淡江大學統計學系

數據科學組碩一

第三次作業

613890176 魏祺紘

繳交時間:10/26

使用程式 : R、Python

Exercise 3.4

COV(Miles, Weight) = -3732.025

Corr(Miles, Weight) = -0.82

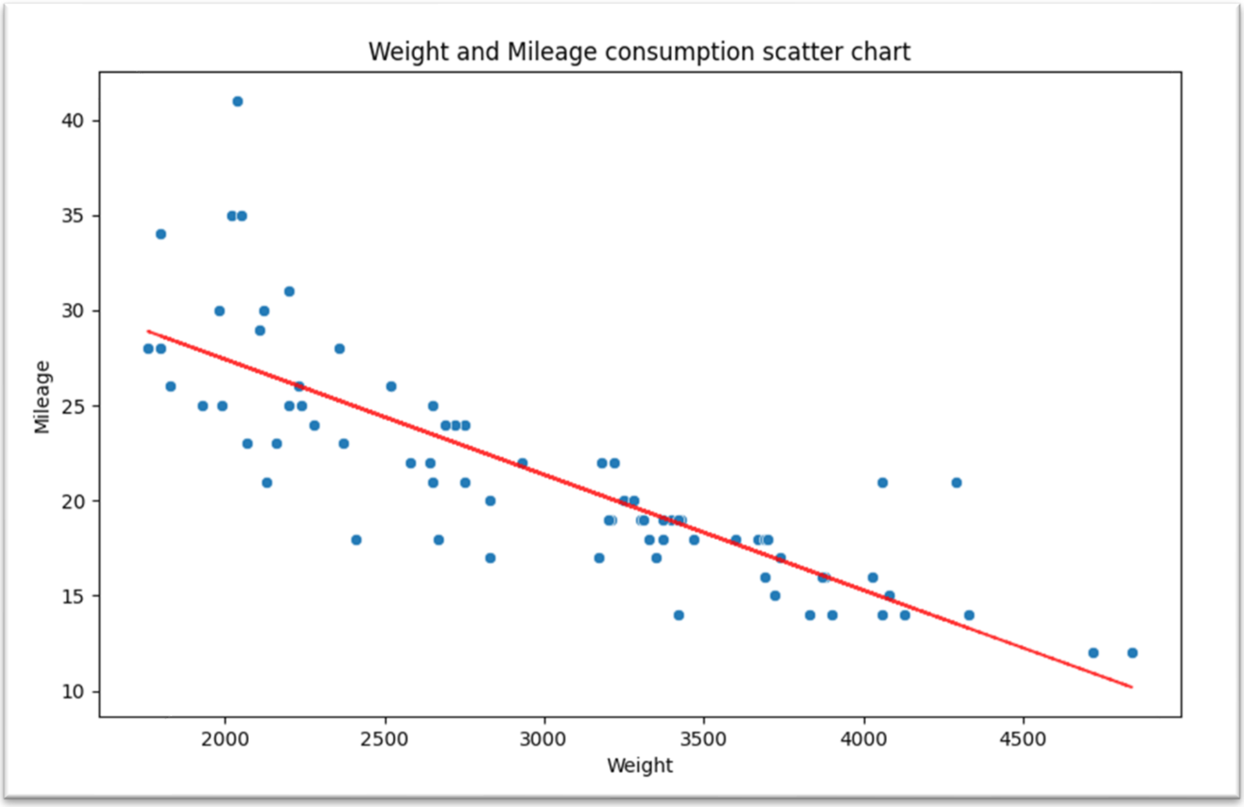

圖 一. 重量與油耗

其共變異數計算結果為 -3732.025，這個負值說明：

結合共變異數、相關係數表示， Mileage (油耗效率，每加侖能行駛的英里數) 和 Weight (重量) 呈現反向變動關係。也就是當車重增加時，油耗效率會下降；反之當車重減輕時，油耗效率會提升！我認為這個結果很合理，因為較重的車需要更多能量來推動 相同油量下，重車會比輕車行駛更短的距離。

Exercise 3.25

	length	left	right	bottom	top	diag
Length	0.1418	0.0314	0.0231	-0.1032	-0.0185	0.0843
Left	0.0314	0.1303	0.1084	0.2158	0.1050	-0.2093
Right	0.0231	0.1084	0.1633	0.2841	0.1300	-0.2405
Bottom	-0.1032	0.2158	0.2841	2.0869	0.1645	-1.0370
Top	-0.0185	0.1050	0.1300	0.1645	0.6447	-0.5496
Diag	0.0843	-0.2093	-0.2405	-1.0370	-0.5496	1.3277

表一. 瑞士銀行真偽鈔共變異數矩陣

進行 Jordan decomposition

$S = \Gamma\Lambda\Gamma^{T}$

$$= \begin{bmatrix} -0.044 & 0.011 & -0.326 & -0.562 & 0.753 & 0.098 \\ 0.112 & 0.071 & -0.259 & -0.455 & -0.347 & -0.767 \\ 0.139 & 0.066 & -0.348 & -0.415 & -0.535 & 0.632 \\ 0.768 & -0.563 & -0.218 & 0.186 & 0.100 & -0.022 \\ 0.201 & 0.653 & -0.557 & 0.450 & 0.102 & -0.349 \\ -0.579 & -0.489 & -0.592 & 0.258 & -0.084 & -0.046 \end{bmatrix} \text{X}$$

$$\begin{bmatrix} 3.000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.936 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.243 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.195 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.085 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.035 \end{bmatrix} \text{X}$$

$$\begin{bmatrix} -0.044 & 0.112 & 0.139 & 0.768 & 0.201 & -0.579 \\ 0.011 & 0.071 & 0.066 & -0.563 & 0.653 & -0.489 \\ -0.326 & -0.259 & -0.348 & -0.218 & -0.557 & -0.592 \\ -0.562 & -0.455 & -0.415 & 0.186 & 0.450 & 0.258 \\ 0.753 & -0.347 & -0.535 & 0.100 & 0.102 & -0.084 \\ 0.098 & -0.767 & 0.632 & -0.222 & -0.394 & -0.046 \end{bmatrix}$$

特徵值都為正的原因：因為共變異數矩陣 S 是正定矩陣(positive definite)

Exercise 3.26

$V = a^T S\, a$

$$= [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1] \text{ X } \begin{bmatrix} 0.1418 & 0.0312 & 0.231 & -01032 & -0.0185 & 0.0843 \\ 0.0314 & 0.1303 & 0.1084 & 0.2158 & 0.1050 & -0.2093 \\ 0.0231 & 0.1084 & 0.1633 & 0.2841 & 0.1300 & -0.2405 \\ -0.032 & 0.2158 & 0.2841 & 2.087 & 0.1645 & -1.0370 \\ -0.0185 & 0.1050 & 0.1300 & 0.1645 & 0.6447 & -0.5496 \\ 0.084 & -0.2093 & -0.2405 & -1.037 & -0.5496 & 1.3 \end{bmatrix} \text{ X } \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$= 2.471859$

V = 2.4719 這個值代表了經過線性轉換後的變異程度，正值表示資料有正向的變異性，反映了當我們將一張鈔票的 6 個測量值相加後，這個"總和"特徵在所有鈔票中的變異程度，這個轉換後的共變異數值提供了一個綜合的衡量標準，有助於我們從整體角度理解鈔票特徵的變異模式。

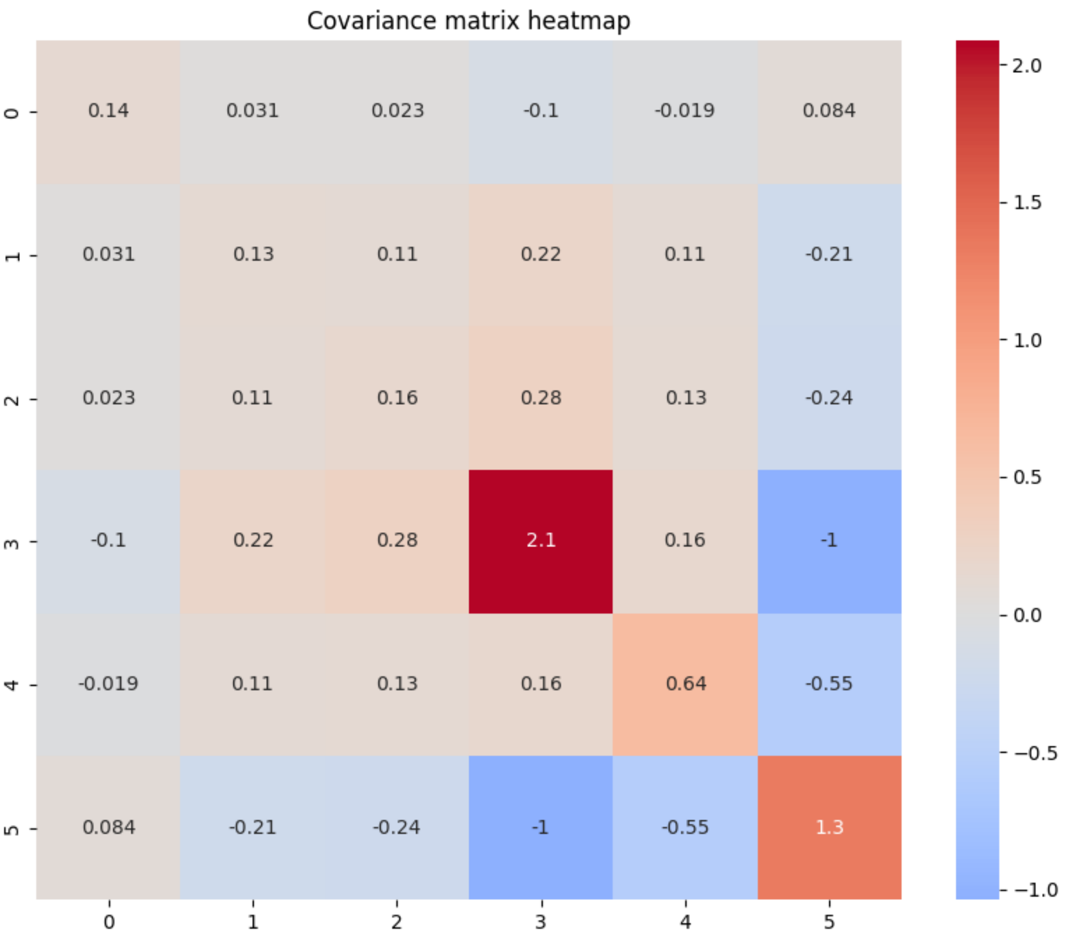

圖 二 . 共變異數熱圖

R 程式碼

```
library(dplyr)

library(ggplot2)


# EX 3.4

car <- read.csv("D:/iphone/iCloudDrive/碩士班/MDA/HW3/carc.csv")


cov(car$Mileage, car$Weight)


#EX 3.25

bank <- read.csv("D:/iphone/iCloudDrive/碩士班/MDA/HW3/swiss bank notes.csv")


# 移除 type 欄位並轉換為矩陣

X <- as.matrix(bank[, -1])   # 移除第一欄的 type 標籤


# 1. 計算共變異數矩陣

S <- cov(X)

print("共變異數矩陣：")

print(S)


# 2. 計算 Jordan 分解
```

```
eigen_decomp <- eigen(S)

eigenvalues <- eigen_decomp$values

eigenvectors <- eigen_decomp$vectors




print("特徵值：")

print(eigenvalues)


print("特徵值 Λ (Lambda):")

print(eigenvalues)


print("特徵向量矩陣 P:")

print(eigenvectors)

t(eigenvectors)


# 驗證分解

# 建立對角矩陣 Λ

Lambda <- diag(eigenvalues)


# 計算 PΛP^T

reconstructed_S <- eigenvectors %*% Lambda %*% t(eigenvectors)
```

```
print("重構的共變異數矩陣 (P Λ P^T):")

print(reconstructed_S)



#EX 3.36

# 定義轉換向量 a

a <- c(1,1,1,1,1,1)



# 計算轉換後的共變異數

transformed_cov <- t(a) %*% S %*% a

print(transformed_cov)
```

python 程式碼

```
import pandas as pd

import numpy as np

from numpy import linalg as LA

import matplotlib.pyplot as plt

import seaborn as sns


#EX 3.4

# 讀取資料集

df = pd.read_csv('D:/iphone/iCloudDrive/碩士班/MDA/HW3/carc.csv')


# 計算共變異數和 相關係數

covariance = df.loc[:, 'Mileage'].cov(df.loc[:, 'Weight'])

correlation = df.loc[:, 'Mileage'].corr(df.loc[:, 'Weight'])


print(f"\n 共變異數：{covariance:.2f}")

print(f"相關係數：{correlation:.2f}")


# 繪製散佈圖

plt.figure(figsize=(10, 6))
```

```python
sns.scatterplot(data=df, x='Weight', y='Mileage')

plt.title('Weight and Mileage consumption scatter chart')

plt.xlabel('Weight')

plt.ylabel('Mileage')


# 添加趨勢線

x = df['Weight']

y = df['Mileage']

z = np.polyfit(x, y, 1)

p = np.poly1d(z)

plt.plot(x, p(x), "r--", alpha=0.8)


plt.savefig('mileage_weight_relationship.png')

plt.close()


#EX 3.25

# 讀取數據

data = pd.read_csv('D:/iphone/iCloudDrive/碩士班/MDA/HW3/swiss bank notes.csv')


# 只選擇數值型特徵（排除 'type' 欄位）

features = ['length', 'left', 'right', 'bottom', 'top', 'diag']
```

```
X = data[features]


# 計算共變異數矩陣
S = np.cov(X.T)


# 計算特徵值和特徵向量
eigenvalues, eigenvectors = LA.eig(S)


# 排序特徵值和特徵向量
idx = eigenvalues.argsort()[::-1]
eigenvalues = eigenvalues[idx]
eigenvectors = eigenvectors[:,idx]


print("共變異數矩陣：")
print(pd.DataFrame(S, columns=features, index=features))
print("\n 特徵值：")
print(eigenvalues)
print("\n 特徵向量：")
print(pd.DataFrame(eigenvectors, columns=[f"PC{i+1}" for i in range(len(features))], index=features))


#EX 3.26
```

```
# 使用向量 a = (1,1,1,1,1,1)ᵀ 進行線性轉換
a = np.ones(len(features))
transformed_var = a.T @ S @ a

print("\n 使用向量 a=(1,1,1,1,1,1)ᵀ 線性轉換後的變異數：")
print(transformed_var)

# 繪製共變異數矩陣熱圖
plt.figure(figsize=(10, 8))
sns.heatmap(pd.DataFrame(S), annot=True, cmap='coolwarm', center=0)
plt.title('Covariance matrix heatmap')
plt.savefig('covariance_heatmap.png')
plt.close()

# 解釋特徵值為何都是正的
print("\n 為什麼特徵值都是正的？")
print("1. 協方差矩陣是對稱矩陣（symmetric matrix）")
print("2. 協方差矩陣是半正定的（positive semi-definite），因為對任何非零向量 v，v^T S v ≥ 0")
print("3. 這表示任何線性組合的變異數都必須是非負的")
print("4. 在實際數據中，由於變數之間存在線性獨立性，協方差矩陣是正定的（positive definite）")
print("5. 正定矩陣的所有特徵值都是正的")
```