Python for Data Analysis
資料結構及基礎語法

講者:楊翔斌
n07061033@mail.ncku.edu.tw

大綱

1. 基本運算
2. numpy套件介紹
3. pandas套件介紹
4. 資料整理演練-以腎臟病資料為例

基本運算

宣告變數:

變數名稱 = 數值

ex:n3=45

n1,n2=1,10

n4=n5=456

n6=7;sum=8

n7=3.1415#float

sw=True

sw1=False

title= “happy”

title1= ‘happy’

wrong:6m=45

True=100#True為保留字

Array	acos	and	asin	atan	assert
close	break	class	continue	Data	cos
e	def	del	elif	except	else
fabs	exec	exp	float	floor	finally
is	input	int	if	in	import
or	not	open	lambda	log10	log
return	raise	range	pass	print	pi
while	try	type	sin	tan	sqrt
global	for	from	write	zeros	

若為保留字通常會顯示有顏色

刪除變數:

ex:del n3

基本運算

print:

```
In [5]: print(100,"abc",60,sep="/")
100/abc/60
```

```
In [10]: name="abcd"
    ...: scor=61
    ...: print("%s score is %d" %(name,scor))
abcd score is 61
```

%s:字串
%d:整數 %f:浮點數

%4d:列印4字元，若整數少於4位數，則填空白，大於4位數全印。
%4s:列印4字元，若字串少於4位數，則填空白，大於4位數全印。
%4.2f:列印4字元，小數列印2位數，若整數少於4位數，則填空白，小數少於2位數，補0。

用format法表示

```
In [11]: print("{} score is {}".format(name,scor))
abcd score is 61
```

基本運算

運算子:

	意義	範例	結果
+	相加	1+5	6
-	相減	1-5	-4
*	相乘	2*5	10
/	相除	10/5	2
%	取餘數	33%5	3
//	取整除商數	33//5	6
**	次方	2**3	8
==	是否等於	6==7	False
!=	是否不等於	6!=8	True
>	是否大於	8>1	True
<	是否小於	8<1	False
>=	是否大於等於	8>=1	True
<=	是否小於等於	8<=1	False

基本運算

判斷式:

1. if

```
g=10
if(g==10):
    print("good")
```

2. if … else

```
g=11
if(g==10):
    print("good")
else:
    print("bad")
```

3. if … elif … else

```
g=13
if(g==10):
    print("good")
elif(g==11):
    print("bad")
elif(g==12):
    print("better")
else:
    print("not bad")
```

基本運算

1. 元組(tuple):名稱=(元素1,元素2…)，不能修改元素。
   ex: a=(1,"ab",2,9,10)#a[0] 1, a[1] "ab"
   #a[2] 2
2. 串列(list):名稱=[元素1,元素2…]，可修改元素。
   ex: b=[1,"ab",2,9,10]#a[0] 1, a[1] "ab"
   #a[3] 9, a[-1] 10, a[-2] 9
3. 串列與元組可互換:c=list(a) d=tuple(b)
4. range函式:變數=range(起始值,終止值,間隔值)
   li4=range(8,0,-1);list(l4)
   Out: [8, 7, 6, 5, 4, 3, 2, 1]
   l4=range(0,10,2);list(l4)
   Out: [0, 2, 4, 6, 8]
5. for: for 變數 in 串列:
   ……

```
In [76]: for i in range(1,10):
    ...:     print(i,end=",")
    ...:
    ...:
1,2,3,4,5,6,7,8,9,
```

基本運算

List相關函式用法：範例　lists=[2,4,8,6,10]　xs=[3,5]

CODE	意義	範例	結果
lists*n	重複n次	lists*2	[2, 4, 8, 6, 10, 2, 4, 8, 6, 10]
lists[n1:n2]	取n1到n2-1元素	lists[1:5]	[4, 8, 6, 10]
lists[n1:n2:n3]	同上取出間隔n3	lists[1:5:2]	[4, 6]
del lists[n1:n2]	刪n1到n2-1元素	del lists[0:2]	[8, 6, 10]
del lists[n1:n2:n3]	同上刪間隔為n3	del lists[0:4:2]	[4, 6, 10]
len(lists)	算元素總數目	len(lists)	5
min(lists)	取最小值	min(lists)	2
max(lists)	取最大值	max(lists)	10
lists.index(n1)	n1元素之值	lists.index(8)	2
lists.count(n1)	n1元素出現次數	lists.count(2)	1
lists.append(n1)	將n1作為元素加最後	lists.append(100)	[2, 4, 8, 6, 10, 100]
lists.extend(x)	將x中元素作為元素加在串列最後	lists.extend(xs)	[2, 4, 8, 6, 10, 3, 5]
lists.insert(n,n1)	於位置n加入n1元素	lists.insert(2,55)	[2, 4, 55, 8, 6, 10]
lists.pop()	刪除最後一個元素	lists.pop()	[2, 4, 8, 6]
lists.remove(n1)	移除n1元素	lists.remove(2)	[4, 8, 6, 10]
lists.reverse()	反轉	lists.reverse()	[10, 6, 8, 4, 2]
lists.sort()	由小到大排列	lists.sort()	[2, 4, 6, 8, 10]

基本運算

1. 字典:名稱={鍵1:值1,鍵2:值2 …}

ex: dict1={"a":50,"b":40,"c":45,"d":45}

print(dict1["a"]) 50

CODE	意義	範例	結果
len(dict1)	字典元素個數	n=len(dict1)	4
dict1.clear()	移除所有元素	dict2=dict1.clear()	空字典
dict1.copy()	複製字典	dict2=dict1.copy()	{'a': 50, 'b': 40, 'c': 45, 'd': 45}
dict1.get	取得鍵對應的值	n=dict1.get("a")	n=50
鍵 in dict1	檢查鍵是否存在	a="b" in dict1	True
dict1.items()	取得鍵值的組合	dict1.items()	dict_items([('a', 50), ('b', 40), ('c', 45), ('d', 45)])
dict1.key()	取得以鍵為元素之組合	dict1.keys()	dict_keys(['a', 'b', 'c', 'd'])
dict1.values()	取得以值為元素之組合	dict1.values()	dict_values([50, 40, 45, 45])
dict1.setdefault(鍵,值)	若鍵不存在以參數的鍵值建立新元素	dict1.setdefault("cc",100)	100

基本運算

處理檔案及目錄相關套件(作影像辨識很常用!!)

1. os.path

basename()	傳回檔案路徑名稱最後的檔案或路徑名
exists()	檢查檔案是否存在
split()	將檔案路徑名稱做分割
join()	將路徑和檔案名稱做結合

2. glob:針對該路徑找檔名有012.png之檔案並全部列出

```
In [5]: glob.glob(r'E:\temp\images\*012.png')
Out[5]:
['E:\\temp\\images\\00000013_012.png',
 'E:\\temp\\images\\00000032_012.png',
 'E:\\temp\\images\\00000061_012.png',
 'E:\\temp\\images\\00000099_012.png',
 'E:\\temp\\images\\00000116_012.png',
 'E:\\temp\\images\\00000118_012.png',
 'E:\\temp\\images\\00000143_012.png',
 'E:\\temp\\images\\00000181_012.png',
 'E:\\temp\\images\\00000193_012.png',
 'E:\\temp\\images\\00000211_012.png',
 'E:\\temp\\images\\00000231_012.png',
 'E:\\temp\\images\\00000246_012.png',
 'E:\\temp\\images\\00000248_012.png',
 'E:\\temp\\images\\00000250_012.png',
```

numpy套件介紹

numpy套件常用之功能介紹

1. np.array: 下圖為一 array

```
In [37]: b = np.array((0,1,2,3,4,5,9,7,10,11,12,13,14,15,10));b
Out[37]: array([ 0,  1,  2,  3,  4,  5,  9,  7, 10, 11, 12, 13, 14, 15, 10])
```

b[0] b[1] b[2] b[7] b[9] b[14]

2. 用reshape將array變成矩陣

```
In [72]: c=b.reshape(3,5);c
Out[72]:
array([[ 0,  1,  2,  3,  4],
       [ 5,  9,  7, 10, 11],
       [12, 13, 14, 15, 10]])
```

c[1,2]

row index

col. index

Axis=1

Axis=0

	col 0	col 1	col 2	col 3
row 0				
row 1				
row 2				
row 3				
row 4				
row 5				
row 6				
row 7				

3. np.linspace:固定間隔取數字，
   ex: np.linspace(0, 100, 8).astype(int)

pandas套件介紹

pandas套件常用之功能介紹

1. 寫入資料:常用read_table或read_csv

(1)從電腦資料夾讀取:

```
import pandas as pd
age_df = pd.read_csv("E:\\temp\\123.csv")
#可以用/或\\，但注意不能用\
```

(2)從網路上載

```
f="http://archive.ics.uci.edu/ml/machine-learning-databases/00198/Faults.NNA"
data1 = pd.read_table(f,header=None)
data1 = pd.read_csv(f,header=None,sep="\s")
#網路上資料集格式多樣，用適當的函式去讀取資料
#sep可以用"\s" "\t" ","
#詳細參數說明可以參考www.cnblogs.com/datablog/p/6127000.html
```

pandas套件介紹

2. 依照列欄選取資料:

(1)loc()，以label為主體

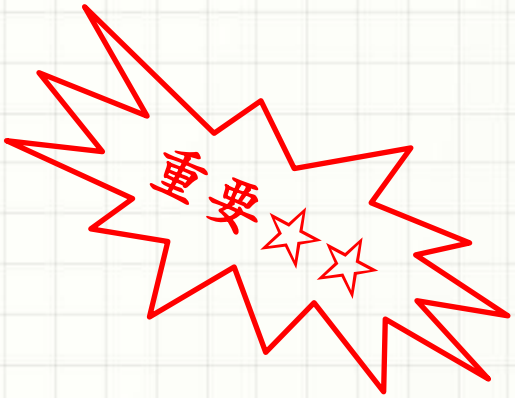
```
data1.loc[0:10, "age"] #表示data1中取欄位為age
                       #列為0~10的資料
data1.loc[:, "age"]    #表示data1中取欄位為age
                       #列未指定則選取全部資料
```

(2)iloc()，以位置為主體

```
data1.iloc[4]          #選取index為4的整列資料
data1.iloc[0:4,1:5]    #選取index為0~3的資料
                       #選取1~4欄的資料
```

```
In [182]: data1.iloc[0:4,1:5]
Out[182]:
   blood pressure specific gravity albumin sugar
0              80             1.02       1     0
1              50             1.02       4     0
2              80             1.01       2     3
3              70            1.005       4     0
```

pandas套件介紹



3. 改變資料格式:

(1)資料會分int、float、str等。

(2)資料集會因為遺漏值或其他因素導致格式不一致，故需要重新定義。

(3)用info()可以查看資料格式，以下指令改格式

data1[[欄名,欄名…]]

=data1[[欄名,欄名…]].astype()

(4)data1[colname].dtype:檢查單一欄之資料格式。

pandas套件介紹

4. 處理遺漏值或空白:

(1)data.info():可以用此計算各欄之非NaN個數。

(2)data = data.replace("null",np.NaN)

將遺漏值統一取代成NaN以利後續處理，""可以放"?"、"Na"、"NA"

(3)data1.isnull().sum():計算各欄的NaN個數

(4)data1.dropna(subset = ["class"]):刪除含有缺失值的特定的列(class)

(5)data1.iloc[:,0].fillna(value=10000):在第0欄中將遺漏值填補成10000

(6)data1.dropna():刪含有缺失值的行(不建議)

(7)data1.dropna(axis=1):刪含有缺失值的欄(不建議)

pandas套件介紹

4. 處理遺漏值或空白:

(8)
```
from sklearn import preprocessing
imr = preprocessing.Imputer(strategy='median')
data1.iloc[:,0] =
imr.fit_transform(pd.DataFrame(data1.iloc[:,0]))
```

利用sklearn利用套件作遺漏值填補，填補的策略是用median來補，用imr.fit_transform將data1中第0欄的NaN補成median。

pandas套件介紹

4. 處理遺漏值或空白:

(9)dropna(thresh = n):保留非NA之個數大於等於n的行。

ex:

```
In [444]: df
Out[444]:
     A    B    C    D     E
0  NaN  2.0  NaN  0.0  10.0
1  3.0  5.0  4.0  NaN   1.0
2  NaN  NaN  NaN  5.0   NaN
3  NaN  NaN  NaN  NaN   NaN
4  4.0  NaN  NaN  5.0   NaN
```

```
In [445]: df.dropna(thresh=2)
Out[445]:
     A    B    C    D     E
0  NaN  2.0  NaN  0.0  10.0
1  3.0  5.0  4.0  NaN   1.0
4  4.0  NaN  NaN  5.0   NaN
```

thresh=2表示若該列非NA個數大於等於2，該列才保留，反之刪除。

```
In [462]: df.dropna(thresh=3)
Out[462]:
     A    B    C    D     E
0  NaN  2.0  NaN  0.0  10.0
1  3.0  5.0  4.0  NaN   1.0
```

pandas套件介紹

5. 啞變數轉換:

(9) get_dummies:將類別變數變成稀疏矩陣。

類別轉稀疏矩陣

	A
1	A_1
2	A_4
3	A_2
4	A_1、A_3
5	A_5
6	A_1
7	A_2

	A_1	A_2	A_3	A_4	A_5
1	1	0	0	0	0
2	0	0	0	1	0
3	0	1	0	0	0
4	1	0	1	0	0
5	0	0	0	0	1
6	1	0	0	0	0
7	0	1	0	0	0

```
ex:raw_data={"sex":["male","female","male",
                    "female","female"]}
   df=pd.DataFrame(raw_data,columns=["sex"])
   df_new = pd.concat([df, df_sex], axis=1)
```

Index	sex	female	male
0	male	0	1
1	female	1	0
2	male	0	1
3	female	1	0
4	female	1	0

資料整理演練–以腎臟病資料為例

讀入資料/去除沒有要分析的資料欄位

↓

檢查資料形式並調整

↓

填補或去除遺漏值

↓

資料分成待預測值、類別變數、連續變數

↓

類別變數：檢視是否有誤植的類別項目
連續變數：對各欄位數值畫分佈圖並處理離群值

↓

檢查資料集是否有數據不平衡問題