Homework 1-2

生工所 李世耀 F08622011

3.

問題分析

以序號總和測試分析臺北站 1961~1980 年、1990~2009 年的 7 月日最高溫，α = 0.05 顯著水準下，判斷兩組 7 月日最高溫紀錄分布是否相同。

序號總和測試(Wilcoxon rank-sum test or Mann–Whitney U test)為非參數化檢定，主要測試兩組樣本分布是否相同，虛無假設為兩組樣本分布相同，因測試統計值只使用序號而非樣本數值，故為非參數化檢定。將兩組樣本混合並排序，若兩組樣本分布相同，我們預期兩組樣本在排序資料中為隨機分布，如圖 1 (a)；若其中一組小於或大於另一組，分布則可能呈現如圖 1 (b)。

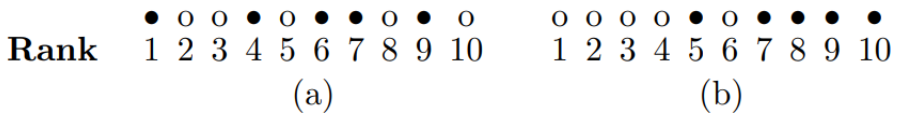

圖 1 樣本來自相同分布與不同分布示意圖 (from: Chris Wild)[1]

計算樣本個數較小的一組所有樣本的序號總和 W，當兩組樣本數足夠大(> 10)時，W 的分布可以使用常態分布近似，將 W 標準化後可得到標準常態分布的測試統計值 z_{rs}。若$|z_{rs}| > z_{1-\alpha/2}$，拒絕虛無假設，兩組樣本分布不相同。

分析方法

給予 1961~1980 年、1990~2009 年兩組 7 月日最高溫樣本不同標籤後，將兩組樣本混合並排序，因兩組樣本個數皆為 620 筆，因此可任選一組計算樣本序號總和 W，利用下式計算測試統計值 z_{rs}，詳細程式請參見附錄。

$$z_{rs} = \begin{cases} \dfrac{W - 0.5 - \mu}{\sigma} & W > \mu \\ 0 \quad \text{if} & W = \mu \\ \dfrac{W + 0.5 - \mu}{\sigma} & W < \mu \end{cases} \qquad \mu = \frac{n(N+1)}{2}, \sigma = \sqrt{\frac{mn(N+1)}{12}}$$

由圖 2 可知數值相等(水平線段)的組數(1134 組)並非少數，則上式標準偏差 σ 需改由下式估計。

$$\sigma = \sqrt{\frac{mn}{N(N-1)}\sum_{k=1}^{N} R_k^2 - \frac{mn(N+1)^2}{4(N-1)}}$$

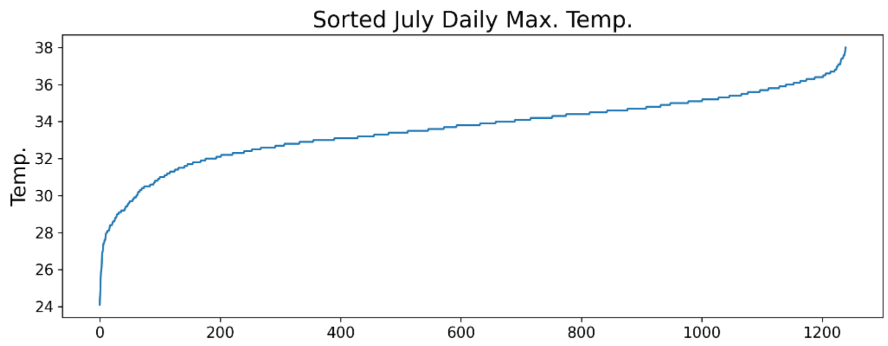

圖 2 兩組 7 月日最高溫混合排序資料呈現(共 1240 筆)

[1] https://www.stat.auckland.ac.nz/~wild/ChanceEnc/index.shtml

結果分析

測試統計值 $z_{rs} = 3.567 > z_{0.975} = 1.96$，故**拒絕虛無假設，兩組 7 月日最高溫樣本分布不相同**，兩組樣本直方圖與盒鬚圖呈現於圖 3，1990~2009 年 7 月日最高溫略高於 1961~1980 年 7 月日最高溫。(使用 `scipy.stats.ranksums()`與 `scipy.stats.mannwhitneyu()`進行檢定，結果相同。)

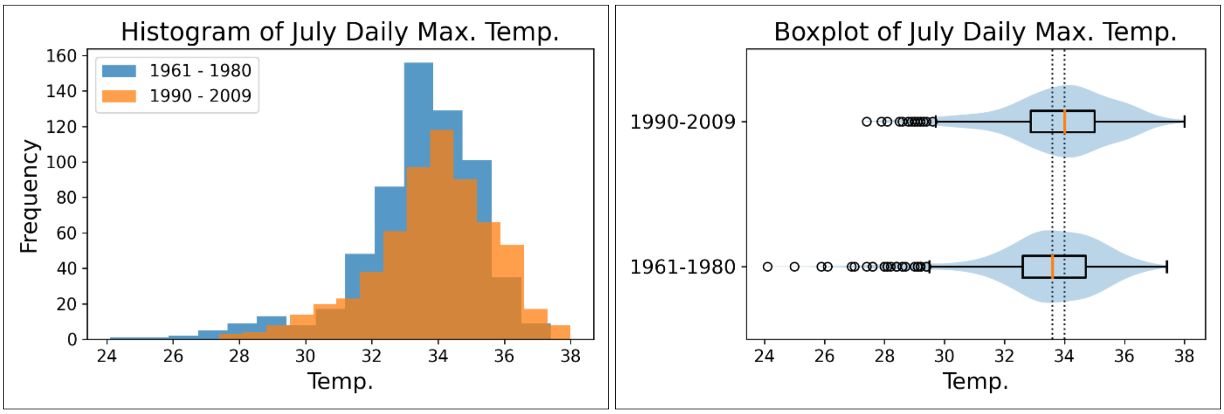

圖 3 1961~1980 年、1990~2009 年 7 月日最高溫直方圖與盒鬚圖

4.

問題分析

樣本資料呈極端值第一型分布，但虛無假設為「隨機樣本呈常態分布」的「假設錯誤」類別問題中，顯著水準 $\alpha = 0.05$ 下，利用蒙地卡羅法，分析在不同樣本數的條件下，卡方檢定發生第二型錯誤的機率及其隨樣本數變化的曲線。

常態分布為統計分析常用的假設分布，而樣本資料是否服從常態分布可以透過卡分檢定測試，但卡方檢定對於樣本數相當敏感，若樣本數太少，在「假設錯誤」的狀況下，卡方檢定則無法有效拒絕虛無假設(第二型錯誤發生機率 β 高)。利用蒙地卡羅法，產生不同樣本數量的資料各 10,000 組，分別進行卡方檢定，可以找出使卡方檢定第二型錯誤發生機率 $\beta < 0.05$ 所需的最小樣本數。常態分布 8 個等機率區間(每個區間累積機率為 0.125)如圖 4 所示。

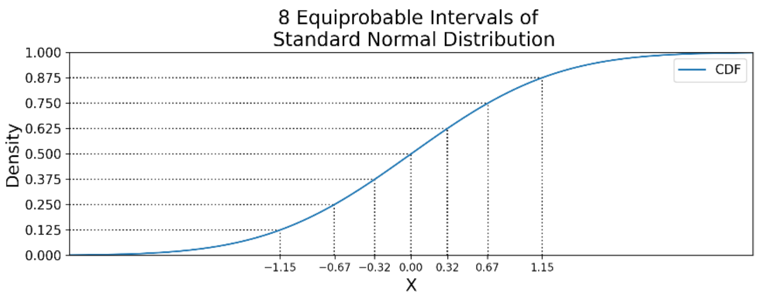

圖 4 標準常態分布等機率區間

分析方法

極端值第一型分布機率密度函數f和累積分布函數F如下式，

$$f(x;\mu,\alpha) = \frac{1}{\alpha} e^{-\left(\frac{x-\mu}{\alpha} + e^{-\frac{x-\mu}{\alpha}}\right)} \qquad F(x;\mu,\alpha) = \int_{-\infty}^{\infty} f(x)dx = e^{-e^{-\frac{x-\mu}{\alpha}}}$$

極端值第一型分布樣本可以透過產生 0 ~ 1 的均勻分布樣本(即累積機率)與利用極端值第一型分布累積分布函數之反函數轉換獲得，如下式與圖 5 所示。

$$y = \frac{x-\mu}{\alpha} \qquad y = -\ln\left(\ln\left(\frac{1}{F(x)}\right)\right)$$

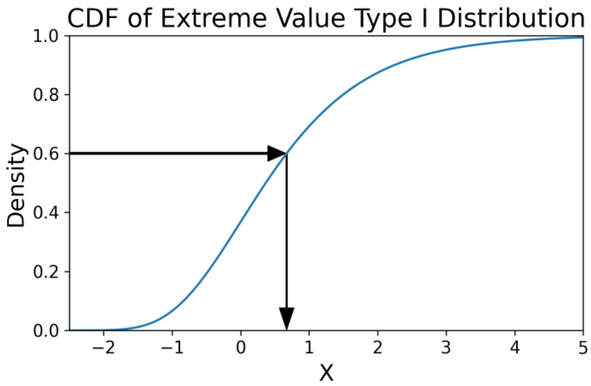

圖 5　極端值第一型分布樣本產生示意圖

假設樣本平均值為 0、標準差為 1，利用 `scipy.stats.uniform.rvs()`產生 size = (10000, $8\times n$) (n=2,3,4,...)的均勻分布樣本，再透過 `scipy.stats.gumbel_r.ppf()`(即累積分布函數之反函數)將其轉換為(10000, $8\times n$)的極端值第一型分布樣本，對不同樣本數($8\times n$)的樣本進行 10,000 次卡方檢定，並計算第二型錯誤發生機率，詳細程式請參見附錄。

$$\beta=\frac{10{,}000\text{ 次卡方檢定中不拒絕虛無假設}H_0\text{的次數}}{10000}$$

結果分析

不同樣本數($8\times n$)樣本 10,000 次卡方檢定，第二型錯誤發生機率隨樣本數變化曲線如下圖 6，可以發現在樣本增加初期，第二型錯誤發生機率下降較快，而要使 $\beta<0.05$ 連續 3 次，n 需要增加至 42 或 43，不同參數假設(圖 7)下獲得的結果一致。

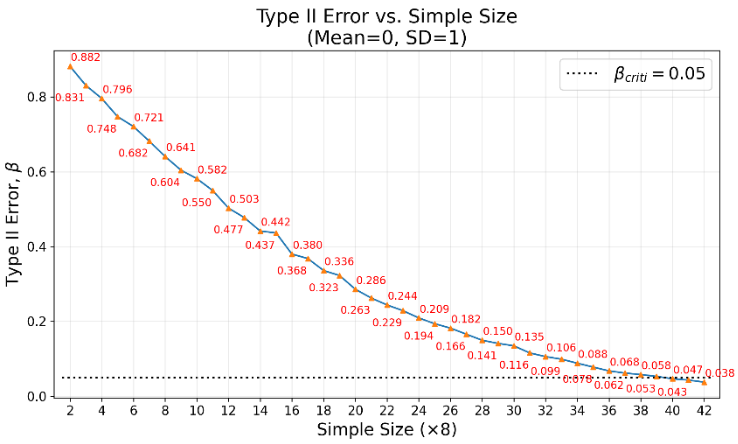

圖 6　第二型錯誤發生機率 β 隨樣本數變化曲線

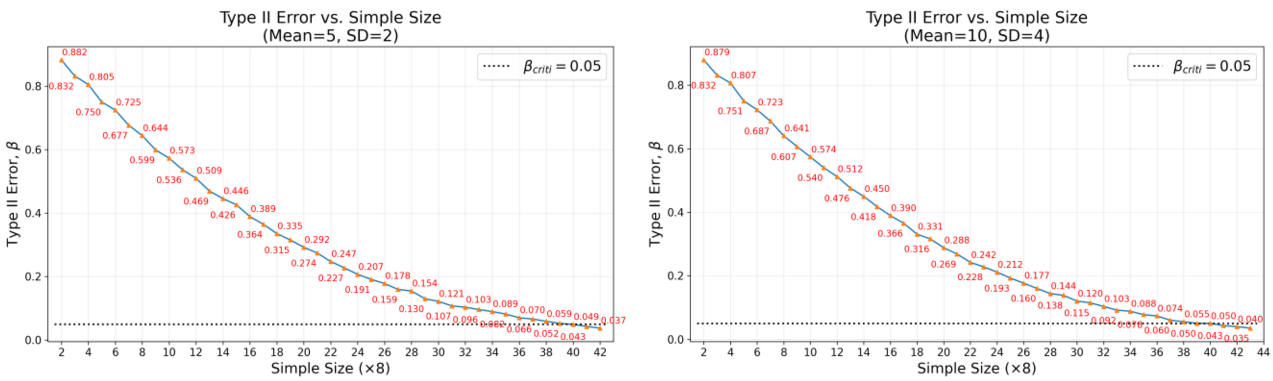

圖 7　不同參數假設第二型錯誤發生機率 β 隨樣本數變化曲線

5.

問題分析

五個溫度測站 O(0,0)、A(20,10)、B(25, 50)、C(-80、30)、D(-20, -60)，日均溫均為常態分佈的隨機變數，期望值均為 30 度、標準偏差均為 3 度，相關係數為距離的函數 $\rho(d) = exp(-d / 30)$。

第 A 小題

某日 O 站和 D 站日均溫分別為 33.3 度和 29.7 度，請「最佳化估計」A、B、C 三站的溫度，計算三個估計值不確定性的變異數，及$\boldsymbol{\rho_{AB|OD}}$、$\boldsymbol{\rho_{AC|OD}}$、$\boldsymbol{\rho_{BC|OD}}$是否與$\boldsymbol{\rho_{AB}}$、$\boldsymbol{\rho_{AC}}$、$\boldsymbol{\rho_{BC}}$相同。

多元常態分布可以下列矩陣形式表示，

$$\boldsymbol{Z} = (Z_1 \quad Z_2 \quad \dots \quad Z_k)^T \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\mu} = (\mu_1 \quad \mu_2 \quad \dots \quad \mu_k)^T$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{12} & \cdots & \sigma_1\sigma_k\rho_{12} \\ \sigma_2\sigma_1\rho_{12} & \sigma_2^2 & \cdots & \sigma_2\sigma_k\rho_{12} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_k\sigma_1\rho_{12} & \sigma_k\sigma_2\rho_{12} & \cdots & \sigma_k^2 \end{bmatrix}$$

給定部分隨機變數觀測值時，可以將 $\boldsymbol{Z}$ 分拆為兩部分，以下列矩陣形式表示，

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \end{bmatrix}, \text{given } \boldsymbol{X} = \boldsymbol{a}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu_X} \\ \boldsymbol{\mu_Y} \end{bmatrix} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma_{XX}} & \boldsymbol{\Sigma_{XY}} \\ \boldsymbol{\Sigma_{YX}} & \boldsymbol{\Sigma_{YY}} \end{bmatrix}$$

$$\boldsymbol{Y_{|X=a}} \sim \mathcal{N}(\boldsymbol{\mu_{Y|X=a}}, \boldsymbol{\Sigma_{Y|X=a}})$$

$$\boldsymbol{\mu_{Y|X=a}} = \boldsymbol{\mu_Y} + \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}}(\boldsymbol{a} - \boldsymbol{\mu_X})$$

$$\boldsymbol{\Sigma_{Y|X=a}} = \boldsymbol{\Sigma_{YY}} - \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}}$$

其中，條件機率的平均值$\boldsymbol{\mu_{Y|X=a}}$為給定 $\boldsymbol{X}$ 時 $\boldsymbol{Y}$ 的最小方均誤差估計量(minimum-mean-square-error estimator)，即給定 $\boldsymbol{X}$ 時 $\boldsymbol{Y}$ 的最佳估計，而$\boldsymbol{\Sigma_{Y|X=a}}$為 $\boldsymbol{X}$ 時 $\boldsymbol{Y}$ 的共變異數矩陣，其對角線上的值即為估計值不確定性的變異數。

相關係數計算公式為下式，給定部分隨機變數觀測值時，剩餘隨機變數共變異數矩陣($\boldsymbol{\Sigma_{Y|X=a}}$)也會改變，因此相關係數也可能會隨之改變。

$$\rho_{Y_iY_j} = \frac{cov(Y_i, Y_j)}{\sigma_{Y_i}\sigma_{Y_j}}$$

第 B 小題

若某日 O 站缺測，擬用 $\boldsymbol{\widehat{T_O} = \sum_i w_i T_i}$，$\boldsymbol{i}$ = A、B、C、D 補遺估計，決定四測站的權重係數值 w_i。

已知$\mu_{T_O|T_i=t_i}$，i = A, B, C, D 為給定四測站觀測值時 O 站的最小方均誤差估計量，可以透過將$\boldsymbol{\mu_{Y|X=a}} = \boldsymbol{\mu_Y} + \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}}(\boldsymbol{a} - \boldsymbol{\mu_X})$轉換為$\widehat{T_O} = \sum_i w_i T_i$，求得 A、B、C、D 四測站的權重係數值 w_i，

$$\boldsymbol{\mu_{Y|X=a}} - \boldsymbol{\mu_Y} = \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}}(\boldsymbol{a} - \boldsymbol{\mu_X}) \quad \boldsymbol{Y} = [0], \boldsymbol{X} = [A \quad B \quad C \quad D]^T$$

$$\widehat{T_O} - \mu = \sum\nolimits_i w_i(T_i - \mu) \quad \Rightarrow \quad w = \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}}$$

$$\Rightarrow \widehat{T_O} = \sum\nolimits_i w_i T_i + \mu\left(1 - \sum\nolimits_i w_i\right)$$

可以發現上式$\widehat{T_O}$的估計式比原始估計式$\widehat{T_O} = \sum_i w_i T_i$多出一項$\mu(1 - \sum_i w_i)$，是因為上式並不是不偏估計(Unbiased estimator)，若加入$\sum_i w_i = 1$(unbiased condition)條件，則與原始估計式相同，但需引入拉格朗日乘數(Lagrange multiplier)求解權重係數 w_i。

分析方法

第 A 小題

O、A、B、C、D 五站日均溫平均值皆為 30 度、標準偏差為 3 度，透過$cov(U,V) = \rho_{UV}\sigma_U\sigma_V$計算共變異數矩陣如下式，

$$X = \begin{bmatrix} O \\ D \end{bmatrix} \qquad Y = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix} = \begin{bmatrix} 9.0 & 1.093 & 4.271 & 1.396 & 0.522 \\ 1.093 & 9.0 & 0.613 & 0.171 & 0.245 \\ 4.271 & 0.613 & 9.0 & 2.348 & 0.301 \\ 1.396 & 0.171 & 2.348 & 9.0 & 0.255 \\ 0.522 & 0.245 & 0.301 & 0.255 & 9.0 \end{bmatrix}$$

給定 O、D 兩站日均溫，利用下列三式計算 A、B、C 三站條件機率平均值、共變異數矩震和相關係數矩陣。

$$\boldsymbol{\mu_{Y|X=a}} = \boldsymbol{\mu_Y} + \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}}(\boldsymbol{a} - \boldsymbol{\mu_X})$$

$$\boldsymbol{\Sigma_{Y|X=a}} = \boldsymbol{\Sigma_{YY}} - \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}}$$

$$\rho_{Y_iY_j} = \frac{cov(Y_i, Y_j)}{\sigma_{Y_i}\sigma_{Y_j}}$$

第 B 小題

$$X = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \qquad Y = [O] \qquad \Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix} = \begin{bmatrix} 9.0 & 2.348 & 0.301 & 0.613 & 4.271 \\ 2.348 & 9.0 & 0.255 & 0.171 & 1.396 \\ 0.301 & 0.255 & 9.0 & 0.245 & 0.522 \\ 0.613 & 0.171 & 0.245 & 9.0 & 1.093 \\ 4.271 & 1.396 & 0.522 & 1.093 & 9.0 \end{bmatrix}$$

O 站缺測，給定 A、B、C、D 四站觀測值估計 O 站日均溫，利用下式計算各站權重係數(僅使用最小方均誤差條件)。

$$\widehat{T_O} - \mu = \sum_i w_i(T_i - \mu) \quad \Rightarrow \quad w = \boldsymbol{\Sigma_{YX}\Sigma_{XX}^{-1}}$$

結果分析

第 A 小題

A、B、C 三站最佳估計、估計不確定性變異數(共變異數矩陣對角線值)、相關係數等計算結果如下式，其中$\rho_{Y|X}$與ρ_Y並不相同。

$$m_{Y|X} = \begin{bmatrix} 31.56 \\ 30.51 \\ 30.18 \end{bmatrix} \qquad \Sigma_{Y|X} = \begin{bmatrix} 6.972 & 1.685 & 0.051 \\ 1.685 & 8.783 & 0.174 \\ 0.051 & 0.174 & 8.966 \end{bmatrix} \qquad \rho_{Y|X} = \begin{bmatrix} 1.0 & 0.215 & 0.006 \\ 0.215 & 1.0 & 0.02 \\ 0.006 & 0.02 & 1.0 \end{bmatrix}$$

$$\rho_Y = \begin{bmatrix} 1.0 & 0.261 & 0.033 \\ 0.261 & 1.0 & 0.028 \\ 0.033 & 0.028 & 1.0 \end{bmatrix}$$

第 B 小題

A、B、C、D 各站權重係數(僅使用最小方均誤差條件)計算結果如下式，權重和不等於 1。

$$w_i = \begin{bmatrix} 0.4587 \\ 0.0327 \\ 0.0393 \\ 0.0886 \end{bmatrix}$$

Homework 1-2 附錄

November 11, 2021

[1]:
```
import numpy as np
import pandas as pd
import scipy.stats as ss
from scipy.spatial import distance_matrix
```

1 第三題

[2]:
```
def Wilcoxon_RankSum(x1, x2, correction=True):
    n1, n2 = len(x1), len(x2)
    N = n1 + n2
    Rank = pd.concat([x1, x2], ignore_index=False).rank()
    if n1 < n2:
        W = Rank.loc[x1.index[0]].sum()
        mu = n1 * (N+1) / 2
    else:
        W = Rank.loc[x2.index[0]].sum()
        mu = n2 * (N+1) / 2

    if correction:
        sigma = np.sqrt(n1*n2 * np.sum(Rank**2) / (N*(N-1)) -
                        n1*n2 * (N+1)**2 / (4*(N-1)))
    else:
        sigma = np.sqrt(n1 * n2 * (N+1) / 12)

    if   W > mu: T = (W - 0.5 - mu) / sigma
    elif W < mu: T = (W + 0.5 - mu) / sigma
    else:        T = 0

    print('Statistic T = {:.3f}'.format(T))
    print('Critical Values: +1.96 / -1.96')
```

1.1 Wilcoxon Rank-sum Test by self-defined function

[3]:
```
Data = pd.read_csv('JulyDMax.csv', index_col='Date')
DMax1 = pd.Series(Data.loc[:,'1961':'1980'].values.flatten(), index=[1]*620)
DMax2 = pd.Series(Data.loc[:,'1990':'2009'].values.flatten(), index=[2]*620)
Wilcoxon_RankSum(DMax1, DMax2)
```

```
Statistic T = 3.567
Critical Values: +1.96 / -1.96
```

1.2 Wilcoxon Rank-sum Test in SciPy

[4]:
```
res = ss.ranksums(DMax2, DMax1)
print('Statistic = {:3.3f}'.format(res.statistic))
print('  p-value = {:.5f}'.format(res.pvalue))
```

```
Statistic = 3.566
  p-value = 0.00036
```

1.3 Mann-Whitney U Test in SciPy

[5]:
```
res = ss.mannwhitneyu(DMax2, DMax1)
print('Statistic = {:8.1f}'.format(res.statistic))
print('  p-value = {:.5f}'.format(res.pvalue))
```

```
Statistic = 214683.5
  p-value = 0.00036
```

2 第四題

[6]:
```
chi2 = ss.chi2.ppf(0.95, df=8-3)              # alpha=0.05, df=k(8)-m(2)-1
beta = 1                                       # When beta < 0.05, stop.
n = 2                                          # n = 2,3,4,...
B = []
successive = 0
alpha = np.sqrt(6) / np.pi
scale = alpha * 1                   # Sigma = 1
loc   = 0 - np.euler_gamma * scale #  Mean = 0

while beta >= 0.05 or successive != 3:
    F = ss.uniform.rvs(size=(10000,8*n))
    x = ss.gumbel_r.ppf(F, loc=loc, scale=scale)
    count = 0
    for i in range(10000):
        mean = x[i].mean()
        std  = x[i].std(ddof=1)
        # Equiprobable Intervals
        equiprob = ss.norm.ppf(np.linspace(0, 1, 9), mean, std)
        hist, bin_edges = np.histogram(x[i], bins=equiprob)
        expected = np.ones(8) * n
        statistic, pvalue = ss.chisquare(hist, expected, ddof=2)
        if statistic < chi2: # Do not reject H0. (Type II error)
            count = count + 1
    beta = count / 10000
    B.append(beta)

    if beta < 0.05: successive = successive + 1

    if successive == 3: break
    else: n = n + 1
```

3 第五題

3.1 第 A 小題

[7]:
```
mu = np.ones(5) * 30
sigma = 3
# Coordinates of 5 stations (O-D-A-B-C)
coord = np.array([[0,0], [-20,-60], [20,10], [25,50], [-80,30]])
# Correlation function
corr = lambda d: np.exp(-d/30)
# Distance Matrix of 5 stations (O-D-A-B-C)
dm = distance_matrix(coord, coord)
# Covariance Matrix
cov = corr(dm) * sigma**2
```

$$X = \begin{bmatrix} O \\ D \end{bmatrix} \qquad Y = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix} = \begin{bmatrix} 9.0 & 1.093 & 4.271 & 1.396 & 0.522 \\ 1.093 & 9.0 & 0.613 & 0.171 & 0.245 \\ 4.271 & 0.613 & 9.0 & 2.348 & 0.301 \\ 1.396 & 0.171 & 2.348 & 9.0 & 0.255 \\ 0.522 & 0.245 & 0.301 & 0.255 & 9.0 \end{bmatrix}$$

[8]:
```
obs_OD = np.array([33.3, 29.7])
# Mean of Station A, B & C given obs at O & D
mu_ABC = mu[2:] + cov[2:,:2]@np.linalg.inv(cov[:2,:2])@(obs_OD-mu[:2])
# Covariance of Station A, B & C given obs at O & D
cov_ABC = cov[2:,2:] - cov[2:,:2]@np.linalg.inv(cov[:2,:2])@cov[:2,2:]
corr_ABC = np.eye(3)
for i in range(3):
    for j in range(3):
        if i != j:
            corr_ABC[i,j] = cov_ABC[i,j] / np.sqrt(cov_ABC[i,i]*cov_ABC[j,j])
```

$$\rho_Y = \begin{bmatrix} 1.0 & 0.261 & 0.033 \\ 0.261 & 1.0 & 0.028 \\ 0.033 & 0.028 & 1.0 \end{bmatrix}$$

$$m_{Y|X} = \begin{bmatrix} 31.56 \\ 30.51 \\ 30.18 \end{bmatrix} \qquad \Sigma_{Y|X} = \begin{bmatrix} 6.972 & 1.685 & 0.051 \\ 1.685 & 8.783 & 0.174 \\ 0.051 & 0.174 & 8.966 \end{bmatrix} \qquad \rho_{Y|X} = \begin{bmatrix} 1.0 & 0.215 & 0.006 \\ 0.215 & 1.0 & 0.02 \\ 0.006 & 0.02 & 1.0 \end{bmatrix}$$

3.2 第 B 小題

[9]:
```
# Coordinates of 5 stations (A-B-C-D-O)
coord = np.array([[20,10], [25,50], [-80,30], [-20,-60], [0,0]])
# Distance Matrix of 5 stations (A-B-C-D-O)
dm = distance_matrix(coord, coord)
# Covariance Matrix
cov = corr(dm) * sigma**2
```

$$X = \begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} \qquad Y = [O] \qquad \Sigma = \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma_{YX} & \Sigma_{YY} \end{bmatrix} = \begin{bmatrix} 9.0 & 2.348 & 0.301 & 0.613 & 4.271 \\ 2.348 & 9.0 & 0.255 & 0.171 & 1.396 \\ 0.301 & 0.255 & 9.0 & 0.245 & 0.522 \\ 0.613 & 0.171 & 0.245 & 9.0 & 1.093 \\ 4.271 & 1.396 & 0.522 & 1.093 & 9.0 \end{bmatrix}$$

$$\hat{T_O} - \mu = \Sigma_i w_i (T_i - \mu)$$

$$m_{Y|X} - m_Y = \Sigma_{YX}\Sigma_{XX}^{-1}(x - m_X)$$

$$\Rightarrow w_i = \Sigma_{YX}\Sigma_{XX}^{-1}$$

[10]:
```
weight = cov[4,:4]@np.linalg.inv(cov[:4,:4])
```

$$w_i = \begin{bmatrix} 0.4587 \\ 0.0327 \\ 0.0393 \\ 0.0886 \end{bmatrix}$$