

Question1

我的策略是先觀察 `data1`、`data2` 的類型，發現均為 `array`，且是`[4000,0]`的資料型態。由於 `array` 不能直接修改，所以，我選擇先建立個空的 `list(listdata1、listdata2)`，將 `array` 轉為 `list` 的型態，再用 `for loop` 將每個元素都取絕對值，以 `append` 方式加入空的 `list` 中，最後，把加滿 4000 筆資料的 `listdata1`、`listdata2` 用 `np.array` 的方式，轉回 `array` 的格式。

Question2

`data1_`和 `orig_data1` 都為 `array[4000,0]`的格式，所以若兩者完全相符，則每一個相對位置的元素必須都完全相同，這樣的情況符合 `all` 的使用條件，`all` 是要每一個條件都達成，才會顯示 `True`。所以，我用 `all` 搭配 `if/else` 語法，以判斷 `data1_`和 `orig_data1`，以及 `data2_`和 `orig_data2` 是否完全相同。

Question3

因為若將一個 `array` 轉換成 `set`，會得到不重複元素的集合，所以，用這個方法即可以找出 `data1_`、`data2_`是由哪些不重複元素所組成。

Question4

本題我打算在輸出的時候，表示哪個元素在 `array` 內的數量為多少，所以首先便得從討論要計算哪幾個數字開始。

我選擇使用 `for` 來進行不同種數字在數量上的計算，而上一題已知道用 `set` 可以表示出這個 `array` 由哪幾種不同的數字所組成，但是，`set` 並不支持編號數字取出的功能 (`type error:'set' object does not support indexing`)，所以我又再把 `set` 轉換成 `list`，如我程式碼中的`:list(set(data1_))`，到此已處理完要計算哪幾種數字的問題。

至於要如何計算 `array` 內數字的數量，我選擇使用 `count` 這個函式，因為 `count` 這個函式只支持 `list` 格式，所以我先把 `array` 轉換成為 `list`，如我程式碼內的`:list(data1_)`。

進行至此，搭配 `for` 和 `count` 即可算出 `array` 內各種數字的數量了。

Question 5

要找出哪個數值的頻率最高，`collections` 這個模組的 `most common` 函式剛好符合。在使用 `most common` 的之前，必須先用 `Counter` 對 `list` 進行運算，而使用 `Counter` 可以得到各個不重複元素的數量，所以 `Counter` 也可以解

Question4。若使用 `most_common(前幾名)`這個函式，可以得到`[(最多的物件名稱,數量),(第 2 名多的物件名稱,數量),.....]`，而我使用`.most_common(1)`，所以會得到`[(最多的物件名稱,數量)]`，因為最後的答案只要輸出數量最多的物件，所以我把它取出表示在 `print` 中。(它的位置在`[0][0]`)。

Question 6

最大值函數為 `max()`，最小值函數為 `min()`，用這兩個函式即可求出兩 `array` 的最大與最小值，而全距為最大值-最小值，將全部資料算完後 `print` 出來即可。

Question 7

題目要求用標準差來比較兩個 `array` 的離散程度，我用先 `np.std(array)` 來計算兩者的標準差，再用 `if` 來比大小即可得出何者的標準差較大，以推論何者的離散程度較大。

Question 8

因為題目要進行運算，這和 `numpy` 的功能相符；而要使用代數 `x`，所以也要引進 `sympy`。至於計算梯形求值法的過程，我打算使用 `def` 來精簡重複的程式碼。

我將 `def` 的參數設定為 `n`、`f`、`toPrint`，`n` 為切成 `n` 段、`f` 為函數、`toPrint` 為是否要把結果 `print` 出來的選擇。

首先，若將區間切成 `n` 段，就會有 `n+1` 個端點，為了將每一個端點用 `list` 表示出來，我使用 `np.linspace` 建立 `array`，再把它轉成 `list` 的格式(名稱為 `lista`)。用 `np.linspace` 的寫法為 `(0,1,n+1)`，區間為 `(0,1)`、有 `n+1` 個端點，。

得到每個端點的 `list` 之後，就可以進行函數的運算，首先，我定義 `a=0` 進行運算，之後再把算出來的值加進去。之後，我先扣除 `0`、`1` 兩點函數值的運算，因為其他點帶入公式的函數都為 $2f(x)*(b-a)/2/n$ ，所以，我就用一個 `for loop` 搭配前面做出的端點(在 `lista` 的 `1` 到 `len(lista)-2` 的位置，因為是 `range` 所以要打 `len(lista)-1`)，將這些點在梯形法中的值算出來，加進 `a` 中。接下來，我把 `0`、`1` 兩點的函數算出來，用梯形法的公式： $f(0)*(b-a)/2/n + f(1)*(b-a)/2/n$ 加進去 `a` 中：

最後，我用 `if` 的方式設定是否要 `print(a)`，這樣就把 `def` 的部分處理完了，之所以要設定 `toPrint` 是因為第八題需要 `print(a)` 而第九題不需要 `print(a)`。

之後，我定義 `Tn8(n,f)` 這個函數，將 `toPrint` 設定為 `True`，再定義 `f1(x)=e-x` 和 `f2(x)=sin(x)*e-x` 這兩個函數，帶入 `Tn8` 中計算後即可得到結果。

Question 9

因為題目只要找出當誤差小於 1×10^{-6} 時的最小 `n` 值，並不需要 `print(a)`，所以，我先定義 `Tn9(n,f)`，把前面的 `toPrint` 參數設定為 `False`。

為了確保 $\frac{\text{真值}-\text{估值}}{\text{真值}}$ 都要大於零，我把這個部份代入 `Tn9(n,f)` 以及真實值後，取了絕對值(`abs()`)。

接下來我用 `for loop` 設定 `n` 的 `range` 為 `1` 到 `1000`(如果設定 `0` 的話梯形法會失效)，再用 `if` 判斷誤差是否小於 1×10^{-6} ，如果成立，便 `print` 出 `n` 值後中斷

(break)，成功輸出結果。