



台塑企業
FORMOSA PLASTICS GROUP

AI基礎訓練中級班

一、PYTHON相關操作及實務課程教材

制定部門：總管理處技訓中心
編定日期：2021年6月18日編印
版次：R2

本著作非經著作權人同意，不得轉載、翻印或轉售。

著作權人：台灣塑膠工業股份有限公司
南亞塑膠工業股份有限公司
台灣化學纖維股份有限公司
台塑石化股份有限公司

AI中級班課程項目：

一、PYTHON相關操作及實務

二、資料前置處理

三、指標衡量方法

四、資料視覺化分析(進階)

五、資料預處理(含深度學習網路建模)

課程目的

在本課程中，將介紹Python程式設計，
並使用指標衡量程式範例來說明Python相關
操作及實務。

(一)Python程式設計

(二)指標衡量程式範例

(一) Python程式設計

1. Python的特性	9
2. Python的安裝	10
3. Anaconda的安裝	11
4. Python撰寫規則	12
5. Python特殊資料型別	13
6. 運算子	14
7. 流程控制-判斷結構(Decision Structures)	15
8. 流程控制-迴圈結構(Loop Structures)	16
9. NumPy資料運算	17
10. NumPy統計函式	18
11. NumPy分佈函式範例	19

目 錄

(二) 指標衡量程式範例

1. 創建Pandas資料(手動鍵入).....	21
2. 創建Pandas資料(檔案匯入).....	22
3. 散佈圖繪製.....	23
4. 平均值與標準差.....	24
5. 基本統計指標.....	25
6. 分析結果儲存至檔案.....	26
7. 最小平方法.....	27
8. 參考資料.....	28
9. 本單元需安裝之套件.....	29



(一) Python程式設計

1. Python的特性

- (1) 一種物件導向程式語言
- (2) 不需要宣告變數資料型別
- (3) 簡單、易學、容易撰寫
- (4) 當今AI領域最常用的程式語言之一
- (5) 開源: 在網路上可以找到很多免費的程式碼和相關資源
- (6) 跨平台: 各種主要的作業系統都支援 Python
- (7) 功能強大、容易擴充: 有非常多的第三方函式庫可以使用

Python為一個不需要宣告變數資料型別的物件導向程式語言

2. Python的安裝

Python的安裝除了Python官方網路 (<https://www.python.org>)

所提供的安裝程式外，另一個重要的開發環境就是Anaconda。

(1) Anaconda是目前最受歡迎的Python開發平台。

(2) Anaconda提供超過1000種的Data Science Packages可使用

(3) Anaconda在**Windows**、**Linux**和**MacOS**等不同作業系統環境
都可以安裝

(4) Anaconda提供虛擬環境管理器。

(5) Anaconda對於在安裝、執行及升級複雜的數據科學
(Data Science) 環境上變得簡單快速。

3. Anaconda的安裝

- 請安裝V3.7版（最新版本為V3.8、試教版本為V3.7）

https://repo.anaconda.com/archive/Anaconda3-2020.02-Windows-x86_64.exe

- Windows中的安裝影片（約12分鐘）

簡易安裝步驟

(1)先安裝Anaconda

(2)執行Anaconda Prompt

(3)安裝相關套件

```
pip bioinfokit install tensorflow==2.2.0 seaborn calmap squarify plotly
```

```
pip install opencv-python scikit-learn jieba textblob
```

```
pip install nltk scrapy bs4 pillow requests sympy nose statsmodels
```

4. Python撰寫規則

「#」為Python中的單行註解標記

(1)英文字母大小寫：

A. Python區分英文字母大小寫。

(2)空白：

A. 隱含控制結構

B. 避免額外的空白。

(3)縮排：

A. 縮排所產生的空白

B. 對應程式的控制結構(for, if, ...)

C. 每個縮排層級使用4個空白，

亦可使用 [Tab] 鍵，但不能混合。

(4)範例

九九乘法表

```
allResult = ""
for i in range(1, 10):
    lineResult = ""
    for j in range(1, 10):
        lineResult = lineResult + str(i) + '*' +
            str(j) + '=' + str(i * j) + '\t'
    allResult = allResult + lineResult + '\n'
print(allResult)
```

Python的指令區分大小寫

Python程式中的指令縮排隱含控制結構

Python可用單引號標記字串資料

5. Python特殊資料型別

Array in
NumPy

(1)**list**：表示串列，這是由一連串資料所組成、有順序且可改變內容的序列：

```
A=[1, 3, 2, 4]
```

(2)**tuple**：表示序對，這是由一連串資料所組成、有順序且不可改變內容的序列：

```
B=(1, 'Taipei', 2, 'Taichung')
```

(3)**set**：表示集合，包含沒有順序、沒有重複且可改變內容的多個資料：

```
C={1, 2, 'Taichung', 'Taipei' }
```

(4)**dict**：表示字典，包含沒有順序、沒有重複且可改變內容的多個鍵：

值對 (key: value pair)：

```
D={' name' : ' 王小明', ' ID' : ' A1357' }
```

Data Frame
in Pandas

6. 運算子

(1) 算術運算子

A. + - * /

B. // (整數除法)

7 // 4 == 1

C. %(餘數)

7 % 4 == 3

D. ** (指數)

2 ** 3 == 8

(2) 比較運算子

運算子	語法	說明
>	a > b	若a大於b，就傳回True，否則傳回False。
<	a < b	若a小於b，就傳回True，否則傳回False。
>=	a >= b	若a大於等於b，就傳回True，否則傳回False。
<=	a <= b	若a小於等於b，就傳回True，否則傳回False。
==	a == b	若a等於b，就傳回True，否則傳回False。
!=	a != b	若a不等於b，就傳回True，否則傳回False。

「//」符號為整數除法運算子

「%」符號為餘數運算子

「**」符號為指數運算子，x**0.5等同math.sqrt(x)平方根函式

7. 流程控制-判斷結構(Decision Structures)

(1) 單向if

```
x=2  
y=1  
if x > y:  
    z = x - y  
    print("x比y大", z)
```

(2) 雙向if... else

```
score = 59  
  
if score >= 60:  
    print("及格!")  
else:  
    print("不及格!")
```

(3) 多向if... elif... else

```
score = 80  
  
if score >= 90:  
    print("優等")  
elif score < 90 and score >= 80:  
    print("甲等")  
elif score < 80 and score >= 70:  
    print("乙等")  
elif score < 70 and score >= 60:  
    print("丙等")  
else:  
    print("不及格")
```

Python中沒有「switch」指令

8. 流程控制-迴圈結構(Loop Structures)

```
for var in iterator:  
    statements1  
[else:  
    statements2]
```

◆ 範例 1:

```
for i in range(5):  
    print(i)
```

此迴圈中，
i有5種不同的數值：
0, 1, 2, 3, 4
最大值為 4

◆ 範例 2:

```
list1 = [15, 20, 33, 7, 8]  
sum = 0  
for i in list1:  
    sum = sum + i  
  
print("總和等於", sum)
```


9. NumPy資料運算

Python使用import指令匯入套件

(1) 陣列 (**array**) 是NumPy
套件中的一種list資料結構
，可以用來存放多個資料。

(2) 陣列所存放的資料叫做
元素(element)，而每個
元素有各自的**值(value)**。

(3) 陣列型別的重要屬性：

A. **array.ndim**：陣列的維度。

B. **array.shape**：陣列的形狀。

C. **array.size**：陣列的元素個數。

NumPy的主要資料型別為list

```
>>> import numpy as np
>>> A = np.array([10, 20, 30])
>>> A.ndim
1
>>> A.shape
(3,) # tuple
>>> A.size
3
```

```
>>> A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]])
>>> print(A)
[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]
>>> A.ndim
2
>>> A.shape
(4, 3) # tuple
>>> A.size
12
```

10. NumPy統計函式

- ✓ `amin(a, axis=None)`
- ✓ `amax(a, axis=None)`
- ✓ `nanmin(a, axis=None)`
- ✓ `nanmin(a, axis=None)`
- ✓ `average(a, axis=None, weights=None)`
- ✓ `median(a, axis=None)`
- ✓ `mean(a, axis=None)`
- ✓ `std(a, axis=None)`
- ✓ `var(a, axis=None)`
- ✓ `nanmedian(a, axis=None)`
- ✓ `nanmean(a, axis=None)`
- ✓ `nanstd(a, axis=None)`
- ✓ `nanvar(a, axis=None)`
- ✓ ...

```
>>> score = np.array([[80, 75, 88, 80, 78], [88, 86, 90, 95, 86], [92, 85, 92, 98, 90], [81, 88, 80, 82, 85], [75, 80, 78, 80, 70]])
```

axis=0 -> 同一欄

axis=1 -> 同一列

算術平均

```
>>> np.mean(score, axis = 1)
array([80.2, 89. , 91.4, 83.2, 76.6])
```

加權平均

```
>>> np.average(score, axis = 1, weights = [0.5, 0.2, 0.1, 0.1, 0.1])
array([79.6, 88.3, 91. , 82.8, 76.3])
```

```
>>> np.median(score, axis = 1) # 中位數
array([80. , 88. , 92. , 82. , 78. ])
```

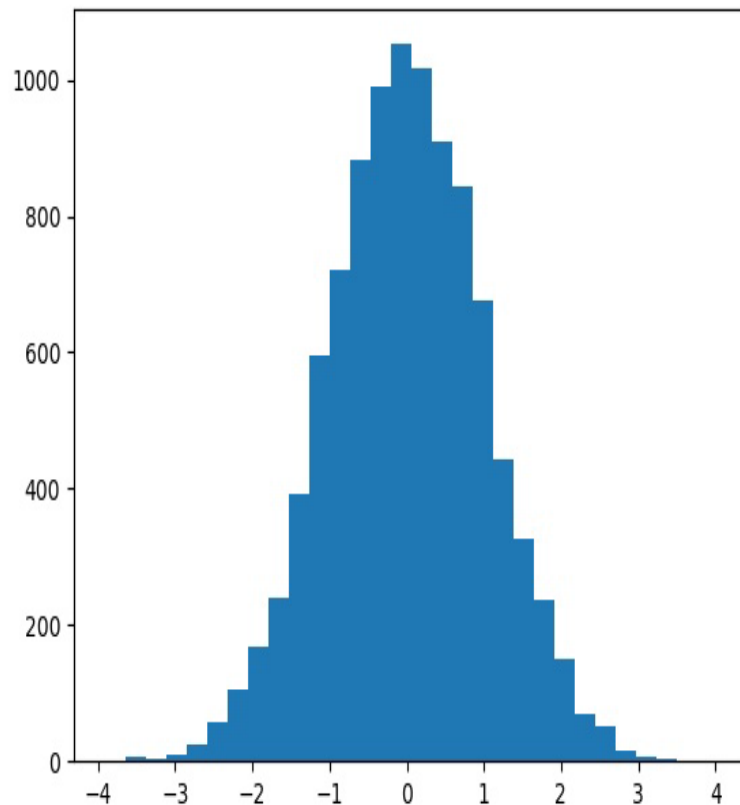
```
>>> np.std(score, axis = 1) # 標準差
array([4.30813185, 3.34664011, 4.1761226 , 2.92574777, 3.77359245])
```

11. NumPy分佈函式範例

標準常態分佈($n=10000$, bins=30)

```
import numpy as np
import matplotlib.pyplot as plt

samples = np.random.normal(size = 10000)
plt.hist(samples, bins = 30)
plt.show()
```





(二)指標衡量程式範例

1. 創建Pandas資料(手動鍵入)

	Time	A	B	C
5	2018/4/12 08:45	49.4	64.1	56.1
6	2018/4/12 08:50	62.6	63.6	54.6
89	2018/4/12 08:55	62.0	63.6	55.7

Data Type:
Dict

```
import pandas as pd
```

```
data = {'Time': ['2018/4/12 08:45', '2018/4/12 08:50 ', '2018/4/12 08:55'],  
        'A': [49.4, 62.6, 62.0],  
        'B': [64.1, 63.6, 63.6],  
        'C': [56.1, 54.6, 55.7]}
```

```
myframe = pd.DataFrame(data, dtype="float64" , index=[5,6,89])
```

Data Frame
in **Pandas**

* pandas的主要資料型別為dict

2. 創建Pandas資料(檔案匯入)

	Time	A	B	C
5	2018/4/12 08:45	49.4	64.1	56.1
6	2018/4/12 08:50	62.6	63.6	54.6
89	2018/4/12 08:55	62.0	63.6	55.7

"C:/AI/Eg1-2.csv"
為Python合法的檔名字串

```
import pandas as pd
```

```
csvFile = "C:/Users/NPLU/Desktop/Eg1-2.csv" # input file name  
myframe = pd.read_csv(csvFile)  
myframe.head()
```

```
myframe.index=[5,6,89] # 改變索引值  
myframe.head()
```

CSV檔案中用逗點符號「,」分隔資料數值
Pandas中的read_csv()函式可將CSV資料檔匯入資料框架

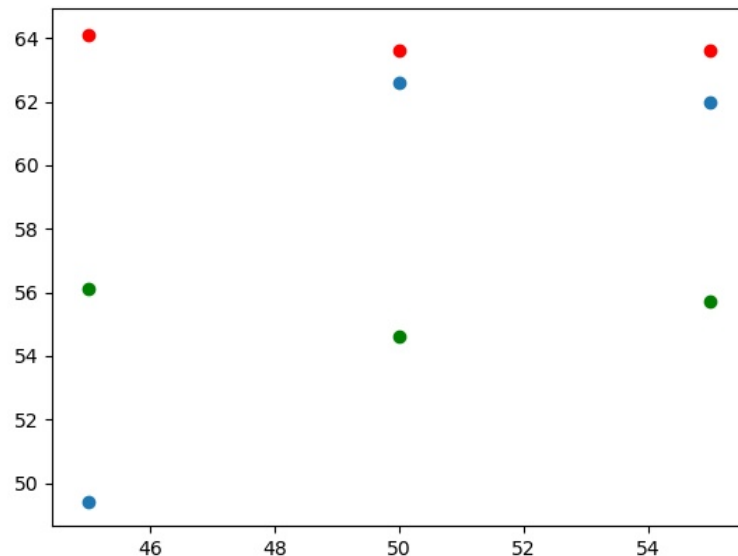
3. 散佈圖繪製

```
import numpy as np  
import matplotlib.pyplot as plt
```

```
x = [45, 50, 55]  
y1 = inFrame.iloc[:,1]  
y2 = inFrame.iloc[:,2]  
y3 = inFrame.iloc[:,3]
```

```
plt.scatter(x, y1)  
plt.scatter(x, y2, color = 'red')  
plt.scatter(x, y3, color = 'green')
```

```
plt.show()
```



	Time	A	B	C
5	2018/4/12 08:45	49.4	64.1	56.1
6	2018/4/12 08:50	62.6	63.6	54.6
89	2018/4/12 08:55	62.0	63.6	55.7

4. 平均值與標準差之函式呼叫

❖ 使用 pandas mean 函式

```
print(inFrame.mean())
```

A 7.453858

B 0.288675

C 0.776745

dtype: float64

❖ 使用 pandas std 函式

```
print(inFrame.std())
```

A 7.453858

B 0.288675

C 0.776745

dtype: float64

	Time	A	B	C
5	2018/4/12 08:45	49.4	64.1	56.1
6	2018/4/12 08:50	62.6	63.6	54.6
89	2018/4/12 08:55	62.0	63.6	55.7

5. 基本統計指標之函式呼叫

❖ 使用 pandas describe 函式

```
outFrame = inFrame.describe()  
print(outFrame)
```

	A	B	C
count	3.000000	3.000000	3.000000
mean	58.000000	63.766667	55.466667
std	7.453858	0.288675	0.776745
min	49.400000	63.600000	54.600000
25%	55.700000	63.600000	55.150000
50%	62.000000	63.600000	55.700000
75%	62.300000	63.850000	55.900000
max	62.600000	64.100000	56.100000

	Time	A	B	C
5	2018/4/12 08:45	49.4	64.1	56.1
6	2018/4/12 08:50	62.6	63.6	54.6
89	2018/4/12 08:55	62.0	63.6	55.7

6. 分析結果儲存至檔案

❖ 儲存outFrame至檔案

Pandas中的to_csv()函式
可將資料框架儲存至CSV檔案

```
outFrame.to_csv("outResult.csv")
```

output file name

	A	B	C
count	3.000000	3.000000	3.000000
mean	58.000000	63.766667	55.466667
std	7.453858	0.288675	0.776745
min	49.400000	63.600000	54.600000
25%	55.700000	63.600000	55.150000
50%	62.000000	63.600000	55.700000
75%	62.300000	63.850000	55.900000
max	62.600000	64.100000	56.100000

	Time	A	B	C
5	2018/4/12 08:45	49.4	64.1	56.1
6	2018/4/12 08:50	62.6	63.6	54.6
89	2018/4/12 08:55	62.0	63.6	55.7

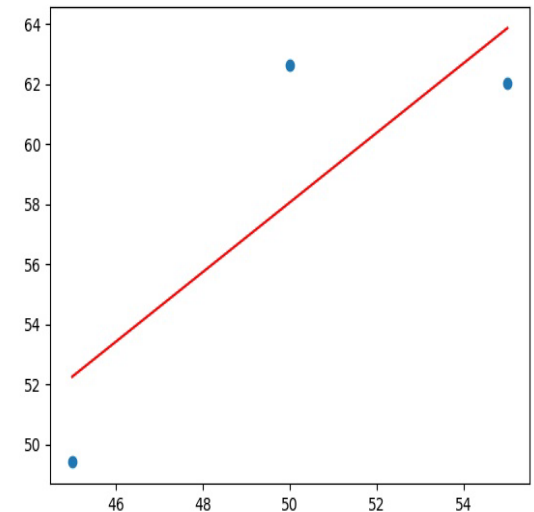
7. 最小平方法 (Least Squares Method)

```
import numpy as np  
import matplotlib.pyplot as plt
```

```
x = [45, 50, 55]  
y = [49.4, 62.6, 62.0]
```

```
cls = np.polyfit(x, y, deg=1)  
p = np.poly1d(cls)  
print(p)  
b0=p[0]  
b1=p[1]
```

```
plt.plot(x, b1 * np.array(x) + b0, color='red')  
plt.scatter(x, y)  
plt.show()
```



	Time	A	B	C
5	2018/4/12 08:45	49.4	64.1	56.1
6	2018/4/12 08:50	62.6	63.6	54.6
89	2018/4/12 08:55	62.0	63.6	55.7

8. 參考資料

(1)Python安裝程序

- 官方版：https://www.youtube.com/watch?v=wqRIKVRUV_k
- Anaconda開發環境：<https://www.youtube.com/watch?v=9LEwsk8dR3o>

(2)Python程式設計

- 陳惠貞，「一步到位！Python程式設計」，旗標，2019。

9. 本單元需安裝之套件

pandas

numpy

matplotlib