Python

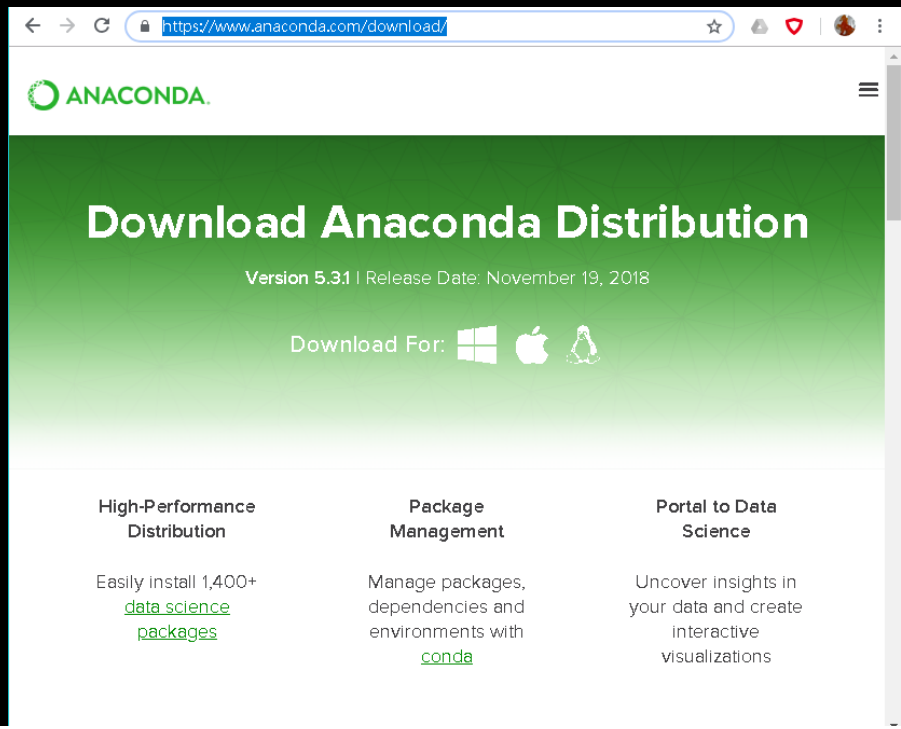
數位創新之風險管理與審計

Y.-H. Huang

2018-12-6

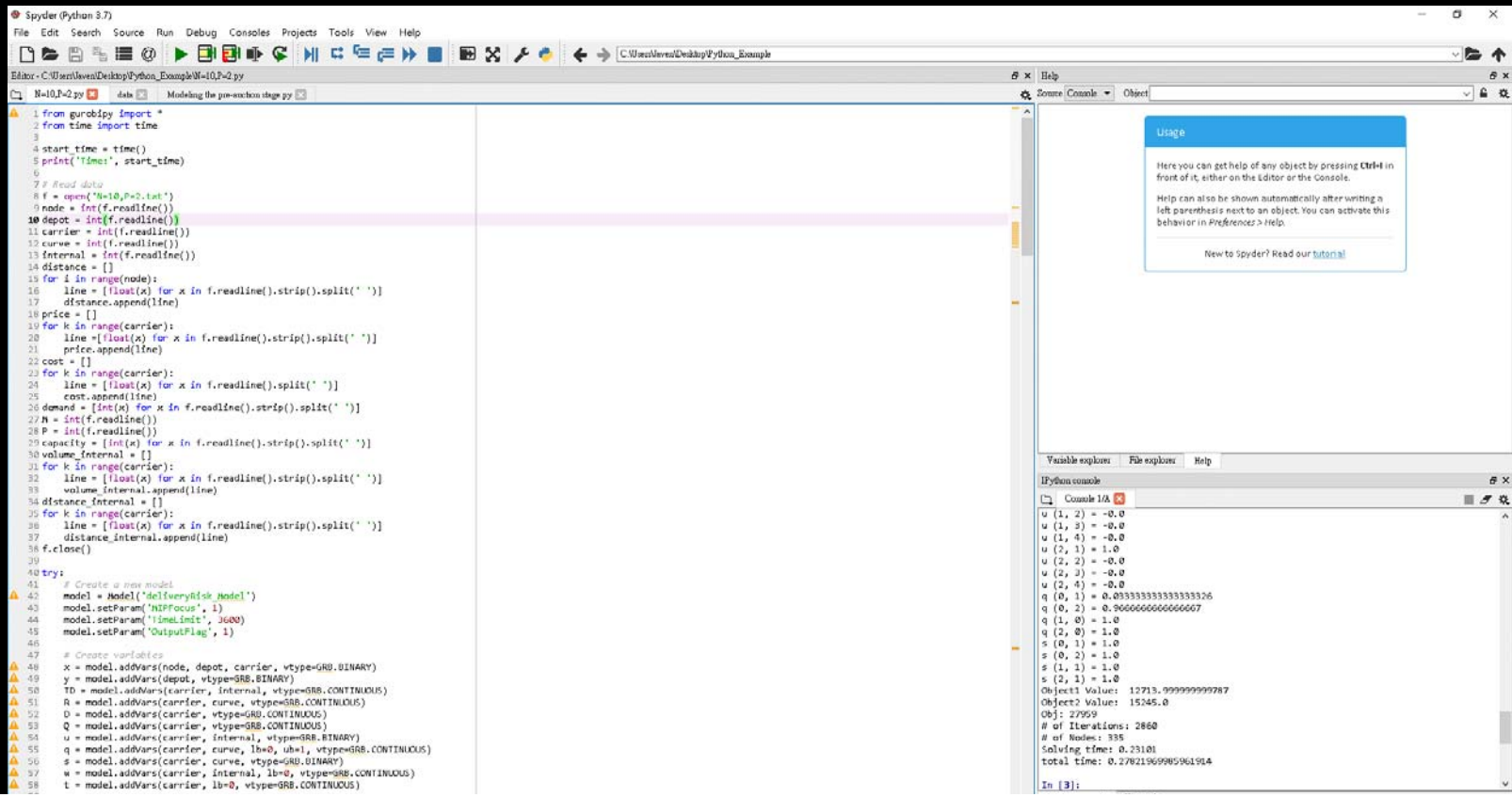
1.1 軟體下載

- **Anaconda**
  - https://www.anaconda.com/download/



1.1 使用軟體

- Spyder (Python 3.7)



1.2 基本Python程式

- 大小寫有別
- # 單行註解
- ’’’ ’’’ : 段落註解
- 每行程式後面不須加任何斷行字元
- 1行程式要寫成多行可以用 \
  sum = 1 + 2 + 3 + \
  4 +5
- 多行程式要寫成1行可以用 ;
  s1 = ‘Apple’; s2 = ‘iPhone’

- 任何 資料 or 變數都是object
  id(x) #印出x的物件ID, id() 函數用于獲取對象的內存地址
- type(x) #印出x的物件type
  isinstance(obj, ClassName) #判斷obj 是否屬於此ClassName
  - print(isinstance("345", str))
  - print(isinstance(345, int))

- PS: 可以用此來學習觀察python裡 object的觀念
  - print(id(s1)) → 2728894511680
  - print(type(s1)) → <class 'str'>

1.3 Assignment & Data Type

- 3種常用的data type: int, float, str
- variable = expression
  - #variable的data type會依後面expression而變化(data type會被後面expression覆蓋)
- Float expression
  - xxx.yyy
  - # 有小數點才表示浮點數, 1.23e20 : 表示1.23*10^20的科學符號
- String expression
  - “ “
  - #表示字串

1.3 Assignment & Data Type

- Example
  - i1 = 1 # i1為int 1
  - f1 = 3.14 # f1為float 3.14, 一定要有小數點才能成為float
  - s1 = 'apple' # s1為string 'apple'
  - s1 = 100 #s1 被改成 int 100

  - var1 = var2 = … = varn = expression #將expression assign 給多個變數
  - i1 = i2 = i3 = 10 #將10 指定給i1, i2, i3

  - var1, var2, …, varn = exp1, exp2, …, expn #同時assign多個變數(data type可不同)
  - i1, f1, s1 = 1, 3.14159, "Apple"

1.4 Data Type轉換

- int(浮點數)
  - #將浮點數轉成整數(無條件捨去小數點)
- float(整數)
  - #將整數轉成浮點數
- str(整數/浮點數)
  - #將整數/浮點數轉成字串
- eval(“string”)
  - #將string轉成 整數/浮點數

1.5 運算子operators

- 依優先順序由高往低如下
  - print(5//3) #整數除法, 小數點無條件捨去, 答案為1
  - print(5%3) #整數取餘數, 答案為2

( )	**括號**
+ -	數值的正負
******	指數
not	Logical operator
*** / // %**	乘 除 整數除法 整數取餘數
+ -	加 減
< <= > >=	大小比較operator
== !=	相等 不等 比較operator
and	Logical operator
or	Logical operator
= += -= *= **= //= %=	指定operator

1.6 載入module

- import A [as x] #載入A module, 也可以同時更名為x
- import A, B #同時載入A & B module

2. Math相關

- **2.1 內建functions**

Function	Meaning
abs(x)	absolute function
max(x1, x2, …, xn)	找最大
min(x1, x2, …, xn)	找最小
pow(a, b)	power function a^b
round(x)	將x小數部分4捨5入(還是浮點數)
round(x,n)	將x小數部分後n位4捨5入

Math module

- import math #載入math module

Function	Meaning
math.pi	常數pi
math.e	常數e
math.ceil(x)	>=x的最小int
math.floor(x)	<=x的最大int
math.exp(x)	自然指數e^x
math.log(x)	自然對數ln(x)
math.log(x,b)	對數b=base
math.degrees(x)	將 x 從radian → degree
math.radians(x)	將 x 從defree → radian
math.sin(x); math.cos(x); math.tan(x); math.asin(x); math.acos(x); math.atan(x);	三角函數

2.3 random module

- import random #載入random module

Function	Meaning
.seed()	用系統時間初始化
.random()	傳回浮點亂數介於 [0,1)
.randint(a,b)	傳回整數亂數介於 [a,b]

3.1 str基本

- 在Python所有的東西都是object, 所以字串也是object, 是屬於str class的object, 建立字串如下
  - s1 = ‘A‘　#表示單一char, 此寫法等同 s1 = str(‘A‘)
  - s2 = ‘Apple’ #表示string, 此寫法等同 s2 = str(‘Apple’)
  - 字串裡面特殊字元

\b	空白	**8**	****	反斜線	**92**
\t	Tab(8格)	9	\’	單引號	39
\n	換行	10	\””	雙引號	34
\f	換頁	12			
\r	回到行首	13			

3.1 str基本

- int, float, str這3種data type其實都是object, 且屬於immutable object, 其內容是不可以被改變的. 為了最佳效率, 若字串內容相同, 則Python會使用同一物件(指向同一地方), 若你 **重新assign 新值給變數** 或是 **改變變數內容值**, Python都會改變 **變數**指向的地方 (似乎沒效率?)
- Example
  - s0 = s1 = 'Apple'
  - print(id(s0) == id(s1)) #True, 指向同一位置
  - s0 = 'Orange'
  - print(id(s0) == id(s1)) #False, 指向不同位置

3.1 str基本

- Example
  - i1 = 10
  - i2 = 10
  - print(id(i1) == id(i2)) #True, 指向同一位置
  - i1 += 1
  - print(id(i1) == id(i2)) #False, 指向不同位置

3.2 字串Operator

- 有許多專為str寫好的operator, 方便使用, 如下所述：

Operator	Meaning
+	可以用 + 來連接char或string, 如下 s1 = “This is Part A.” + “This is Part B.”
*	可以用 * 來重複char或string, 如下 s1 = “Apple.”*3 # s1=”Apple.Apple.Apple.”
[index]	str本身可以看成是1個char array(index from 0), 可以用s1[index]去取的位置index的char. s1 = ‘Apple’ print(s1[0]) #‘A’ print(s1[4]) #‘e’

3.2 字串Operator

Operator	Meaning
[start:end]	分割(slicing) operator, 用此傳回str部分字串 s1[start:] → 傳回字串從位置start以後的字串(s from 0) s1[:end] → 傳回字串從 位置0~位置end-1 的字串 s1[start:end] →傳回字串從 位置start~位置end-1的字串 s1 = 'Apple.' print(s1[2:]) #'ple' print(s1[:2]) #'Ap' print(s1[1:3]) #'pp'
for e1 in str_obj	可將str放在for-loop 的 sequence 位置 e1 會去尋訪str_obj每一個element str_obj也可以為 str[start:end] 格式
in **not in**	檢視字串是否再另一字串內 (或是不再) s1 = 'Welcome' s2 = 'come' s3 = 'Apple' print(s2 in s1) #True print(s3 not in s1) #True

3.2 字串Operator

Operator	Meaning
== !=	比較2字串是否相等 或 不等… 或 大小關係
> >=	s1 = ‘Apple’
< <=	s2 = ’Apple’
	print(s1 == s2) #True, 可以用== 等直接比較字串(這在c是不可以的)

3.3 字串function

- 內建**functions (:type，表示傳回的資料)**
  - print(chr(65))
  - print(ord('A'))

Function	Meaning
ord(ch) : int	傳回ch的ASCII編號
chr(n): char	傳回 ascii編號的char
len(s1): int	傳回字串s1的長度
max(s1): char	傳回字串s1裡面ASCII編號最大的字元
min(s1): char	傳回字串s1裡面ASCII編號最小的字元

3.3 字串function

- 常用**functions**
  - A='ABCABCAAABC'
  - print(A.count("ABC"))

Function	Meaning
.find(s1): int	傳回字串出現s1子字串的最小index(從前面找)
.rfind(s1): int	傳回字串出現s1子字串的最大index(從後面找)
.startswith(s1): bool	若字串內容以s1子字串開頭, 傳回True
.endtswith(s1): bool	若字串內容以s1子字串結尾, 傳回True
.count(s1): int	傳回字串出現s1子字串的次數
.replace(old,new):str	將字串裡面 old子字串 轉成 new子字串
.split(s1): list	將字串裡面依s1切割, 傳回切割好的str list
.strip(): str	將字串前後空白刪除
.lstrip(): str	將字串前面空白刪除
.rstrip(): str	將字串後面空白刪除
.upper(): str	將字串字元全部轉成大寫
.lower(): str	將字串字元全部轉成小寫
.capitalize(): str	將字串第1個字元轉成大寫
.title(): str	將字串每一個word的第1個字元轉成大寫
.swapcase(): str	將字串裡字元小寫變大寫, 大寫變小寫
.center(width): str	再給予寬度下, 將字串往中間靠齊
.ljust(width): str	再給予寬度下, 將字串往左靠齊
.rjust(width): str	再給予寬度下, 將字串往右靠齊

3.3 字串function

- 測試**functions**
  - A='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
  - print(A.isspace())
  - print(A.isalpha())

Function	Meaning
.isspace(): bool	字串是否只有space字元
.isdigit(): bool	字串是否只有數字字元
.isalpha(): bool	字串是否只有字母字元
.isalnum(): bool	字串是否只有字母及數字字元
.isidentifier(): bool	字串是否有Python 保留字
.islower(): bool	字串是否全為小寫字元
.isupper(): bool	字串是否全為大寫字元

3.4 格式化輸出字串

- 數字格式化
  - print("{:.2f}".format(3.1415926));
  - print(“{:5.2f}”.format(3.1415926)); #所使用位置數共5個
  - print("{:6.2f}".format(3.1415926));
  - print("{:7.2f}".format(3.1415926));

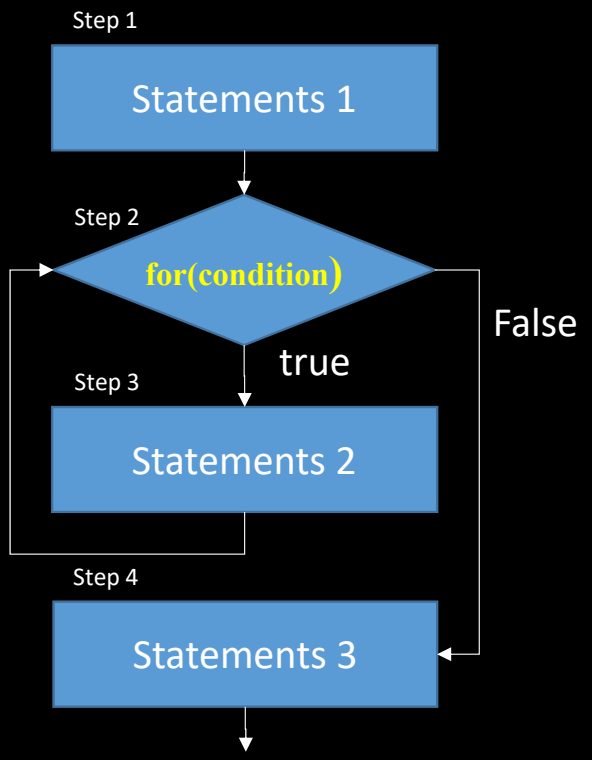
Loop!

Y.-H. Huang

2018-12-13

For-Loop (for迴圈)

- Step 1: 先執行運算式1。
- Step 2: 執行判斷運算式2，
  - 若其值為真（非0），則執行for語句中指定的運算式3 (Step 3)。
  - 若其值為假（為0），則結束迴圈，跳到Step 4。
- Step 3: 執行運算式 3後，再轉回上面判斷運算式2繼續執行。
- Step 4: 迴圈結束，執行 for 語句下面的一個語句。
- 其執行過程可用右圖表示。



For-Loop

- Java

```
int n=100;
for (int i=0; i<n; i++)
{
        //Statement;

System.out.println(n);
}
```

- Python

```
n=100
for i in range(100):
        # Statement
        print(i)
```

For-Loop

- Java

```
int n=100;
for (int i=0; i<n; i++)
{
        //Statement;

System.out.println(n);
}
```

- Python

```
n=100
for i in ragne(0,n,1):
        #Statement
        print(n)
```

For-Loop

- Java

```
int n=100;
for (int i=0; i<n; i++)
{
        /*Statement;

System.out.println(i);*/
        System.out.println(i);
}
```

- Python

```
n=100
for i in range(100):
        """ Statement
        print(i)"""
        print(i)
```

For-Loop

- **Java**

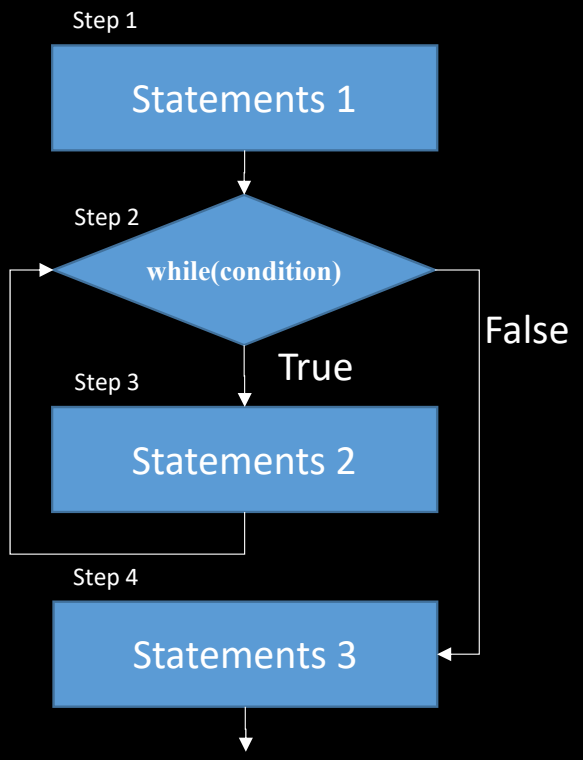
```
for(int i=0;i>100;i--)
{
    System.out.println(i);
}
```

- Python

```
for i in range(1,100,-1):
        print(i)
```

While-Loop (while迴圈)

- Step 1: 先執行運算式1。
- Step 2: 執行判斷運算式2，
  - 若其值為真（非0），則執行for語句中指定的運算式3 (Step 3)。
  - 若其值為假（為0），則結束迴圈，跳到Step 4。
- Step 3: 執行運算式 3後，再轉回上面判斷運算式2繼續執行。
- Step 4: 迴圈結束，執行 for 語句下面的一個語句。
- 其執行過程可用右圖表示。



While-Loop

- Java

```
int total=0;
int i=0;
while (i<=100)
{
        total+=i;
        i++;
}
```

System.***out.println("total="+total);***

- Python

```
i=1
total=0
while i<=100:
    total+=i
    i+=1
print('total=',total)
```

While-Loop

- Java

```
int i=1,j=1;
while(i<10 && j<10)
{
        System.out.println(i+"*"+j+"="
"+i*j);
        i++;
        j+=1;
}
```

- Python

```
j=1; i=1
total=0
while i<10 and j<10:
    print(i,"*",j,"=",i*j);
    i+=1;
    j+=1;
```

While-Loop

- Java

```
int i=1,j=1;
while(i<10 || j<10)
{
        System.out.println(i+"*"+j+
"="+i*j);
        i++;
        j+=2;
}
```

- Python

```
j=1; i=1
total=0
while i<10 or j<10:
    print(i,"*",j,"=",i*j);
    i+=1;
    j+=2;
```

Truth Table

Condition P	Condition Q	P and Q (P && Q)	P or Q (P||Q)
F	F	X	X
F	T	X	O
T	F	X	O
T	T	O	O

Array (陣列)

- **Java**

```
int a1[][]=new int[2][4];
for(int i=0;i<2;i++)
{
        for(int j=0;j<4;j++)
        {
                a1[i][j]=((int)(Math.random()*11));
        }
}
for(int i=0;i<2;i++)
{
        for(int j=0;j<4;j++)
        {
                System.out.print(a1[i][j]+"\t");
        }
        System.out.println();
}
```

- Python

```
import numpy as np
a1=(np.random.randint(0, 11, size=(2,4)))
print(a1)
```

Array (陣列)

Reference:

- https://blog.csdn.net/qq_26948675/article/details/54318917

Code:

```
import numpy
A = numpy.zeros(shape=(3,2),dtype=int) #A=numpy.zeros((3,2),int)
print(A)
```

Result:

```
[[0 0]
 [0 0]
 [0 0]]
```

Array (陣列)

- Reference
  - https://docs.scipy.org/doc/numpy-1.15.1/reference/generated/numpy.random.randint.html

- Code

```
import numpy
X=numpy.random.randint(0, 101, size=(200,5))
print(X)
```

Result

```
[[ 99  43  22  64  52]
 [ 18  38  41  57  45]
…
 [ 81  85  28  59  62]]
```

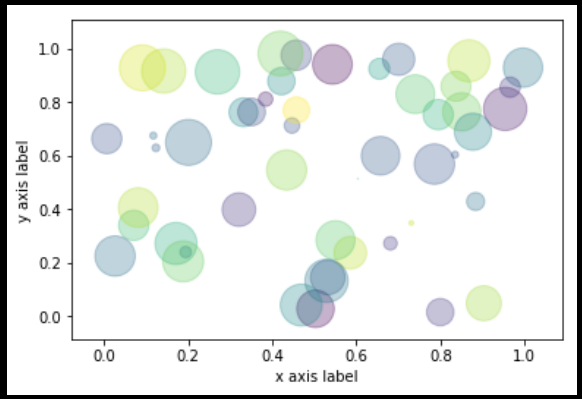
Scatter plot

Reference

- https://jakevdp.github.io/PythonDataScienceHandbook/04.02-simple-scatter-plots.html
- cmap -> https://matplotlib.org/examples/color/colormaps_reference.html

Code

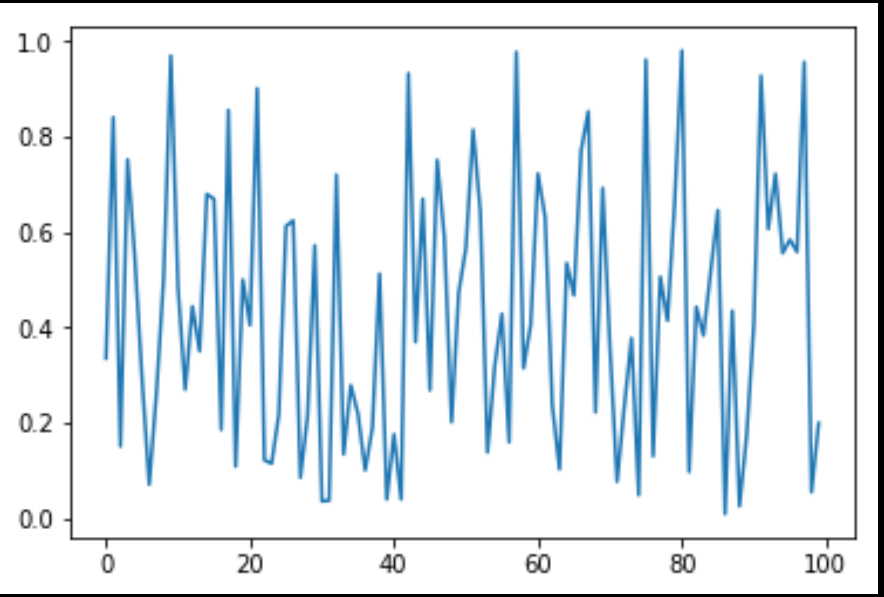
```
import numpy
import matplotlib.pyplot as plt
x = numpy.random.rand(50)
y = numpy.random.rand(50)
colors = np.random.rand(50)
area = 1000 * numpy.random.rand(50)
plt.scatter(x, y, s=area, c=colors, alpha=0.3,  cmap='viridis')
plt.xlabel('x axis label')
plt.ylabel('y axis label')
plt.show()#plt.colorbar()
```



Plot

- Code

```
x=numpy.array(range(0,100))
y = numpy.random.rand(100)
print(x)
plt.plot(x,y)
plt.show()
```



Read (.txt)

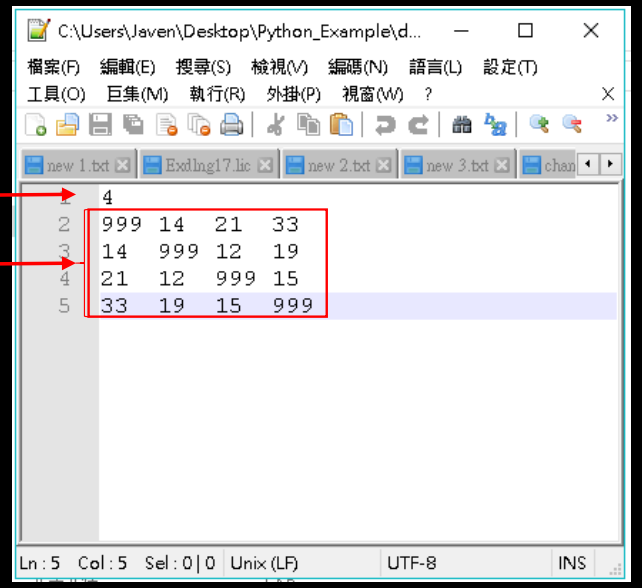
- Code

```
f = open('data_practice_1.txt') #開檔(filename)
n = int(f.readline())   #每次讀取一列值
sigma =numpy.zeros((n,n),int) #使用numpy宣告n*n的二維陣列，其資料型態為int
for i in range(n):
    t=0
    for x in f.readline().strip().split('\t'):
        sigma[i][t]=int(x)
        t=t+1
print(sigma)
```

```
[[999  14  21  33]
 [ 14 999  12  19]
 [ 21  12 999  15]
 [ 33  19  15 999]]
```

n

sigma



Mathematical Problem (Example 1)

$$\text{Min} \quad 3x_1 + 10x_2 + 5x_3 + 2x_4 - 3225x_5 - 3x_1x_3 - 4x_2x_3 + 6x_3x_5$$

$$\text{s.t.} \quad x_1 - x_2 \geq 23,$$

$$x_2 - 2x_3 \geq 25,$$

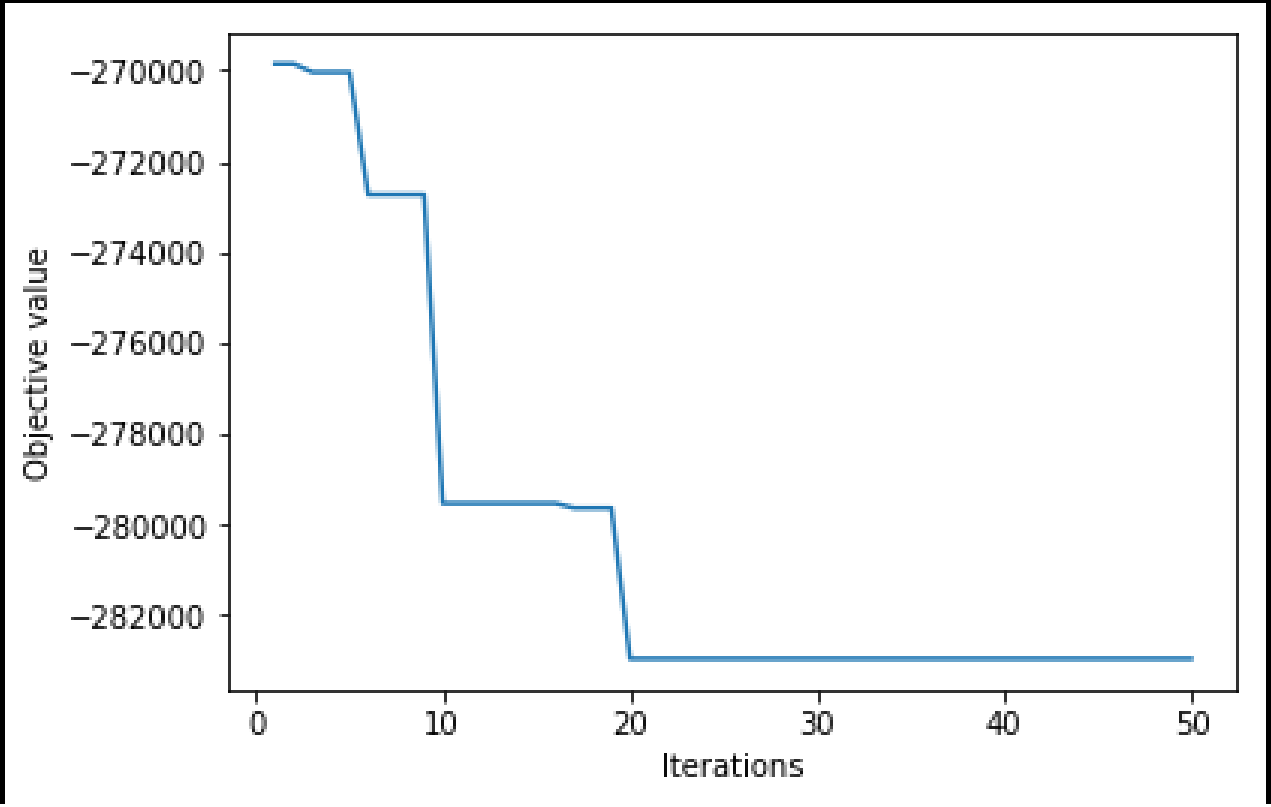
$$x_3 + x_4 \geq 26,$$

$$x_4 - x_5 \geq 12,$$

$$0 \leq x_i \leq 100,\ x_i \in Z^+,\ i = 1,2,3,4,5$$

Solve it! (i.e., Example 1)

- Trend of the solution



Python Code for Example 1

Step 1: Generate *K* two dimensional arrays x[K][5]

```
K=200
X=(np.random.randint(0, 101, size=(K,5)))
Objective=[0]*K
#Declare variables for expressing x_i for i = 1, ... , 5
#x[k][0]=x_1; x[k][1]=x_2 ; x[k][2]=x_3 ; x[k][3]=x_4; x[k][4]=x_5 for k = 1, ... , 200
```

Step 2: Considering four constraints in Example 1,

How to do ?

Incumbent solution

Yours ?

Global Optimal Solution

- Objective value:
  - -284480
- Solution
  - $(x_1, x_2, x_3, x_4, x_5)$=(100,77,26,100,88)

Homework (Your turn!)

- **Project Portfolio Selection Optimization Problem**
- Minimize $\sum_{i=1}^{n} \sum_{j=i}^{n} \sigma_{i,j} x_i x_j$
- s.t. $\sum_{i=1}^{n} \mu_i x_i \geq \rho,$

  $\sum_{i=1}^{n} x_i = h,$

  $0 \leq x_i \leq m$ and $x_i \in Z_+^n$

Dataset

n	10										
m	20										
h	50										
rho	3.7507105										
Sigma		1	2	3	4	5	6	7	8	9	10
	1	0.4523625	0.40467268	0.83315474	0.7824389	0.7993383	0.5205764	0.956171	0.24762513	0.88491124	0.3446189
	2	0	0.73881346	0.32004428	0.44684574	0.67202216	0.07085925	0.8255161	0.29860917	0.2170353	0.7284934
	3	0	0	0.568011	0.6376845	0.4116073	0.75314426	0.4680004	0.9375506	0.20676553	0.04765369
	4	0	0	0	0.6773278	0.55949956	0.21301791	0.8725014	0.50890994	0.24606359	0.8296403
	5	0	0	0	0	0.5437379	0.7770127	0.3006077	0.4805589	0.64890075	0.8665791
	6	0	0	0	0	0	0.49183804	0.8907402	0.01980419	0.5292025	0.18297797
	7	0	0	0	0	0	0	0.2597509	0.4690806	0.16771379	0.35887146
	8	0	0	0	0	0	0	0	0.93764335	0.23369238	0.38195595
	9	0	0	0	0	0	0	0	0	0.37878948	0.3119114
	10	0	0	0	0	0	0	0	0	0	0.74046135
Mu	0.672579	0.8595426	0.75366503	0.8229993	0.73738587	0.70131165	0.5096577	0.9683198	0.6154588	0.86050063	

Press Esc

```
10
20
50
3.7507105
0.4523625     0.40467268    0.83315474    0.7824389     0.7993383     0.5205764     0.956171 0.24762513    0.88491124    0.3446189
0.    0.73881346    0.32004428    0.44684574    0.67202216    0.070859246   0.82551605    0.29860917    0.2170353     0.7284934
0.    0.    0.568011 0.6376845     0.4116073     0.75314426    0.46800038    0.9375506     0.20676553    0.04765369
0.    0.    0.    0.6773278     0.55949956    0.21301791    0.87250143    0.50890994    0.24606359    0.8296403
0.    0.    0.    0.    0.5437379     0.7770127     0.30060774    0.4805589     0.64890075    0.8665791
0.    0.    0.    0.    0.    0.49183804    0.89074016    0.01980419    0.5292025     0.18297797
0.    0.    0.    0.    0.    0.    0.25975093    0.4690806     0.16771379    0.35887146
0.    0.    0.    0.    0.    0.    0.    0.93764335    0.23369238    0.38195595
0.    0.    0.    0.    0.    0.    0.    0.    0.37878948    0.3119114
0.    0.    0.    0.    0.    0.    0.    0.    0.    0.74046135
0.672579 0.8595426     0.75366503    0.8229993     0.73738587    0.70131165    0.5096577     0.96831983    0.6154588     0.86050063
```

Hint (Read .txt)

Code

```
f = open('dataset.txt')
n = int(f.readline())
m = int(f.readline())
h = int(f.readline())
rho = float(f.readline())
sigma =numpy.zeros((n,n),float)
for i in range(n):
    t=0
    for x in f.readline().strip().split('\t'):
        sigma[i][t]=float(x)
        t=t+1
```

```
mu =numpy.zeros((n),float)
t=0
for x in f.readline().strip().split('\t'):
    mu[t]=float(x)
    t=t+1
```