巨量資料分析期末書面報告

B074030025 劉于寧
B084030019 何冠融
B084030022 曾恩琪

一、 **主題：**旱地拔蔥策略

二、 **交易策略介紹：**

1、超過十個交易日股價橫向整理

取 11 個交易日的最低價、最高價、平均收盤價，若 11 天之最高價大於 11MA 不足 2%；且 11 天之最低價低於 11MA 不足 2%，則符合該條件。

2、在整理期間成交量明顯萎縮

若當天的成交量為前一天的 80%以下，則符合該條件。

3、以中長紅的姿態突破整理

若當天收盤價大於開盤價 1%以上；且當天的開盤價大於於前一日的最高價，則符合該條件。

4、突破時代著比先前多出不少的成交量

若當天成交量比前一天多出 20%以上，則符合該條件。

※設定手續費為千分之 1425、交易稅為千分之 3

三、 程式碼說明：

1、爬取 0050 資料

```
import urllib.request,re
x = urllib.request.urlopen("https://www.cnyes.com/twstock/ingredient.aspx?code=0050").read().decode("utf-8")
y=x.split('持股比例')[3]
y1=y.split('</td>\r\n\t\t<td></td>\r\n\t\t<td></td>\r\n\t</tr>\r\n</table>\r\n\r\n</div></div>\r\n\r\n<div class="returnTop">')[0]
y2=y1.split('.htm">')
y3=[i.split('</a>')[0] for i in y2]
codelist=y3[1:-1]#0050成分股
#percentage=y1.split('class="alignr">')
#percentage=[i.split('</td>')[0] for i in percentage]
#percentage=percentage[1:-1]#持股比例

#===取得0050資料==================================
start = datetime.datetime(2018,12,9)
end = datetime.datetime(2020,2,11)
df = pdr.DataReader('0050.TW', 'yahoo', start=start,end=end)
df=df.dropna(axis=0)#刪除移漏值
x=[df.index,df['Volume'],df['Open'], df['High'],df['Low'],df['Close']]
x=np.transpose(x)
info0050=[[pd.Timestamp(test[0]).date(),test[1],round(test[2],2),round(test[3],2),round(test[4],2),round(test[5],2)]for test in x]
```

(1) 取得 0050 歷史股價，並篩選可用資訊及刪除遺漏值。

#使用 import pandas_datareader as pdr 指令，匯入

pandas_datareader 套件。

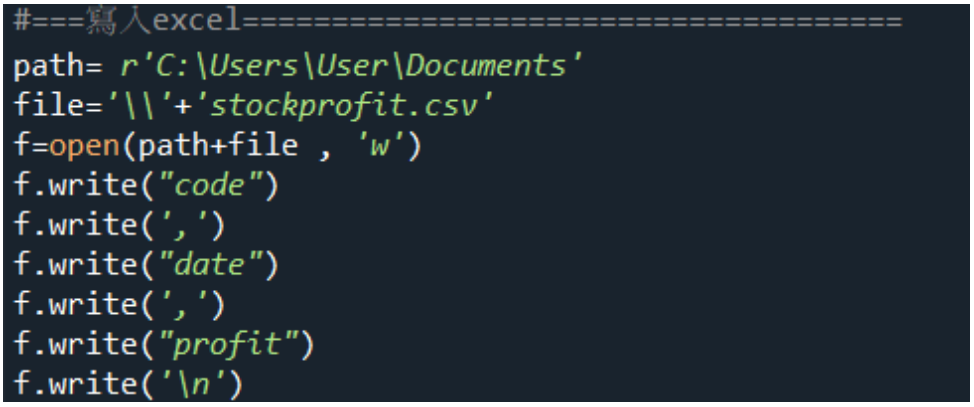
#使用 df = pdr.DataReader('0050.TW', 'yahoo', start=start,end=end)

指令，從 yahoo 上爬取資料。

(2)爬取 0050 成分股，並將股票代碼存入 codelist 中以便後續使用。

#使用 urllib.request.urlopen() 指令從鉅亨網爬取資料。

2、建立名為“stockprofit”的.csv 檔，放置後續回測績效



```
#===寫入excel=====================================
path= r'C:\Users\User\Documents'
file='\\'+'stockprofit.csv'
f=open(path+file , 'w')
f.write("code")
f.write(',')
f.write("date")
f.write(',')
f.write("profit")
f.write('\n')
```

3、將取得的資料匯入資料庫

```
#===匯入資料庫=====================================
import pymysql
connection = pymysql.connect(host="127.0.0.1",
                             user="root",
                             password="grho5439",
                             db="stock")
sql = "create table "+'s'+str(code)+"(date char(15),volumn int,open float,high float,low float,close float);"
sql2="drop table if exists "+'s'+str(code)+";"
cursor=connection.cursor()#建立連接點
cursor.execute(sql2)#先刪除table如果有同檔名
cursor.execute(sql)
connection.commit()

# Insert to the database
try:
    with connection.cursor() as cursor:
        for row in target_info:
            sql = "INSERT INTO "+'s'+str(code)+" VALUES (%s,%s,%s,%s,%s,%s)"
            cursor.execute(sql, (row[0],row[1],row[2],row[3],row[4],row[5]))
    connection.commit()
finally:
    connection.close()
```

以成分股代碼為名稱於先前創建好的 stock 資料庫中創建資料表，並匯入爬取的資料(日期、成交量、開盤價、最高價、最低價、收盤價)，如已有相同名稱的資料表，則先 將原始資料表刪除再進行匯入。

#以股票代碼創建資料表是為了方便查詢單一個股資料

4、進場判斷程式碼

(1)以『中長紅』姿態突破並且突破時帶著比先前不少的成交量

```
if(target_info[i][5]<target_info[i][2]*rule3):#3.以"中長紅"的姿態突破整理
    continue
else:#4.突破時帶著比先前多出不少的成交量
    if(target_info[i][1]<target_info[i-1][1]*rule4):
        continue
```

若當天收盤價大於開盤價 1%以上；且當天成交量比前一天多出 20%以上，則進行下一條件，否則該日不進場。

#rule3 為中長紅定義常數(在這裡為 1.01)。

#rule4 為『不少的成交量』定義常數(在這裡為 0.2)。

(2)超過十個交易日股價橫向整理

```
for test in range(1,12):#1.超過十個交易日股價橫向整理-1
    minp.append(target_info[i-test][4])
    maxp.append(target_info[i-test][3])
    closep.append(target_info[i-test][5])
periodmax=max(maxp)
periodmin=min(minp)
average_close=np.mean(closep)
if(periodmax>average_close*rule1_1 and periodmin<average_close*rule1_2):
    continue
```

1. 將 11 天交易日的最低、最高、收盤價分別加入 minp[]、maxp[]、closep[]。
2. 取出 11 天來的最高價、最低價及平均收盤價。
3. 若 11 天最高價大於 11MA 不足 2%；且 11 天最低價低於 11MA 不足-2%，則檢視下一條件；否則該日不進場。

#rule1_1、rule1_2 為震盪區間定義常數(這裡為 1.02 及 0.98)

(3) 以中長紅的姿態『突破』整理

```
for back in range(1,12):
    if(target_info[i][2]<target_info[i-back][3]):#3.以中長紅的姿態"突破"整理
        select=False
        break
```

若當天的開盤價小於前一日的最高價，則中止迴圈，否則繼續。

(4) 在整理期間成交量明顯萎縮

```
if(target_info[i-back][1]>target_info[i][1]*rule2):#2.在整理期間成交量明顯萎縮
    select=False
    break
```

若當天的成交量為前一天的 80%以上，則中止迴圈，否則繼續。

5、進場程式碼

```
if(trade == False and select == True):
    trade = True
    outdate = i
    overallcost.append(target_info[i][5])
    costlist.append(target_info[i][5])
    cost = target_info[i][5]
    continue
```

先設定一個變數 trade 來判別是否買進，若達進場條件(select = True)，

則對該檔股票進行交易。

6、停利、停損程式碼

```
for i in range(12,len(target_info)-2):
    minp=[]
    maxp=[]
    closep=[]
    select=True
    if(trade == True):
        if(cost*stoploss > target_info[i][5]):
            outdate = None
            trade = False
            profit=target_info[i][5]-cost
            profitlist.append(profit*(1-TC))
            target_date.append(target_info[i][0])
            print("Ordertime:",date,"Covertime:",target_info[i][0],"buyprice:",cost,"gain:",round(profit,3))
            f.write(str(codestr))
            f.write(',')
            f.write(str(target_info[i][0]))
            f.write(',')
            f.write(str(profit))
            f.write('\n')
        elif(cost*stopgain < target_info[i][5]):
            outdate = None
            trade = False
            profit=target_info[i][5]-cost
            profitlist.append(profit*(1-TC))
            target_date.append(target_info[i][0])
            print("Ordertime:",date,"Covertime:",target_info[i][0],"buyprice:",cost,"gain:",round(profit,3))
            f.write(str(codestr))
            f.write(',')
            f.write(str(target_info[i][0]))
            f.write(',')
            f.write(str(profit))
            f.write('\n')
```

(1)停利條件：

若當天收盤價格大於買入價格 13%以上，則平倉且將損益寫入.csv；

否則繼續持有。　#stoploss 為停利定義常數(這裡為 1.13)

(2)停損條件：

若當天收盤價格小於買入價格 3%以上，則平倉且將損益寫入.csv；

否則繼續持有。　#stopgain 為停損定義常數(這裡為 0.97)

```
if(trade==True):
    if(i == len(target_info)-1):
        outdate = None
        trade = False
        outsign = False
        profit=target_info[i][5]-cost
        profitlist.append(profit)
        target_date.append(target_info[i][0])
        print("Ordertime:",date,"Covertime:",target_info[i][0],"buyprice:",cost,"(hold for two days)gain:",round(profit,3))
        f.write(str(codestr))
        f.write(',')
        f.write(str(target_info[i][0]))
        f.write(',')
        f.write(str(profit))
        f.write('\n')
```

(3)若未達停損/停利條件，則直接持有至回測期間結束，並自動結算後寫入.csv 檔。

7、在可以零股交易的前提下，以該股的價格買進對應的 0050 張數。

```
if(firsttrade==True and select==True):
    firsttrade=False
    holdprofit.append(target_info[len(target_info)-1][5]-target_info[i][5])
    holdcost.append(target_info[i][5])
    hold0050profit.append((info0050[len(target_info)-1][5]-info0050[i][5])*(target_info[i][5]/info0050[i][5]))
    hold0050cost.append(target_info[i][5])
    #print(target_info[len(target_info)-1][5],"+",target_info[i][5])

if(trade == False and select == True):
    trade = True
    outdate = i
    overallcost.append(target_info[i][5])
    costlist.append(target_info[i][5])
    cost = target_info[i][5]
    continue
```

8、計算各權值股的總進場獲利，及平均報酬率

```
    if(len(profitlist)!=0):
        overallprofit.append(sum(profitlist))
        print("total gain:",round(sum(profitlist),2))
        print("return:",round(sum(profitlist)/sum(costlist),5))
        print("\n")
    else:
        print("nothing match the condition!\n")
```

計算各權值股的總進場獲利，及平均報酬率，若此檔權值股無進場紀錄，則印出「無符合條件」。

9、印出績效

```
print("-----------------------------------------")
print(" overall gain:",sum(overallprofit))
print(" return:",sum(overallprofit)/sum(overallcost))
print("-----------------------------------------")
print("(buy and hold) overall gain:",sum(holdprofit))
print("(buy and hold) return:",sum(holdprofit)/sum(holdcost))
print("-----------------------------------------")
print("(hold 0050 instead) overall gain:",sum(hold0050profit))
print("(hold 0050 instead) return:",sum(hold0050profit)/sum(hold0050cost))
```

分別計算出:

(1)策略成立下，執行停利/停損的總收益及總報酬率。

(2)策略成立下，不執行停利/停損的總收益及總報酬率。

(3)策略成立下，相同金額 0050 至期末結算的收益及報酬率。

四、 回測績效

1、 各權值股績效顯示

```
code: 2327
nothing match the condition!

code: 2330
2019-12-11 buyprice: 280.0 gain: 39.0
total gain: 39.0
return: 0.13929


code: 2357
nothing match the condition!

code: 2382
2019-05-06 buyprice: 60.7 gain: -2.3
2020-01-30 buyprice: 64.2 gain: -2.6
total gain: -4.9
return: -0.03923
```

擷取部分輸出:

(1) 2330 達停利標準 13%。

(2) 2382 達停損標準-3% (兩次平均-3.9%)。

(3) 2327、2357 無符合「旱地拔蔥」的條件。

2、總績效分析顯示

```
overall gain: 98.65999999999998
return: 0.0390738863189912
-----------------------------------------
(buy and hold) overall gain: 0.2499999999999929
(buy and hold) return: 0.00011537538535378382
-----------------------------------------
(hold 0050 instead) overall gain: 60.58841002460314
(hold 0050 instead) return: 0.027961644618247378
```

在旱地拔蔥策略之下，執行停利/停損的報酬率優於另外二者。

五、 K 線圖說明

註：這裡使用我們製作的另一個用來產生 K 線的程式碼來生成 K 線。

1、使用 mpl_finance 套件畫圖

```
import matplotlib.pyplot as plt
import mpl_finance as mpf
%matplotlib inline
```

```
for idate in target_date:

    start = idate-datetime.timedelta(days=20)
    end = idate+datetime.timedelta(days=30)
    df = pdr.DataReader(codestr, 'yahoo', start=start,end=end)
    df=df.dropna(axis=0)#刪除移漏值

    df.index = df.index.format(formatter=lambda x: x.strftime('%Y-%m-%d'))

    fig = plt.figure(figsize=(24, 8))

    ax = fig.add_subplot(1, 1, 1)
    ax.set_xticks(range(0, len(df.index), 10))
    ax.set_xticklabels(df.index[::10])
    mpf.candlestick2_ochl(ax, df['Open'], df['Close'], df['High'],
                          df['Low'], width=0.6, colorup='r', colordown='g', alpha=0.75);
```

匯入單一個股股票資訊(開、高、低、收)及調整匯出樣式便可得下面的

K 線圖。

2、匯出範例

(1)以 2303 為例

```
code: 2303
Ordertime: 2019-04-25 Covertime: 2019-05-28 buyprice: 13.1 gain: -0.65
total gain: -0.65
return: -0.04962


code: 2308
nothing match the condition!

code: 2317
Ordertime: 2019-04-01 Covertime: 2019-04-17 buyprice: 80.8 gain: 11.0
total gain: 11.0
return: 0.13614
```

股票代碼 : 2303
購買價格 : 13.1元
進場時間 : 2019/04/25
出場時間 : 2019/05/28
損失-0.65元，報酬率-5%

生成之 K 線圖：

(2)以 2317 為例

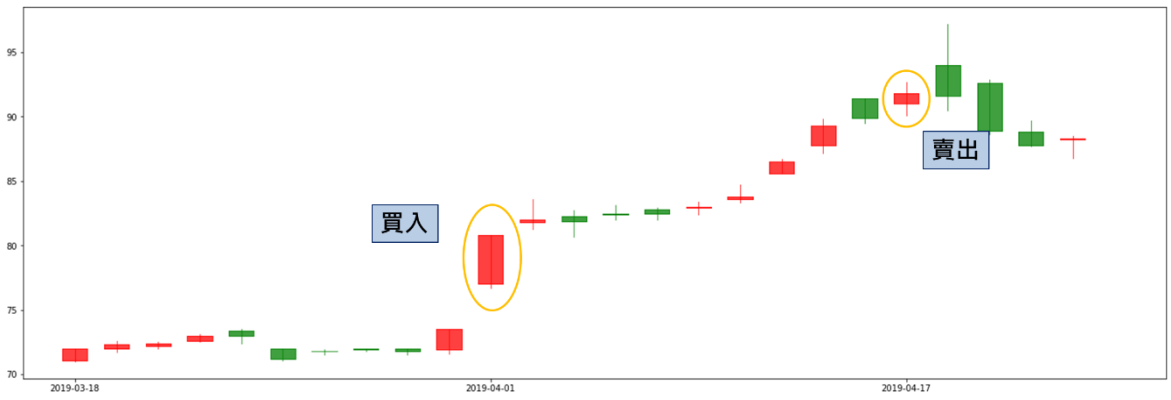
```
code: 2303
Ordertime: 2019-04-25 Covertime: 2019-05-28 buyprice: 13.1 gain: -0.65
total gain: -0.65
return: -0.04962


code: 2308
nothing match the condition!

code: 2317
Ordertime: 2019-04-01 Covertime: 2019-04-17 buyprice: 80.8 gain: 11.0
total gain: 11.0
return: 0.13614
```

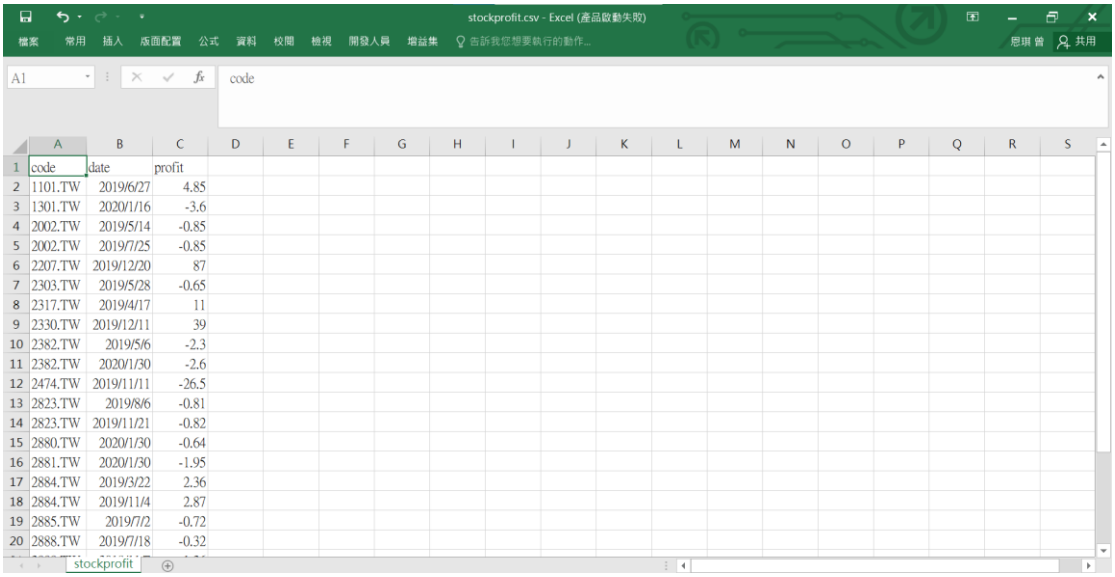
股票代碼 : 2317
購買價格 : 80.8元
進場時間 : 2019/04/01
出場時間 : 2019/04/17
獲利11.0元，報酬率13.6%

生成之 K 線圖：



六、.csv 檔匯出示意圖

stockprofit.csv　　2022/1/1 下午 11:54　　Microsoft Excel 逗點...　　1 KB



	A	B	C
1	code	date	profit
2	1101.TW	2019/6/27	4.85
3	1301.TW	2020/1/16	-3.6
4	2002.TW	2019/5/14	-0.85
5	2002.TW	2019/7/25	-0.85
6	2207.TW	2019/12/20	87
7	2303.TW	2019/5/28	-0.65
8	2317.TW	2019/4/17	11
9	2330.TW	2019/12/11	39
10	2382.TW	2019/5/6	-2.3
11	2382.TW	2020/1/30	-2.6
12	2474.TW	2019/11/11	-26.5
13	2823.TW	2019/8/6	-0.81
14	2823.TW	2019/11/21	-0.82
15	2880.TW	2020/1/30	-0.64
16	2881.TW	2020/1/30	-1.95
17	2884.TW	2019/3/22	2.36
18	2884.TW	2019/11/4	2.87
19	2885.TW	2019/7/2	-0.72
20	2888.TW	2019/7/18	-0.32