299	0.436542	0.943275	0.845865	0.979739	0.756643	0.468561	0.923670	1.000000

In [34]:
```
print("Sample covariance matrix of monthly log returns:")
log_month_cov = pd.DataFrame(cov_log(monthly))
log_month_cov
```

```
Sample covariance matrix of monthly log returns:
```

Out[34]:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	0.007609	0.004026	0.006282	0.004843	0.004102	0.004570	0.001983	0.003745	0.003533
1	0.004026	0.007308	0.003865	0.003065	0.002024	0.001342	0.003414	0.002467	0.001546
2	0.006282	0.003865	0.005742	0.003968	0.003655	0.003700	0.001923	0.003229	0.003076
3	0.004843	0.003065	0.003968	0.003284	0.002496	0.003082	0.001510	0.002453	0.002091
4	0.004102	0.002024	0.003655	0.002496	0.002398	0.002415	0.000999	0.002034	0.002053
5	0.004570	0.001342	0.003700	0.003082	0.002415	0.003592	0.000655	0.002195	0.001979
6	0.001983	0.003414	0.001923	0.001510	0.000999	0.000655	0.001847	0.001299	0.000856
7	0.003745	0.002467	0.003229	0.002453	0.002034	0.002195	0.001299	0.001945	0.001748
8	0.003533	0.001546	0.003076	0.002091	0.002053	0.001979	0.000856	0.001748	0.001827

In [35]:
```
print("Sample correlation matrix of monthly log returns:")
log_month_corr = pd.DataFrame(corr_log(monthly))
log_month_corr
```

```
Sample correlation matrix of monthly log returns:
```

Out[35]:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	1.000000	0.539883	0.950377	0.968718	0.960392	0.874066	0.528889	0.973369	0.947534
1	0.539883	1.000000	0.596625	0.625677	0.483531	0.261835	0.929212	0.654228	0.423146
2	0.950377	0.596625	1.000000	0.913638	0.984970	0.814652	0.590508	0.966026	0.949549
3	0.968718	0.625677	0.913638	1.000000	0.889587	0.897343	0.612968	0.970428	0.853792
4	0.960392	0.483531	0.984970	0.889587	1.000000	0.822758	0.474618	0.941939	0.980916
5	0.874066	0.261835	0.814652	0.897343	0.822758	1.000000	0.254362	0.830165	0.772551
6	0.528889	0.929212	0.590508	0.612968	0.474618	0.254362	1.000000	0.685373	0.466092
7	0.973369	0.654228	0.966026	0.970428	0.941939	0.830165	0.685373	1.000000	0.927153
8	0.947534	0.423146	0.949549	0.853792	0.980916	0.772551	0.466092	0.927153	1.000000

Comparisons

- The differences between covariance matrices and correlation matrices of monthly percentage returns and log returns are very close to each other

(iv) Comment on the differences between the sample covariance and correlation matrices for daily, weekly, and monthly returns.

- The overall entries of sample covariance matrix for monthly returns are larger than those for weekly returns, but with a few exceptions

In [36]:
```
pct_month_cov - pct_week_cov
```

Out[36]:

	0	1	2	3	4	5	6	7	8
0	0.005333	0.003461	0.004323	0.003506	0.003068	0.002882	0.001522	0.002939	0.002572
1	0.003461	0.006751	0.002939	0.002241	0.001514	0.000372	0.002858	0.002055	0.001130
2	0.004323	0.002939	0.003662	0.002449	0.002498	0.001903	0.001206	0.002349	0.002131
3	0.003506	0.002241	0.002449	0.001893	0.001634	0.001695	0.000838	0.001700	0.001364
4	0.003068	0.001514	0.002498	0.001634	0.001651	0.001455	0.000596	0.001496	0.001489
5	0.002882	0.000372	0.001903	0.001695	0.001455	0.001872	-0.000086	0.001427	0.001142
6	0.001522	0.002858	0.001206	0.000838	0.000596	-0.000086	0.001274	0.000955	0.000526
7	0.002939	0.002055	0.002349	0.001700	0.001496	0.001427	0.000955	0.001436	0.001289
8	0.002572	0.001130	0.002131	0.001364	0.001489	0.001142	0.000526	0.001289	0.001183

- All entries of sample covariance matrix for weekly returns are larger than those for daily returns

In [37]:
```
pct_week_cov - pct_day_cov
```

Out[37]:

	0	1	2	3	4	5	6	7	8
0	0.001515	0.000640	0.001287	0.000873	0.000587	0.001217	0.000357	0.000476	0.000530
1	0.000640	0.001413	0.000995	0.000806	0.000493	0.000720	0.000806	0.000479	0.000393
2	0.001287	0.000995	0.001490	0.001056	0.000757	0.001292	0.000589	0.000566	0.000552
3	0.000873	0.000806	0.001056	0.001011	0.000546	0.001032	0.000551	0.000508	0.000429
4	0.000587	0.000493	0.000757	0.000546	0.000438	0.000613	0.000291	0.000305	0.000297
5	0.001217	0.000720	0.001292	0.001032	0.000613	0.001346	0.000518	0.000497	0.000510
6	0.000357	0.000806	0.000589	0.000551	0.000291	0.000518	0.000556	0.000277	0.000224
7	0.000476	0.000479	0.000566	0.000508	0.000305	0.000497	0.000277	0.000303	0.000231
8	0.000530	0.000393	0.000552	0.000429	0.000297	0.000510	0.000224	0.000231	0.000353

- The differences of correlation matrices for different time frequencies are trivial, the entries are close to each other.

In [38]:
```
pct_month_corr - pct_week_corr
```

Out[38]:

	0	1	2	3	4	5	6	7	8
0	-4.440892e-16	7.726875e-02	6.354686e-02	2.226060e-01	0.184830	5.115514e-03	9.022504e-02	2.327642e-01	1.844625e-01
1	7.726875e-02	-3.330669e-16	-6.985945e-02	-4.529689e-02	-0.123610	-3.025133e-01	3.514857e-02	-1.970166e-02	-1.214642e-01
2	6.354686e-02	-6.985945e-02	-2.220446e-16	3.618017e-02	0.060522	-1.156246e-01	-6.067534e-02	1.217600e-01	1.523106e-01
3	2.226060e-01	-4.529689e-02	3.618017e-02	-1.110223e-16	0.059183	1.500974e-03	-1.094366e-01	8.546452e-02	1.015060e-01
4	1.848301e-01	-1.236102e-01	6.052164e-02	5.918259e-02	0.000000	-1.372472e-02	-1.279555e-01	7.226588e-02	1.797854e-01
5	5.115514e-03	-3.025133e-01	-1.156246e-01	1.500974e-03	-0.013725	-1.110223e-16	-3.808148e-01	1.672854e-02	-1.578727e-02
6	9.022504e-02	3.514857e-02	-6.067534e-02	-1.094366e-01	-0.127955	-3.808148e-01	2.220446e-16	2.468153e-02	-7.645425e-02
7	2.327642e-01	-1.970166e-02	1.217600e-01	8.546452e-02	0.072266	1.672854e-02	2.468153e-02	-2.220446e-16	1.479838e-01
8	1.844625e-01	-1.214642e-01	1.523106e-01	1.015060e-01	0.179785	-1.578727e-02	-7.645425e-02	1.479838e-01	1.110223e-16

In [39]:
```
pct_week_corr - pct_day_corr
```

Out[39]:

	0	1	2	3	4	5	6	7	8

0	3.330669e-16	-1.413912e-01	-6.405303e-02	-0.122325	-0.134919	-3.062699e-02	-0.165396	-1.111785e-01	-9.417139e-02
1	-1.413912e-01	2.220446e-16	1.740070e-02	0.016241	-0.055328	-8.978910e-02	0.006976	3.162475e-02	-6.712806e-02
2	-6.405303e-02	1.740070e-02	-2.220446e-16	-0.023197	-0.005558	-6.883288e-03	-0.018408	-2.499738e-02	-7.736786e-02
3	-1.223251e-01	1.624066e-02	-2.319679e-02	0.000000	-0.046106	-2.092739e-02	0.049166	2.923325e-02	-7.976726e-02
4	-1.349186e-01	-5.532774e-02	-5.557575e-03	-0.046106	0.000000	-6.459165e-02	-0.057970	-4.170616e-02	-6.576591e-02
5	-3.062699e-02	-8.978910e-02	-6.883288e-03	-0.020927	-0.064592	-2.220446e-16	-0.046586	-5.912007e-02	-7.006800e-02
6	-1.653963e-01	6.975755e-03	-1.840802e-02	0.049166	-0.057970	-4.658560e-02	0.000000	-6.631285e-03	-1.025162e-01
7	-1.111785e-01	3.162475e-02	-2.499738e-02	0.029233	-0.041706	-5.912007e-02	-0.006631	2.220446e-16	-1.035280e-01
8	-9.417139e-02	-6.712806e-02	-7.736786e-02	-0.079767	-0.065766	-7.006800e-02	-0.102516	-1.035280e-01	2.220446e-16

- The above results hold for both percentage returns and log returns.

Problem 12

Consider a market made of cash and an asset with spot price $\$40$. The continuously compounded risk-free rate is 2%. In four months, the price of the stock will be either $\$45$ or $\$35$.

(i) What is the payoff matrix M in four months?

(ii) Is the market complete, i.e., is the matrix M nonsingular?

(iii) How do you replicate a four months at-the-money call option on this asset, using the cash and the underlying asset?

Solutions:

(i) The value of $1 in 4 months is

$$1 \cdot e^{rt} = e^{0.02 \times \frac{4}{12}} = 1.0067$$

Payoff matrix

$$M = \begin{pmatrix} 1.0067 & 1.0067 \\ 45 & 35 \end{pmatrix}$$

(ii) $\det(M) = 1.0067(35 - 45) = -10.067 \neq 0$

$\therefore M$ is nonsingular.

(iii) The value of call option

$$C(T) = \max(S(T) - K, 0) = \begin{cases} S(T) - K & \text{if } S(T) > K \\ 0 & \text{if } S(T) \leq K \end{cases}$$

$$C(1/3) = \max(S(1/3) - K, 0)$$

state w^1: $S(1/3) = 45$, $C(1/3) = 45 - 40 = 5$

state w^2: $S(1/3) = 35$, $C(1/3) = 0$

$$\therefore C_{1/3} = \begin{pmatrix} 5 & 0 \end{pmatrix}$$

Denote the positions of cash and asset of the replicated portfolio is $\Theta = (\theta_1, \theta_2)$

$$(\theta_1, \theta_2) \begin{pmatrix} 1.0067 & 1.0067 \\ 45 & 35 \end{pmatrix} = (5, 0)$$

$$\therefore \Theta M = C_{1/3} \rightarrow M^t \Theta^t = C_{1/3}^t$$

$$\begin{pmatrix} 1.0067 & 45 \\ 1.0067 & 35 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$$

$$
\begin{aligned}
\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} &= \begin{pmatrix} 1.0067 & 45 \\ 1.0067 & 35 \end{pmatrix}^{-1} \begin{pmatrix} 5 \\ 0 \end{pmatrix} \\
&= \frac{1}{1.0067 \times (35-45)} \begin{pmatrix} 35 & -45 \\ -1.0067 & 1.0067 \end{pmatrix} \begin{pmatrix} 5 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} -17.3835 \\ 0.5 \end{pmatrix}
\end{aligned}
$$

The portfolio replicating the four months ATM call option on the asset is made of a long cash position of $17.3835 and a long 0.5 position in the asset.

Problem 13

In six months, the price of an asset with spot price $\$42$ will be either $\$35, \$38, \$40$, $\$43, \$45$, or $\$50$. Consider a market made of cash, the stock, and a six months at-themoney call and a six months at-the-money put option on one share of the stock. The risk-free interest rate is constant equal to 2%.

(i) What is the payoff matrix in six months?

(ii) Is this market complete?

(iii) Are all the securities in this market non-redundant?

Solutions:

(i)

$$
\begin{aligned}
C(T) &= \max(S(T)-K, 0) \\
P(T) &= \max(K-S(T), 0)
\end{aligned}
$$

State w^1: $S_T = 35\ C_T = 0\ P_T = 7$

State w^2: $S_T = 38\ C_T = 0\ P_T = 4$

State w^3: $S_T = 40\ C_T = 0\ P_T = 2$

State w^4: $S_T = 43\ C_T = 1\ P_T = 0$

State w^5: $S_T = 45\ C_T = 3\ P_T = 0$

State w^6: $S_T = 50\ C_T = 8\ P_T = 0$

The value of $1 in six months would be $e^{0.02 \times \frac{1}{2}} = e^{0.01}$

Payoff matrix

$$
M = \begin{pmatrix} e^{0.01} & e^{0.01} & e^{0.01} & e^{0.01} & e^{0.01} & e^{0.01} \\ 35 & 38 & 40 & 43 & 45 & 50 \\ 0 & 0 & 0 & 1 & 3 & 8 \\ 7 & 4 & 2 & 0 & 0 & 0 \end{pmatrix}
$$

(ii) No, because there're fewer securities than market states, so the payoff matrix is singular. The market is incomplete.

(iii)

$$
\begin{pmatrix} r_1 \\ r_2 + r_4 - r_3 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} e^{0.01} & e^{0.01} & e^{0.01} & e^{0.01} & e^{0.01} & e^{0.01} \\ 42 & 42 & 42 & 42 & 42 & 42 \\ 0 & 0 & 0 & 1 & 3 & 8 \\ 7 & 4 & 2 & 0 & 0 & 0 \end{pmatrix}
$$

r_1 is dependent on $r_2 + r_4 - r_3$. Each row represent an asset in six months $S_{i,1/2}, i = 1:4$

$$
S_{2,1/2} + S_{4,1/2} - S_{3,1/2} = \frac{42}{e^{0.01}} S_{1,1/2}
$$

i.e.

$$
S_{1,1/2} = \frac{e^{0.01}}{42} (S_{2,1/2} + S_{4,1/2} - S_{3,1/2})
$$

Therefore cash is a redundant security.

Reference:

- A Linear Algebra Primer for Financial Engineering

- Backward error in GE; pivoting; iterative refinement