



Pandas資料視覺化

製作人：黃宥輔

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]

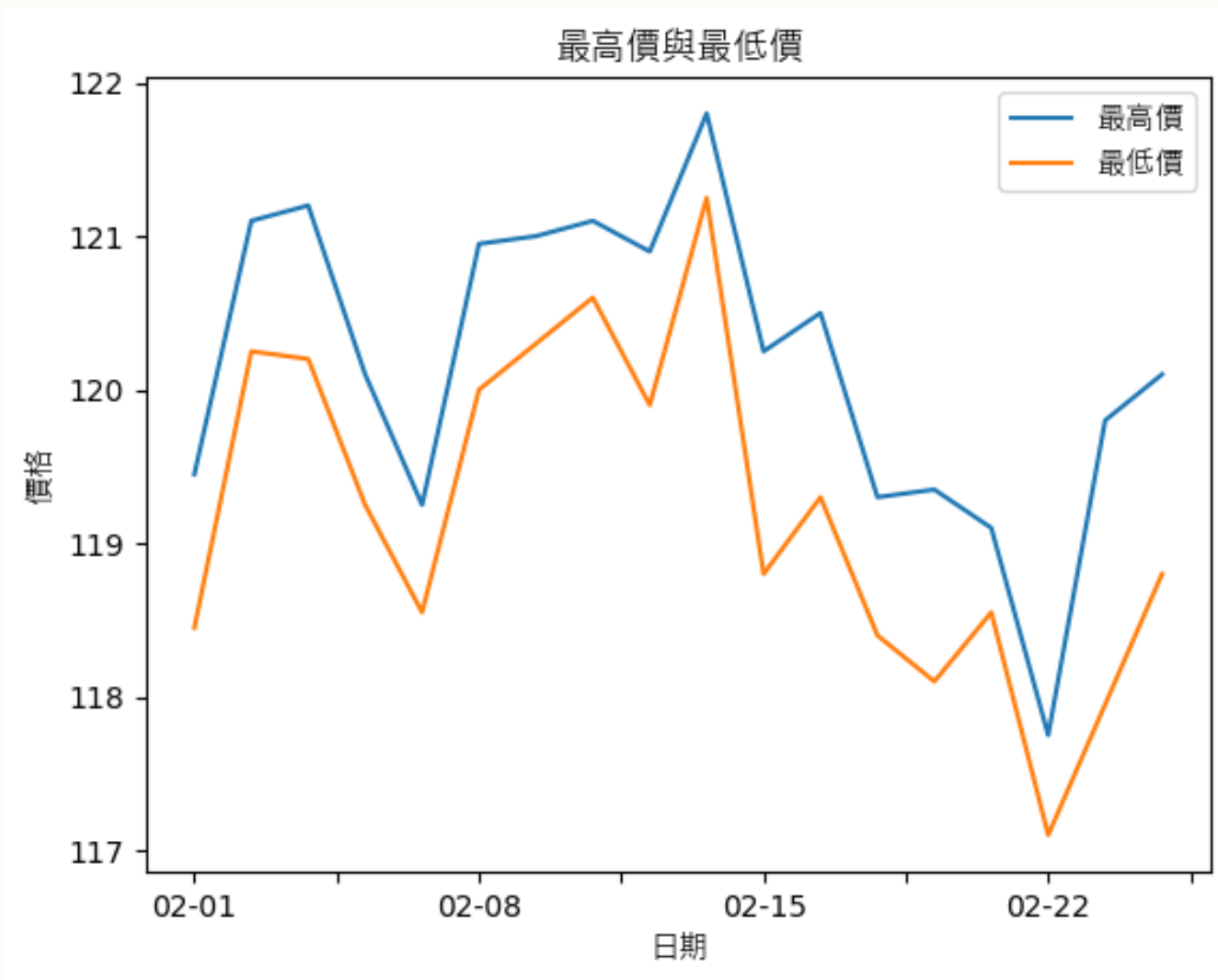
- Pandas可以結合繪圖工具，以plot方法將DataFrame繪製成圖表，預設是使用Matplotlib套件，常用的有下列幾種圖形

代號	圖表類型
line	折線圖
bar	柱狀圖(長條圖)
barh	橫向柱狀圖
hist	直方分布圖
area	區域圖
pie	圓餅圖
scatter	散布圖(二維, DataFrame)

文件：<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html>

使用資料

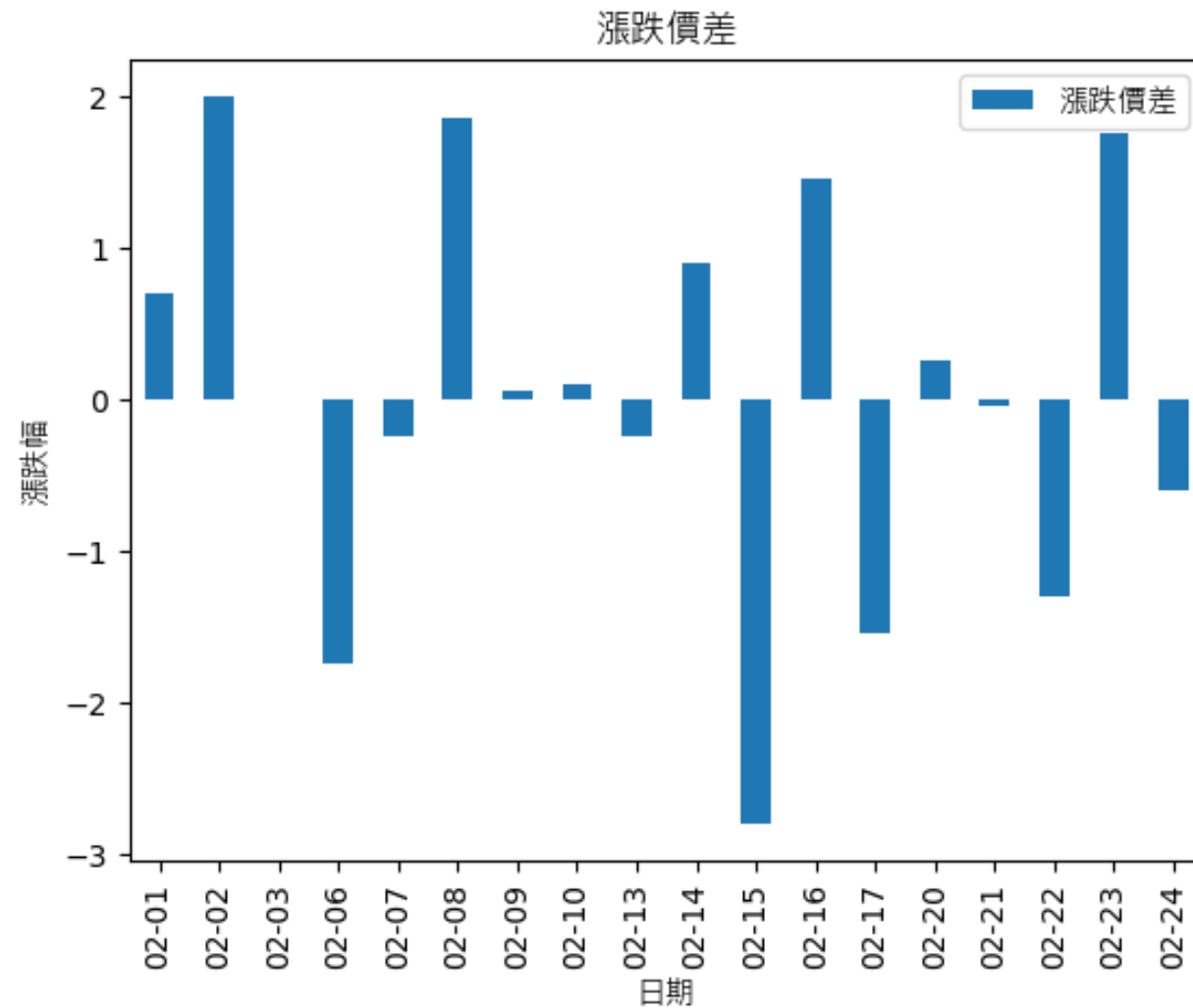
日期	最高價	最低價
2023/02/01	119.45	118.45
2023/02/02	121.1	120.25
2023/02/03	121.2	120.2
2023/02/06	120.1	119.25
2023/02/07	119.25	118.55
2023/02/08	120.95	120
2023/02/09	121	120.3
2023/02/10	121.1	120.6
2023/02/13	120.9	119.9
2023/02/14	121.8	121.25
2023/02/15	120.25	118.8
2023/02/16	120.5	119.3
2023/02/17	119.3	118.4
2023/02/20	119.35	118.1
2023/02/21	119.1	118.55
2023/02/22	117.75	117.1
2023/02/23	119.8	117.95
2023/02/24	120.1	118.8



```
1  # -*- coding: utf-8 -*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      'font.size': 10, # 文字大小
8      "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif'為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
16
17 # df["日期"] = df["日期"].dt.date # 只保留日期
18 df["日期"] = df["日期"].dt.strftime('%m-%d') # 只保留月與日
19
20 # 繪圖
21 df.plot(kind="line", # 圖形類別
22         x="日期", # x軸設定
23         y=['最高價', '最低價'] # y軸設定
24        )
25
26 plt.title('最高價與最低價') # 圖表標題
27 plt.xlabel('日期') # x軸標籤
28 plt.ylabel('價格') # y軸標籤
29 plt.show() # 顯示圖片
```


使用資料

日期	漲跌價差
2023/02/01	0.7
2023/02/02	2
2023/02/03	0
2023/02/06	-1.75
2023/02/07	-0.25
2023/02/08	1.85
2023/02/09	0.05
2023/02/10	0.1
2023/02/13	-0.25
2023/02/14	0.9
2023/02/15	-2.8
2023/02/16	1.45
2023/02/17	-1.55
2023/02/20	0.25
2023/02/21	-0.05
2023/02/22	-1.3
2023/02/23	1.75
2023/02/24	-0.6

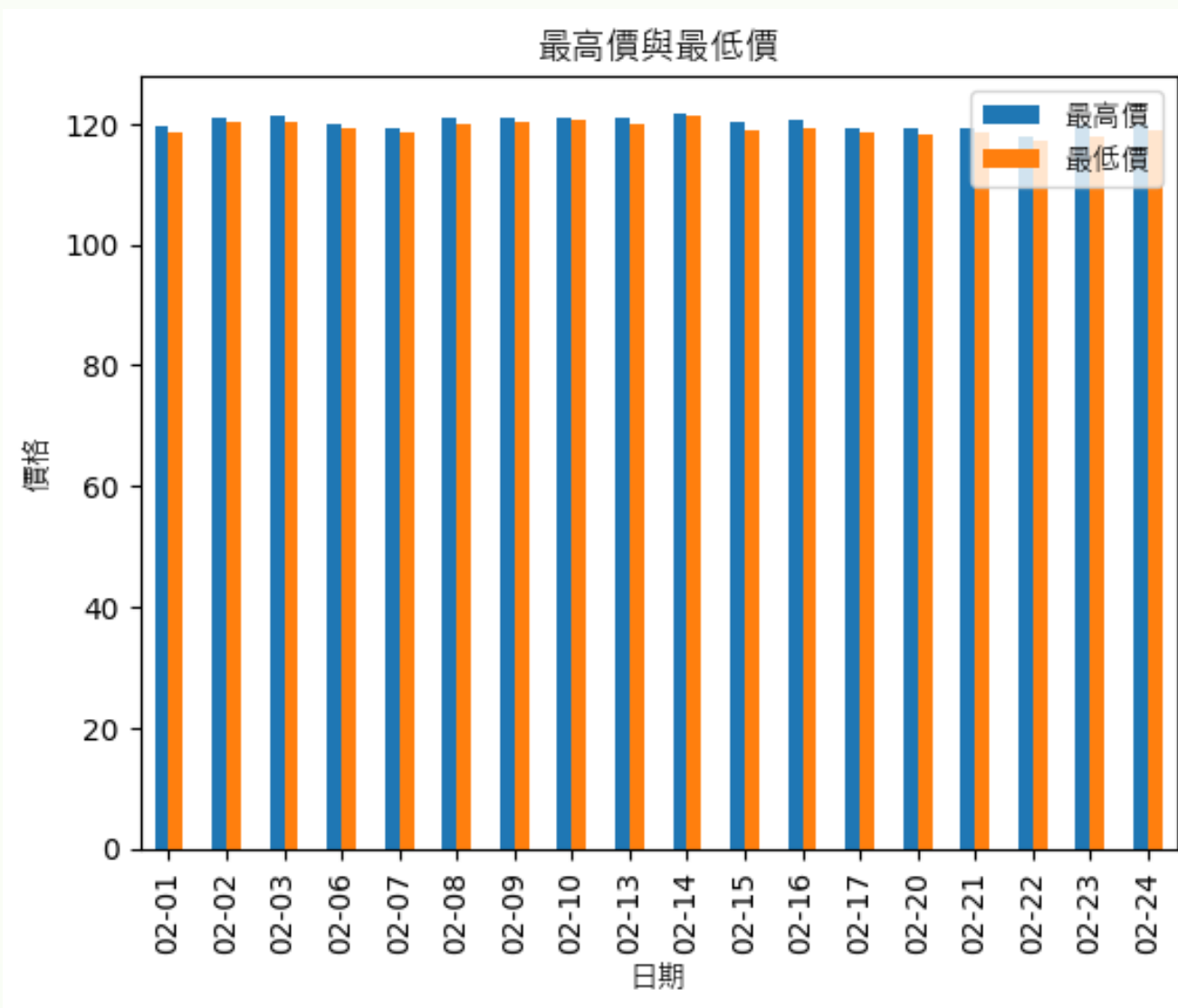


```

1  # -*- coding: utf-8 -*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      'font.size': 10, # 文字大小
8      "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif'為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
16
17 # df["日期"] = df["日期"].dt.date # 只保留日期
18 df["日期"] = df["日期"].dt.strftime('%m-%d') # 只保留月與日
19
20 # 繪圖
21 df.plot(kind="bar", # 圖形類別
22         x="日期", # x軸設定
23         y='漲跌價差' # y軸設定
24        )
25
26 plt.title('漲跌價差') # 圖表標題
27 plt.xlabel('日期') # x軸標籤
28 plt.ylabel('漲跌幅') # y軸標籤
29 plt.show() # 顯示圖片
    
```


使用資料

日期	最高價	最低價
2023/02/01	119.45	118.45
2023/02/02	121.1	120.25
2023/02/03	121.2	120.2
2023/02/06	120.1	119.25
2023/02/07	119.25	118.55
2023/02/08	120.95	120
2023/02/09	121	120.3
2023/02/10	121.1	120.6
2023/02/13	120.9	119.9
2023/02/14	121.8	121.25
2023/02/15	120.25	118.8
2023/02/16	120.5	119.3
2023/02/17	119.3	118.4
2023/02/20	119.35	118.1
2023/02/21	119.1	118.55
2023/02/22	117.75	117.1
2023/02/23	119.8	117.95
2023/02/24	120.1	118.8



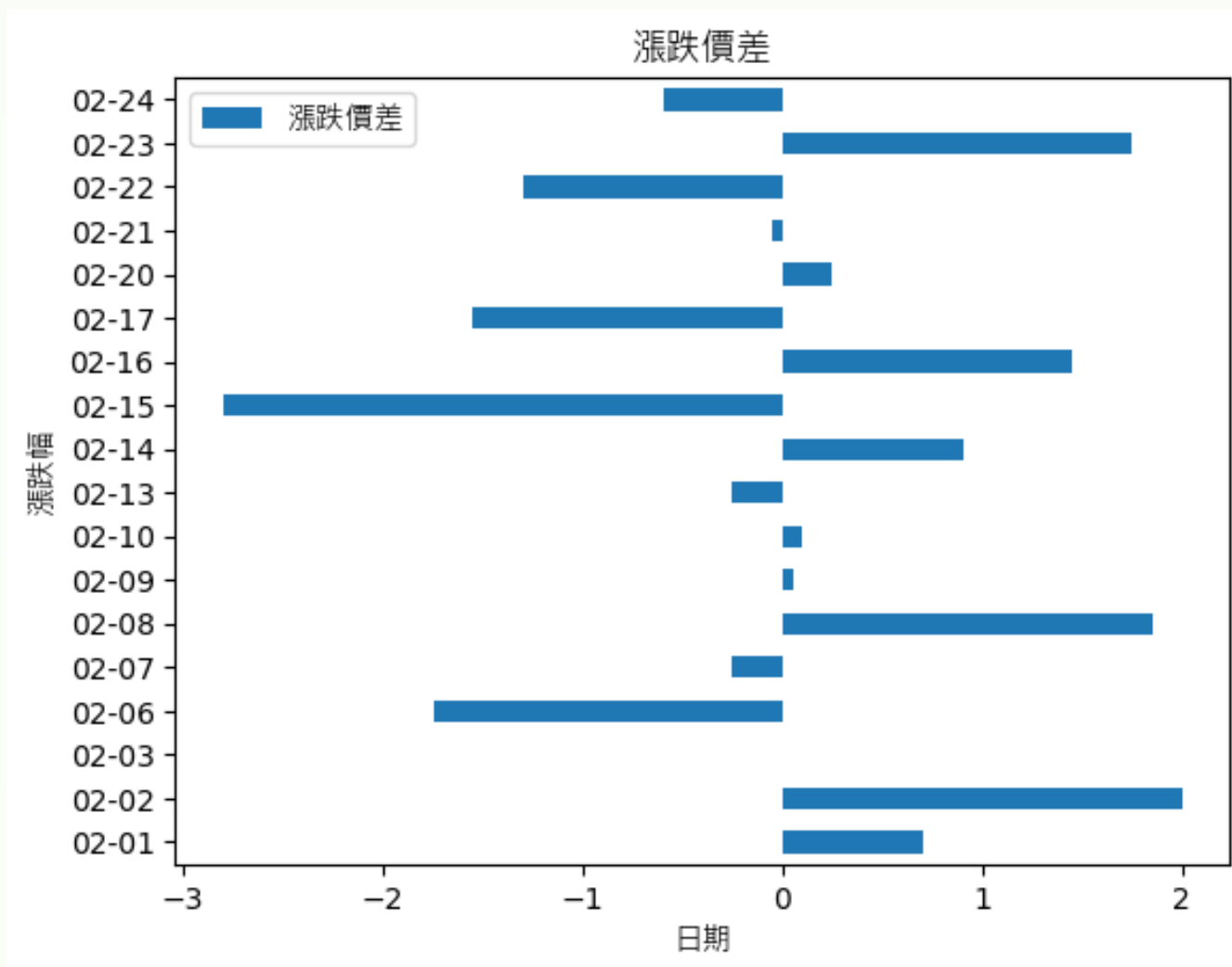
```
1  # -*- coding: utf-8 -*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      'font.size': 10, # 文字大小
8      "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif'為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
16
17 # df["日期"] = df["日期"].dt.date # 只保留日期
18 df["日期"] = df["日期"].dt.strftime('%m-%d') # 只保留月與日
19
20 # 繪圖
21 df.plot(kind="bar", # 圖形類別
22         x="日期", # x軸設定
23         y=['最高價', '最低價'] # y軸設定
24         )
25
26 plt.title('最高價與最低價') # 圖表標題
27 plt.xlabel('日期') # x軸標籤
28 plt.ylabel('價格') # y軸標籤
29 plt.show() # 顯示圖片
```



Pandas繪圖 - barh

使用資料

日期	漲跌價差
2023/02/01	0.7
2023/02/02	2
2023/02/03	0
2023/02/06	-1.75
2023/02/07	-0.25
2023/02/08	1.85
2023/02/09	0.05
2023/02/10	0.1
2023/02/13	-0.25
2023/02/14	0.9
2023/02/15	-2.8
2023/02/16	1.45
2023/02/17	-1.55
2023/02/20	0.25
2023/02/21	-0.05
2023/02/22	-1.3
2023/02/23	1.75
2023/02/24	-0.6





Pandas繪圖 - barh

```

1  # -*- coding: utf-8 -*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      ... 'font.size': 10, # 文字大小
8      ... "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif'為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
16
17 # df["日期"] = df["日期"].dt.date # 只保留日期
18 df["日期"] = df["日期"].dt.strftime('%m-%d') # 只保留月與日
19
20 # 繪圖
21 df.plot(kind="barh", # 圖形類別
22         x="日期", # x軸設定
23         y='漲跌價差' # y軸設定
24        )
25
26 plt.title('漲跌價差') # 圖表標題
27 plt.xlabel('日期') # x軸標籤
28 plt.ylabel('漲跌幅') # y軸標籤
29 plt.show() # 顯示圖片

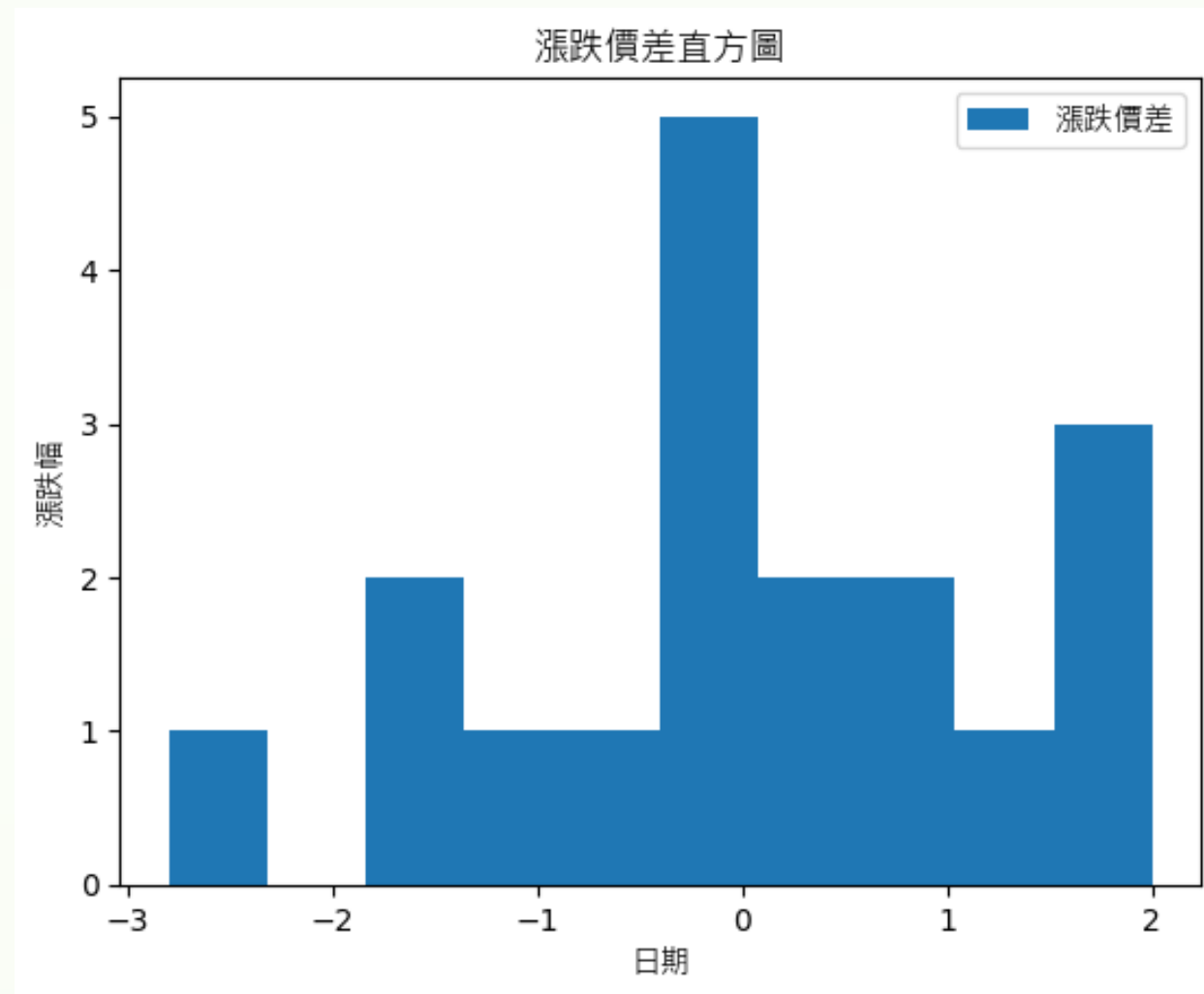
```



Pandas繪圖 - hist

使用資料

日期	漲跌價差
2023/02/01	0.7
2023/02/02	2
2023/02/03	0
2023/02/06	-1.75
2023/02/07	-0.25
2023/02/08	1.85
2023/02/09	0.05
2023/02/10	0.1
2023/02/13	-0.25
2023/02/14	0.9
2023/02/15	-2.8
2023/02/16	1.45
2023/02/17	-1.55
2023/02/20	0.25
2023/02/21	-0.05
2023/02/22	-1.3
2023/02/23	1.75
2023/02/24	-0.6





Pandas繪圖 - hist

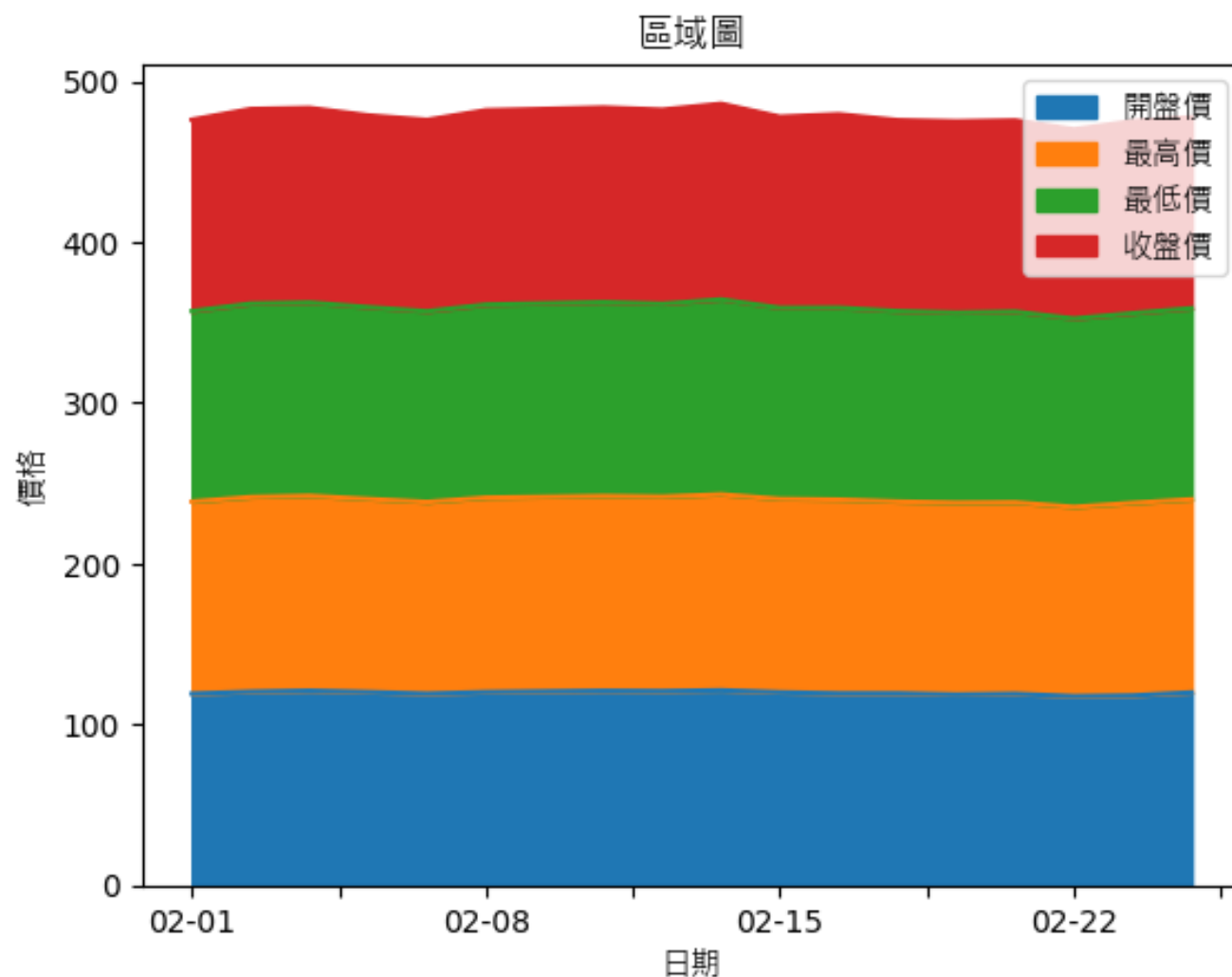
```

1  # -*- coding: utf-8 -*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      'font.size': 10, # 文字大小
8      "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif'為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
16
17 # df["日期"] = df["日期"].dt.date # 只保留日期
18 df["日期"] = df["日期"].dt.strftime('%m-%d') # 只保留月與日
19
20 # 繪圖
21 df.plot(kind="hist", # 圖形類別
22         x="日期", # x軸設定
23         y='漲跌價差', # y軸設定
24         bins=10, # bin為要彙總成多少個柱狀圖
25         )
26
27 plt.title('漲跌價差直方圖') # 圖表標題
28 plt.xlabel('日期') # x軸標籤
29 plt.ylabel('漲跌幅') # y軸標籤
30 plt.show() # 顯示圖片

```


使用資料

日期	開盤價	最高價	最低價	收盤價
2023/02/01	119.1	119.45	118.45	119
2023/02/02	120.3	121.1	120.25	121
2023/02/03	120.95	121.2	120.2	121
2023/02/06	120.1	120.1	119.25	119.25
2023/02/07	119.1	119.25	118.55	119
2023/02/08	120.1	120.95	120	120.85
2023/02/09	120.5	121	120.3	120.9
2023/02/10	120.9	121.1	120.6	121
2023/02/13	120.7	120.9	119.9	120.75
2023/02/14	121.25	121.8	121.25	121.65
2023/02/15	119.95	120.25	118.8	118.85
2023/02/16	119.3	120.5	119.3	120.3
2023/02/17	119.3	119.3	118.4	118.75
2023/02/20	118.55	119.35	118.1	119
2023/02/21	119	119.1	118.55	118.95
2023/02/22	117.5	117.75	117.1	117.65
2023/02/23	117.95	119.8	117.95	119.4
2023/02/24	119.9	120.1	118.8	118.8



```
1  #-*-coding:utf-8-*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      'font.size': 10, # 文字大小
8      "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif'為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
16
17 # df["日期"] = df["日期"].dt.date # 只保留日期
18 df["日期"] = df["日期"].dt.strftime('%m-%d') # 只保留月與日
19
20 # 繪圖
21 df.plot(kind="area", # 圖形類別
22         x="日期", # x軸設定
23         y=['開盤價', '最高價', '最低價', '收盤價'], # y軸設定
24         )
25
26 plt.title('區域圖') # 圖表標題
27 plt.xlabel('日期') # x軸標籤
28 plt.ylabel('價格') # y軸標籤
29 plt.show() # 顯示圖片
```

Pandas繪圖 - pie

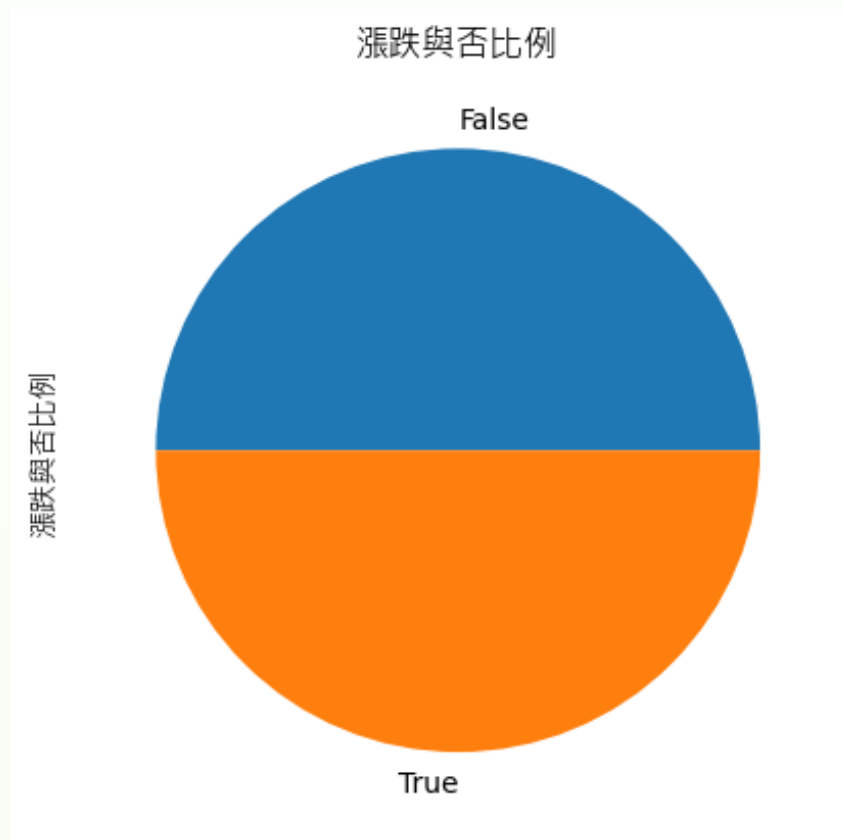
使用資料

漲跌價差
0.7
2
0
-1.75
-0.25
1.85
0.05
0.1
-0.25
0.9
-2.8
1.45
-1.55
0.25
-0.05
-1.3
1.75
-0.6

漲跌與否
TRUE
TRUE
FALSE
FALSE
FALSE
TRUE
TRUE
TRUE
FALSE
TRUE
FALSE
TRUE
FALSE
FALSE
TRUE
FALSE

漲跌與否	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
FALSE	9	9	9	9	9	9	9	9	9
TRUE	9	9	9	9	9	9	9	9	9

漲跌與否	日期
FALSE	9
TRUE	9



```
1  #-*-coding:utf-8-*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      ... 'font.size': 10, # 文字大小
8      ... "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif'為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
16
17 #df["日期"] = df["日期"].dt.date # 只保留日期
18 df["日期"] = df["日期"].dt.strftime('%m-%d') # 只保留月與日
19
20 df.index = df["日期"] # 將索引換成日期，以便繪圖時被設定為x軸的標籤
```

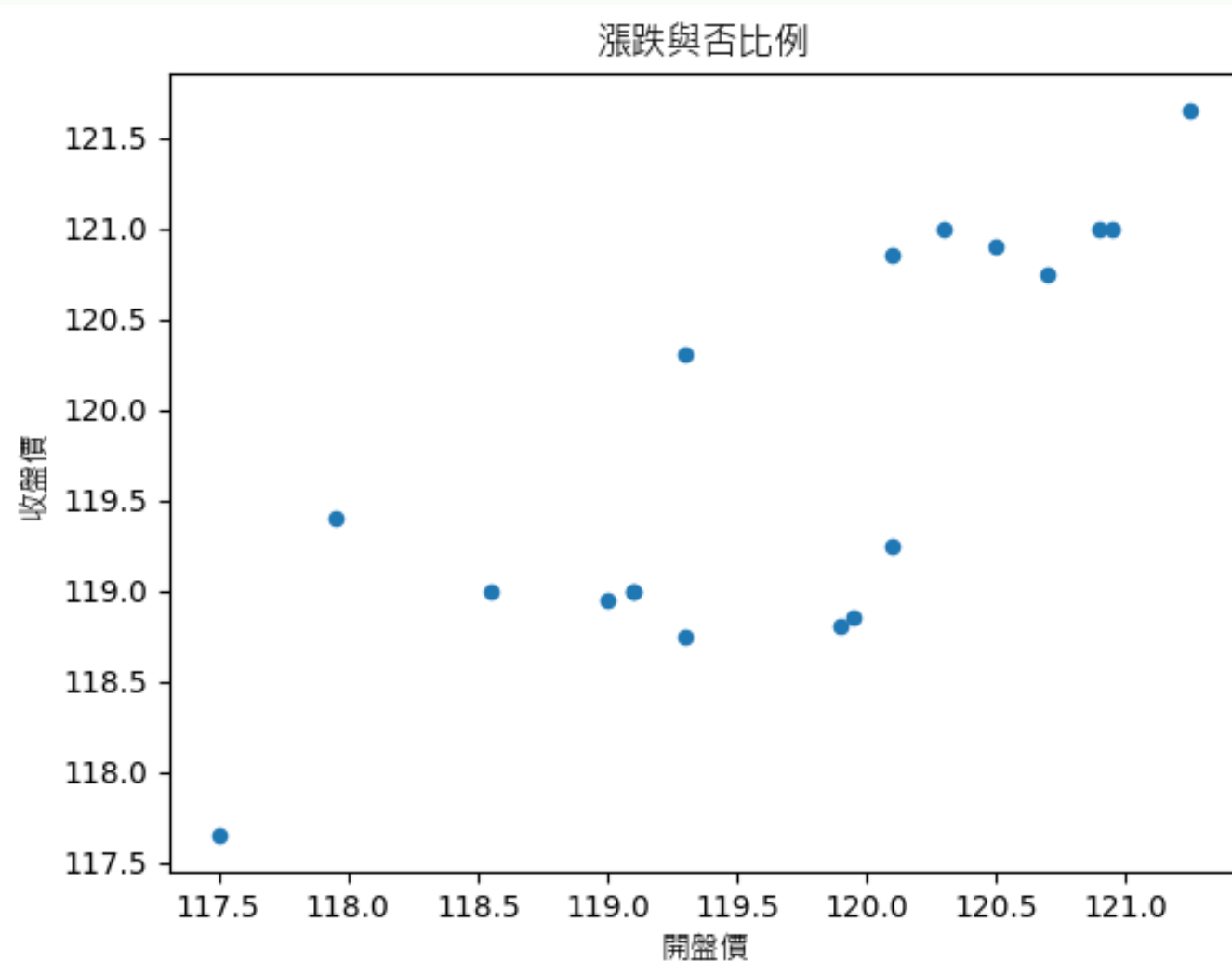
```
22 # 統計漲跌
23 df["漲跌與否"] = df["漲跌價差"] > 0 # 判斷是否有上漲
24 df_group = df.groupby(by = ['漲跌與否']) # 根據是否有上漲進行分組
25 df_rise_count = df_group.count() # 將各組進行數量計算
26
27 print(df_rise_count) # 會將每一欄進行統計
28 df_rise_count.to_excel("123.xlsx")
29 print(df_rise_count['日期']) # 因為數值都一樣，所以抓其中一欄進行繪圖即可
30
31 # 進行圓餅圖繪圖
32 df_rise_count['日期'].plot(kind = "pie")
33
34 plt.title('漲跌與否比例') # 圖表標題
35 plt.ylabel('漲跌與否比例') # Y軸標籤
36 plt.show() # 顯示圖片
```



Pandas繪圖 - scatter

使用資料

開盤價	收盤價
119.1	119
120.3	121
120.95	121
120.1	119.25
119.1	119
120.1	120.85
120.5	120.9
120.9	121
120.7	120.75
121.25	121.65
119.95	118.85
119.3	120.3
119.3	118.75
118.55	119
119	118.95
117.5	117.65
117.95	119.4
119.9	118.8





Pandas繪圖 - scatter

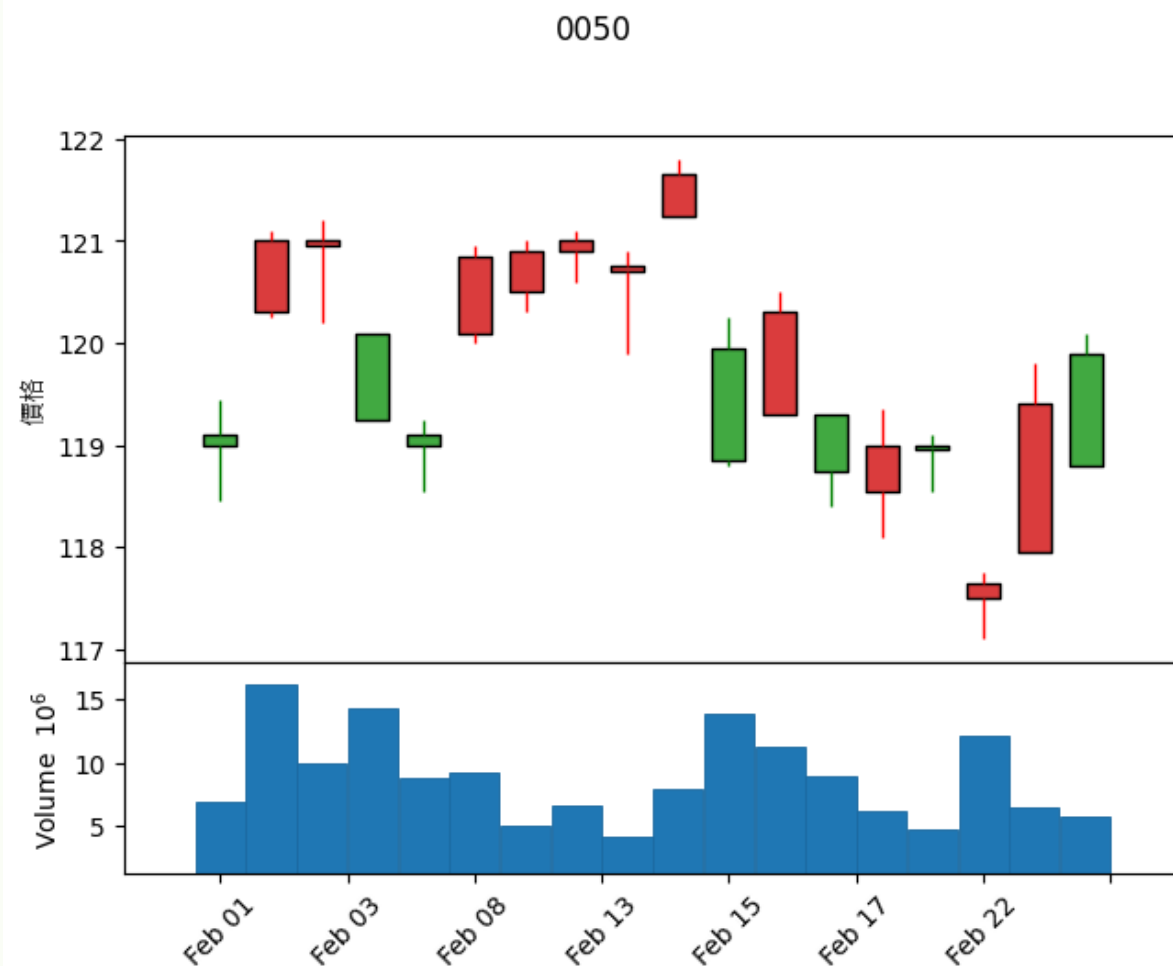
```
1  # -*- coding: utf-8 -*-
2  import pandas as pd
3  import matplotlib.pyplot as plt
4
5  # 設定繪圖的參數
6  plt.rcParams.update({
7      .... 'font.size': 10, # 文字大小
8      .... "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
9  })
10 # 'sans-serif' 為英文字體，不加入的話負號會顯示不出來
11 # 微軟正黑體 Microsoft JhengHei
12 # 標楷體 DFKai-SB
13
14 df = pd.read_excel("0050.xlsx") # 讀取資料
15
16 df.plot(kind="scatter",
17         .... x="開盤價",
18         .... y="收盤價",
19         .... )
20
21 plt.title('漲跌與否比例') # 圖表標題
22 plt.show() # 顯示圖片
```



Pandas繪圖 - candlesticks

使用資料

日期	開盤價	最高價	最低價	收盤價	成交股數
2023/02/01	119.1	119.45	118.45	119	6,903,171
2023/02/02	120.3	121.1	120.25	121	16,194,958
2023/02/03	120.95	121.2	120.2	121	9,934,984
2023/02/06	120.1	120.1	119.25	119.25	14,297,030
2023/02/07	119.1	119.25	118.55	119	8,787,291
2023/02/08	120.1	120.95	120	120.85	9,239,321
2023/02/09	120.5	121	120.3	120.9	5,032,245
2023/02/10	120.9	121.1	120.6	121	6,572,526
2023/02/13	120.7	120.9	119.9	120.75	4,149,906
2023/02/14	121.25	121.8	121.25	121.65	7,965,786
2023/02/15	119.95	120.25	118.8	118.85	13,852,368
2023/02/16	119.3	120.5	119.3	120.3	11,233,273
2023/02/17	119.3	119.3	118.4	118.75	8,960,233
2023/02/20	118.55	119.35	118.1	119	6,230,136
2023/02/21	119	119.1	118.55	118.95	4,772,315
2023/02/22	117.5	117.75	117.1	117.65	12,041,784
2023/02/23	117.95	119.8	117.95	119.4	6,448,658
2023/02/24	119.9	120.1	118.8	118.8	5,828,709



```
1 # -*- coding: utf-8 -*-
2 import pandas as pd
3 import mplfinance as fplt
4
5 df = pd.read_excel("0050.xlsx") # 讀取資料
6
7 df["日期"] = pd.to_datetime(df["日期"]) # 將文字轉datetime格式
8 df.index = df["日期"]
9
10 print(df.columns)
11 # 重新命名欄為名稱，因為套件只接受固定的欄位名稱
12 df.rename(columns={"開盤價": "Open",
13                  "最高價": "High",
14                  "最低價": "Low",
15                  "收盤價": "Close",
16                  "成交股數": "Volume"},
17           inplace=True)
18
19 # 調整圖表標示顏色
20 mc = fplt.make_marketcolors(
21     up='tab:red', down='tab:green', # 上漲為紅，下跌為綠
22     wick={'up': 'red', 'down': 'green'}, # 影線上漲為紅，下跌為綠
23     volume='tab:blue', # 交易量顏色
24 )
```

```
26 # 定義圖表風格
27 s = fplt.make_mpf_style(marketcolors = mc,
28 rc = {
29     'font.size': 10, # 文字大小
30     "font.family": ['sans-serif', "Microsoft JhengHei"] # 字型
31 }
32 )
33
34 fplt.plot(
35     df, # 開高低收量的資料
36     type = 'candle', # 類型為蠟燭圖，也就是 K 線圖
37     style = s, # 套用圖表風格
38     title = "0050", # 設定圖表標題
39     ylabel = '價格', # 設定 Y 軸標題
40     volume = True,
41     # savefig = 'stock_kbar.png', # 儲存檔案
42 )
```

再圖片中加入savefig，即可儲存圖片，
但不會顯示再spyder中

雷達圖



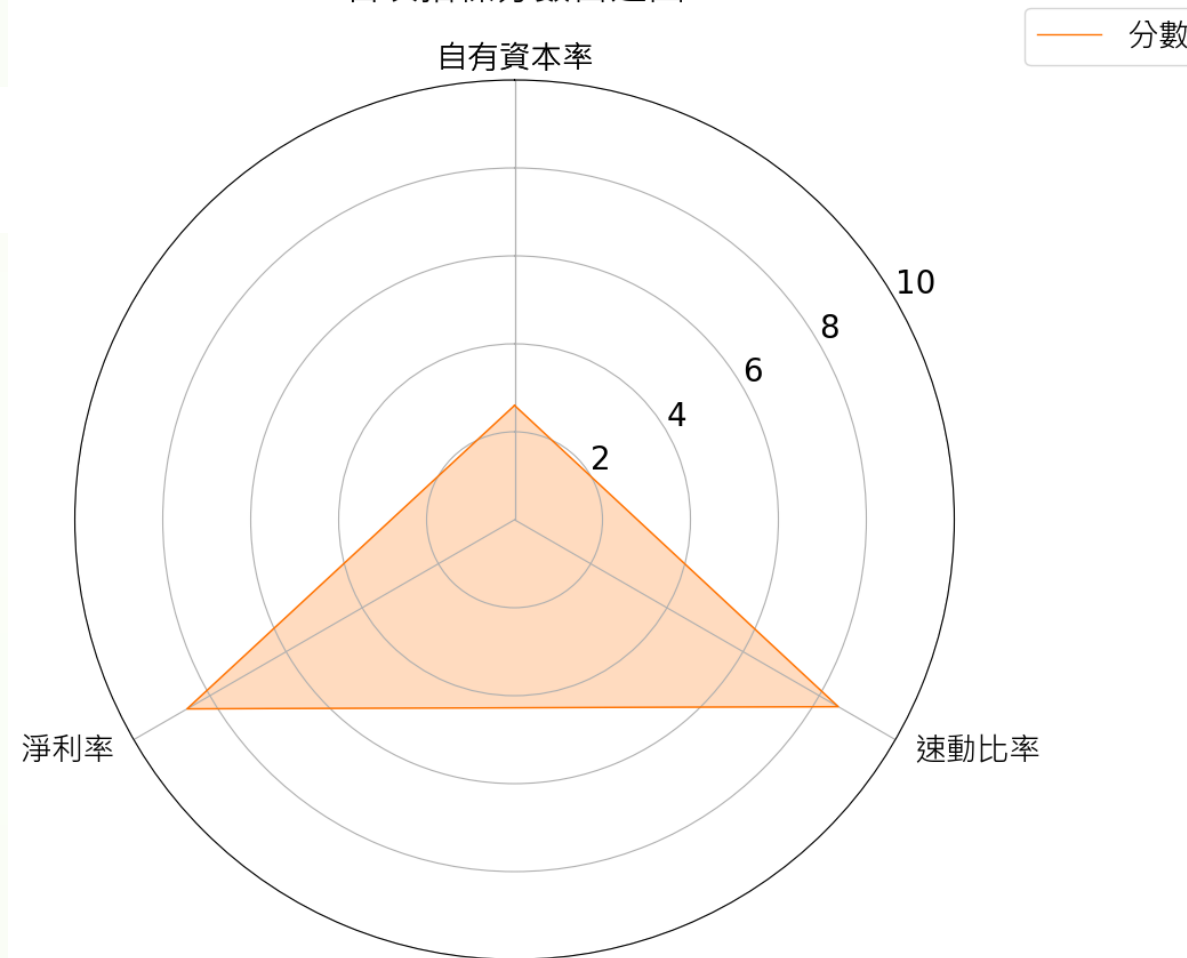
AI.FINTECH

AI 金融 科技 中心

使用資料

```
labels = ['自有資本率', '速動比率', '淨利率']  
values = [2.6, 8.5, 8.6]
```

各項指標分數雷達圖



```
1 # -*- coding: utf-8 -*-
2 #https://www.pythoncharts.com/matplotlib/radar-charts/
3
4 # coding=utf-8
5 import numpy as np
6 import matplotlib.pyplot as plt
7 import matplotlib.font_manager as fm
8
9 def main(labels, values):
10     plt.rcParams.update({'font.size': 20})
11     ....
12     font_path = "mingliu.ttc" → #中文字型路徑
13     font_prop = fm.FontProperties(fname=font_path) → → #調整字型
14     ....
15     # 項目標題
16     #labels = ['Acceleration', 'Displacement', 'Horsepower', 'MPG', 'Weight']
17     # 有幾個項目
18     num_vars = len(labels)
19     # 雷達圖各個項目角度
20     angles = np.linspace(0, 2 * np.pi, num_vars, endpoint=False).tolist()
21     # 將第一個角度放到最後，這樣圖形才會閉鎖
22     angles.append(angles[0])
23     # 設定圖表的大小與座標軸
24     fig, ax = plt.subplots(figsize=(10, 10), subplot_kw=dict(polar=True))
```



```
9 def main(labels, values):  
    ...  
26     # 將資料放入圖表  
27     def add_to_radar(label, color):  
28         # values = [10, 20, 30, 40, 50] # 加入資料  
29         values.append(values[0]) # 在最後加入第一筆資料，讓圖形閉合  
30         ...  
31         ax.plot(angles, values, color=color, linewidth=1, label=label)  
32         ax.fill(angles, values, color=color, alpha=0.25)  
33         ...  
34         # Add each car to the chart.  
35         add_to_radar('score', '#1aaf6c')  
36         # add_to_radar('peugeot 504 1979', '#429bf4')  
37         # add_to_radar('ford granada 1977', '#d42cea')  
38         ...  
39         # Fix axis to go in the right order and start at 12 o'clock.  
40         ax.set_theta_offset(np.pi / 2)  
41         ax.set_theta_direction(-1)  
42         ...  
43         # 將標籤名稱加上第一筆資料配合圖表閉合  
44         labels.append(labels[0])  
45         # 設定  
46         ax.set_thetagrids(np.degrees(angles), labels, fontproperties=font_prop)
```

```
9 def main(labels, values):  
    ...  
48     ... # 根據角度調整圖表項目標籤的位置  
49     ... for label, angle in zip(ax.get_xticklabels(), angles):  
50     ...     if angle in (0, np.pi):  
51     ...         label.set_horizontalalignment('center')  
52     ...     elif 0 < angle < np.pi:  
53     ...         label.set_horizontalalignment('left')  
54     ...     else:  
55     ...         label.set_horizontalalignment('right')  
56     ...  
57     ... # 設定數值的上下限  
58     ... ax.set_ylim(0, 10)  
59     ...  
60     ... # 將Y數值的數值標籤，置中在第一與第二個項目之間  
61     ... ax.set_rlabel_position(180 / num_vars)  
62     ...  
63     ... # Add title.  
64     ... ax.set_title('各項指標分數雷達圖', y=1.08, fontproperties=font_prop)  
65     ...  
66     ... # Add a legend as well.  
67     ... ax.legend(loc='upper right', bbox_to_anchor=(1.3, 1.1))  
68     ...  
69     ... return fig
```

```
71 if __name__ == "__main__":  
72     labels = ['自有資本率', '速動比率', '淨利率']  
73     values = [2.6, 8.5, 8.6]  
74     fig = main(labels, values)  
75     fig.show()
```



章節到此結束，有任何問題歡迎提出來討論！

