

量化金融導論

繪製均線、k 棒和日成交量圖表

檢測公司是否有內線交易

目錄

1. 專題題目：
 - (一) 繪製均線、k 棒及成交量圖表
 - (二) 檢測公司是否有內線交易
2. 摘要
3. Data
4. 研究方法
5. 完整程式碼和解釋過程
6. 繪製之圖形
7. 結論

1. 專題題目：

(一) 繪製均線、k 棒及成交量圖表

(二) 檢測公司是否有內線交易 (觀察股票在事件發生前後十天的股價變化，判斷是否可能有內線交易的發生)

2. 摘要：

這學期有修習另一門課叫做投資組合管理，而我們每一周都固定針對四檔股票 (PG、MMM、MSCI、CME) 做不同的模型分析 (Sharpe Performance Model、Residual Earning Model 等等)，但是 excel 的功能也有限，較無法將許多數據以圖表的方式一次展出，而使用 Python 程式語言可以快速地用 function 的功能一次寫好，之後再將工作表直接進行代換。所以我想運用兩堂課分別學到的不同的知識加以融合，針對股票做一些分析和繪圖。

而第二個提案則是和股價反應有關，選定公司為 Twitter，觀測因為 Q3 淨利大跌之利空消息釋出 ($t=0$) 之前後十天內 ($-10, +10$) 股價的變化。

第二個提案所採用的新聞事件：

〈財報〉Twitter Q3淨利大跌95% 盤前股價崩逾20%

鉅亨網新聞中心 2019/10/24 20:42



鉅亨買基金

配息放

聯博全

限時**0**手續
再送**終身優**

🕒 主編精選

美國擬對31億美
英國商品徵收新



3. Data :

從 yahoo finance 上下載 PG、MMM、MSCI、CME 四檔股票從 2018-6-25 至 2020-6-23 共兩年的日資料，主要使用檔案中開盤價、最高價、最低價、收盤價、以及成交量的資料。而均線的資料要寫程式才能得出。檔案名稱為 MA_Candlesticks。第二個的資料也是由 yahoo finance 下載 TWTR 的資料，共 20 天。檔案名稱為 Insider_trading。

4. 研究方法：

使用 Python 語言，主要使用到的 package 有 pandas, matplotlib, mplfinance。而第二個方法先從日資料中計算得出報酬率，再算出異常收益率（異常收益率是指某種證券的實際收益率與市場預期收益率之間的差值能反映該種證券收益情況及該種證券與市場的關係。我使用的證券資料是 Twitter 的 Adjust close price，而市場資料是 S&P 500 的 Adjust Close Price）。最後再得出累積異常收益率後繪製成圖表。

5. 完整程式碼和解釋過程：

(一) 第一部分

```
1  import pandas as ps
2  import matplotlib.pyplot as plt
3  import mplfinance as mpf
4
5  #-----PART 1-----
6  #繪製7均線、k棒和日成交量圖表
7
8  MA_C= ps.ExcelFile('MA_Candlesticks.xlsx') #獲取MA_Candlesticks中sheet的名稱: 'PG_D', 'MMM_D', 'MSCI_D', 'CME_D'
9  k= MA_C.sheet_names #資料為從2018-6-25至2020-6-23共兩年的日資料，503筆
10
11 def func_READ(xlsx: str, sheet: str): #寫入excel檔案
12     data_f= ps.read_excel(f'{xlsx}.xlsx', sheet_name = sheet)
13     data_f['Date']= ps.to_datetime(data_f['Date']) #更改為日期格式
14     data_f.set_index(['Date'], inplace=True)
15     return data_f
16
17 PG_D= func_READ('MA_Candlesticks', 'PG_D')
18 MMM_D= func_READ('MA_Candlesticks', 'MMM_D')
19 MSCI_D= func_READ('MA_Candlesticks', 'MSCI_D')
20 CME_D= func_READ('MA_Candlesticks', 'CME_D')
21
22
23 #計算繪製均線所需的數據
24 #func_MA([5,10,15], CME_D)-> 可得到5日、10日、15日均線所需的資料（每n日之資料平均）
25
26 def func_MA(days: List, sheet):
27     for d in days: #days為想要繪製的均線天數的串列
28         ma_days= [None]*d #在串列中加入和天數相同筆的None為資料，才能和sheet的rows數相同
29         for k in range(len(sheet['Close'])-d): #減去d筆已填入None的
30             average_days= sum(sheet['Close'][k:k+d])/d #計算ma資料
31             ma_days.append(average_days) #加入串列
32             sheet[f'ma{d}']= ma_days #在sheet中新增ma的資料
33
34     return sheet
35
36 #繪製均線、k棒和日成交量圖表
37
38 def plot_ma_candle(days: List, sheet):
39     ma_sheets= []
40     for d in days:
41         ma_sheets.append(f'ma{d}') #將要繪製均線的ma的欄資料先獨立出來
42     ma_lines= mpf.make_addplot(sheet[ma_sheets]) #使用mplfinance的addplot功能
43     plot= mpf.plot(sheet, type= 'candle', addplot= ma_lines, volume= True, figscale= 5, style= 'starsandstripes')
44     #用計算出的ma資料繪製7均線、k棒和日成交量圖表，figscale值愈大圖表愈清晰，而style則是可以選擇圖表輸出的顏色樣式
45     #由於此api很新，因此尚未開發出加title的功能（在開發程式文件中找不到相關程式碼）
46     return plot
47
48 funcs1= [func_MA, plot_ma_candle]
49 funcs1[0]([5,10,20,30,60,120,250], CME_D) #日均線數值可變
50 funcs1[1]([5,10,20,30,60,120,250], CME_D)
51
52 '''
53 將Part 1產出的圖表打叉，會跳出Part 2的圖表
54 '''
```

(二) 第二部分

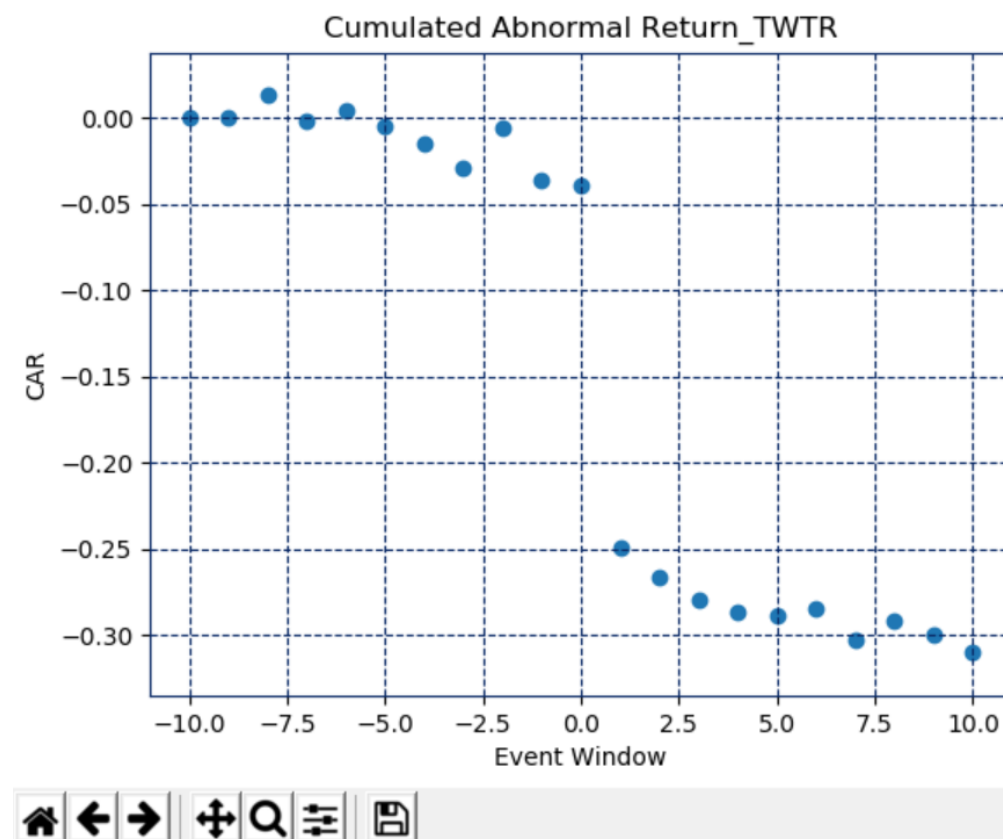
```
54 #-----PART 2-----
55 #檢測是否有Insider Trading(內線交易)
56 #選定公司為Twitter，觀測因為Q3淨利大跌之利空消息釋出(t=0)之前後十天內(-10,+10)股價的變化
57 #已知事件發生日為2019-10-24，股價反應視窗為前後十天之資料
58
59
60 def func_READ2(xlsx: str, sheet: str): #寫入excel檔案
61     data_f= ps.read_excel(f'{xlsx}.xlsx', sheet_name = sheet)
62     return data_f
63
64 In_trading= func_READ2('Insider_Trading', 'In_trading')
65
66 def rate_of_return(sheet): #計算rate of return
67     names= ['TWTR', 'S&P 500'] #TWTR跟S&P500都要
68     for name in names:
69         r_of_r= [None]*1 #為符合資料長度
70         for k in range(len(sheet[name])-1): #減去1筆已填入None的
71             ror= (sheet[name][k]-sheet[name][k+1])/sheet[name][k+1] #計算rate of return
72             r_of_r.append(ror) #加入串列
73             sheet[f'ror{name}']= r_of_r
74     return sheet
75
76 #已知預估alpha=0.0009, slope= 0.6175, standard error=0.0111 (rorTWTR, rorS&P500)
77 alpha= 0.0009
78 slope= 0.6175
79 standard_error=0.0111
80
81 #expected_return= alpha+slope*rorS&P 500
82 #abnormal_return= rorTWTR- expected return
83
84 def abnormal_return(sheet):
85     abnormal_r= []
86
87     #expected_return= alpha+slope*rorS&P 500
88     #abnormal_return= rorTWTR- expected return
89
90     def abnormal_return(sheet):
91         abnormal_r= []
92         for k in range(1,len(sheet['rorS&P 500'])): #減去1筆已填入None的
93             a_r= sheet['rorTWTR'][k]-(alpha+slope*sheet['rorS&P 500'][k])
94             abnormal_r.append(a_r) #加入串列
95         abnormal_r.append(0) #符合sheet長度
96         sheet['abnormal_r']= abnormal_r[::-1] #做串列反轉，等一下才能夠累加(cumulate)
97         return sheet
98
99     def cumulative_abnormal_return(sheet):
100         cumulate_r= []
101         for k in range(len(sheet['abnormal_r'])):
102             cum_r= sum(sheet['abnormal_r'][0:k]) #將abnormal return 累加
103             cumulate_r.append(cum_r)
104             sheet['cumulate_r']= cumulate_r
105         return sheet
106
107     funcs2= [rate_of_return, abnormal_return, cumulative_abnormal_return]
108     funcs2[0](In_trading)
109     funcs2[1](In_trading)
110     funcs2[2](In_trading)
111
112     t_reverse= In_trading['t'][::-1] #因為事件要從-10開始算到+10(0為事件發生當天)，原始資料日期要反轉
113     cumulate_r= In_trading['cumulate_r']
114     #畫圖，x軸為t(Event Window)，y軸為CAR(Cumulated Abnormal Return)
115     mlb.scatter(t_reverse, cumulate_r)
116     mlb.title('Cumulated Abnormal Return_TWTR') #title
117     mlb.xlabel('Event Window') #xlabel
118     mlb.ylabel('CAR') #ylabel
119
120     mlb.show()
121
122     #結論：由Cumulated Abnormal Return_TWTR可觀察出，股價在事件發生前(t<0)已經有微幅下降的情況，可能隱含著內線交易的發生
```

6. 繪製之圖形

(一) CME 過去兩年的日資料所繪製的 K 棒，MA=5, 10, 20, 30, 60, 120, 250 天共七條均線，以及日成交量的圖表



(二) Twitter 在利空事件發生前後十天的累計超額收益率



7. 結論

由 Cumulated Abnormal Return_TWTR 中可以看出，在事件發生前累計超額收益率就有微幅下降，可以做為市場中可能有 Insider Trading 的現象。