

依據情緒指標來建立模型，並以台灣股市為投資標的，
來達到穩定獲利之效果



台北大學 指導教授 涂登才 老師
台北大學 資工所 宋晉德
台北大學 國企所 陳映如
台北大學 國企所 黃柏毅
台北大學 國企所 郭伯軒

一、金融市場分析

20 世紀 60 年代，有效市場假說（EMH）是金融市場的主流理論，其定義為在證券市場中，價格可以完全反映所有可以獲得的訊息。但後來發現有效市場假說很成功地解釋流動資產收益缺乏可預測性的這個方面與許多金融從業者的傳統“尋求價值”產生了一些矛盾。到了 20 世紀 80 年代中期，才開始認真探索流動金融市場並不總是像有效市場倡導者所建議的那樣有序的可能性，因此，開始有了其他的理論，例如，Black 與 DeLong 等人的“噪音交易者理論”，噪音是指證券價格與價值之間存在的一個偏差，如果一些投資者以與基本面無關的“噪音”信號進行交易，那麼資產價格將偏離其內在價值。但此理論支持者的大多數證據都具有爭論。有其他的實證研究也記錄了股票市場異常情況，例如市場反應過度和過度反應（Gregory W. Brown and Michael T. Cliff, 2004）。

DeLong、Shleifer、Summers 和 Waldmann (1990) 提出了一個假設是投資者受到情緒的影響。近年來研究發現，投資者情緒（investor sentiment）在國際市場波動（international market volatility）中發揮重要作用，並產生收益可預測性的形式與過度反應的糾正一致。在國家層面收益（country-level returns）方面，全球情緒是逆向預測因素。全球市場情緒和本地市場情緒是市場內橫截面收益時間序列的逆向預測因素（contrarian predictors of the time-series of cross-sectional returns within markets）：當情緒高漲時，套利相對較難且難以估值的股票，未來收益率較低。私人資本流動有如一種機制，透過這樣的機制，情緒在市場中傳播並形成全球情緒（Malcolm Baker, Jeffrey Wurgler, Yu Yuan, 2012）。另外，在 20 世紀 90 年代末期，曾有一段意想不到的投資者情緒將投機性和難以估值的科技股的價格推向了不可思議的水平（Malcolm Baker and Jeffrey Wurgler, 2007）。

投資者情緒是否會影響股票價格等已不再是大家討論的問題，如何衡量投資者情緒並量化其影響才是現在大家所討論的，例如過度自信，代表性和保守主義，來解釋個人投資者對過去的回報或基本面的反應過度或過度反應（Malcolm Baker and Jeffrey Wurgler, 2007）。因此，我們可以了解到投資者情緒對於股票市場、選擇權市場、期貨市場等十分重要，接下來我們將使用不同的情緒指標與模型來做為我們本次模擬交易的根據。

二、本期操作策略

在「Investor sentiment and the near-term stock market」論文中，有介紹不同情緒指標的變數，並且試著比較他們與 survey 中的 SENTp(法人機構的情緒指標)和 SENTa(個體戶的情緒指標)的關係。跟 SENTp 相關的有 HI/LO, ADV/DEC, ARMS, SPECIAL, VOL, FUTp 和 SHORTSLS。跟 SENTa 相關的有 HI/LO, ADV/DEC, ARMS, MKTVANE, CEFD, change in CEFD, ODDLOT 和 FUTa。ARMS 是 $(\text{Adv}t/\text{Adv Volt})/(\text{Dect}/\text{Dec Volt})$ (Brown and Cliff, 2004)。

Survey sentiment and technical variables

		Monthly SENT ^p	Weekly SENT ^p	SENT ^a
Trading volume	ADV/DEC	48.5813***	-0.8145*	0.9775
	ARMS	-13.9606*	0.8817	6.3919
	HI/LO	-0.0567	0.0802*	0.3499***
Type of trade	Δ MARGIN	-0.0279		
	Δ SHORTIR	19.0383***		
	SPECIAL	24.5495*	12.1160**	5.8992
	ODDLOT	6.1902***	-2.2866***	-4.9089***
	SHORTSLS		1.1186	-10.1899
Derivatives	PUT/CALL		-0.4326	0.6284
	VOL		-0.5776	3.9741**
	FUT ^p		0.0000	-0.0000
	FUT ^a		0.0000	0.0001
	MKTVANE		0.0177	0.0864
Other	CEFD	-0.2481***	-0.1096	-0.2508
	IPORET	-0.0620		
	IPON	0.0572**		
	FUNDFLOW	3.1082***		
	FUNDCASH	-0.9419***		
Control variables	R_{BIG}	-0.8269*	0.4876***	1.4093***
	$R_{\text{BIG},t-1}$	0.2638	0.7444***	0.9913***
	$R_{\text{BIG},t-2}$		0.9747***	-0.1523
	$R_{\text{BIG},t-3}$		0.4899***	0.4056
	R_{SOB}	-0.2286	0.1158	-0.0208
	$R_{\text{SOB},t-1}$	0.2239	-0.0273	0.5448
	$R_{\text{SOB},t-2}$		0.0742	-0.0614
	$R_{\text{SOB},t-3}$		0.1289	0.3544
	CONST	-51.2297***	-2.5817	-10.3119
R^2		0.7105	0.9261	0.5475

Regressions of survey sentiment data on the technical variables. The monthly regression uses 336 observations from January 1970 to December 1997. The weekly regressions use 596 observations from July 1987 through December 1998. Newey and West (1987) standard errors use four and eight lags for the monthly and weekly regressions, respectively.

* Indicate significance at the 10% level.

** Indicate significance at the 5% level.

*** Indicate significance at the 1% level.

weekly samples.²⁰ The correlation between the filtered estimates from the institutional and individual samples is only 0.59 despite the fact that they share several common inputs.

(一)、HILO

股市投資人情緒指標的變數是創新高的股票家數除以創新低的股票家數(HILO)。HILO 指數的算法，是將創新高的股票家數除以創新低的股票家數，並以此作為衡量投資者情緒的變數。其中，創新高的股票家數為所有上市與上櫃公司股價創過去一年新高之家數；創新低的股票家數為所有上市與上櫃公司股價創過去一年新低之家數。(Brown and Cliff, 2004)。

過去有眾多文獻提及財務變數對於股票市場之影響，例如公司規模、益本比、帳面權益對市值比、財務槓桿以及研發支出比等。而除了這些變數之外，Baker and Wurgler (2006) 還納入股利對帳面權益比、固定資產比以及銷售成長率進行股票報酬影響變數之研究。但是因為台灣股票市場投資參與者結構屬散戶居多的淺碟型市場，不像歐美國家一般成熟，所以財務特徵變數對股票報酬之解釋力不穩定的現象，在我國尤其可能發生，投資者常常有追高殺低等不理性的行為，而根據投資人情緒對投機型股票報酬之影響 (2009 TOPCO) 論文的研究，投資人情緒對投機性愈大之股票的影響愈大。這表示，因為投機型股票的價值難以被散戶投資人正確評價，因此股價容易受到主觀情緒所影響，且該類股票之價格在較大的區間內變動，將增加套利難度而造成投機型股票價格有較大的偏離程度。

我們也使用了 HILO 情緒指標作為我們選擇投資的依據，因為在其他篇參考的論文當中，HILO 是其中顯著性較佳的指數，例如：Malcolm Baker and Jeffrey Wurgler, 2007，以及 Gregory W. Brown, Michael T. Cliff, 2004。在過去的論文中，除了 HILO，還有用許多其他的變數來估計股票市場上投資人的情緒指標，例如：市場周轉率、月 IPO 家數與月平均 IPO 報酬率、新股發行比率、上漲家數除以下跌家數、ARMS 指標、融資變動比率、融券變動比率與放空比率……等等眾多的情緒指標。

(二)、漲跌趨勢線(Advance Decline Line)

漲跌趨勢線 (Advance Decline Line)，也稱為 A/D 線。用於繪製每日上漲和下跌股票數量之間的差異。指標是累積的，正數加到前一個數字，或者如果數字是負數，則從前一個數字中減去。A / D 線用於顯示市場情緒，因為它告訴交易者是否有更多的股票上漲或下跌。它用於確認主要指數的價格趨勢，也可以在出現分歧時發出逆轉警告。A / D 線是一個廣度指標，用於顯示有多少股票參與股市反彈或下跌。當主要股指上漲時，A / D 線上升證實上升趨勢顯示出強勁的參與。如果主要股指正在反彈且 A / D 線下跌，則表明參與反彈的股票較少，這意味著指數可能接近其反彈的結束。當主要

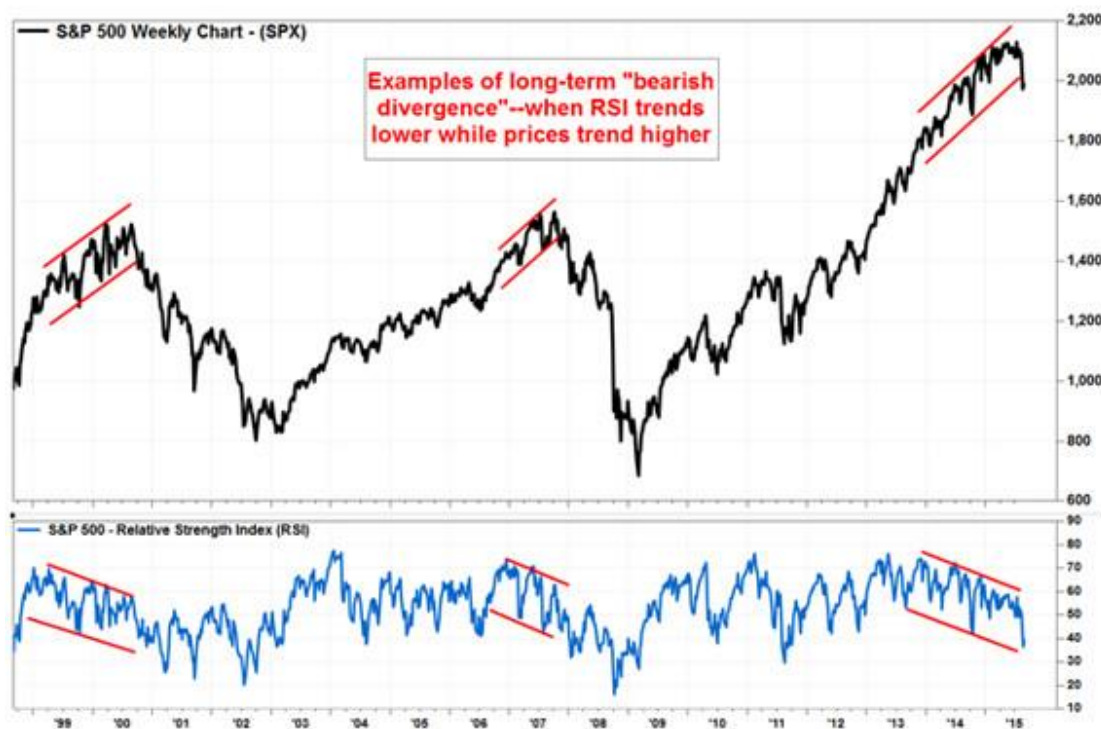
股指下跌時，下跌的 A/D 線會確認下跌趨勢。

漲跌趨勢線(Advance Decline Line)公式：

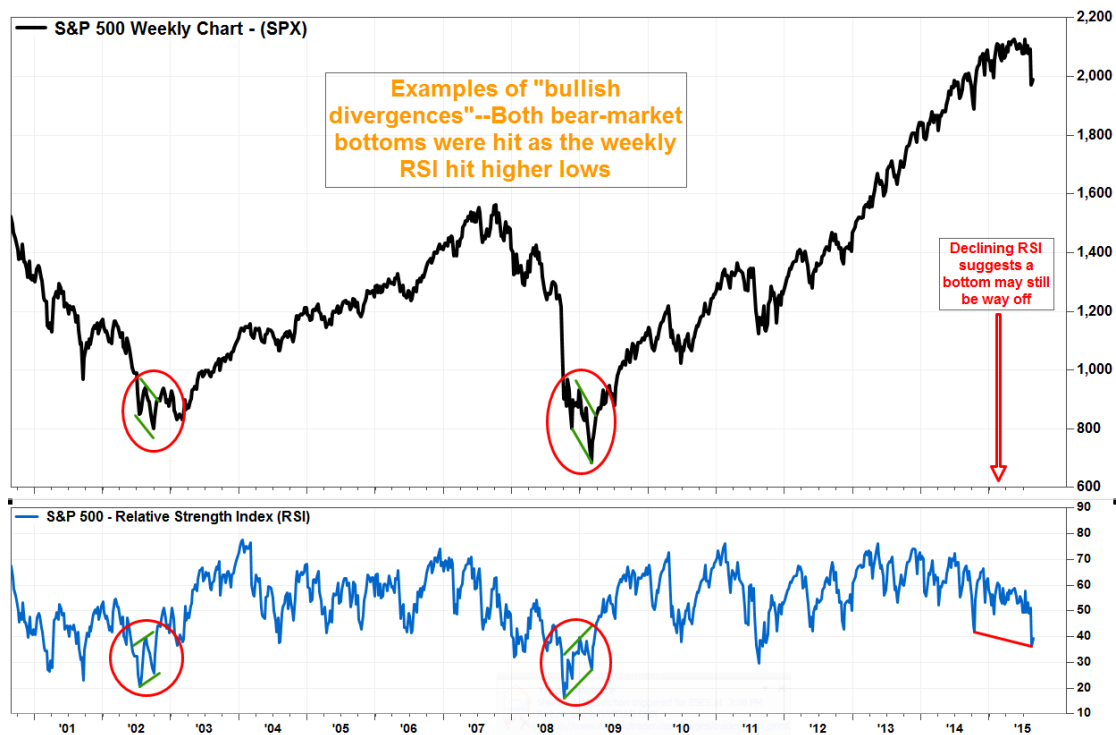
$$A/D = \text{Net Advances} + \text{Previous Advances}$$

其中，Net Advances = (今日上漲股票家數 - 今日下跌股票家數)，Previous Advances = 前一日的 A/D 指數

A / D 線用於確認當前趨勢的強度及其反轉的可能性。該指標顯示大多數股票是否參與市場方向。如果指數向上移動但 A / D 線向下傾斜，稱為看跌背離，則表明市場正在失去其廣度並且可能即將逆轉方向。如果 A / D 線的斜率上升且市場呈上升趨勢，那麼便可以說市場是健康的。相反的，如果指數繼續走低並且 A / D 線向上，稱為看漲背離，則可能表明賣方正在失去信念。如果 A / D 線和市場一起走低，那麼價格下跌的可能性就會更大。



(看跌背離 bearish divergence 範例)



(看漲背離 bullish divergence 範例)

(三)、ARMS

阿姆斯指標(ARMS)，又被稱為買賣指標、交易者指數(Trading Index)，是對 ADV/DEC 指標做了修正而發展出的一個指標(Gregory W. Brown and Michael T. Cliff, 2004)，其中多考慮了成交量值。ARMS 是一個過度買進和過度賣出的指標，衡量方法是分辨出每日行情中，上漲股票與下跌股票的那一方擁有較大的成交量比重。以下為 ARMS 的公式：

$$\text{Arms Index (TRIN)} = \frac{\text{Advancing Stocks/Declining Stocks}}{\text{Advancing Volume/Declining Volume}}$$

其中，

Advancing Stocks：當日市場中上漲家數值(Number of stocks that are higher on the day)。

Declining Stocks：當日市場中下跌家數值(Number of stocks that are lower on the day)。

Advancing Volume (AV)：當日市場中上漲的成交量值(Total volume of all

advancing stocks)。

Declining Volume (DV)：當日市場中下跌的成交量值(Total volume of all declining stocks)。

在這邊 on the day 的時間長度可以自己設定。Avancing Stock，代表有哪幾檔股票上漲。Declining Stock，代表哪幾檔股票下跌。Advancing Volume，代表全部上漲股票的交易量。Declining Volume，代表全部下跌股票的交易量。相除之後，便可以得到 ARMS 指標。一般來說，ARMS 指標值通常都在 3 以下。

ARMS < 1	家數的漲跌比值小於成交量的漲跌比值。	表示市場處於牛市（多頭市場）。指股市或經濟呈現長期上漲的趨勢，市場充滿樂觀氣氛。
ARMS = 1	家數的漲跌比值和成交量的漲跌比值相等。	
ARMS > 1	家數的漲跌比值大於成交量的漲跌比值。	表示市場處於熊市（空頭市場）。指股市或經濟呈現長期下滑的趨勢。

另外，我們看見了關於其他國家使用了 ARMS 作為超買超賣的指標，當指標下降到超買水平線時，代表有賣出機會；當指標上升至超賣水平線時，代表有買入機會。因此我們也做了參考，如以下圖表所示。

		超賣界限	超買界限
深證指數	4 天 ARMS 指標	1.4	0.35
	21 天 ARMS 指標	1.4	0.6
上證指數	4 天 ARMS 指標	2.4	1.1

	21 天 ARMS 指標	因參考性低，所以不列入。	0.7
--	--------------	--------------	-----

三、本期資本配置與投資組合

目前是先以製作 HILO，之後會再繼續製作 ADV/DEC 和 ARMS。資產配置則是採取被動式交易，參考 <http://www.taifex.com.tw/cht/2/weightedProportion>。第 1 名台積電權重是給 22%、第 2 名至第 11 名權重是給 2%、第 11 名至 132 名權重是給 0.5%。

原始資料，則是使用上面的網址 911 檔股票(再扣掉幾檔資料有誤)，時間長度則是從 2018 年 1 月 1 日至 2019 年 4 月 12 日。抓取方式如下:(catch_data.py)

```
from pandas_datareader import data as web
import datetime as dt
import fix_yahoo_finance as yf
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import csv
import time

local_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)
```



```
# Read Excel csv file
exampleFile = open('StockInfo.csv')
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)

yf.pdr_override()

start = dt.datetime(2018, 1, 1)
end = dt.datetime(2019, 4, 12)

for i in range(1,len(exampleData)):
    try:

print(str(i)+'\t 股票代號:'+exampleData[i][0]+'\t 股票名
稱:'+exampleData[i][1])
        df = web.get_data_yahoo([exampleData[i][0] + '.TW'], start, end)

df.to_csv('D:\\test\\portfolio\\data\\'+exampleData[i][0]+'_'+exampleData[
i][1]+'_TW.csv')
    except:
        break
```

```
# Get the last time
final_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
      local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)
print('Final Time is ', final_time.tm_year, '/', final_time.tm_mon, '/',
      final_time.tm_mday, '/', final_time.tm_hour,
      ":", final_time.tm_min)
# Save and Close Excel file
exampleFile.close()
```

建立 HI/LO 指標，由於原始論文(Brown and Cliff，2004) 長度是一年，但是在這邊我們分別是使用九天交易日和二十六天交易日來建立我們的情緒指標，並且也記錄當日個股是否分別為九天和二十六天交易日的最高或最低點。以下則是程式碼:(build_index.py)

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import csv
import time
```

```
local_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)

exampleFile = open('StockInfo.csv')
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)

for i in range(1,len(exampleData)):
    date_stk = []
    open_stk = []
    high_stk = []
    low_stk = []
    close_stk = []
    volume_stk = []
    hi9 = []
    lo9 = []
    hi26 = []
    lo26 = []
    hi_check9 = []
    lo_check9 = []
    hi_check26 = []
    lo_check26 = []

    twFile = open('data\\' + exampleData[i][0] + '_' + exampleData[i][1] +
'_TW.csv')
    twReader = csv.reader(twFile)
    twData = list(twReader)
```

```

print(str(i) + '\t 股票代號:' + exampleData[i][0] + '\t 股票名稱:' +
exampleData[i][1])

    # Create Excel file

    indexFile = open('index\\' + exampleData[i][0] + '_' + exampleData[i][1]
+ '_TW_index.csv', 'w', newline='')

    csvCursor = csv.writer(indexFile)

    # write header to csv file

    csvHeader =
['date','hi9','lo9','hi26','lo26','hi_check9','lo_check9','hi_check26','lo_check26','o
pen','close','volume']

    csvCursor.writerow(csvHeader)


    for j in range(1,len(twData)):
        date_stk.append(twData[j][0])
        open_stk.append(float(twData[j][1]))
        high_stk.append(float(twData[j][2]))
        low_stk.append(float(twData[j][3]))
        close_stk.append(float(twData[j][4]))
        volume_stk.append(float(twData[j][6]))

        if j >=9:
            temp_hi = max(high_stk[j-9:j-1])
            temp_lo = min(low_stk[j-9:j-1])
            hi9.append(temp_hi)
            lo9.append(temp_lo)


            if temp_hi <= float(twData[j][2]):
                hi_check9.append(1)


                lo_check9.append(1)
            else:
                lo_check9.append(0)

```

else:

```
hi9.append(0)
lo9.append(0)
hi_check9.append(0)
lo_check9.append(0)
```

if j >= 26:

```
temp_hi = max(high_stk[j-26:j-1])
temp_lo = min(low_stk[j-26:j-1])
hi26.append(temp_hi)
lo26.append(temp_lo)
```

if temp_hi <= float(twData[j][2]):

```
hi_check26.append(1)
```

else:

```
hi_check26.append(0)
```

if temp_lo >= float(twData[j][3]):

```
lo_check26.append(1)
```

else:

```
lo_check26.append(0)
```

else:

```
hi26.append(0)
lo26.append(0)
hi_check26.append(0)
lo_check26.append(0)
```

```

for j in range(0,len(date_stk)):
    csvContent =
[date_stk[j],hi9[j],lo9[j],hi26[j],lo26[j],hi_check9[j],lo_check9[j],hi_check26[j],
lo_check26[j],open_stk[j],close_stk[j],volume_stk[j]]
    #csvHeader = ['date', 'hi9', 'lo9', 'hi26', 'lo26', 'hi_check9',
'lo_check9', 'hi_check26', 'lo_check26']
    csvCursor.writerow(csvContent)
    indexFile.close()
    twFile.close()

# Get the last time
final_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)
print('Final Time is ', final_time.tm_year, '/', final_time.tm_mon, '/',
final_time.tm_mday, '/', final_time.tm_hour,
      ":", final_time.tm_min)
# Save and Close Excel file
exampleFile.close()

```

利用前面建立好的資料，接著統計九日交易日和二十六日交易日的達到最高點股票個數和達到最低點股票個數。並且做相除，但是我們在這邊有做一點變形，方便之後我們做偵測進出場的訊號。HIL09，九日交易日最高點股票個數/(九日交易日最高點股票個數+九日交易日最低點股票個數)。HIL026，二十六日交易日最高點股票個數/(二十六日交易日最高點股票個數+二十六日交易日最低點股票個數)。程式碼如下:(count_hi_lo.py)。ARMS9 和 ARMS26 以及 AD9 和 AD26 則是參照前面公式製作。

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import csv
import time

local_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
      local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)

exampleFile = open('StockInfo.csv')
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)
print(str(1) + '\t 股票代號:' + exampleData[1][0] + '\t 股票名稱:' +
      exampleData[1][1])
indexFile = open('index\\' + exampleData[1][0] + '_' + exampleData[1][1] +
'_TW_index.csv')
indexReader = csv.reader(indexFile)
indexData = list(indexReader)

date = []
hi_check9_count = []
lo_check9_count = []
hi_check26_count = []
lo_check26_count = []
hi_9_volume = []
lo_9_volume = []
```

```

hi_26_volume = []
lo_26_volume = []
#csvHeader =
['date','hi9','lo9','hi26','lo26','hi_check9','lo_check9','hi_check26','lo_check26']
for i in range(1,len(indexData)):
    date.append(str(indexData[i][0]))
    hi_check9_count.append(int(indexData[i][5]))
    lo_check9_count.append(int(indexData[i][6]))
    hi_check26_count.append(int(indexData[i][7]))
    lo_check26_count.append(int(indexData[i][8]))
    if int(indexData[i][5]) > 0:
        hi_9_volume.append(float(indexData[i][11]))
    else:
        hi_9_volume.append(0)
    if int(indexData[i][6]) > 0:
        lo_9_volume.append(float(indexData[i][11]))
    else:
        lo_9_volume.append(0)
    if int(indexData[i][7]) > 0:
        hi_26_volume.append(float(indexData[i][11]))
    else:
        hi_26_volume.append(0)
    if int(indexData[i][8]) > 0:
        lo_26_volume.append(float(indexData[i][11]))
    else:
        lo_26_volume.append(0)
indexFile.close()

```



```

for i in range(2,len(exampleData)):
    try:
        print(str(i) + '\t 股票代號:' + exampleData[i][0] + '\t 股票名稱:' +
exampleData[i][1])
        indexFile = open('index\\' + exampleData[i][0] + '_' +
exampleData[i][1] + '_TW_index.csv')
        indexReader = csv.reader(indexFile)
        indexData = list(indexReader)
        for j in range(1,len(indexData)):
            if date[j-1]==str(indexData[j][0]):
                hi_check9_count[j-1] =
hi_check9_count[j-1]+int(indexData[j][5])
                lo_check9_count[j-1] =
lo_check9_count[j-1]+int(indexData[j][6])
                hi_check26_count[j-1] =
hi_check26_count[j-1]+int(indexData[j][7])
                lo_check26_count[j-1] =
lo_check26_count[j-1]+int(indexData[j][8])
                if int(indexData[j][5]) > 0:
                    hi_9_volume[j-1]+=float(indexData[j][11])
                if int(indexData[j][6]) > 0:
                    lo_9_volume[j-1]+=float(indexData[j][11])
                if int(indexData[j][7]) > 0:
                    hi_26_volume[j-1]+=float(indexData[j][11])
                if int(indexData[j][8]) > 0:
                    lo_26_volume[j-1]+=float(indexData[j][11])
            indexFile.close()
    except:

```

break

Create Excel file

indexadFile = open('index_advance\\hi_lo.csv', 'w', newline='')

csvCursor = csv.writer(indexadFile)

write header to csv file

csvHeader =

['date','hi_check9_count','lo_check9_count','hilo9','hi_check26_count','lo_check26_count','hilo26','arms9','arms26','ad9','ad26']

csvCursor.writerow(csvHeader)

ad_9=0.0

ad_26=0.0

for i in range(25,len(date)):

 print(date[i])

 print(hi_check9_count[i])

 print(lo_check9_count[i])

 hilo9=hi_check9_count[i]/(lo_check9_count[i]+hi_check9_count[i])

arms9=(hi_check9_count[i]/lo_check9_count[i])/(hi_9_volume[i]/lo_9_volume[i])

 ad_9 = ad_9 + hi_check9_count[i] - lo_check9_count[i]

 print(hilo9)

 print(hi_check26_count[i])

 print(lo_check26_count[i])

 hilo26=hi_check26_count[i]/(lo_check26_count[i]+hi_check26_count[i])

 arms26 = (hi_check26_count[i] / lo_check26_count[i]) / (hi_26_volume[i] / lo_26_volume[i])

```

ad_26 = ad_26 + hi_check26_count[i] - lo_check26_count[i]
    print(hilo26)
    csvContent =
[date[i],hi_check9_count[i],lo_check9_count[i],hilo9,hi_check26_count[i],lo_ch
eck26_count[i],hilo26,arms9,arms26,ad_9,ad_26]
        csvCursor.writerow(csvContent)
indexadFile.close()
# Get the last time
final_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
local_time.tm_mday, '/', local_time.tm_hour,
        ":", local_time.tm_min)
print('Final Time is ', final_time.tm_year, '/', final_time.tm_mon, '/',
final_time.tm_mday, '/', final_time.tm_hour,
        ":", final_time.tm_min)
# Save and Close Excel file
exampleFile.close()

```

四、風險管理機制(理論或模型)

(一)面對風險時的三種不同態度:

1. 風險趨避(risk aversion):

是一個經濟學、金融學和心理學的一個概念，用來解釋在不確定狀況下消費者和投資者的行為。風險厭惡是指一個人面對不確定收益的交易時，更傾向於選擇較保險但是也可能具有較低期望收益的交易。例如一個風險厭惡的投資者，會選擇將他的錢存在銀行以獲得較低但確定的利息，而不願意將錢用於購買股票，承擔損失的風險以獲得較高的期望收益。

2. 風險中立 (Risk neutral) :

風險中性是相對於風險偏好和風險厭惡的概念, 風險中性的投資者對自己承擔的風險並不要求風險補償. 我們把每個人都是風險中性的世界稱之為風險中性世界 (Risk-Neutral World), 這樣的世界里, 投資者對風險不要補償, 所有證券的預期收益率都是無風險利率. 需要強調的是, 風險中性假設下得到的衍生物估值同樣可以應用於非風險中性的世界. 真實世界里的投資者儘管在風險偏好方面存在差異, 但當套利機會出現時, 投資者無論風險偏好如何都會採取套利行為, 消除套利機會後的均衡價格與投資者的風險偏好無關, 風險中性者並不介意一項投機是否具有比較確定或者不那麼確定的結果。他們只是根據預期的貨幣價值來選擇投機, 特別而言, 他們要使期望貨幣價值最大化。

3. 風險愛好(Risk-preferent):

風險愛好者是指那些在投資時願意承擔更多風險, 以求獲取更高收益的投資者。當面對具有相同預期貨幣價值的投機時, 風險愛好者喜歡結果不那麼確定的投機, 而不喜歡較穩定但低收益的結果。

4. 透過風險評估問卷(衡量自然人在風險下的選擇)

問卷: 透過此份問卷可以得到風險承受程度的分類, 透過區分的方式將受試者分為三個區塊, 也就是風險趨避、風險中立和風險偏好並給與其相對應商品作投資。由於每一個自然人在模型中的波動大小不同, 也就可以根據他們的風險偏好程度加以分類並給予商品作選擇。

(二)、檢測風險屬性問卷

(1) 年齡:

1. 20 以下
2. 20~29
3. 30~39
4. 40~49

5. 50~59

6. 60~69

7. 70 以上

(2) 請問您的教育程度？

1. 其他

2. 國小

3. 國中

4. 高中職

5. 專科

6. 大學

7. 碩士

8. 博士

(3) 請問您的年收入？

1. 0~30 萬

2. 31~60 萬

3. 61~80 萬

4. 81~100 萬

5. 101~150 萬

6. 151~300 萬

7. 301~500 萬

8. 501~1000 萬

(4) 請問您的資金來源？

1. 薪資/固定收入

2. 退休金

3. 閒置資金

4. 投資收益

5. 年終獎金

6. 轉換資標的

(5) 請問您目前可隨時動用的資金約有(新台幣)?

1. 0~30 萬
2. 31~60 萬
3. 61~80 萬
4. 81~100 萬
5. 101~150 萬
6. 151~300 萬
7. 301~500 萬
8. 501~1000 萬

(6) 請問至今您已經接觸理財多久時間?

1. 未滿半年
2. 半年以上~未滿一年
3. 一年以上~未滿三年
4. 三年以上~未滿五年
5. 五年以上~未滿十年
6. 十年以上

(7)請問您目前投資之資產中，約有多少是持有具波動性的商品?

1. 0%
2. 0%~10%
3. 10%~25%
4. 25%~50%
5. 超過 50%

(8)在一般情況下，您能接受之價格波動，大約在哪種程度?

1. 價格波動介於-5%~+5%
2. 價格波動介於-10%~+10%
3. 價格波動介於-15%~+15%
4. 價格波動介於-20%~+20%
5. 價格波動超過正負 20%

(9)如您持有之整體投資資產下跌超過 15%，請問對您的生活影響程度為何？(風險評估-承受力)(現金 流量期望)

1. 無法承受
2. 影響程度大
3. 中度影響
4. 影響程度小
5. 沒有影響

(10)當您的投資超過預設的停損或停利點時，請問您會採取那種處置方式？(風險評估-偏好)(現金流量 期望)

1. 立即賣出所有部位
2. 先賣出一半或一半以上部位
3. 先賣出一半以內部位
4. 暫時觀望，視情況再因應
5. 繼續持有至回本或不漲為止

(11)當您的投資組合預期平均報酬率達到多少時才會考慮賣出？(風險評估)(現金流量期望)

1. 5%
2. 10%
3. 15%
4. 20%
5. 25%以上

(12)若有臨時且非預期之事件發生時，請問您的備用金相當於您幾個月的家庭開支？(備用金係指在沒有違約金的前提下可隨時動用的存款)(風險評估-承受力)

1. 無備用金儲蓄
2. 3 個月以下
3. 3 個月(含)以上 ~ 6 個月
4. 6 個月(含)以上~9 個月

5. 9 個月(含)以上

(13)請問您偏好以下那類風險及報酬率之投資組合？(期望報酬)

1. 沒有概念
2. 絕對低度風險投資組合+穩健保本(低度風險，只要保本就好)
3. 低度風險投資組合+低度回報(低風險承擔下，追求低的投資報酬)
4. 中度風險投資組合+中度回報(在中等風險承擔下，要求中等水準的合理報酬)
5. 高風險投資組合+高度回報(願意承擔高度風險，也期待創造超額報酬)

【對投資風險的整體計分表】1. 上述問題計分方式:1(2 分)、2(4 分)、3(6 分)、4(8 分)、5(10 分) 2. 下表是根據您對上述問題回答的總分，藉此評估您依自身投資屬性在面對風險時所持有的一般承受度。

分數	風險屬性分類	定義	建議產品風險等級
20≤總評分≤27 或 1-2 及 2-7 勾選 a 選項者	第一級 保守型	您能承受的資產波動風險極低。極度保守的您十分注重本金的保護，寧可讓資產隨著利率水準每年獲取穩定的孳息收入，也不願冒風險追求可能的可觀報酬。您的理財目的可利用銀行存款，或具有穩定收益的產品來達成。	RR1
28≤總評分≤36	第二級 安穩型	您能承受的資產波動風險低。除了注重本金的保護外，您願意承受有限的風險，以獲得比定存高的報酬。所以除了定存和貨幣市場工具之外，建議可將部分資金配置在投資等級的固定收益或平衡型商品。	RR1～ RR2
37≤總評分≤47	第三級 穩健型	您能承受的資產波動風險中庸。穩健的您期望在本金、固定孳息與資本增長達致平	RR1～ RR3

		衡。您可以接受短期間的市場波動，並且瞭解投資現值可能因而減損。穩健的投資組合可以包括多種類別的資產，透過風險分散的方式獲得穩健的投資報酬，但仍須留意個別產品類型的波動性。	
48 ≤ 總評分 ≤ 60	第四級 成長型	您能承受的資產波動風險高。為了達成長期的資本增長，您願意忍受較大幅度的市場波動與短期下跌風險。成長的投資組合可以包括各種類別，且預期報酬率較高的資產，但建議您採取分批投入，且設定停損停利點以便循序漸進達到您的投資目標。	RR1 ~ RR4
61 ≤ 總評分 ≤ 100	第五級 積極型	您能承受的資產波動風險極高。非常積極的您如獵鷹般不停尋找獲利市場，並願意大筆投資在風險屬性較高的商品。積極的投資組合中資產類別包羅萬象，且在必要時利用槓桿操作來提高獲利，但因市場反轉所造成的資本下跌風險偏高，建議嚴格執行停損停利的投資原則，才能達到長期資產增值的目的。	RR1 ~ RR5

參照以上定義，根據分析結果，您的投資風險屬性為：_____ 型

附錄

目前我們這邊是使用 HI/LO 來做為進出場的機制，也會製作 ADV/DEC 和 ARMS。之後在下次報告，也會導入停利停損的機制，並且討論停利停損比例 2 倍至 4 倍。

(一)、HI/LO

進出場機制，進場:昨日 HILO9 小於 HILO26，當日 HILO9 大於 HILO26

日。出場:昨日 HILO9 大於 HILO26，當日 HILO9 小於 HILO26 日。隔日開盤
價格買進或賣出。程式碼如下:(hilo_model.py)

手續費則是設定，滑價 0.5%。(賣出價格乘上*0.995)

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import csv
import time

local_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
      local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)

exampleFile = open('StockInfo.csv')
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)

indexadFile = open('index_advance\\hi_lo.csv')
indexadReader = csv.reader(indexadFile)
indexadData = list(indexadReader)

indexFile = open('index\\' + exampleData[1][0] + '_' + exampleData[1][1] +
'_TW_index.csv')
indexReader = csv.reader(indexFile)
indexData = list(indexReader)
earn_date=[]
earn_profit=[]
```

```

for i in range(0,len(indexData)):
    earn_date.append(indexData[i][0])
    earn_profit.append(float(0.0))
stock_weight=[]
stock_weight.append(float(22.0))
for i in range(1,10):
    stock_weight.append(float(2.0))
for i in range(10,131):
    stock_weight.append(float(0.5))

indexFile.close()

for i in range(1,132):
    print(exampleData[i][0] + '_' + exampleData[i][1])
    indexFile = open('index\\' + exampleData[i][0] + '_' +
exampleData[i][1] + '_TW_index.csv')
    indexReader = csv.reader(indexFile)
    indexData = list(indexReader)
    print(indexData)
    buy_price = []
    buy_date_index = []
    sell_price = []
    sell_date_index = []

    #print(indexadData[2])
    #print(indexData[27])
    buy_check=0

```

```

for j in range(3,len(indexadData)):
    last_hilo9=float(indexadData[j-2][3])
    current_hilo9=float(indexadData[j-1][3])
    last_hilo26=float(indexadData[j-2][6])
    current_hilo26=float(indexadData[j-1][6])
    if buy_check==0 and last_hilo9<last_hilo26 and
current_hilo9>current_hilo26:
        buy_date_index.append(int(j+25))
        buy_price.append(float(indexData[j+25][9]))
        buy_check=1
    elif buy_check==1 and last_hilo9>last_hilo26 and
current_hilo9<current_hilo26:
        sell_date_index.append(int(j+25))
        sell_price.append(float(indexData[j+25][9]))
        buy_check=0

for j in range(0,len(sell_date_index)):

    print("buy:")
    print(buy_date_index[j])
    print(buy_price[j])
    print("sell:")
    print(sell_date_index[j])
    print(sell_price[j]*0.995)
    print("earn_per_NT1:")
    print(sell_date_index[j])
    print((sell_price[j]*0.995-buy_price[j])/buy_price[j])

```

```

earn_profit[sell_date_index[j]]+=((sell_price[j]*0.995-buy_price[j])/buy_price
[j])*(stock_weight[i-1]/100.0)
        print(earn_profit[sell_date_index[j]])
    indexFile.close()

sum_earn_profit=[]
for i in range(0,len(earn_date)):
    sum_earn_profit.append(sum(earn_profit[0:i]))

final_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)
print('Final Time is ', final_time.tm_year, '/', final_time.tm_mon, '/',
final_time.tm_mday, '/', final_time.tm_hour,
      ":", final_time.tm_min)
# Save and Close Excel file

indexFile.close()
indexadFile.close()
exampleFile.close()

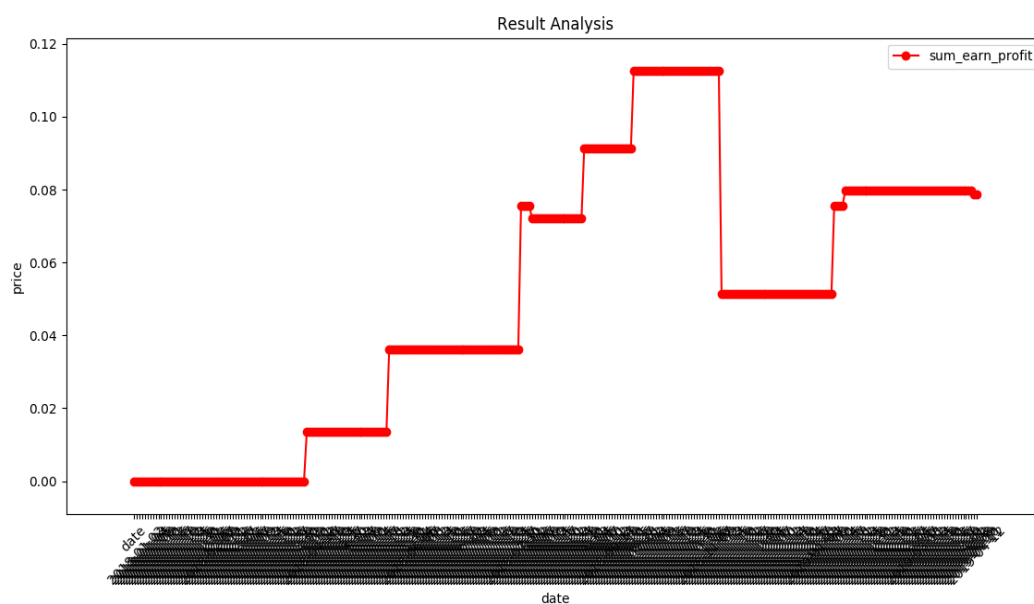
plt.title('Result Analysis')
earn_date_temp = range(len(earn_date))
plt.plot(earn_date_temp, sum_earn_profit, 'ro-', label='sum_earn_profit')
plt.xticks(earn_date_temp, earn_date, rotation=45)

```

```
plt.legend()

plt.xlabel('date')
plt.ylabel('price')
plt.margins(0.08)
plt.subplots_adjust(bottom=0.15)
plt.show()
```

被動式交易下，採取 HILO 全進全出策略，假設一元投資下，獲利表現如附圖：



(二)、ARMS

用 0.8 作為 ARMS9 和 ARMS26 進出場機制。(arms_model.py)

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
import csv
import time

local_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
      local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)

exampleFile = open('StockInfo.csv')
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)

indexadFile = open('index_advance\\hi_lo.csv')
indexadReader = csv.reader(indexadFile)
indexadData = list(indexadReader)

indexFile = open('index\\' + exampleData[1][0] + '_' + exampleData[1][1] +
'_TW_index.csv')
indexReader = csv.reader(indexFile)
indexData = list(indexReader)
earn_date=[]
earn_profit=[]
for i in range(0,len(indexData)):
    earn_date.append(indexData[i][0])
    earn_profit.append(float(0.0))
stock_weight=[]
stock_weight.append(float(22.0))
for i in range(1,10):
```

```

stock_weight.append(float(2.0))
for i in range(10,131):
    stock_weight.append(float(0.5))

indexFile.close()

for i in range(1,132):
    print(exampleData[i][0] + '_' + exampleData[i][1])
    indexFile = open('index\\' + exampleData[i][0] + '_' +
exampleData[i][1] + '_TW_index.csv')
    indexReader = csv.reader(indexFile)
    indexData = list(indexReader)
    print(indexData)
    buy_price = []
    buy_date_index = []
    sell_price = []
    sell_date_index = []

    #print(indexadData[2])
    #print(indexData[27])
    buy_check=0

    for j in range(3,len(indexadData)):
        last_hilo9=float(indexadData[j-2][7])
        current_hilo9=float(indexadData[j-1][7])
        last_hilo26=float(indexadData[j-2][8])
        current_hilo26=float(indexadData[j-1][8])
        if buy_check==0 and (current_hilo26 > 0.8 or current_hilo9

```



```

> 0.8) and (last_hilo9 > current_hilo9 and last_hilo26 > current_hilo26):
    buy_date_index.append(int(j+25))
    buy_price.append(float(indexData[j+25][9]))
    buy_check=1
    elif buy_check==1 and (current_hilo26 < 0.8 or current_hilo9 <
0.8) and (last_hilo9 < current_hilo9 and last_hilo26 < current_hilo26):
        sell_date_index.append(int(j+25))
        sell_price.append(float(indexData[j+25][9]))
        buy_check=0

for j in range(0,len(sell_date_index)):

    print("buy:")
    print(buy_date_index[j])
    print(buy_price[j])
    print("sell:")
    print(sell_date_index[j])
    print(sell_price[j]*0.995)
    print("earn_per_NT1:")
    print(sell_date_index[j])
    print((sell_price[j]*0.995-buy_price[j])/buy_price[j])

earn_profit[sell_date_index[j]]+=((sell_price[j]*0.995-buy_price[j])/buy_price
[j])*(stock_weight[i-1]/100.0)
    print(earn_profit[sell_date_index[j]])
    indexFile.close()

sum_earn_profit=[]

```

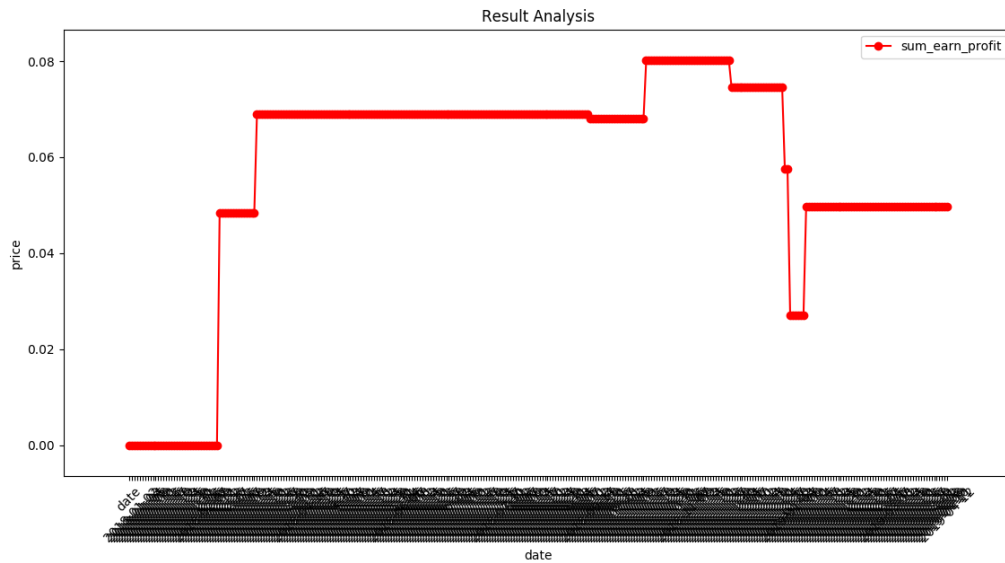
```
for i in range(0,len(earn_date)):
    sum_earn_profit.append(sum(earn_profit[0:i]))

final_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)
print('Final Time is ', final_time.tm_year, '/', final_time.tm_mon, '/',
final_time.tm_mday, '/', final_time.tm_hour,
      ":", final_time.tm_min)
# Save and Close Excel file

indexFile.close()
indexadFile.close()
exampleFile.close()

plt.title('Result Analysis')
earn_date_temp = range(len(earn_date))
plt.plot(earn_date_temp, sum_earn_profit, 'ro-', label='sum_earn_profit')
plt.xticks(earn_date_temp, earn_date, rotation=45)
plt.legend()

plt.xlabel('date')
plt.ylabel('price')
plt.margins(0.08)
plt.subplots_adjust(bottom=0.15)
plt.show()
```



(三)、AD

用 0.8 作為 AD9 和 AD26 進出場機制。(ad_model.py)

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import csv
import time

local_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
      local_time.tm_mday, '/', local_time.tm_hour,
      ":", local_time.tm_min)

exampleFile = open('StockInfo.csv')
exampleReader = csv.reader(exampleFile)
exampleData = list(exampleReader)
```

```

indexadFile = open('index_advance\\hi_lo.csv')
indexadReader = csv.reader(indexadFile)
indexadData = list(indexadReader)

indexFile = open('index\\' + exampleData[1][0] + '_' + exampleData[1][1] +
'_TW_index.csv')
indexReader = csv.reader(indexFile)
indexData = list(indexReader)
earn_date=[]
earn_profit=[]
for i in range(0,len(indexData)):
    earn_date.append(indexData[i][0])
    earn_profit.append(float(0.0))
stock_weight=[]
stock_weight.append(float(22.0))
for i in range(1,10):
    stock_weight.append(float(2.0))
for i in range(10,131):
    stock_weight.append(float(0.5))

indexFile.close()

for i in range(1,132):
    print(exampleData[i][0] + '_' + exampleData[i][1])
    indexFile = open('index\\' + exampleData[i][0] + '_' +
exampleData[i][1] + '_TW_index.csv')
    indexReader = csv.reader(indexFile)

```

```

indexData = list(indexReader)
print(indexData)
buy_price = []
buy_date_index = []
sell_price = []
sell_date_index = []

#print(indexadData[2])
#print(indexData[27])
buy_check=0

for j in range(3,len(indexadData)):
    last_hilo9=float(indexadData[j-2][9])
    current_hilo9=float(indexadData[j-1][9])
    last_hilo26=float(indexadData[j-2][10])
    current_hilo26=float(indexadData[j-1][10])
    if buy_check==0 and (current_hilo9/last_hilo9 > 0.8 and
current_hilo26/last_hilo26 > 0.8):
        buy_date_index.append(int(j+25))
        buy_price.append(float(indexData[j+25][9]))
        buy_check=1
    elif buy_check==1 and(current_hilo9/last_hilo9 < 0.8 or
current_hilo26/last_hilo26 < 0.8):
        sell_date_index.append(int(j+25))
        sell_price.append(float(indexData[j+25][9]))
        buy_check=0

for j in range(0,len(sell_date_index)):

```

```

print("buy:")
    print(buy_date_index[j])
    print(buy_price[j])
    print("sell:")
    print(sell_date_index[j])
    print(sell_price[j]*0.995)
    print("earn_per_NT1:")
    print(sell_date_index[j])
    print((sell_price[j]*0.995-buy_price[j])/buy_price[j])

earn_profit[sell_date_index[j]]+=((sell_price[j]*0.995-buy_price[j])/buy_price
[j])*(stock_weight[i-1]/100.0)
    print(earn_profit[sell_date_index[j]])
    indexFile.close()

sum_earn_profit=[]
for i in range(0,len(earn_date)):
    sum_earn_profit.append(sum(earn_profit[0:i]))

final_time = time.localtime(time.time())
print('Start Time is ', local_time.tm_year, '/', local_time.tm_mon, '/',
local_time.tm_mday, '/', local_time.tm_hour,
    ":", local_time.tm_min)
print('Final Time is ', final_time.tm_year, '/', final_time.tm_mon, '/',
final_time.tm_mday, '/', final_time.tm_hour,
    ":", final_time.tm_min)
# Save and Close Excel file

```

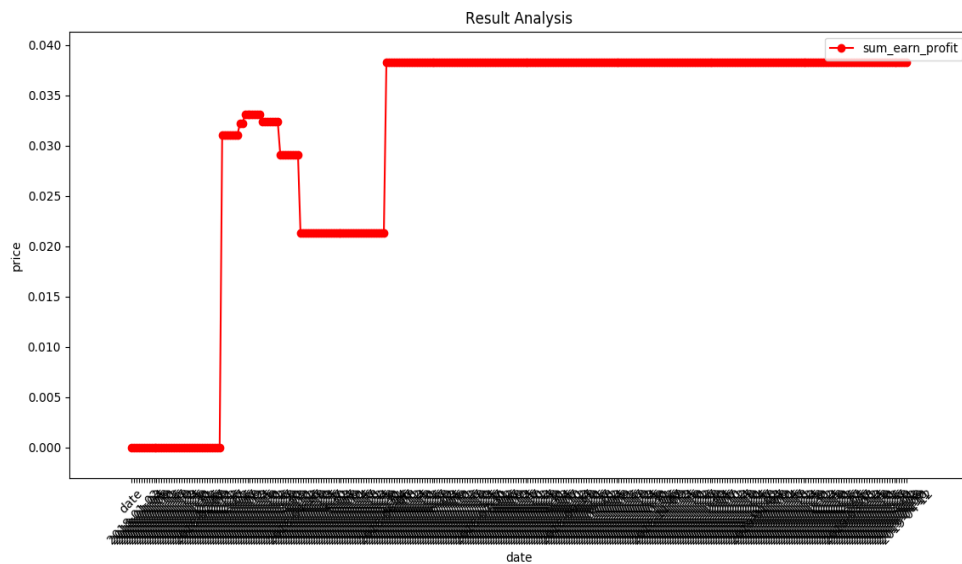
```

indexFile.close()
indexadFile.close()
exampleFile.close()

plt.title('Result Analysis')
earn_date_temp = range(len(earn_date))
plt.plot(earn_date_temp, sum_earn_profit, 'ro-', label='sum_earn_profit')
plt.xticks(earn_date_temp, earn_date, rotation=45)
plt.legend()

plt.xlabel('date')
plt.ylabel('price')
plt.margins(0.08)
plt.subplots_adjust(bottom=0.15)
plt.show()

```



資料來源：

- [file:///C:/Users/user/Desktop/PDF\(C\)%EF%BC%9A%E6%B0%B8%E8%B1%90%E6%8A%95%E9%A1%A7%E6%8A%95%E8%B3%87%E4%BA%BA%E9%A2%A8%E9%9A%AA%E5%B1%AC%E6%80%A7%E5%88%86%E6%9E%90%E5%95%8F%E5%8D%B7-%E8%87%AA%E7%84%B6%E4%BA%BA.pdf](file:///C:/Users/user/Desktop/PDF(C)%EF%BC%9A%E6%B0%B8%E8%B1%90%E6%8A%95%E9%A1%A7%E6%8A%95%E8%B3%87%E4%BA%BA%E9%A2%A8%E9%9A%AA%E5%B1%AC%E6%80%A7%E5%88%86%E6%9E%90%E5%95%8F%E5%8D%B7-%E8%87%AA%E7%84%B6%E4%BA%BA.pdf)
- <https://www.taishinbank.com.tw/TS/TS02/TS0207/TS020701/TS02070102/TSB0536052>
- https://www.ctbcbank.com/html/fileUpload/WM_Advisory/currency_unit2-1.html
- <https://www.yuantabank.com.tw/bank/tcbankMergerArea/changeItem/wealthManagement/list2.do>
- stockchart:
https://stockcharts.com/school/doku.php?id=chart_school:market_indicators:ad_line
- gupiaozmw: <https://www.gupiaozmw.com/jsfx/121154267.html>
- MoneyDJ : <https://reurl.cc/EYX8A>
- 每日頭條 : <https://reurl.cc/QNWp0>
- <https://wiki.mbalib.com/zh-tw/%E9%A3%8E%E9%99%A9%E7%88%B1%E5%A5%BD%E8%80%85>
- <https://wiki.mbalib.com/zh-tw/%E9%A3%8E%E9%99%A9%E4%B8%AD%E6%80%A7>
- <https://zh.wikipedia.org/wiki/%E9%A3%8E%E9%99%A9%E5%8E%8C%E6%81%B6>
- 股訊網 : <https://www.guxunw.com/article-1621-1.html>