Introduction to Data Science

Pandas

TA: 李姵徵

Why pandas?

- 標準表格文件容易載入及保存
  - CSV (Comma-separated Values)
  - TSV (Tab-separated Values)
  - Excel files
  - Database formats
- 序列及表格的索引彈性且方便聚合
- 數值和統計運算快速
- 漂亮且直觀的視覺化

用以對表格型、序列型資料
載入、操作及視覺化的工具

Some Links

- Official Pandas website
  - https://pandas.pydata.org/
- Official documentation
  - https://pandas.pydata.org/docs/
- 10 minutes to Pandas
  - https://pandas.pydata.org/docs/user_guide/10min.html
- Pandas cookbook
  - https://github.com/jvns/pandas-cookbook

Today

- Install and import pandas
- Read data and some basic
- Create Series and DataFrame
- Missing values
- Extract rows, columns, and elements
- Mask and filter
- Operator broadcast and compute statistics
- Group
- Plot

不明白或想多學習怎麼辦？
google大神是你的好朋友！

Dataset – Iris.csv

- 三種鳶尾花 (Species: 150筆資料)
  - Iris-setosa (50筆)
  - Iris-versicolor (50筆)
  - Iris-virginica (50筆)
- 四個數值特徵值
  - SepalLengthCm (萼片長)
  - SepalWidthCm (萼片寬)
  - PetalLengthCm (花瓣長)
  - PetalWidthCm (花瓣寬)

了解資料很重要！
很重要！很重要！

Image Source: https://www.datacamp.com/community/tutorials/machine-learning-in-r

Install and import pandas

CMD

```
>pip install pandas

>pip list
```

pip install <package>	下載package到環境
pip list	列出環境內所有package

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

import <package>	載入package到程式內
%matplotlib inline	圖表內嵌到jupyter

Read data and some basic

```
df = pd.read_csv('Iris.csv')
df.head()
```

read_csv()　讀csv檔
read_excel() 讀excel檔

head() 列出前 5 筆資料
tail()　列出後 5 筆資料

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

DataFrame

Series

Read data and some basic

```
df.columns
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

```
df.describe()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
df.shape
```

```
(150, 6)
```

```
df.dtypes
```

```
Id                int64
SepalLengthCm   float64
SepalWidthCm    float64
PetalLengthCm   float64
PetalWidthCm    float64
Species          object
dtype: object
```

shape() DataFrame的大小 (#row, #column)
dtypes() Series元素的資料類型
describe() 連續型變數的基本統計量
columns() 所有的變數名稱 (column name)

Create Series and DataFrame

```
# like a list
pd.Series([20, 50])
```

```
0    20
1    50
dtype: int64
```

```
# like a list
pd.Series([20, 50], index=['a', 'b'])
```

```
a    20
b    50
dtype: int64
```

```
# like a dict
pd.Series({'a': 20, 'b': 50})
```

```
a    20
b    50
dtype: int64
```

```
# from a dictionary of Series
df_0 = pd.DataFrame({'x': pd.Series([0, 1], index=[0, 1]),
                     'y': pd.Series([2, 3], index=[1, 2]),
                     'z': pd.Series([4, 5, 6])})
df_0
```

	x	y	z
0	0.0	NaN	4
1	1.0	2.0	5
2	NaN	3.0	6

Not-a-Number (NaN)
表示未定義或不可表示的值

```
# from a dictionary of Lists
pd.DataFrame({'x': [0, 1],
              'y': [2, 3],
              'z': [4, 5]})
```

	x	y	z
0	0	2	4
1	1	3	5

Series()	創建一個Series，index若未設定則為[0, …, n]
DataFrame()	創建一個DataFrame

Missing values

```
df.isnull()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
145	False	False	False	False	False	False
146	False	False	False	False	False	False
147	False	False	False	False	False	False
148	False	False	False	False	False	False
149	False	False	False	False	False	False

150 rows × 6 columns

```
df.isnull().sum()
```

```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

isnull()	是否存在遺失值
isnull().sum()	各行遺失值總數

Missing values

```
# from a dictionary of Series
df_0 = pd.DataFrame({'x': pd.Series([0, 1], index=[0, 1]),
                     'y': pd.Series([2, 3], index=[1, 2]),
                     'z': pd.Series([4, 5, 6])})
df_0
```

	x	y	z
0	0.0	NaN	4
1	1.0	2.0	5
2	NaN	3.0	6

```
df_0.isnull()
```

	x	y	z
0	False	True	False
1	False	False	False
2	True	False	False

```
df_0.isnull().sum()
```

```
x    1
y    1
z    0
dtype: int64
```

```
df_0.fillna(-1)
```

	x	y	z
0	0.0	-1.0	4
1	1.0	2.0	5
2	-1.0	3.0	6

```
df_0.dropna()
```

	x	y	z
1	1.0	2.0	5

```
df_0.dropna(axis=1)
```

	z
0	4
1	5
2	6

isnull()	是否存在遺失值
isnull().sum()	各行遺失值總數
fillna(value)	為遺失值補上某個 value
dropna()	將存在遺失值的列刪除，axis若未設定則為0代表row，設定為1代表column

ps. 若想改變DataFrame記得加上inplace = True！

Extract row, column, and element

```
df.loc[0]
```

```
Id                      1
SepalLengthCm           5.1
SepalWidthCm            3.5
PetalLengthCm           1.4
PetalWidthCm            0.2
Species         Iris-setosa
Name: 0, dtype: object
```

```
df.iloc[0]
```

```
Id                      1
SepalLengthCm           5.1
SepalWidthCm            3.5
PetalLengthCm           1.4
PetalWidthCm            0.2
Species         Iris-setosa
Name: 0, dtype: object
```

```
df[4:8]
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa

```
bool_vec = [] # get rows 4~8
for i in range(len(df)):
    if i >= 4 and i <= 8:
        bool_vec.append(True)
    else:
        bool_vec.append(False)
df[bool_vec]
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa

df.loc[label]	藉由label提取row的資訊
df.iloc[loc]	藉由integer location提取row的資訊
df[start:end]	取出從start到end的資訊
df[bool_vec]	取出boolean vector中，True的資訊

Extract row, column, and element

```
df['Species']
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

```
df.Species
```

```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             ...
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

```
df[['SepalLengthCm', 'SepalWidthCm']]
```

	SepalLengthCm	SepalWidthCm
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6
...	...	...
145	6.7	3.0
146	6.3	2.5
147	6.5	3.0
148	6.2	3.4
149	5.9	3.0

150 rows × 2 columns

df[col]	提取一個column的資訊
df.col	提取一個column的資訊
df[[col1, col2]]	提取多個column的資訊

Extract row, column, and element

```
df.loc[0]['Species']
```

```
'Iris-setosa'
```

```
df.iloc[0]['Id']
```

```
1
```

```
df['PetalLengthCm'][0]
```

```
1.4
```

```
df['PetalWidthCm'][0]
```

```
0.2
```

```
df.SepalLengthCm[0]
```

```
5.1
```

```
df.SepalWidthCm[0]
```

```
3.5
```

Select element(s)以下方法皆可
(可先選列再選行，也可先選行再選列)
df.loc[label][col],
df.iloc[loc][col],
df[col][label],
df[col][loc],
df.col[label],
df.col[loc]

Mask and filter

```
df['PetalLengthCm']
```

```
0      1.4
1      1.4
2      1.3
3      1.5
4      1.4
      ...
145    5.2
146    5.0
147    5.2
148    5.4
149    5.1
Name: PetalLengthCm, Length: 150, dtype: float64
```

```
# mask
df['PetalLengthCm'] <= 1.3
```

```
0      False
1      False
2       True
3      False
4      False
       ...
145    False
146    False
147    False
148    False
149    False
Name: PetalLengthCm, Length: 150, dtype: bool
```

```
# filter
df['PetalLengthCm'][df['PetalLengthCm'] <= 1.3]
```

```
2     1.3
13    1.1
14    1.2
16    1.3
22    1.0
35    1.2
36    1.3
38    1.3
40    1.3
41    1.3
42    1.3
Name: PetalLengthCm, dtype: float64
```

```
# filter
df[df['PetalLengthCm'] <= 1.3]
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
2	3	4.7	3.2	1.3	0.2	Iris-setosa
13	14	4.3	3.0	1.1	0.1	Iris-setosa
14	15	5.8	4.0	1.2	0.2	Iris-setosa
16	17	5.4	3.9	1.3	0.4	Iris-setosa
22	23	4.6	3.6	1.0	0.2	Iris-setosa
35	36	5.0	3.2	1.2	0.2	Iris-setosa
36	37	5.5	3.5	1.3	0.2	Iris-setosa
38	39	4.4	3.0	1.3	0.2	Iris-setosa
40	41	5.0	3.5	1.3	0.3	Iris-setosa
41	42	4.5	2.3	1.3	0.3	Iris-setosa
42	43	4.4	3.2	1.3	0.2	Iris-setosa

除了常見的>、<、==、!=，
也可以搭配 and、or、 not

Operator broadcast and compute statistics

```
df['PetalWidthCm']
```

```
0      0.2
1      0.2
2      0.2
3      0.2
4      0.2
      ...
145    2.3
146    1.9
147    2.0
148    2.3
149    1.8
Name: PetalWidthCm, Length: 150, dtype: float64
```

```
# operator broadcast
df['PetalWidthCm']*10  # cm -> mm
```

```
0       2.0
1       2.0
2       2.0
3       2.0
4       2.0
       ...
145    23.0
146    19.0
147    20.0
148    23.0
149    18.0
Name: PetalWidthCm, Length: 150, dtype: float64
```

```
# compute statistics
print('sum:', df['PetalWidthCm'].sum())
print('max:', df['PetalWidthCm'].max())
print('min:', df['PetalWidthCm'].min())
print('mean:', df['PetalWidthCm'].mean())
print('var:', df['PetalWidthCm'].var())
print('std:', df['PetalWidthCm'].std())
print('median:', df['PetalWidthCm'].median())
```

```
sum: 179.8
max: 2.5
min: 0.1
mean: 1.1986666666666668
var: 0.582414317673378
std: 0.7631607417008411
median: 1.3
```

argmax()	最大值的位置
argmin()	最小值的位置
count()	計算各元素的個數
prod()	連乘 (product)
cumsum()	累加(cumulative sum)

Group

```
df.groupby('Species')
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001F40EB38400>
```

```
df.groupby('Species').mean()
```

Species	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Iris-setosa	25.5	5.006	3.418	1.464	0.244
Iris-versicolor	75.5	5.936	2.770	4.260	1.326
Iris-virginica	125.5	6.588	2.974	5.552	2.026

```
df.groupby('Species').aggregate([pd.DataFrame.mean, pd.DataFrame.std])
```

	Id		SepalLengthCm		SepalWidthCm		PetalLengthCm		PetalWidthCm	
Species	mean	std	mean	std	mean	std	mean	std	mean	std
Iris-setosa	25.5	14.57738	5.006	0.352490	3.418	0.381024	1.464	0.173511	0.244	0.107210
Iris-versicolor	75.5	14.57738	5.936	0.516171	2.770	0.313798	4.260	0.469911	1.326	0.197753
Iris-virginica	125.5	14.57738	6.588	0.635880	2.974	0.322497	5.552	0.551895	2.026	0.274650

```
df.groupby('Species').get_group('Iris-virginica').head()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
100	101	6.3	3.3	6.0	2.5	Iris-virginica
101	102	5.8	2.7	5.1	1.9	Iris-virginica
102	103	7.1	3.0	5.9	2.1	Iris-virginica
103	104	6.3	2.9	5.6	1.8	Iris-virginica
104	105	6.5	3.0	5.8	2.2	Iris-virginica

groupby() 分群
get_group() 獲得某群資料
aggregate() 聚合

Group

```
df_2 = pd.DataFrame({'Gender': ['M', 'F', 'M', 'F', 'M', 'F', 'M'],
                     'Degree': ['BS', 'BS', 'BS', 'MS', 'MS', 'PhD', 'phD'],
                     'Pet': ['Cat', 'Cat', 'Dog', 'Dog', 'Dog', 'Cat', 'Cat'],
                     'Team': ['A', 'A', 'A', 'B', 'B', 'B', 'B']})
df_2
```

	Gender	Degree	Pet	Team
0	M	BS	Cat	A
1	F	BS	Cat	A
2	M	BS	Dog	A
3	F	MS	Dog	B
4	M	MS	Dog	B
5	F	PhD	Cat	B
6	M	phD	Cat	B

groupby() 是個很好用的指令，除了使用已知的column names來分群外，也可以搭配第12頁ppt的boolean vector或者第15頁ppt提到的mask自己定義想要的分群方式

```
df_2.groupby(['Team', 'Gender']).count()
```

		Degree	Pet
Team	Gender		
A	F	1	1
	M	2	2
B	F	2	2
	M	2	2

提醒！get_group時，只能使用tuple的方式！

```
df_2.groupby(['Team', 'Gender']).get_group(('A', 'M'))
```

	Gender	Degree	Pet	Team
0	M	BS	Cat	A
2	M	BS	Dog	A

Plot

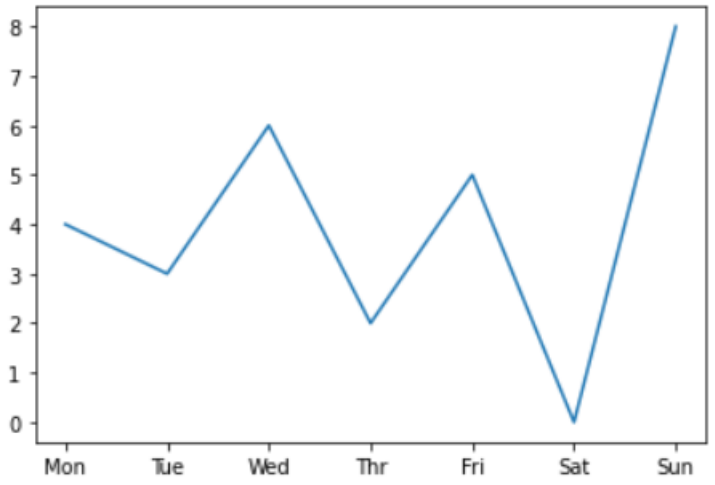
記得搭配matplotlib.pyplot

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

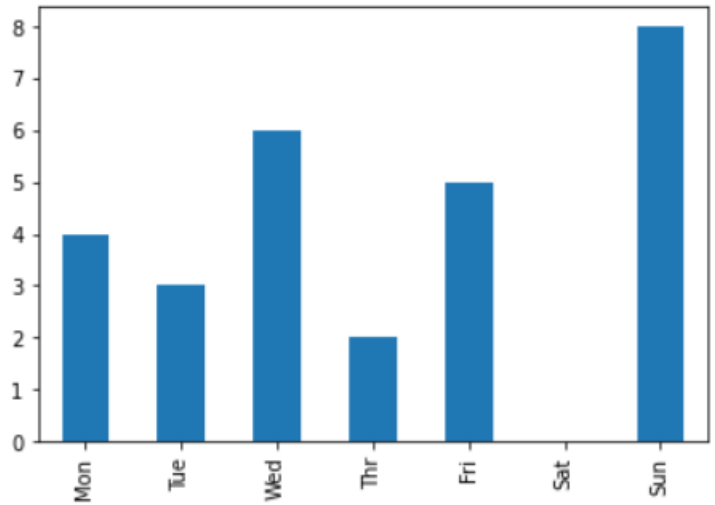
```
days = ['Mon', 'Tue', 'Wed', 'Thr', 'Fri', 'Sat', 'Sun']
studyHours = [4, 3, 6, 2, 5, 0, 8]
s = pd.Series(studyHours, index=days)
s
```

```
Mon    4
Tue    3
Wed    6
Thr    2
Fri    5
Sat    0
Sun    8
dtype: int64
```

```
s.plot()
plt.show()
```




```
s.plot(kind='bar')
plt.show()
```



End

謝謝聽到這裡還醒著的你們 (> /// <)

希望大家都能順利完成HW1的前兩題，加油！