Python & Data - Week 17

Pandas

打包下載

1. 使用 Pandas 讀取資料
2. DataFrame 簡介
3. DataFrame 操作方式
    - Access 存取
    - Transform 轉變
    - Slicing 切割
    - Sorting 排序
    - Filtering 篩選
    - 使用資料做計算
4. 實作
    - 找出學生的平均年齡
    - 找出兩科合計最高分的學生及找出頭3名
    - 找出學號為質數的同學 (如何找質數: https://www.geeksforgeeks.org/prime-numbers/)

使用 Pandas 讀取資料

1. 讀取 Excel 或 CSV

In [ ]:
```
import pandas as pd
pd.read_excel('./inputs/student_scores.xlsx')
```

Out[ ]:

	student_id	name	age	chinese_score	english_score
0	1	David	15	60	80
1	2	Peggy	14	74	55
2	3	Michael	15	80	90
3	4	Sarah	16	99	88
4	5	Paul	15	77	74
5	6	Lisa	13	55	68
6	7	Jeffrey	14	98	75
7	8	Isaac	15	65	67
8	9	Danny	15	88	85
9	10	Taylor	15	75	86

1. 讀取 XML

In [ ]:
```
import pandas as pd
pd.read_xml('./inputs/test_data.xml')
```

Out[ ]:

	first-name	last-name	age
0	Peter	Chan	15
1	Mary	Lui	20

1. 讀取 JSON

In [ ]:
```
import pandas as pd
pd.read_json('./inputs/random_data.json')
```

Out[ ]:

	first_name	last_name	age
0	Peter	Chan	15
1	Mary	Lui	20

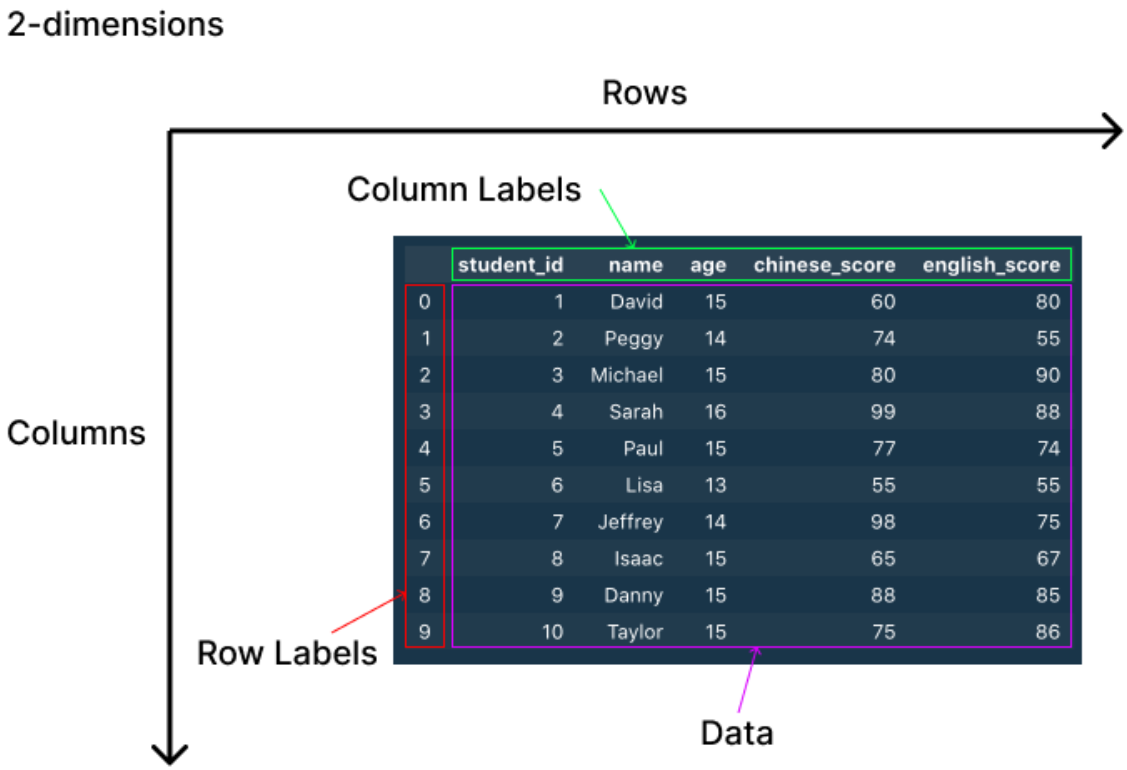
1. 讀取 HTML Table

In [ ]:
```
import pandas as pd
# 讀取一個或多個表格
df = pd.read_html('./inputs/table.html')
df[0]
```

Out[ ]:

	name	city	age	py-score
0	Xavier	Mexico City	41	88.0
1	Ann	Toronto	28	79.0
2	Jana	Prague	33	81.0
3	Yi	Shanghai	34	80.0
4	Robin	Manchester	38	68.0
5	Amal	Cairo	31	61.0
6	Nori	Osaka	37	84.0

DataFrame 簡介



DataFrame 的操作方式

1. 存取 Column

In [ ]:
```
import pandas as pd
score_df = pd.read_excel('./inputs/student_scores.xlsx')
new_df = score_df.age # Access age column with label
score_df['age'] # Alternative
```

Out[ ]:
```
0    15
1    14
2    15
3    16
4    15
5    13
6    14
7    15
8    15
9    15
Name: age, dtype: int64
```

1.1 存取 Row

In [ ]:
```
score_df.loc[0] # Row labels
score_df.iloc[0] # Index (Position, starts from 0)
```

Out[ ]:
```
student_id            1
name              David
age                  15
chinese_score        60
english_score        80
Name: 0, dtype: object
```

1. Transform

In [ ]:
```
score_df['total_score'] = score_df['chinese_score'] + score_df['english_sc
score_df
```

Out[ ]:

	student_id	name	age	chinese_score	english_score	total_score
0	1	David	15	60	80	140
1	2	Peggy	14	74	55	129
2	3	Michael	15	80	90	170
3	4	Sarah	16	99	88	187
4	5	Paul	15	77	74	151
5	6	Lisa	13	55	68	123
6	7	Jeffrey	14	98	75	173
7	8	Isaac	15	65	67	132
8	9	Danny	15	88	85	173
9	10	Taylor	15	75	86	161

1. Slicing

In [ ]:
```
score_df[1:]
```

Out[ ]:

	student_id	name	age	chinese_score	english_score	total_score
1	2	Peggy	14	74	55	129
2	3	Michael	15	80	90	170
3	4	Sarah	16	99	88	187
4	5	Paul	15	77	74	151
5	6	Lisa	13	55	68	123
6	7	Jeffrey	14	98	75	173
7	8	Isaac	15	65	67	132
8	9	Danny	15	88	85	173
9	10	Taylor	15	75	86	161

In [ ]:
```
score_df[::2]
```

Out[ ]:

	student_id	name	age	chinese_score	english_score	total_score
0	1	David	15	60	80	140
2	3	Michael	15	80	90	170
4	5	Paul	15	77	74	151
6	7	Jeffrey	14	98	75	173
8	9	Danny	15	88	85	173

1. Sorting

In [ ]:

```
# Ascending 順序，反之為倒序
score_df.sort_values('chinese_score', ascending=False)
```

Out[ ]:

	student_id	name	age	chinese_score	english_score	total_score
3	4	Sarah	16	99	88	187
6	7	Jeffrey	14	98	75	173
8	9	Danny	15	88	85	173
2	3	Michael	15	80	90	170
4	5	Paul	15	77	74	151
9	10	Taylor	15	75	86	161
1	2	Peggy	14	74	55	129
7	8	Isaac	15	65	67	132
0	1	David	15	60	80	140
5	6	Lisa	13	55	68	123

1. Filtering

In [ ]:

```
chinse_passed_df = score_df['chinese_score'] >= 60
chinse_passed_df
```

Out[ ]:

```
0     True
1     True
2     True
3     True
4     True
5    False
6     True
7     True
8     True
9     True
Name: chinese_score, dtype: bool
```

In [ ]:

```
score_df[chinse_passed_df]
```

Out[ ]:

	student_id	name	age	chinese_score	english_score	total_score
0	1	David	15	60	80	140
1	2	Peggy	14	74	55	129
2	3	Michael	15	80	90	170
3	4	Sarah	16	99	88	187
4	5	Paul	15	77	74	151
6	7	Jeffrey	14	98	75	173
7	8	Isaac	15	65	67	132
8	9	Danny	15	88	85	173
9	10	Taylor	15	75	86	161

In [ ]:

```
# 另一種寫法
score_df[score_df['chinese_score'] >= 60]
```

Out[ ]:

	student_id	name	age	chinese_score	english_score	total_score
0	1	David	15	60	80	140
1	2	Peggy	14	74	55	129
2	3	Michael	15	80	90	170
3	4	Sarah	16	99	88	187
4	5	Paul	15	77	74	151
6	7	Jeffrey	14	98	75	173
7	8	Isaac	15	65	67	132
8	9	Danny	15	88	85	173
9	10	Taylor	15	75	86	161

💡 試試找出不及格的同學?

1. 使用資料做計算

In [ ]:

```
# 平均分數
average_chinese_score = score_df['chinese_score'].mean()
print(f'Average chinese score {average_chinese_score}')

# 最低分數
min_chinese_score = score_df['chinese_score'].min()
print(f'Lowest chinese score {min_chinese_score}')

# 最高分數
min_chinese_score = score_df['chinese_score'].max()
print(f'Highest chinese score {min_chinese_score}')

# 基本統計資料
score_df.describe()
```

```
Average chinese score 77.1
Lowest chinese score 55
Highest chinese score 99
```

Out[ ]:

	student_id	age	chinese_score	english_score	total_score
count	10.00000	10.000000	10.000000	10.000000	10.000000
mean	5.50000	14.700000	77.100000	76.800000	153.900000
std	3.02765	0.823273	14.850739	11.163432	22.087955
min	1.00000	13.000000	55.000000	55.000000	123.000000
25%	3.25000	14.250000	67.250000	69.500000	134.000000
50%	5.50000	15.000000	76.000000	77.500000	156.000000
75%	7.75000	15.000000	86.000000	85.750000	172.250000
max	10.00000	16.000000	99.000000	90.000000	187.000000