



Python 中的 Excel Pandas 介紹

製作人：黃宥輔



Python中的EXCEL - Pandas

什麼是 Pandas?



Pandas為Python中處理數據的套件，擁有豐富的數據分析工具，從欄位間的四則運算、累加，到敘述統計、資料分組都可以實現，其核心數據結構是Series和DataFrame，分別用以處理一維資料與二維資料，並提供讀寫EXCEL檔、CSV檔、HTML的表格等，重要的是，Pandas是一個開源的程式庫，有廣泛的使用者和開發者社群，能夠保障長期開發和更新。



官方網站 <https://pandas.pydata.org/>

- ▶ Python 中用於資料處理，容易上手的強大套件
- ▶ 可以輕鬆讀取 xlsx、csv、JSON、SQL、HTML 格式檔案
- ▶ 可以進行資料清洗、資料加工、敘述統計 ... 等
- ▶ 從多個角度計算數據的統計值、分位數、頻率等，以便更好地瞭解數據的分佈情況
- ▶ pandas內建的視覺化工具可以方便地繪製數據圖表，如散點圖、線圖、長條圖等，並且也支援自定義圖表風格，**結合Matplotlib套件，可以plot函數進行繪圖**
- ▶ 可以與很多套件交互應用，如 Ta-Lib、Keras ... 等

- ▶ Anaconda 中內建
- ▶ 使用pip手動安裝
 - ▶ `pip install pandas`

項目	說明	對應EXCEL
Series	一維資料	欄
DataFrame	二維資料	工作表

Series				
	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

DataFrame				
	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

DataFrame 建構

**AI.FINTECH**

A I 金 融 科 技 中 心

項目	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	data_dict = {
4	"項目": ["蘋果", "橘子", "西瓜"],
5	"數量": [2, 3, 1],
6	}
7	
8	df = pd.DataFrame(data_dict)
9	print(df)

	項目	數量
0	蘋果	2
1	橘子	3
2	西瓜	1

常用格式	語法
EXCEL	<code>df = pd.read_excel (file_path)</code>
CSV	<code>df = pd.read_csv (file_path)</code>
SQL	<code>df = pd.read_sql (sql_cmd, con = conn)</code>
HTML	<code>df = pd.read_html (html_table)</code>

項目	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	
4	df = pd.read_excel("0050.xlsx")
5	print(df)

0050.xlsx								
日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價	漲跌價差	成交筆數
2023/02/01	6,903,171	821,124,932	119.1	119.45	118.45	119	0.7	8,921
2023/02/02	16,194,958	1,956,007,315	120.3	121.1	120.25	121	2	15,931
2023/02/03	9,934,984	1,200,856,481	120.95	121.2	120.2	121	0	10,997
2023/02/06	14,297,030	1,709,758,781	120.1	120.1	119.25	119.25	-1.75	25,811
2023/02/07	8,787,291	1,045,314,027	119.1	119.25	118.55	119	-0.25	13,540
2023/02/08	9,239,321	1,114,132,382	120.1	120.95	120	120.85	1.85	8,739
2023/02/09	5,032,245	607,543,546	120.5	121	120.3	120.9	0.05	7,141
2023/02/10	6,572,526	794,912,205	120.9	121.1	120.6	121	0.1	9,368
2023/02/13	4,149,906	499,573,936	120.7	120.9	119.9	120.75	-0.25	8,262
2023/02/14	7,965,786	968,395,750	121.25	121.8	121.25	121.65	0.9	8,497
2023/02/15	13,852,368	1,653,181,275	119.95	120.25	118.8	118.85	-2.8	33,160
2023/02/16	11,233,273	1,349,768,658	119.3	120.5	119.3	120.3	1.45	14,038
2023/02/17	8,960,233	1,064,028,251	119.3	119.3	118.4	118.75	-1.55	22,127
2023/02/20	6,230,136	741,121,308	118.55	119.35	118.1	119	0.25	11,499
2023/02/21	4,772,315	567,003,197	119	119.1	118.55	118.95	-0.05	9,128
2023/02/22	12,041,784	1,414,076,263	117.5	117.75	117.1	117.65	-1.3	24,996
2023/02/23	6,448,658	768,858,699	117.95	119.8	117.95	119.4	1.75	8,273
2023/02/24	5,828,709	695,430,436	119.9	120.1	118.8	118.8	-0.6	9,038

```
1 # -*- coding: utf-8 -*-
2 import pandas as pd
3
4 df = pd.read_excel( "0050.xlsx" )
5 print( df )
6
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

Variable explorer Help Plots Files Find Code Analysis

Console I/A

20230324演講')

	日期	成交股數	成交金額	開盤價	...	最低價	收盤價	漲跌價差	成交筆數
0	2023/02/01	6903171	821124932	119.10	...	118.45	119.00	0.70	8921
1	2023/02/02	16194958	1956007315	120.30	...	120.25	121.00	2.00	15931
2	2023/02/03	9934984	1200856481	120.95	...	120.20	121.00	0.00	10997
3	2023/02/06	14297030	1709758781	120.10	...	119.25	119.25	-1.75	25811
4	2023/02/07	8787291	1045314027	119.10	...	118.55	119.00	-0.25	13540
5	2023/02/08	9239321	1114132382	120.10	...	120.00	120.85	1.85	8739
6	2023/02/09	5032245	607543546	120.50	...	120.30	120.90	0.05	7141
7	2023/02/10	6572526	794912205	120.90	...	120.60	121.00	0.10	9368
8	2023/02/13	4149906	499573936	120.70	...	119.90	120.75	-0.25	8262
9	2023/02/14	7965786	968395750	121.25	...	121.25	121.65	0.90	8497
10	2023/02/15	13852368	1653181275	119.95	...	118.80	118.85	-2.80	33160
11	2023/02/16	11233273	1349768658	119.30	...	119.30	120.30	1.45	14038
12	2023/02/17	8960233	1064028251	119.30	...	118.40	118.75	-1.55	22127
13	2023/02/20	6230136	741121308	118.55	...	118.10	119.00	0.25	11499
14	2023/02/21	4772315	567003197	119.00	...	118.55	118.95	-0.05	9128
15	2023/02/22	12041784	1414076263	117.50	...	117.10	117.65	-1.30	24996
16	2023/02/23	6448658	768858699	117.95	...	117.95	119.40	1.75	8273
17	2023/02/24	5828709	695430436	119.90	...	118.80	118.80	-0.60	9038

[18 rows x 9 columns]

語法	說明	結果範例
df.shape	取得 DataFrame 的形狀	(20, 9)
df.columns	取得 DataFrame 的欄位名稱	Index(['日期', '成交股數', '成交金額', '開盤價', '最高價', '最低價', '收盤價', '漲跌價差', '成交筆數'], dtype='object')
df.index	取得 DataFrame 的欄位索引列	RangeIndex(start=0, stop=20, step=1)



DataFrame 的資訊

df.columns

df.index

	日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價	漲跌價差	成交筆數
0	2021/1/4	6,307,525	778,088,986	122.2	124.5	122	124.35	2.1	5434
1	2021/1/5	4,962,514	616,480,760	124.2	124.65	123.75	124.6	0.25	6179
2	2021/1/6	10,859,755	1,367,952,768	126.1	127.15	124.1	125.95	1.35	9245
3	2021/1/7	6,863,448	877,015,237	126.55	129	126.55	128.9	2.95	6299
4	2021/1/8	13,298,515	1,733,821,488	130.6	131.25	129.55	131.2	2.3	10629
5	2021/1/11	10,492,804	1,375,120,183	131.2	132.3	130.5	132.3	1.1	10995
6	2021/1/12	11,223,083	1,484,059,869	132.3	133.15	131.6	132.1	-0.2	11184
7	2021/1/13	10,912,897	1,465,070,002	132.75	135.45	132.7	135.3	3.2	8059
8	2021/1/14	12,390,589	1,660,294,742	135	135.05	133.35	134.05	-1.25	11405
9	2021/1/15	21,933,080	2,983,569,891	137.15	138	134.4	134.5	0.45	13533
10	2021/1/18	14,495,824	1,937,130,062	134.6	134.95	132.1	134.4	-0.1	11800
11	2021/1/19	12,262,208	1,675,696,411	135.2	138.1	135.2	137.95	3.55	8222
12	2021/1/20	30,367,315	4,202,770,386	138.1	139.5	137.65	138.2	0.25	25701
13	2021/1/21	14,029,981	1,984,425,275	138.4	143.25	138.35	143	4.8	12122
14	2021/1/22	24,714,046	3,413,371,452	138.2	139	135	137.45	X0.00	26235
15	2021/1/25	14,258,565	1,931,915,211	137.2	137.3	134.2	135.6	-1.85	15890
16	2021/1/26	15,080,041	2,016,627,426	135.55	136.6	131.8	132.85	-2.75	20506
17	2021/1/27	8,124,071	1,082,040,695	133.3	134.4	132.25	133.05	0.2	8939
18	2021/1/28	15,242,813	1,983,766,728	130	131.3	129.4	130.35	-2.7	20286
19	2021/1/29	14,709,917	1,910,423,184	131.6	131.85	128.2	128.2	-2.15	18761

df.shape

項目	程式碼
1	<code># -*- coding: utf-8 -*-</code>
2	<code>import pandas as pd</code>
3	
4	<code>df = pd.read_excel("0050.xlsx")</code>
5	<code>print(df.shape)</code> # 取得 DataFrame 的形狀
6	<code>print("-----")</code>
7	<code>print(df.columns)</code> # 取得 DataFrame 的欄位名稱
8	<code>print("-----")</code>
9	<code>print(df.index)</code> # 取得 DataFrame 的欄位索引列

上機題2參考答案.py × 1. DataFrame 建構.py × 2. 讀取資料.py × 3. DataFrame 的資訊.py ×

```
1 # -*- coding: utf-8 -*-
2 import pandas as pd
3
4 df = pd.read_excel( "0050.xlsx" )
5 print( df.shape ) # 取得 DataFrame 的形狀
6 print( "-----" )
7 print( df.columns ) # 取得 DataFrame 的欄位名稱
8 print( "-----" )
9 print( df.index ) # 取得 DataFrame 的欄位索引
10
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our [tutorial](#)

Help Variable Explorer Plots Files Code Analysis

Console 2/A ×

IPython 8.12.0 -- An enhanced Interactive Python.

```
In [1]: runfile('C:/Users/user/Desktop/Pandas相關/pandas 介紹/3. DataFrame 的資
訊.py', wdir='C:/Users/user/Desktop/Pandas相關/pandas 介紹')
(20, 9)
```

```
Index(['日期', '成交股數', '成交金額', '開盤價', '最高價', '最低價', '收盤價', '漲跌價
差', '成交筆數'], dtype='object')
```

```
RangeIndex(start=0, stop=20, step=1)
```

IPython Console History

conda: base (Python 3.11.3) ✎ Completions: conda(base) ✓ LSP: Python Line 10, Col 1 UTF-8 CRLF RW Mem 93%

語法	說明
<code>df.loc [index , (column)]</code>	以欄位、序列名稱擷取資料
<code>df.iloc [index , (column)]</code>	以欄位、序列順序擷取資料
<code>df.at [index , (column)]</code>	以欄位、序列名稱擷取資料格
<code>df.iat [index , (column)]</code>	以欄位、序列順序擷取資料格

項目	說明
參數	index、columns 可以填入變數或 list，挑選資料
	可以只輸入 index，會回傳 Series，且 Series 也可用以上這些方法

項目	程式碼	結果																
1	# -*- coding: utf-8 -*-																	
2	import pandas as pd																	
3																		
4	df = pd.read_excel("0050.xlsx")																	
5	print(df.loc[[0, 4, 9, 14] , ["日期", "收盤價"]])	<table><tr><th></th><th>日期</th><th>收盤價</th></tr><tr><td>0</td><td>2021-01-04</td><td>124.35</td></tr><tr><td>4</td><td>2021-01-08</td><td>131.20</td></tr><tr><td>9</td><td>2021-01-15</td><td>134.50</td></tr><tr><td>14</td><td>2021-01-22</td><td>137.45</td></tr></table>		日期	收盤價	0	2021-01-04	124.35	4	2021-01-08	131.20	9	2021-01-15	134.50	14	2021-01-22	137.45	
	日期		收盤價															
0	2021-01-04		124.35															
4	2021-01-08		131.20															
9	2021-01-15	134.50																
14	2021-01-22	137.45																
6	print(df.iloc[[0, 4, 9, 14] , [0, 6]])																	
7	print("-----")																	
8	print(df.iloc[1:4 , 2:5])	<table><tr><th></th><th>成交金額</th><th>開盤價</th><th>最高價</th></tr><tr><td>1</td><td>616480760</td><td>124.20</td><td>124.65</td></tr><tr><td>2</td><td>1367952768</td><td>126.10</td><td>127.15</td></tr><tr><td>3</td><td>877015237</td><td>126.55</td><td>129.00</td></tr></table>		成交金額	開盤價	最高價	1	616480760	124.20	124.65	2	1367952768	126.10	127.15	3	877015237	126.55	129.00
	成交金額		開盤價	最高價														
1	616480760		124.20	124.65														
2	1367952768	126.10	127.15															
3	877015237	126.55	129.00															
9	print("-----")																	


```
5 print( df.loc[ [0, 4, 9, 14] , [ "日期", "收盤價" ] ] )
```

```
6 print( df.iloc[ [0, 4, 9, 14] , [ 0, 6 ] ] )
```

	日期	收盤價
0	2021-01-04	124.35
4	2021-01-08	131.20
9	2021-01-15	134.50
14	2021-01-22	137.45

	日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價
0	2021/1/4	6,307,525	778,088,986	122.2	124.5	122	124.35
1	2021/1/5	4,962,514	616,480,760	124.2	124.65	123.75	124.6
2	2021/1/6	10,859,755	1,367,952,768	126.1	127.15	124.1	125.95
3	2021/1/7	6,863,448	877,015,237	126.55	129	126.55	128.9
4	2021/1/8	13,298,515	1,733,821,488	130.6	131.25	129.55	131.2
5	2021/1/11	10,492,804	1,375,120,183	131.2	132.3	130.5	132.3
6	2021/1/12	11,223,083	1,484,059,869	132.3	133.15	131.6	132.1
7	2021/1/13	10,912,897	1,465,070,002	132.75	135.45	132.7	135.3
8	2021/1/14	12,390,589	1,660,294,742	135	135.05	133.35	134.05
9	2021/1/15	21,933,080	2,983,569,891	137.15	138	134.4	134.5
10	2021/1/18	14,495,824	1,937,130,062	134.6	134.95	132.1	134.4
11	2021/1/19	12,262,208	1,675,696,411	135.2	138.1	135.2	137.95
12	2021/1/20	30,367,315	4,202,770,386	138.1	139.5	137.65	138.2
13	2021/1/21	14,029,981	1,984,425,275	138.4	143.25	138.35	143
14	2021/1/22	24,714,046	3,413,371,452	138.2	139	135	137.45

```
8 print( df.iloc[ 1:4 , 2:5 ] )
```

	成交金額	開盤價	最高價
1	616480760	124.20	124.65
2	1367952768	126.10	127.15
3	877015237	126.55	129.00

	日期	成交股數	成交金額	開盤價
0	2021/1/4	6,307,525	778,088,986	122.2
1	2021/1/5	4,962,514	616,480,760	124.2
2	2021/1/6	10,859,755	1,367,952,768	126.1
3	2021/1/7	6,863,448	877,015,237	126.55

項目	程式碼	結果															
10	print(df.at[3, "收盤價"])	<div>128.9</div>															
11	print(df.iat[3, 6])																
12	print("-----")	<table><tr><th></th><th>日期</th><th>收盤價</th></tr><tr><td>2</td><td>2021-01-06</td><td>125.95</td></tr><tr><td>4</td><td>2021-01-08</td><td>131.20</td></tr><tr><td>6</td><td>2021-01-12</td><td>132.10</td></tr><tr><td>8</td><td>2021-01-14</td><td>134.05</td></tr></table>		日期	收盤價	2	2021-01-06	125.95	4	2021-01-08	131.20	6	2021-01-12	132.10	8	2021-01-14	134.05
	日期		收盤價														
2	2021-01-06	125.95															
4	2021-01-08	131.20															
6	2021-01-12	132.10															
8	2021-01-14	134.05															
13	print(df.loc[:, [“日期”, “收盤價”]].iloc[2 : 10 : 2])																

```
10 print( df.at[ 3, "收盤價" ] )
```

```
11 print( df.iat[ 3, 6 ] )
```

128.9

	日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價
0	2021/1/4	6,307,525	778,088,986	122.2	124.5	122	124.35
1	2021/1/5	4,962,514	616,480,760	124.2	124.65	123.75	124.6
2	2021/1/6	10,859,755	1,367,952,768	126.1	127.15	124.1	125.95
3	2021/1/7	6,863,448	877,015,237	126.55	129	126.55	128.9

13 `print(df.loc[:, [“日期”, “收盤價”]].iloc[2 : 10 : 2])`

	日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價
0	2021/1/4	6,307,525	778,088,986	122.2	124.5	122	124.35
1	2021/1/5	4,962,514	616,480,760	124.2	124.65	123.75	124.6
2	2021/1/6	10,859,755	1,367,952,768	126.1	127.15	124.1	125.95
3	2021/1/7	6,863,448	877,015,237	126.55	129	126.55	128.9
4	2021/1/8	13,298,515	1,733,821,488	130.6	131.25	129.55	131.2
5	2021/1/11	10,492,804	1,375,120,183	131.2	132.3	130.5	132.3
6	2021/1/12	11,223,083	1,484,059,869	132.3	133.15	131.6	132.1
7	2021/1/13	10,912,897	1,465,070,002	132.75	135.45	132.7	135.3
8	2021/1/14	12,390,589	1,660,294,742	135	135.05	133.35	134.05
9	2021/1/15	21,933,080	2,983,569,891	137.15	138	134.4	134.5
10	2021/1/18	14,495,824	1,937,130,062	134.6	134.95	132.1	134.4
11	2021/1/19	12,262,208	1,675,696,411	135.2	138.1	135.2	137.95
12	2021/1/20	30,367,315	4,202,770,386	138.1	139.5	137.65	138.2
13	2021/1/21	14,029,981	1,984,425,275	138.4	143.25	138.35	143
14	2021/1/22	24,714,046	3,413,371,452	138.2	139	135	137.45
15	2021/1/25	14,258,565	1,931,915,211	137.2	137.3	134.2	135.6
16	2021/1/26	15,080,041	2,016,627,426	135.55	136.6	131.8	132.85
17	2021/1/27	8,124,071	1,082,040,695	133.3	134.4	132.25	133.05
18	2021/1/28	15,242,813	1,983,766,728	130	131.3	129.4	130.35
19	2021/1/29	14,709,917	1,910,423,184	131.6	131.85	128.2	128.2

	日期	收盤價
0	2021/1/4	124.35
1	2021/1/5	124.6
2	2021/1/6	125.95
3	2021/1/7	128.9
4	2021/1/8	131.2
5	2021/1/11	132.3
6	2021/1/12	132.1
7	2021/1/13	135.3
8	2021/1/14	134.05
9	2021/1/15	134.5
10	2021/1/18	134.4
11	2021/1/19	137.95
12	2021/1/20	138.2
13	2021/1/21	143
14	2021/1/22	137.45
15	2021/1/25	135.6
16	2021/1/26	132.85
17	2021/1/27	133.05
18	2021/1/28	130.35
19	2021/1/29	128.2

	日期	收盤價
2	2021-01-06	125.95
4	2021-01-08	131.20
6	2021-01-12	132.10
8	2021-01-14	134.05

語法	說明
$df[\text{“欄位1”}] + df[\text{“欄位2”}]$	兩個欄位相加
$df[\text{“欄位1”}] - df[\text{“欄位2”}]$	兩個欄位相減
$df[\text{“欄位1”}] * df[\text{“欄位2”}]$	兩個欄位相乘
$df[\text{“欄位1”}] / df[\text{“欄位2”}]$	兩個欄位相除

項目	程式碼
1	import pandas as pd
2	df = pd.read_excel("0050.xlsx")
3	
4	df["加"] = df["開盤價"] + df["收盤價"]
5	df["減"] = df["最高價"] + df["最低價"]
6	df["乘"] = df["成交股數"] * df["收盤價"]
7	df["除"] = df["最高價"] / df["最低價"]
8	print(df[["加", "減", "乘", "除"]])

	加	減	乘	除
0	246.55	246.50	7.843407e+08	1.020492
1	248.80	248.40	6.183292e+08	1.007273
2	252.05	251.25	1.367786e+09	1.024577
3	255.45	255.55	8.846984e+08	1.019360
4	261.80	260.80	1.744765e+09	1.013122
5	263.50	262.80	1.388198e+09	1.013793
6	264.40	264.75	1.482569e+09	1.011778
7	268.05	268.15	1.476515e+09	1.020723
8	269.05	268.40	1.660958e+09	1.012748
9	271.65	272.40	2.949999e+09	1.026786
10	269.00	267.05	1.948239e+09	1.021575
11	273.15	273.30	1.691572e+09	1.021450
12	276.30	277.15	4.196763e+09	1.013440
13	281.40	281.60	2.006287e+09	1.035417
14	275.65	274.00	3.396946e+09	1.029630
15	272.80	271.50	1.933461e+09	1.023100
16	268.40	268.40	2.003383e+09	1.036419
17	266.35	266.65	1.080908e+09	1.016257
18	260.35	260.70	1.986901e+09	1.014683
19	259.80	260.05	1.885811e+09	1.028471

語法	說明
df.mean()	計算單一或所有欄位的平均值
df.sum()	計算單一或所有欄位的總和值
df.max()	計算單一或所有欄位的最大值
df.min()	計算單一或所有欄位的最小值
df.std()	計算單一或所有欄位的標準差
df.describe()	顯示單一或所有欄位的敘述統計
df.count()	計算單一或所有欄位的資料數

項目	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	
4	df = pd.read_excel("0050.xlsx")
5	print(df.mean())
6	print("-----")
7	print(df.sum())
8	print("-----")

df.mean()

```
成交股數    1.362645e+07
成交金額    1.823982e+09
開盤價      1.325100e+02
最高價      1.338375e+02
最低價      1.311325e+02
收盤價      1.327150e+02
成交筆數    1.307120e+04
dtype: float64
```

df.sum()

```
成交股數    2.725290e+08
成交金額    3.647964e+10
開盤價      2.650200e+03
最高價      2.676750e+03
最低價      2.622650e+03
收盤價      2.654300e+03
成交筆數    2.614240e+05
dtype: float64
```

項目	程式碼
9	<code>print(df.max())</code>
10	<code>print("-----")</code>
11	<code>print(df.min())</code>
12	<code>print("-----")</code>
13	<code>print(df.std())</code>
14	<code>print("-----")</code>

df.max()

```

日期          2021-01-29 00:00:00
成交股數          30367315
成交金額          4202770386
開盤價              138.4
最高價              143.25
最低價              138.35
收盤價              143.0
成交筆數              26235
dtype: object

```

df.min()

```

日期          2021-01-04 00:00:00
成交股數          4962514
成交金額          616480760
開盤價              122.2
最高價              124.5
最低價              122.0
收盤價              124.35
成交筆數              5434
dtype: object

```

df. std()

```

日期          8 days 03:49:48.779306003
成交股數          6148015.832372
成交金額          869350532.209508
開盤價              4.779716
最高價              4.9614
最低價              4.467699
收盤價              4.777169
成交筆數          6206.249479
dtype: object

```

項目	程式碼
15	<code>print(df.describe())</code>
16	<code>df.describe().to_excel("des.xlsx")</code>
17	<code>print("-----")</code>
18	<code>print(df.count())</code>

df.describe()

```

      成交股數      成交金額      ...      收盤價      成交筆數
count  2.000000e+01  2.000000e+01  ...  20.000000    20.000000
mean   1.362645e+07  1.823982e+09  ...  132.715000  13071.200000
std     6.148016e+06  8.693505e+08  ...    4.777169   6206.249479
min     4.962514e+06  6.164808e+08  ...  124.350000   5434.000000
25%    1.076802e+07  1.373328e+09  ...  129.987500   8759.750000
50%    1.284455e+07  1.704759e+09  ...  132.950000  11294.500000
75%    1.480245e+07  1.983931e+09  ...  135.375000  16607.750000
max     3.036732e+07  4.202770e+09  ...  143.000000  26235.000000

```

[8 rows x 7 columns]

df.count()

```

日期      20
成交股數    20
成交金額    20
開盤價     20
最高價     20
最低價     20
收盤價     20
漲跌價差    20
成交筆數    20
dtype: int64

```

des.xlsx								
	A	B	C	D	E	F	G	H
1		成交股數	成交金額	開盤價	最高價	最低價	收盤價	成交筆數
2	count	20	20	20	20	20	20	20
3	mean	13626450	1.82E+09	132.51	133.8375	131.1325	132.715	13071.2
4	std	6148016	8.69E+08	4.779716	4.9614	4.467699	4.777169	6206.249
5	min	4962514	6.16E+08	122.2	124.5	122	124.35	5434
6	25%	10768017	1.37E+09	130.45	131.2875	129.1	129.9875	8759.75
7	50%	12844552	1.7E+09	133.025	134.675	131.95	132.95	11294.5
8	75%	14802448	1.98E+09	135.95	137.475	134.25	135.375	16607.75
9	max	30367315	4.2E+09	138.4	143.25	138.35	143	26235

`df.mean(axis = 1))`

項目	程式碼
20	<code>df = df.loc[:,["開盤價", "最高價", "最低價", "收盤價"]]</code>
21	<code>print(df.mean(axis = 1))</code>

```

0      123.2625
1      124.3000
2      125.8250
3      127.7500
4      130.6500
5      131.5750
6      132.2875
7      134.0500
8      134.3625
9      136.0125
10     134.0125
11     136.6125
12     138.3625
13     140.7500
14     137.4125
15     136.0750
16     134.2000
17     133.2500
18     130.2625
19     129.9625
dtype: float64

```

語法	說明
<code>df[df["收盤價"] > 135]</code>	使用關係運算式，篩選符合條件的資料 範例為篩選出 df 中收盤價大於 135 的資料
<code>df.head()</code>	顯示df的前 5 列，也可以填入整數顯示 N 列
<code>df.tail()</code>	顯示df的後 5 列，也可以填入要顯示 N 列
<code>df.sort_values(by = ["收盤價", "日期"], ascending = [True, True])</code>	依照收盤價為主，日期為輔，由小到大排序， ascending = False則是由大到小
<code>df.drop_duplicates(subset = [], keep = "first")</code>	移除重複，subset中填入以那些欄位為主，不填入則是全部欄位 first 代表若有重複，保留第一筆資料；在keep 輸入 last 則是保留最後一筆
<code>df.dropna(subset = [])</code>	移除空值，subset中填入以那些欄位為主，不填入則是全部欄位
<code>df.fillna(0)</code>	將空值填入 0
<code>df.shift(1)</code>	將資料欄位往下移動一格， 輸入負值則是往上移動

資料處理

項目	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	
4	df = pd.read_excel("0050.xlsx")
5	
6	print(df[df["收盤價"] > 135])
7	print("-----")

df[df["收盤價"] > 135]

	日期	成交股數	成交金額	開盤價	...	最低價	收盤價	漲跌價差
成交筆數								
7	2021-01-13	10912897	1465070002	132.75	...	132.70	135.30	3.2
11	2021-01-19	12262208	1675696411	135.20	...	135.20	137.95	3.55
12	2021-01-20	30367315	4202770386	138.10	...	137.65	138.20	0.25
13	2021-01-21	14029981	1984425275	138.40	...	138.35	143.00	4.8
14	2021-01-22	24714046	3413371452	138.20	...	135.00	137.45	X0.00
15	2021-01-25	14258565	1931915211	137.20	...	134.20	135.60	-1.85

[6 rows x 9 columns]

項目

程式碼

8

print(df.head())

9

print("-----")

10

print(df.tail())

df.head()

	日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價	漲跌價
差	成交筆數							
0	2021-01-04	6307525	778088986	122.20	124.50	122.00	124.35	2.1
1	2021-01-05	4962514	616480760	124.20	124.65	123.75	124.60	0.25
2	2021-01-06	10859755	1367952768	126.10	127.15	124.10	125.95	1.35
3	2021-01-07	6863448	877015237	126.55	129.00	126.55	128.90	2.95
4	2021-01-08	13298515	1733821488	130.60	131.25	129.55	131.20	2.3
								10629

df.tail()

	日期	成交股數	成交金額	開盤價	...	最低價	收盤價	漲跌價差
成交筆數								
15	2021-01-25	14258565	1931915211	137.20	...	134.20	135.60	-1.85
16	2021-01-26	15080041	2016627426	135.55	...	131.80	132.85	-2.75
17	2021-01-27	8124071	1082040695	133.30	...	132.25	133.05	0.2
18	2021-01-28	15242813	1983766728	130.00	...	129.40	130.35	-2.7
19	2021-01-29	14709917	1910423184	131.60	...	128.20	128.20	-2.15
								18761


```
df.sort_values(by = ["收盤價", "日期"], ascending = [False, True])
```

項目	程式碼
11	<pre>print(df.sort_values(by = ["收盤價", "日期"], ascending = [True, True]))</pre>
12	<pre>print("-----")</pre>

	日期	成交股數	成交金額	開盤價	...	最低價	收盤價	漲跌價差
成交筆數								
13	2021-01-21	14029981	1984425275	138.40	...	138.35	143.00	4.8
12	2021-01-20	30367315	4202770386	138.10	...	137.65	138.20	0.25
11	2021-01-19	12262208	1675696411	135.20	...	135.20	137.95	3.55
14	2021-01-22	24714046	3413371452	138.20	...	135.00	137.45	X0.00
15	2021-01-25	14258565	1931915211	137.20	...	134.20	135.60	-1.85
7	2021-01-13	10912897	1465070002	132.75	...	132.70	135.30	3.2
9	2021-01-15	21933080	2983569891	137.15	...	134.40	134.50	0.45
10	2021-01-18	14495824	1937130062	134.60	...	132.10	134.40	-0.1
8	2021-01-14	12390589	1660294742	135.00	...	133.35	134.05	-1.25
17	2021-01-27	8124071	1082040695	133.30	...	132.25	133.05	0.2
16	2021-01-26	15080041	2016627426	135.55	...	131.80	132.85	-2.75
5	2021-01-11	10492804	1375120183	131.20	...	130.50	132.30	1.1
6	2021-01-12	11223083	1484059869	132.30	...	131.60	132.10	-0.2
4	2021-01-08	13298515	1733821488	130.60	...	129.55	131.20	2.3
18	2021-01-28	15242813	1983766728	130.00	...	129.40	130.35	-2.7
3	2021-01-07	6863448	877015237	126.55	...	126.55	128.90	2.95
19	2021-01-29	14709917	1910423184	131.60	...	128.20	128.20	-2.15
2	2021-01-06	10859755	1367952768	126.10	...	124.10	125.95	1.35
1	2021-01-05	4962514	616480760	124.20	...	123.75	124.60	0.25
0	2021-01-04	6307525	778088986	122.20	...	122.00	124.35	2.1

項目	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	data_dict = {
4	"項目": ["蘋果", "橘子", "西瓜", "西瓜"],
5	"數量": [2, 3, 1, 1],
6	"儲藏倉庫": ["A", None, "B", "C"]
7	}
8	
9	df = pd.DataFrame(data_dict)
10	print(df)
11	print("-----")

	項目	數量	儲藏倉庫
0	蘋果	2	A
1	橘子	3	None
2	西瓜	1	B
3	西瓜	1	C

項目	程式碼
12	<code>print(df.drop_duplicates(subset = ["項目"], keep = "first"))</code>
13	<code>print("-----")</code>
14	<code>print(df.dropna(subset = ["儲藏倉庫"]))</code>
15	<code>print("-----")</code>
16	<code>print(df.fillna(0))</code>
17	<code>print("-----")</code>
18	<code>print(df.shift(1))</code>
19	<code>print("-----")</code>
20	<code>print(df["數量"].shift(1))</code>

	項目	數量	儲藏倉庫
0	蘋果	2	A
1	橘子	3	None
2	西瓜	1	B

	項目	數量	儲藏倉庫
0	蘋果	2	A
2	西瓜	1	B
3	西瓜	1	C

	項目	數量	儲藏倉庫
0	蘋果	2	A
1	橘子	3	A
2	西瓜	1	B
3	西瓜	1	C

	項目	數量	儲藏倉庫
0	橘子	3.0	None
1	西瓜	1.0	B
2	西瓜	1.0	C
3	None	NaN	None

```

0    NaN
1    2.0
2    3.0
3    1.0
Name: 數量, dtype: float64

```

語法	說明
<code>for c in df.columns:</code>	走訪每個 column
<code>for c in df.index:</code>	走訪每個 index
<code>for index, row in df.iterrows():</code>	一行一行走訪

項目	程式碼
1	<code># -*- coding: utf-8 -*-</code>
2	<code>import pandas as pd</code>
3	<code>data_dict = {</code>
4	<code> "項目": ["蘋果", "橘子", "西瓜", "西瓜"],</code>
5	<code> "數量": [2, 3, 1, 1],</code>
6	<code> "儲藏倉庫": ["A", None, "B", "C"]</code>
7	<code>}</code>
8	
9	<code>df = pd.DataFrame(data_dict)</code>

引用前例的 DataFrame

項目	程式碼
11	for c in df.columns:
12	print(df[c])
13	print("-----")
14	
15	for i in df.index:
16	print(df.loc[i])
17	print("-----")

```

0  蘋果
1  橘子
2  西瓜
3  西瓜
Name: 項目, dtype: object
0  2
1  3
2  1
3  1
Name: 數量, dtype: int64
0  A
1  None
2  B
3  C
Name: 儲藏倉庫, dtype: object
    
```

```

項目    蘋果
數量    2
儲藏倉庫 A
Name: 0, dtype: object
項目    橘子
數量    3
儲藏倉庫 None
Name: 1, dtype: object
項目    西瓜
數量    1
儲藏倉庫 B
Name: 2, dtype: object
項目    西瓜
數量    1
儲藏倉庫 C
Name: 3, dtype: object
    
```

	項目	數量	儲藏倉庫
0	蘋果	2	A
1	橘子	3	
2	西瓜	1	B
3	西瓜	1	C

項目	程式碼
19	for index, row in df.iterrows():
20	print(index)
21	print("\\\\/\\\/\\\/\\\/")
22	print(row)
23	print("^^^\\^^^\\^^^\\^^")

0

```
\\\/\\\/\\\/\\\/
項目      蘋果
數量      2
儲藏倉庫   A
```

Name: 0, dtype: object
^^^^^^

1

```
\\\/\\\/\\\/\\\/
項目      橘子
數量      3
儲藏倉庫   None
```

Name: 1, dtype: object
^^^^^^

2

```
\\\/\\\/\\\/\\\/
項目      西瓜
數量      1
儲藏倉庫   B
```

Name: 2, dtype: object
^^^^^^

3

```
\\\/\\\/\\\/\\\/
項目      西瓜
數量      1
儲藏倉庫   C
```

Name: 3, dtype: object
^^^^^^

	項目	數量	儲藏倉庫
0	蘋果	2	A
1	橘子	3	
2	西瓜	1	B
3	西瓜	1	C

項目	程式碼
25	for index, row in df.iterrows():
26	for c in df.columns:
27	print("%10s%2s" % ("index: ", index) , end = "")
28	print("%10s%5s" % ("column: ", c) , end = "")
29	print("%10s%5s" % ("內容: ", row[c]))

	項目	數量	儲藏倉庫
0	蘋果	2	A
1	橘子	3	
2	西瓜	1	B
3	西瓜	1	C

```

index: 0  column: 項目  內容: 蘋果
index: 0  column: 數量  內容: 2
index: 0  column: 儲藏倉庫  內容: A
index: 1  column: 項目  內容: 橘子
index: 1  column: 數量  內容: 3
index: 1  column: 儲藏倉庫  內容: None
index: 2  column: 項目  內容: 西瓜
index: 2  column: 數量  內容: 1
index: 2  column: 儲藏倉庫  內容: B
index: 3  column: 項目  內容: 西瓜
index: 3  column: 數量  內容: 1
index: 3  column: 儲藏倉庫  內容: C

```


常用格式	語法
EXCEL	<code>df = pd.to_excel (file_path, header = True, index = True)</code>
CSV	<code>df = pd.to_csv (file_path, header = True, index = True)</code>

項目	程式碼
1	# -*- coding: utf-8 -*-
2	import pandas as pd
3	
4	df = pd.read_excel("0050.xlsx")
5	df.describe().to_excel("describe_df.xlsx")
6	
7	df = df.loc[:, ["日期","收盤價"]].iloc[2:10:2]
8	df.to_excel("processed_df.xlsx")

篩選(processed_df.xlsx)		
	日期	收盤價
2	2021-01-06 00:00:00	125.95
4	2021-01-08 00:00:00	131.2
6	2021-01-12 00:00:00	132.1
8	2021-01-14 00:00:00	134.05

敘述統計(describe_df.xlsx)							
	成交股數	成交金額	開盤價	最高價	最低價	收盤價	成交筆數
count	20	20	20	20	20	20	20
mean	13626449.55	1823982038	132.51	133.8375	131.1325	132.715	13071.2
std	6148015.832	869350532.2	4.779715913	4.961400349	4.467699131	4.777168838	6206.249479
min	4962514	616480760	122.2	124.5	122	124.35	5434
25%	10768017.25	1373328329	130.45	131.2875	129.1	129.9875	8759.75
50%	12844552	1704758950	133.025	134.675	131.95	132.95	11294.5
75%	14802448	1983931365	135.95	137.475	134.25	135.375	16607.75
max	30367315	4202770386	138.4	143.25	138.35	143	26235

項目	內容
標的	0050
時間	2021 年 1 月
策略	收盤價大於 130 時，隔日以開盤價買進一張，持有 3 天後收盤價賣出
KPI	總獲益、標準差、夏普指數、平均報酬率...等



實務範例

項目	程式碼	日期	收盤價	隔日開盤價	賣出收盤價
1	# -*- coding: utf-8 -*-	0 2021-01-04	124.35	124.20	128.90
2	import pandas as pd	1 2021-01-05	124.60	126.10	131.20
3		2 2021-01-06	125.95	126.55	132.30
4	df = pd.read_excel("0050.xlsx")	3 2021-01-07	128.90	130.60	132.10
5		4 2021-01-08	131.20	131.20	135.30
6	df["隔日開盤價"] = df["開盤價"].shift(-1) # 隔天買入的開盤價	5 2021-01-11	132.30	132.30	134.05
7	df["賣出收盤價"] = df["收盤價"].shift(-3) # 出場時的收盤價	6 2021-01-12	132.10	132.75	134.50
8		7 2021-01-13	135.30	135.00	134.40
9	df = df.loc[:,["日期", "收盤價", "隔日開盤價", "賣出收盤價"]] # 選擇本次策略會用到的欄位，節省運算時間與降低資料複雜度	8 2021-01-14	134.05	137.15	137.95
10	print(df)	9 2021-01-15	134.50	134.60	138.20
11	print("-----")	10 2021-01-18	134.40	135.20	143.00
		11 2021-01-19	137.95	138.10	137.45
		12 2021-01-20	138.20	138.40	135.60
		13 2021-01-21	143.00	138.20	132.85
		14 2021-01-22	137.45	137.20	133.05
		15 2021-01-25	135.60	135.55	130.35
		16 2021-01-26	132.85	133.30	128.20
		17 2021-01-27	133.05	130.00	NaN
		18 2021-01-28	130.35	131.60	NaN
		19 2021-01-29	128.20	NaN	NaN

項目	程式碼
12	
13	<code>df = df.dropna()</code> # 去除空值，空的隔日開盤價與賣出收盤價無法計算損益
14	<code>print(df)</code>
15	<code>print("-----")</code>
16	
17	<code>df = df[df["收盤價"] > 130]</code> # 篩選收盤價大於 130 的資料
18	<code>print(df)</code>
19	<code>print("-----")</code>

df = df.dropna()

	日期	收盤價	隔日開盤價	賣出收盤價
0	2021-01-04	124.35	124.20	128.90
1	2021-01-05	124.60	126.10	131.20
2	2021-01-06	125.95	126.55	132.30
3	2021-01-07	128.90	130.60	132.10
4	2021-01-08	131.20	131.20	135.30
5	2021-01-11	132.30	132.30	134.05
6	2021-01-12	132.10	132.75	134.50
7	2021-01-13	135.30	135.00	134.40
8	2021-01-14	134.05	137.15	137.95
9	2021-01-15	134.50	134.60	138.20
10	2021-01-18	134.40	135.20	143.00
11	2021-01-19	137.95	138.10	137.45
12	2021-01-20	138.20	138.40	135.60
13	2021-01-21	143.00	138.20	132.85
14	2021-01-22	137.45	137.20	133.05
15	2021-01-25	135.60	135.55	130.35
16	2021-01-26	132.85	133.30	128.20

df[df ["收盤價"] > 130]

	日期	收盤價	隔日開盤價	賣出收盤價
4	2021-01-08	131.20	131.20	135.30
5	2021-01-11	132.30	132.30	134.05
6	2021-01-12	132.10	132.75	134.50
7	2021-01-13	135.30	135.00	134.40
8	2021-01-14	134.05	137.15	137.95
9	2021-01-15	134.50	134.60	138.20
10	2021-01-18	134.40	135.20	143.00
11	2021-01-19	137.95	138.10	137.45
12	2021-01-20	138.20	138.40	135.60
13	2021-01-21	143.00	138.20	132.85
14	2021-01-22	137.45	137.20	133.05
15	2021-01-25	135.60	135.55	130.35
16	2021-01-26	132.85	133.30	128.20



實務範例

項目	程式碼	df["損益"]				
21	df["損益"] = (df["賣出收盤價"] - df["隔日開盤價"]) * 1000 # 計算損益					
22	print(df)					
23	print("-----")					
24						
25	total_profit = df["損益"].sum() # 將損益加總					
26	avg_profit = df["損益"].mean() # 將損益平均					
27	trade_times = df["損益"].count() # 交易次數					
28						
29	print("總損益", total_profit)					
30	print("平均損益", avg_profit)					
31	print("交易次數", trade_times)					

df["損益"]

	日期	收盤價	隔日開盤價	賣出收盤價	損益
4	2021-01-08	131.20	131.20	135.30	3314.33750
5	2021-01-11	132.30	132.30	134.05	968.30125
6	2021-01-12	132.10	132.75	134.50	965.66875
7	2021-01-13	135.30	135.00	134.40	-1387.09500
8	2021-01-14	134.05	137.15	137.95	-5.86750
9	2021-01-15	134.50	134.60	138.20	2796.66000
10	2021-01-18	134.40	135.20	143.00	6974.56500
11	2021-01-19	137.95	138.10	137.45	-1455.00875
12	2021-01-20	138.20	138.40	135.60	-3597.25000
13	2021-01-21	143.00	138.20	132.85	-6134.79625
14	2021-01-22	137.45	137.20	133.05	-4934.25625
15	2021-01-25	135.60	135.55	130.35	-5969.95750
16	2021-01-26	132.85	133.30	128.20	-5857.23750

KPI 指標

總損益 -4050.000000000012

平均損益 -311.53846153846246

交易次數 13



章節到此結束，有任何問題歡迎提出來討論！

