MA304 Multivariate Statistical Analysis

12112627 李乐平

Answer to Question 10.3, Assignment 3

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
# Load and display the dataset
data = pd.read_csv("10_3.csv")
data
```

Out[2]:

	x1	x2	y1	y2
0	191	155	179	145
1	195	149	201	152
2	181	148	185	149
3	183	153	188	149
4	176	144	171	142
5	208	157	192	152
6	189	150	190	149
7	197	159	189	152
8	188	152	197	159
9	192	150	187	151
10	179	158	186	148
11	183	147	174	147
12	174	150	185	152
13	190	159	195	157
14	188	151	187	158
15	163	137	161	130
16	195	155	183	158
17	186	153	173	148
18	181	145	182	146
19	175	140	165	137
20	192	154	185	152
21	174	143	178	147
22	176	139	176	143
23	197	167	200	158
24	190	163	187	150

In [3]:
```python
# Standardize the data
data = (data - data.mean()) / data.std()
```

In [4]:
```python
# Compute the sample covariance matrix
cov = data.cov()
cov
```

Out[4]:

	x1	x2	y1	y2
x1	1.000000	0.734556	0.710752	0.703981
x2	0.734556	1.000000	0.693157	0.708550
y1	0.710752	0.693157	1.000000	0.839252
y2	0.703981	0.708550	0.839252	1.000000

In [5]:
```python
"""
Partition the covariance matrix
Cov([x' y']') =

    sig11 | sig12
    -------------
    sig21 | sig22

"""
l = 2
sig11 = cov.iloc[0:l, 0:l].values
sig12 = cov.iloc[0:l, l:2 * l].values
sig21 = cov.iloc[l:2 * l, 0:l].values
sig22 = cov.iloc[l:2 * l, l:2 * l].values
```

In [6]:
```python
# Calculate the A^(-1/2)
def sqrtinv(mat):
    inv_mat = np.linalg.inv(mat)
    eigenvalues, eigenvectors = np.linalg.eig(inv_mat)
    sqrt_eigenvalues = np.diag(np.sqrt(eigenvalues))
    sqrt_inv_mat = eigenvectors @ sqrt_eigenvalues @ np.linalg.inv(eiger
    return sqrt_inv_mat
```

In [7]:
```python
# Compute the canonical correlations and cononical variable pairs
A = sqrtinv(sig11) @ sig12 @ np.linalg.inv(sig22) @ sig21 @ sqrtinv(sig1
B = sqrtinv(sig22) @ sig21 @ np.linalg.inv(sig11) @ sig12 @ sqrtinv(sig2
C = np.linalg.inv(sig11) @ sig12 @ np.linalg.inv(sig22) @ sig21
D = np.linalg.inv(sig22) @ sig21 @ np.linalg.inv(sig11) @ sig12
A_eig, alpha = np.linalg.eig(A)
B_eig, beta = np.linalg.eig(B)
beta = np.array([beta[:, 1],beta[:, 0]])
B_eig = np.array([B_eig[1], B_eig[0]])
# alpha = sqrtinv(sig11) @ sig12 @ sqrtinv(sig22) @ beta * np.array([1 /
beta = sqrtinv(sig22) @ sig21 @ sqrtinv(sig11) @ alpha * np.array([1 / r
a = sqrtinv(sig11) @ alpha
b = sqrtinv(sig22) @ beta
```

In [8]:
```
a = np.array(a)
b = np.array(b)
```

In [9]:
```
# Verification of the validity
# a' sig11 a == I, b' sig22 b == I

def print_matrix(matrix):
    for row in matrix:
        print("[" + ", ".join(f"{val:.4f}" for val in row) + "]")

print_matrix(a.T @ sig11 @ a)
print_matrix(b.T @ sig22 @ b)
```

```
[1.0000, -0.0000]
[-0.0000, 1.0000]
[1.0000, -0.0000]
[-0.0000, 1.0000]
```

In [10]:
```
# Compute the correlation coefficient between the first pair of canonica
diag_cov = np.sqrt(np.diag(np.diag(cov)))
D1 = diag_cov[0:l, 0:l]
D2 = diag_cov[l:2 * l, l:2 * l]

cov_x_u = np.linalg.inv(D1) @ sig11 @ a[:, 0]
cov_x_v = np.linalg.inv(D1) @ sig12 @ b[:, 0]
cov_y_u = np.linalg.inv(D2) @ sig21 @ a[:, 0]
cov_y_v = np.linalg.inv(D2) @ sig22 @ b[:, 0]
```

In [11]:
```
print(f"""
First canonical correlation:
    ρ1 = {np.sqrt(B_eig[0]): .4f}
First pair of canonical variables:
    u1 = {a[0][0]: .4f}x1 + {a[1][0]: .4f}x2
    v1 = {b[0][0]: .4f}y1 + {b[1][0]: .4f}y2

Second canonical correlation:
    ρ2 = {np.sqrt(B_eig[1]): .4f}
Second pair of canonical variables:
    u2 = {a[0][1]: .4f}x1 + {a[1][1]: .4f}x2
    v2 = {b[0][1]: .4f}y1 + {b[1][1]: .4f}y2

The correlation coefficients between the first pair of canonical variabl
    ρ(x, u1) = {cov_x_u}'
    ρ(x, v1) = {cov_x_v}'
    ρ(y, u1) = {cov_y_u}'
    ρ(y, v1) = {cov_y_v}'
""")
```

```
First canonical correlation:
    ρ1 =  0.7885
First pair of canonical variables:
    u1 =  0.5522x1 +  0.5215x2
    v1 =  0.5044y1 +  0.5383y2

Second canonical correlation:
    ρ2 =  0.0537
Second pair of canonical variables:
    u2 = -1.3664x1 +  1.3784x2
    v2 = -1.7686y1 +  1.7586y2

The correlation coefficients between the first pair of canonical variables and th
e original variables are
    ρ(x, u1) = [0.93528768 0.92715117]'
    ρ(x, v1) = [0.73748174 0.73106604]'
    ρ(y, u1) = [0.75397708 0.75826626]'
    ρ(y, v1) = [0.95620737 0.96164698]'
```