Project Title:

Investigation of a Trading Strategy Using The Day of The Week and the Weather

Lee Lik Choi

Supervisor

Prof. David Rossiter

Table of Content

1. Abstract

Current Development in stock market is shifting towards trading by Algorithms, gradually replacing brokers in tradition. For professionals in financial market, technical indicators of the stock are frequently used. However, areas outside the indicators may also have a substantial effect in the market, due to the possible effect on the psychology of traders. This project aims to explore the possibilities of stock analysis outside general practice, in particular weather and day of the week.

2. Introduction

2.1 Background

Adoption of algorithms in trading has become increasingly popular. In US, the NYSE has nearly 90% [1] of their trading volume created by algorithmic trading. It shows the importance of developing a comprehensive algorithm to maximize the gain of traders. While traders may depend heavily on the traditional technical indicators, a number of parameters with mathematical importance, they may also eager to include factors outside those numbers. Weather and Week of the days are two of the many examples. These factors may have effects on the psychology of human traders, as their mental status can affect their decision to make a trade.

2.2 Concept

In this project, external factors are investigated, in hopes of providing hint to a better algorithm. It may be obvious that the mental status of the traders may cloud their judgment and influence the subsequent actions. Other than the price actions of the stocks, external factors may have an effect. Investigation on some external factors may be a possible way to optimize the trading gain.

Although the trading in US may be less affected by human, the trading by brokers may still be dominant in other countries. Even if the trading is entirely by algorithm, other factors such as release of breaking news can be affected by external factors. Therefore, these external factors are examined. In this project, Weather and Day of the Week are the two focusing areas.

3. Methodology

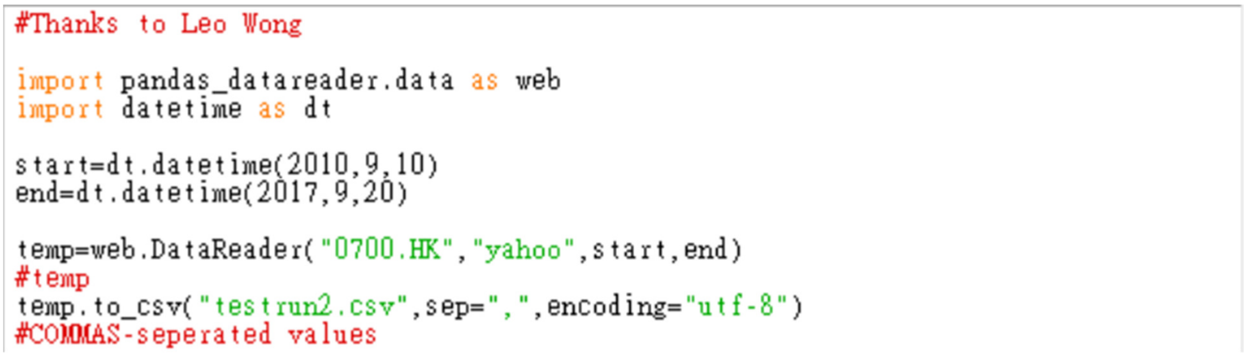
3.1 System

A stock analysis program developed by Prof. David Rossiter is used in this project. The program is in Python. A number of libraries in data science is adopted. They are Pandas data reader, Numpy, Matplotlib. Anacoda 3 is also used for easier development.

3.2 Obtain Data

Stock

To ensure the accuracy of the methods, historical stock data are obtained through Yahoo Finance. As downloading all the desired stock data manually is time-consuming, a block of code based on the one below is integrated into the system. It requests the desired stock data from Yahoo Finance and the file will be in .csv format for easier handling of data.



```
#Thanks to Leo Wong

import pandas_datareader.data as web
import datetime as dt

start=dt.datetime(2010,9,10)
end=dt.datetime(2017,9,20)

temp=web.DataReader("0700.HK","yahoo",start,end)
#temp
temp.to_csv("testrun2.csv",sep=",",encoding="utf-8")
#COMMAS-seperated values
```

Figure 1: Requesting Yahoo Finance stock data through Pandas Data Reader

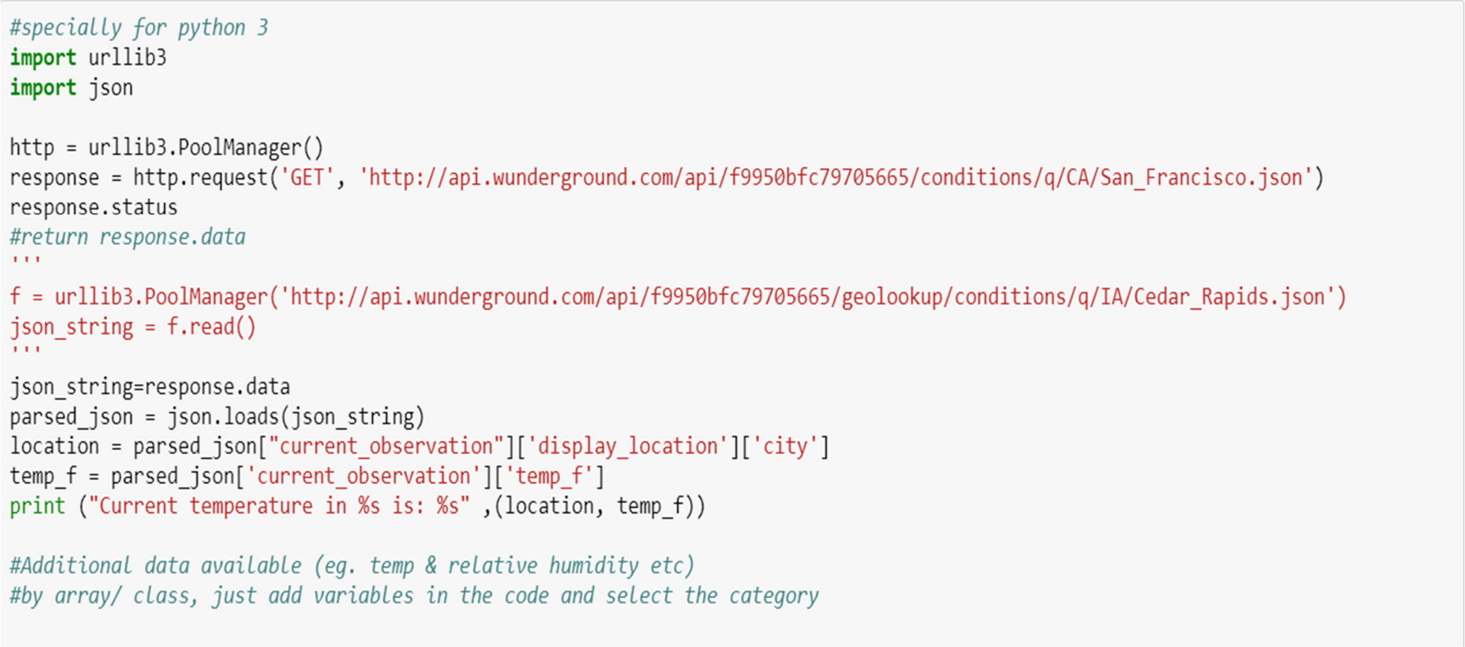
API and Changing parameters in the URL are attempted. However, it was not successful to obtain the data. API often have a query limit and may cost a little sum of money. While for downloading the stock data, the URL often has a randomized parameter to prevent computer access and to protect their servers. It is a problem of getting data from Google Finance.

Weather

Two methods are attempted to get the Historical weather data of Hong Kong. One from Underground Weather (U Weather) and another one from Hong Kong Observatory (HKO). Both provide detailed data in addition to temperature and humidity.

- U Weather

API query is available to students for free. However, the query limit is limited to 10 calls per minute and 500 calls per day. Downloading years of weather data for one location may require weeks. Therefore, it is not adopted, even though the function has been created.

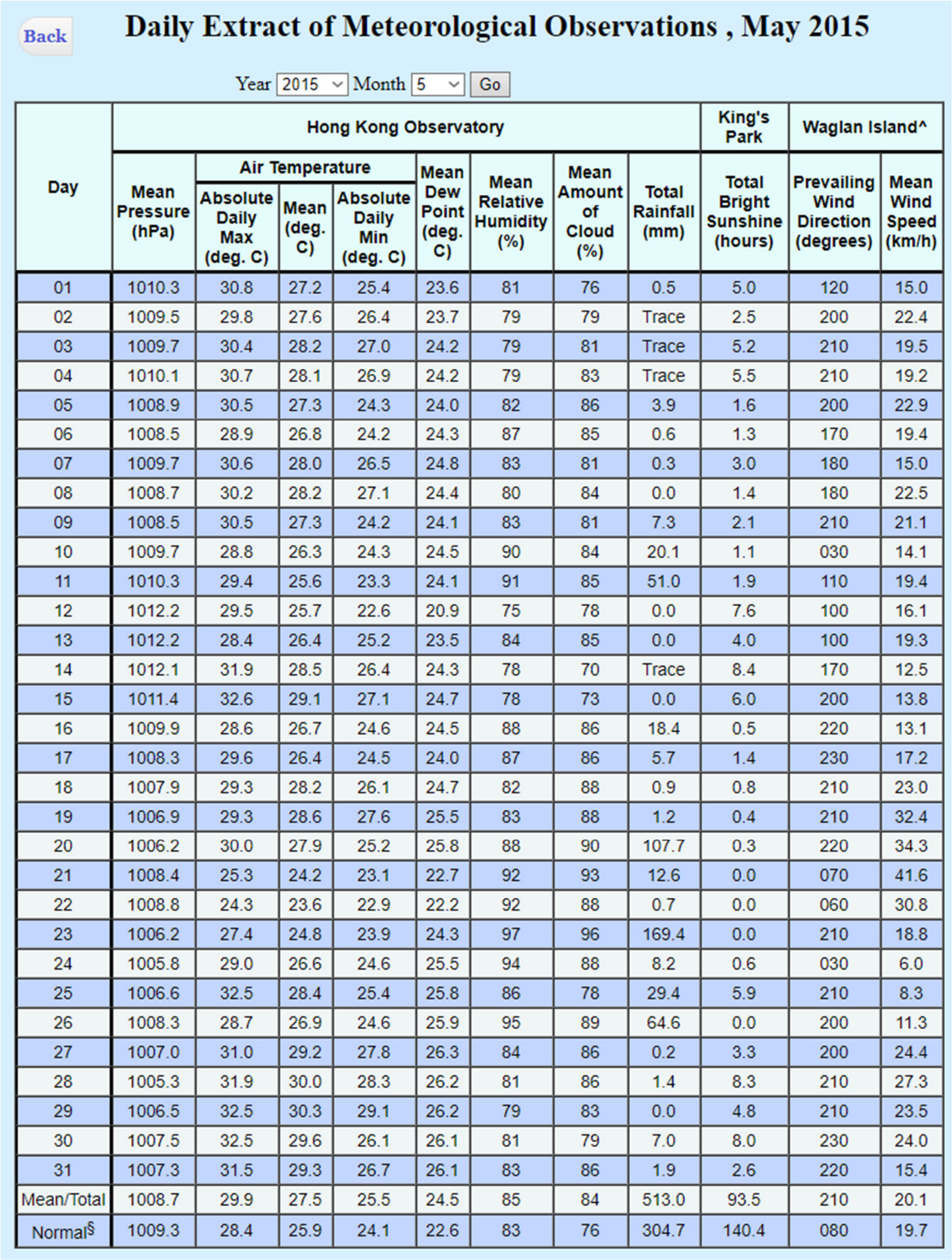


```
#specially for python 3
import urllib3
import json

http = urllib3.PoolManager()
response = http.request('GET', 'http://api.wunderground.com/api/f9950bfc79705665/conditions/q/CA/San_Francisco.json')
response.status
#return response.data
'''
f = urllib3.PoolManager('http://api.wunderground.com/api/f9950bfc79705665/geolookup/conditions/q/IA/Cedar_Rapids.json')
json_string = f.read()
'''
json_string=response.data
parsed_json = json.loads(json_string)
location = parsed_json["current_observation"]['display_location']['city']
temp_f = parsed_json['current_observation']['temp_f']
print ("Current temperature in %s is: %s" ,(location, temp_f))

#Additional data available (eg. temp & relative humidity etc)
#by array/ class, just add variables in the code and select the category
```

Figure 2: U Weather API call for Python 3

- HKO

Hong Kong Observatory is the most reliable source as it is the source of weather data in Hong Kong. The government has a website(data.gov.hk) for accessing publicly available data. API is also supported for the ease of obtaining data. However, the data retrieved was by hours, in comparison with our daily stock data. Such a data set is difficult to handle. Therefore, another approach to scrap data from the website is considered. Daily extract of the weather in Hong Kong is available on the HKO website. However, the html code of the website may try to prevent the data scrapping process by hiding the source of those data. Nodejs was considered to perform such a task.



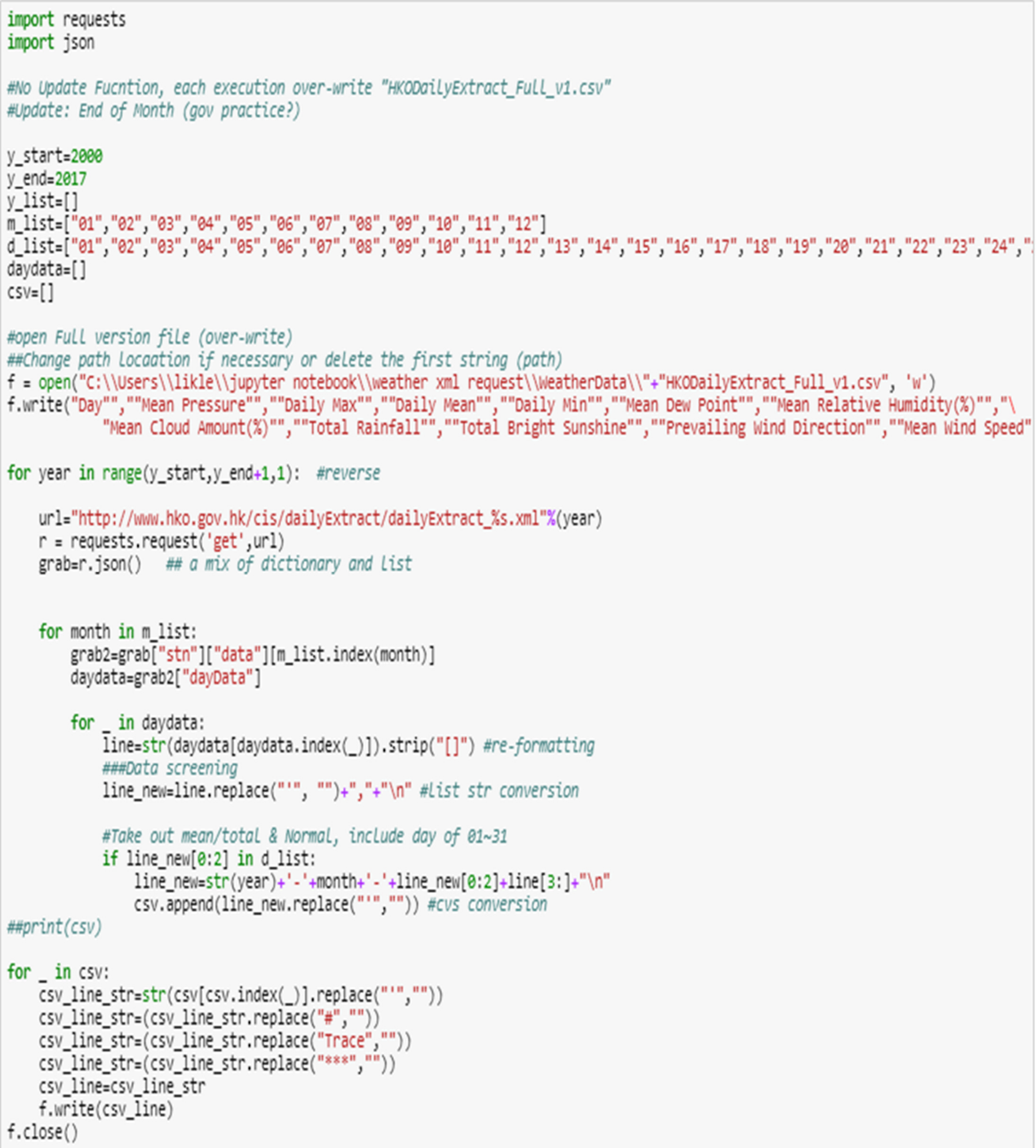
Daily Extract of Meteorological Observations , May 2015

Year 2015 Month 5 Go

Day	Hong Kong Observatory								King's Park	Waglan Island^	
	Mean Pressure (hPa)	Air Temperature			Mean Dew Point (deg. C)	Mean Relative Humidity (%)	Mean Amount of Cloud (%)	Total Rainfall (mm)	Total Bright Sunshine (hours)	Prevailing Wind Direction (degrees)	Mean Wind Speed (km/h)
		Absolute Daily Max (deg. C)	Mean (deg. C)	Absolute Daily Min (deg. C)							
01	1010.3	30.8	27.2	25.4	23.6	81	76	0.5	5.0	120	15.0
02	1009.5	29.8	27.6	26.4	23.7	79	79	Trace	2.5	200	22.4
03	1009.7	30.4	28.2	27.0	24.2	79	81	Trace	5.2	210	19.5
04	1010.1	30.7	28.1	26.9	24.2	79	83	Trace	5.5	210	19.2
05	1008.9	30.5	27.3	24.3	24.0	82	86	3.9	1.6	200	22.9
06	1008.5	28.9	26.8	24.2	24.3	87	85	0.6	1.3	170	19.4
07	1009.7	30.6	28.0	26.5	24.8	83	81	0.3	3.0	180	15.0
08	1008.7	30.2	28.2	27.1	24.4	80	84	0.0	1.4	180	22.5
09	1008.5	30.5	27.3	24.2	24.1	83	81	7.3	2.1	210	21.1
10	1009.7	28.8	26.3	24.3	24.5	90	84	20.1	1.1	030	14.1
11	1010.3	29.4	25.6	23.3	24.1	91	85	51.0	1.9	110	19.4
12	1012.2	29.5	25.7	22.6	20.9	75	78	0.0	7.6	100	16.1
13	1012.2	28.4	26.4	25.2	23.5	84	85	0.0	4.0	100	19.3
14	1012.1	31.9	28.5	26.4	24.3	78	70	Trace	8.4	170	12.5
15	1011.4	32.6	29.1	27.1	24.7	78	73	0.0	6.0	200	13.8
16	1009.9	28.6	26.7	24.6	24.5	88	86	18.4	0.5	220	13.1
17	1008.3	29.6	26.4	24.5	24.0	87	86	5.7	1.4	230	17.2
18	1007.9	29.3	28.2	26.1	24.7	82	88	0.9	0.8	210	23.0
19	1006.9	29.3	28.6	27.6	25.5	83	88	1.2	0.4	210	32.4
20	1006.2	30.0	27.9	25.2	25.8	88	90	107.7	0.3	220	34.3
21	1008.4	25.3	24.2	23.1	22.7	92	93	12.6	0.0	070	41.6
22	1008.8	24.3	23.6	22.9	22.2	92	88	0.7	0.0	060	30.8
23	1006.2	27.4	24.8	23.9	24.3	97	96	169.4	0.0	210	18.8
24	1005.8	29.0	26.6	24.6	25.5	94	88	8.2	0.6	030	6.0
25	1006.6	32.5	28.4	25.4	25.8	86	78	29.4	5.9	210	8.3
26	1008.3	28.7	26.9	24.6	25.9	95	89	64.6	0.0	200	11.3
27	1007.0	31.0	29.2	27.8	26.3	84	86	0.2	3.3	200	24.4
28	1005.3	31.9	30.0	28.3	26.2	81	86	1.4	8.3	210	27.3
29	1006.5	32.5	30.3	29.1	26.2	79	83	0.0	4.8	210	23.5
30	1007.5	32.5	29.6	26.1	26.1	81	79	7.0	8.0	230	24.0
31	1007.3	31.5	29.3	26.7	26.1	83	86	1.9	2.6	220	15.4
Mean/Total	1008.7	29.9	27.5	25.5	24.5	85	84	513.0	93.5	210	20.1
Normal§	1009.3	28.4	25.9	24.1	22.6	83	76	304.7	140.4	080	19.7

Figure 3: The Daily Extract of HKO, website view.

Yet, after looking into the JavaScript and Ajax code of the website, an URL was found to access the desired data. It was then turned into a function to automatically gather the data from the URL, by changing the query parameters. It requests data in JSON format and then the monthly data is handled and put into csv. The weather data of all the required years will be in one single file.



```
import requests
import json

#No Update Fucntion, each execution over-write "HKODailyExtract_Full_v1.csv"
#Update: End of Month (gov practice?)

y_start=2000
y_end=2017
y_list=[]
m_list=["01","02","03","04","05","06","07","08","09","10","11","12"]
d_list=["01","02","03","04","05","06","07","08","09","10","11","12","13","14","15","16","17","18","19","20","21","22","23","24",
daydata=[]
csv=[]

#open Full version file (over-write)
##Change path locaation if necessary or delete the first string (path)
f = open("C:\\Users\\likle\\jupyter notebook\\weather xml request\\WeatherData\\"+"HKODailyExtract_Full_v1.csv", 'w')
f.write("Day"","Mean Pressure"","Daily Max"","Daily Mean"","Daily Min"","Mean Dew Point"","Mean Relative Humidity(%)"","\
          "Mean Cloud Amount(%)"","Total Rainfall"","Total Bright Sunshine"","Prevailing Wind Direction"","Mean Wind Speed"

for year in range(y_start,y_end+1,1):  #reverse

    url="http://www.hko.gov.hk/cis/dailyExtract/dailyExtract_%s.xml"%(year)
    r = requests.request('get',url)
    grab=r.json()   ## a mix of dictionary and list


    for month in m_list:
        grab2=grab["stn"]["data"][m_list.index(month)]
        daydata=grab2["dayData"]

        for _ in daydata:
            line=str(daydata[daydata.index(_)]).strip("[]") #re-formatting
            ###Data screening
            line_new=line.replace("'", "")+","+"\n" #List str conversion

            #Take out mean/total & Normal, include day of 01~31
            if line_new[0:2] in d_list:
                line_new=str(year)+'-'+month+'-'+line_new[0:2]+line[3:]+"\n"
                csv.append(line_new.replace("'","")) #cvs conversion
##print(csv)

for _ in csv:
    csv_line_str=str(csv[csv.index(_)].replace("'",""))
    csv_line_str=(csv_line_str.replace("#",""))
    csv_line_str=(csv_line_str.replace("Trace",""))
    csv_line_str=(csv_line_str.replace("***",""))
    csv_line=csv_line_str
    f.write(csv_line)
f.close()
```

Figure 4: A function to download and handle the JSON format data.

3.3 Comparison

Two areas are investigated to give an idea whether such an external effect exist significantly in the market.

- Day of the week

A few graphs are generated to check whether trading at a particular day of the week gives an advantage. It is assumed that psychology of human is affected. “Monday Blue” and “TGIF” (Thanks God It’s Friday) are two examples to illustrate day of the week importance.

- Weather

The weather data is loaded into the stock analysis system as a kind of stock. It requires extra data handling to fit the special formatting in the system and lines of code are added into some functions to make the extra options compatible to the existing system. Then a back-test is run to give a simulated result of trading under those given instructions, which is adopted and adjusted from the code of Prof. Rossiter.

4. Result and Interpretation

4.1 Day of the Week

The Graphs are generated by using Matplotlib, Numpy and Pandas Data Reader. The graph shows the percentage gain or loss compared to the previous trading day, with “1” as Monday and “5” as Friday. The historical data of Apple Inc. (AAPL.US) is used.

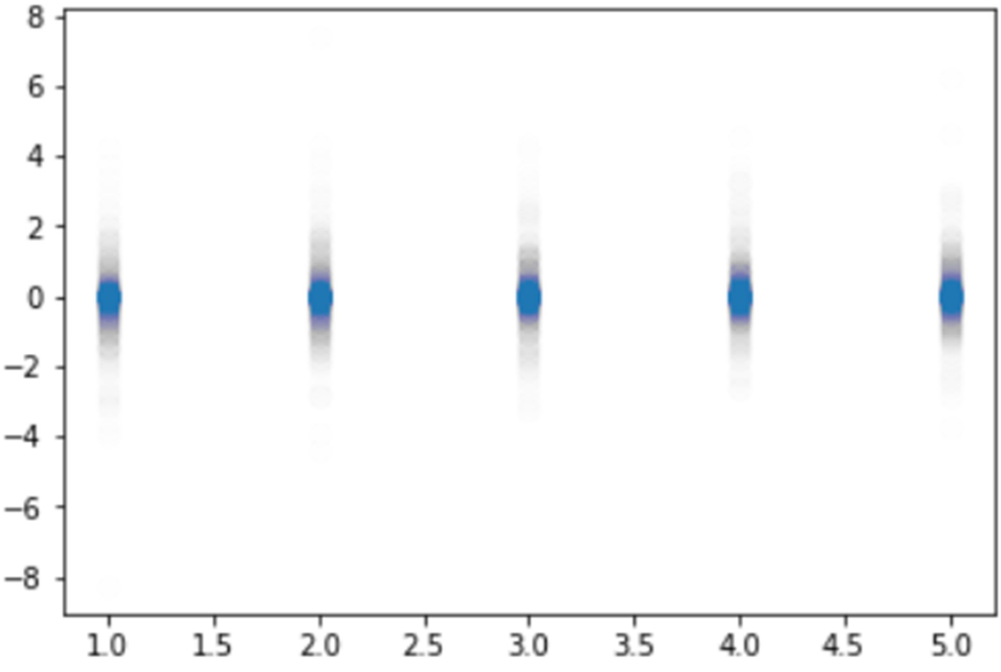


Figure 5: A graph showing the distribution of daily stock change.

However, the difference is not too obvious to observe. Then it was converted into a box and whisker diagram. The vertical axis shows the percentage in float number (0.05 equals to 5% etc.). It shows the upper and lower quartile as well as the median. Minor difference exists as those three mentioned do not lie on the same position.

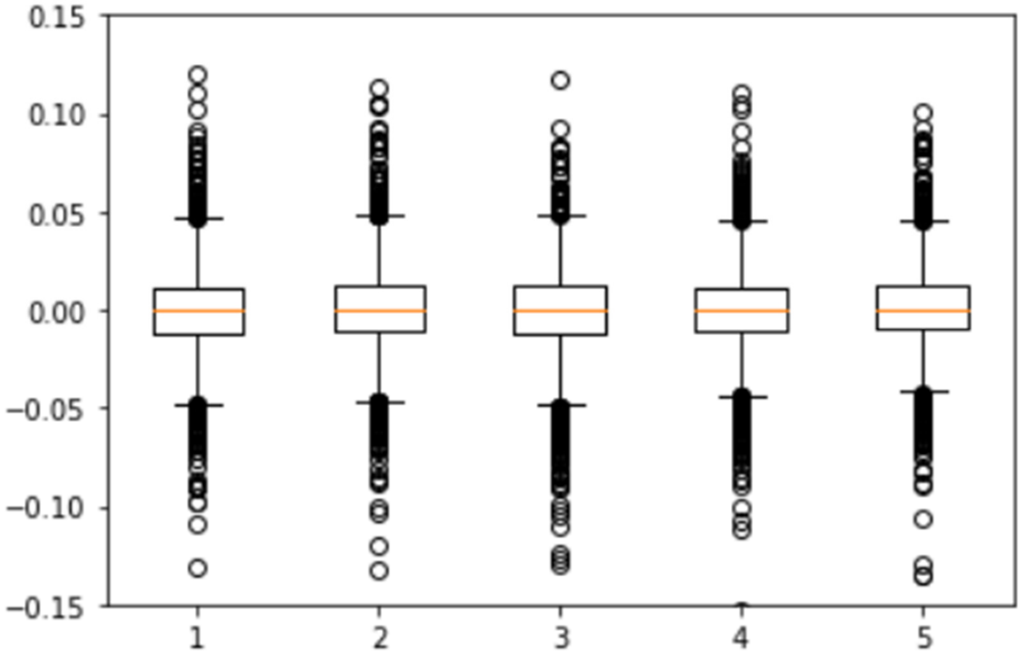


Figure 6: A box and whisker diagram to show the detailed distribution.

This one below is another graph showing the absolute value of the percentage change. The overall distribution ranger differs, in which Friday may have the narrowest range. It is possible that traders limit their trading as more variation (news or other factors) may arise over the weekend.

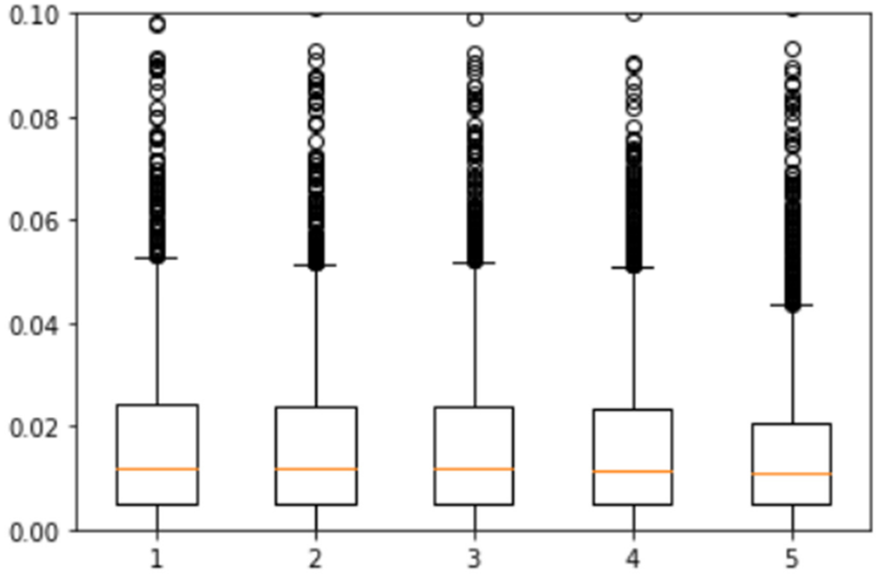


Figure 7: A diagram showing the absolute value of the change in stock price.

The “box” part of the diagram is further enlarged for easier observation. The diagram shows that the median of the days differs, with Monday and Wednesday higher and Friday lower. The difference is about 0.1% to 0.2%, which might be a great difference for a day trade.

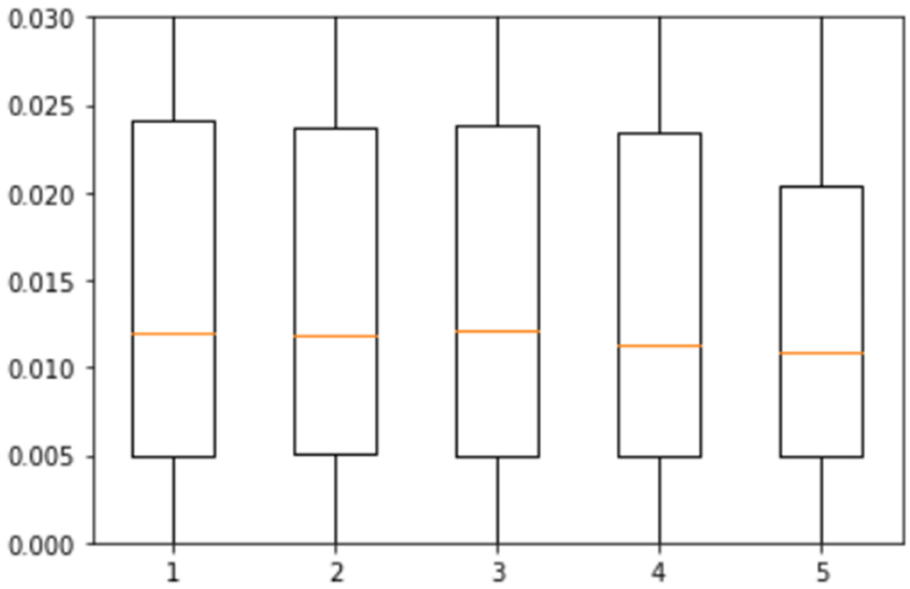


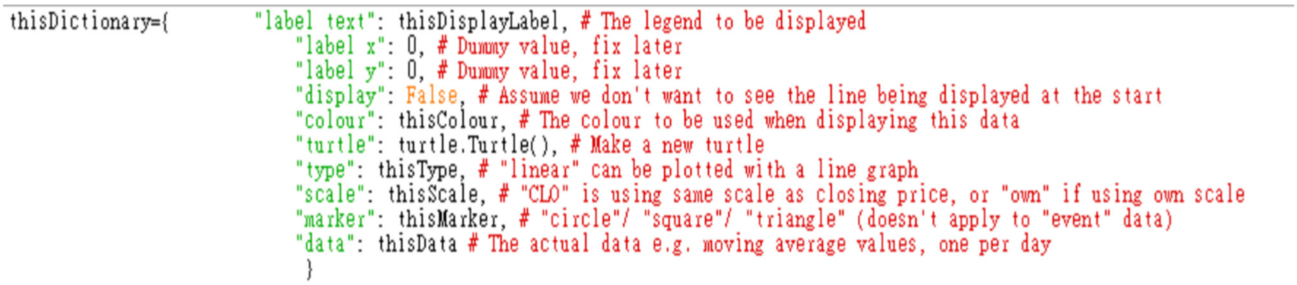
Figure 8: An enlarged view of the previous graph.

4.2 Weather

In this part, an extra program called “WeatherExtension.py” (WE)is created. It serves as an extension to the system Stock_v20. It has several functions. First, it helps to convert the Hong Kong weather data into a type of “stock” for comparison, as the previous system has several checks on the data validity, which it cannot be passed easily. It also contains a part to create WCI index and a back-test, which will be explained later.

Data Conversion

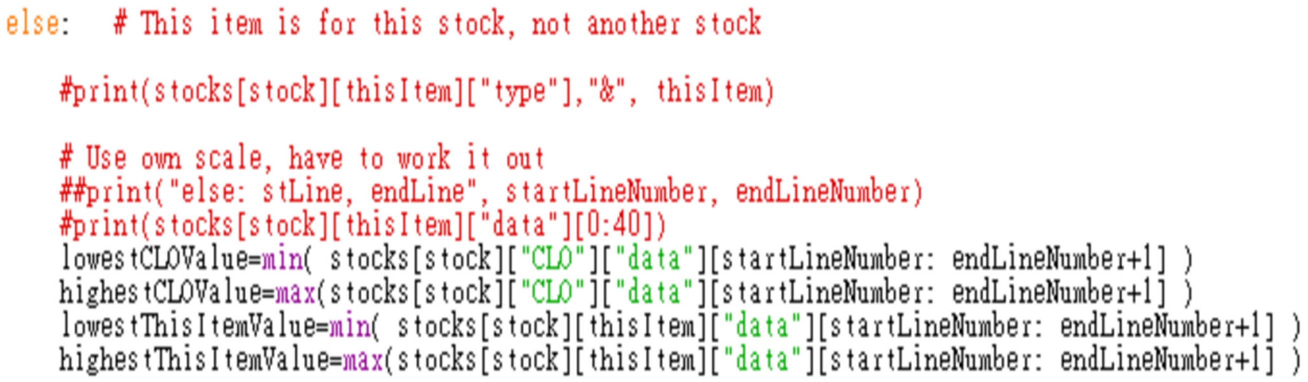
To properly display the weather information and reduce the complexity of the program, the weather is treated as a type of stock. In WE, it copies the sample stock “BLNK.US” (a blank stock) and fill in zero to get the desired dictionary in each of the six items. (Volume, Open price etc.). Special data handling is required to handle a few different situations.



```
thisDictionary={        "label text": thisDisplayLabel, # The legend to be displayed
                           "label x": 0, # Dummy value, fix later
                           "label y": 0, # Dummy value, fix later
                           "display": False, # Assume we don't want to see the line being displayed at the start
                           "colour": thisColour, # The colour to be used when displaying this data
                           "turtle": turtle.Turtle(), # Make a new turtle
                           "type": thisType, # "linear" can be plotted with a line graph
                           "scale": thisScale, # "CLO" is using same scale as closing price, or "own" if using own scale
                           "marker": thisMarker, # "circle"/ "square"/ "triangle" (doesn't apply to "event" data)
                           "data": thisData # The actual data e.g. moving average values, one per day
                            }
```

Figure 9: The data format in the program

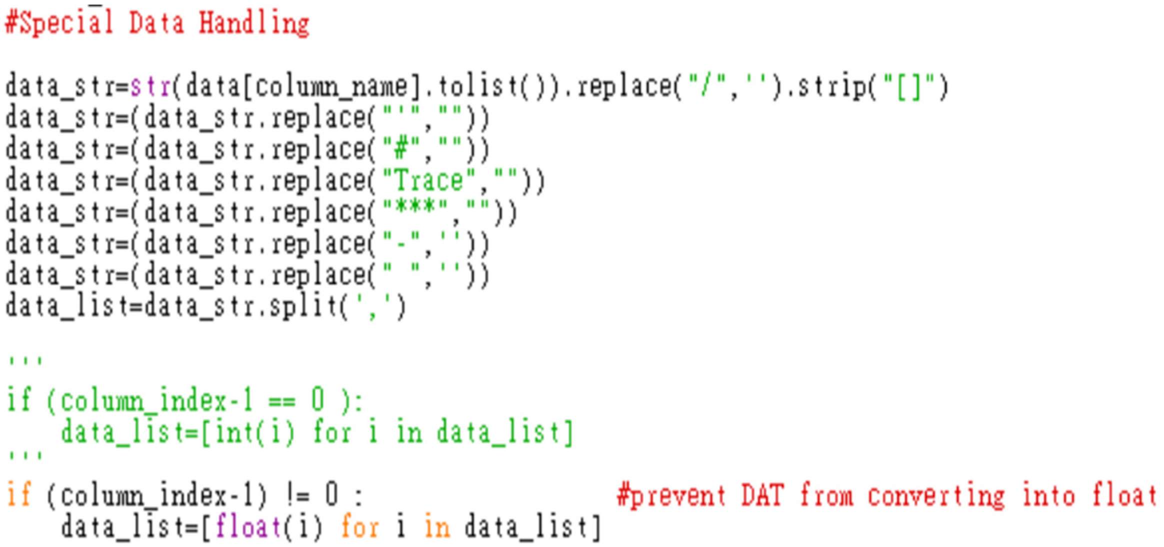
The graph cannot be drawn if all the data are zeros, as the min and max will be the same, resulting in “float division by zero” error.



```
else:   # This item is for this stock, not another stock

    #print(stocks[stock][thisItem]["type"],"&", thisItem)

    # Use own scale, have to work it out
    ##print("else: stLine, endLine", startLineNumber, endLineNumber)
    #print(stocks[stock][thisItem]["data"][0:40])
    lowestCLOValue=min( stocks[stock]["CLO"]["data"][startLineNumber: endLineNumber+1] )
    highestCLOValue=max(stocks[stock]["CLO"]["data"][startLineNumber: endLineNumber+1] )
    lowestThisItemValue=min( stocks[stock][thisItem]["data"][startLineNumber: endLineNumber+1] )
    highestThisItemValue=max(stocks[stock][thisItem]["data"][startLineNumber: endLineNumber+1] )
```

Figure 10: The lines of code with bug.

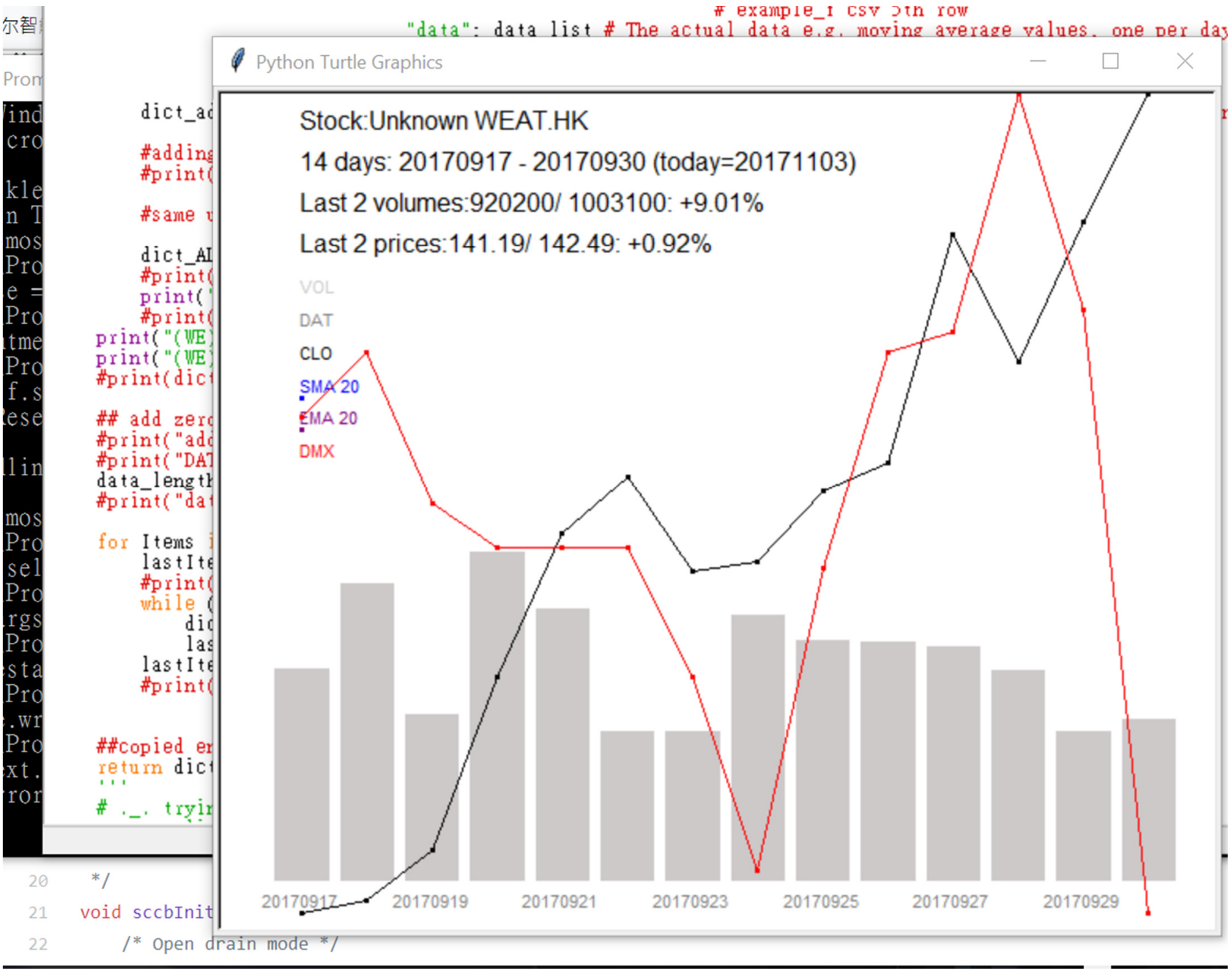
Another data handling process, removing the special case in the HKO weather csv document and converting the date to the format desired.



```
#Special Data Handling

data_str=str(data[column_name].tolist()).replace("/",'').strip("[]")
data_str=(data_str.replace("'",""))
data_str=(data_str.replace("#",""))
data_str=(data_str.replace("Trace",""))
data_str=(data_str.replace("***",""))
data_str=(data_str.replace("-",''))
data_str=(data_str.replace(" ",''))
data_list=data_str.split(',')

'''
if (column_index-1 == 0 ):
    data_list=[int(i) for i in data_list]
'''
if (column_index-1) != 0 :                     #prevent DAT from converting into float
    data_list=[float(i) for i in data_list]
```

Figure 11: Code for the data handling process.

After a complicated process of debug and data handling, the weather data is successfully added into the stock (WEAT.HK). It can be displayed with another stock.



```
# example_1 csv 5th row
"data": data_list # The actual data e.g. moving average values, one per day
```

```
Stock:Unknown WEAT.HK
14 days: 20170917 - 20170930 (today=20171103)
Last 2 volumes:920200/ 1003100: +9.01%
Last 2 prices:141.19/ 142.49: +0.92%
```

```
*/
void sccbInit
    /* Open drain mode */
```

Figure 12: the program displaying the weather data.

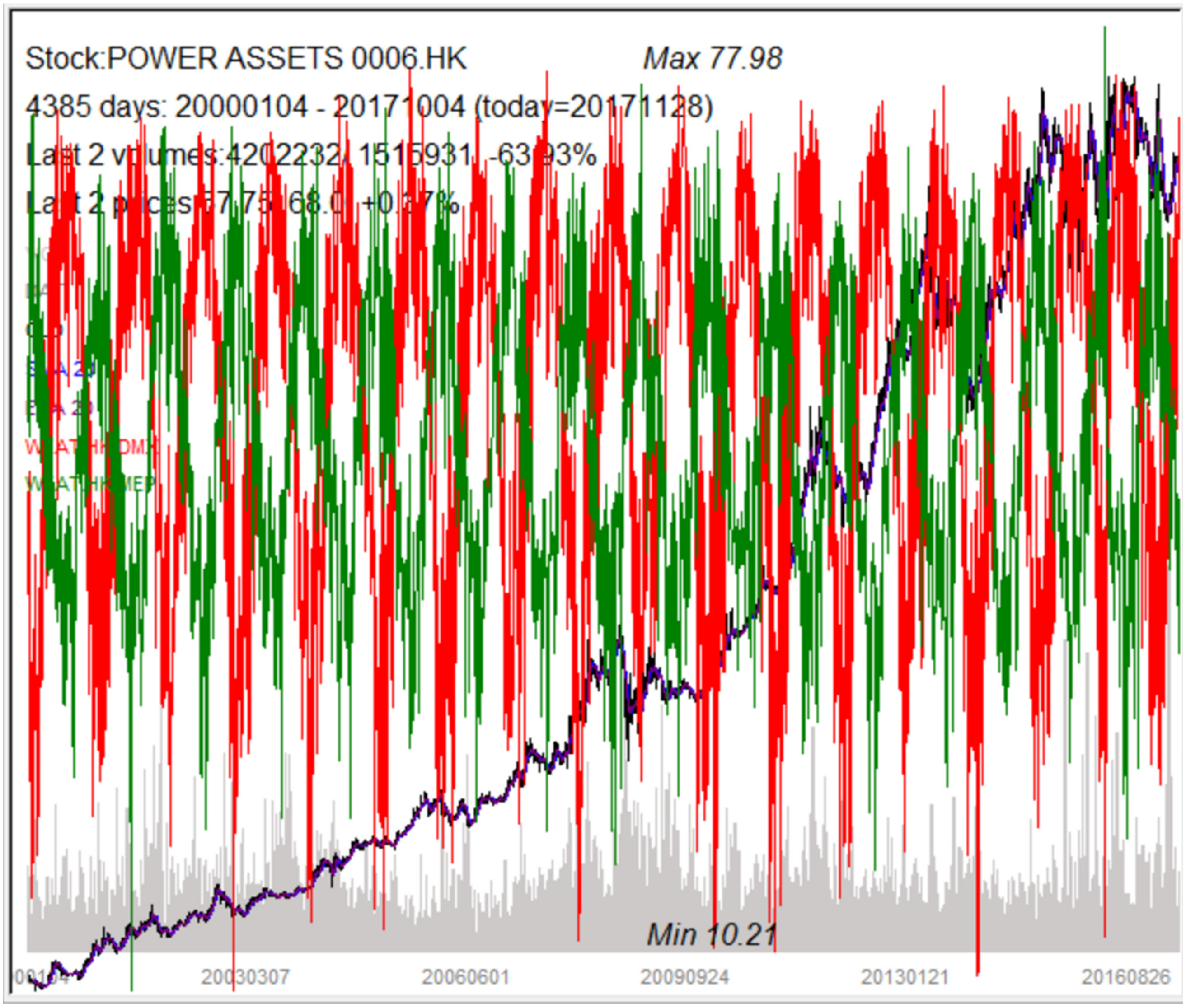


Figure 13: Display in comparison with another stock.
Red line is Daily Max Temperature, DMX. Green line is Mean Pressure, MEP.

4.3 WCI index and Back-test

4.3.1 WCI Indices

To turn the weather conditions into a parameter for trading, an Weather Comfy Index (WCI) is created to measure how comfortable is the weather. It is assumed that when people feel comfortable, they are more likely to buy and when they are uncomfortable, they are more likely to sell.

Best Weather
Temperature:

72F (22C) is considered the most comfortable temperature [2] and 86F is “thermal neutral”, meaning any temperature above will cause uncomfortable feelings, as it affects the heat transfer.

Humidity:

Relative Humidity affects heat dissipation. Below 30% and above 52% perceived as "dry" and "moist" to most human [3]. The most comfortable percentage is 45.

Dew Point:

The Dew point is a relationship of humidity and temperature. It can give a more accurate "feeling" as it represents the heat dissipation of human in some extent. 50 to 64 F (10 ~17.78 C) is the comfortable range for human [4]. The calculation can be approximated as **Td = T - ((100 - RH) /5)** [5].

Wind Direction and speed:

Wind direction may also affect the pollutant content in air. However, with reference to the Air Quality Index from EPD [6], there is no significant correlation between the air speed and direction, with the AQI.
With insignificant evidence support, these factors are excluded from the index.

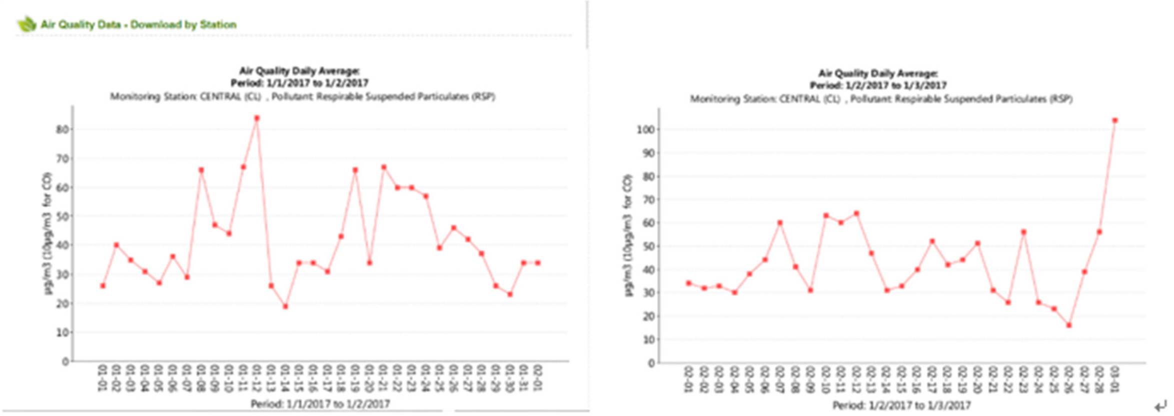


Figure 14: The Air Quality Index daily values in each month.

WCI Index (weather only)
Therefore, WCI index is created to measure the comfort level of the weather on that day. Its composition is as follow.

Temperature 20%

"best temp" +/- 1	+20
Deviation within 3C	+10
Deviation above 3C	+0

Mean Dew Point 50%

10 ~ 17.8 C	+50
Deviation within 1C outside the range	+20
Deviation above 1C outside the range	+0

Relative Humidity 30%

45% +/- 3%	+30
30~52%	+15
Outside the range	+0

As Dew Point gives a better approximation of human feeling to weather, it has 50% portion in the index. While extreme weather may happen at some time of the day, leading to lower level of comfort, they are also considered.

WCI Index (weather and day of the week)

This WCI Index is the same as the one above, with inclusion of week of the day as a factor. There are two versions of this index, WCI-A and WCI-B. The WCI-B is the reverse version of WCI-A in value. It serves as a comparison to show the possible positive effect of Friday and the possible negative effect of Monday and Wednesday.

WCI-A

Week of the Day additional 5%

Monday or Wednesday	-5
Friday	+5
Other	+0

WCI-B

Week of the Day additional 5%

Monday or Wednesday	+5
Friday	-5
Other	+0

This extra index is to test the minor effect of the “week of the day” factor. As found above, in day trading, there is a larger variation on Monday and Wednesday, in comparison with Friday.

4.3.2 Back-Test Part I

The back-test will buy when the index is larger than or equal to 70 (WCI > =70) and there is no stock in hand. It will sell when the WCI is lower than 40 (WCI <40) and there is stock in hand. Both transaction are done using the closing price of that day. Yellow line represents the worth over time (1 million capital at start). Green letter “B” means a Buy transaction, Red letter “S” means a Sell transaction. “Buy and Hold” strategy, which buys on the first day and hold until the last day, will be a benchmark for comparison.

Two stocks are chosen to carry out the back-test. 0006.HK is a blue chip that may represent the energy industry. While 2800.HK follows the changes in Hang Seng Index (HSI), such that it may represent the overall market trend. It may help to find out whether the application of such method is more applicable to a single stock or a general market performance.

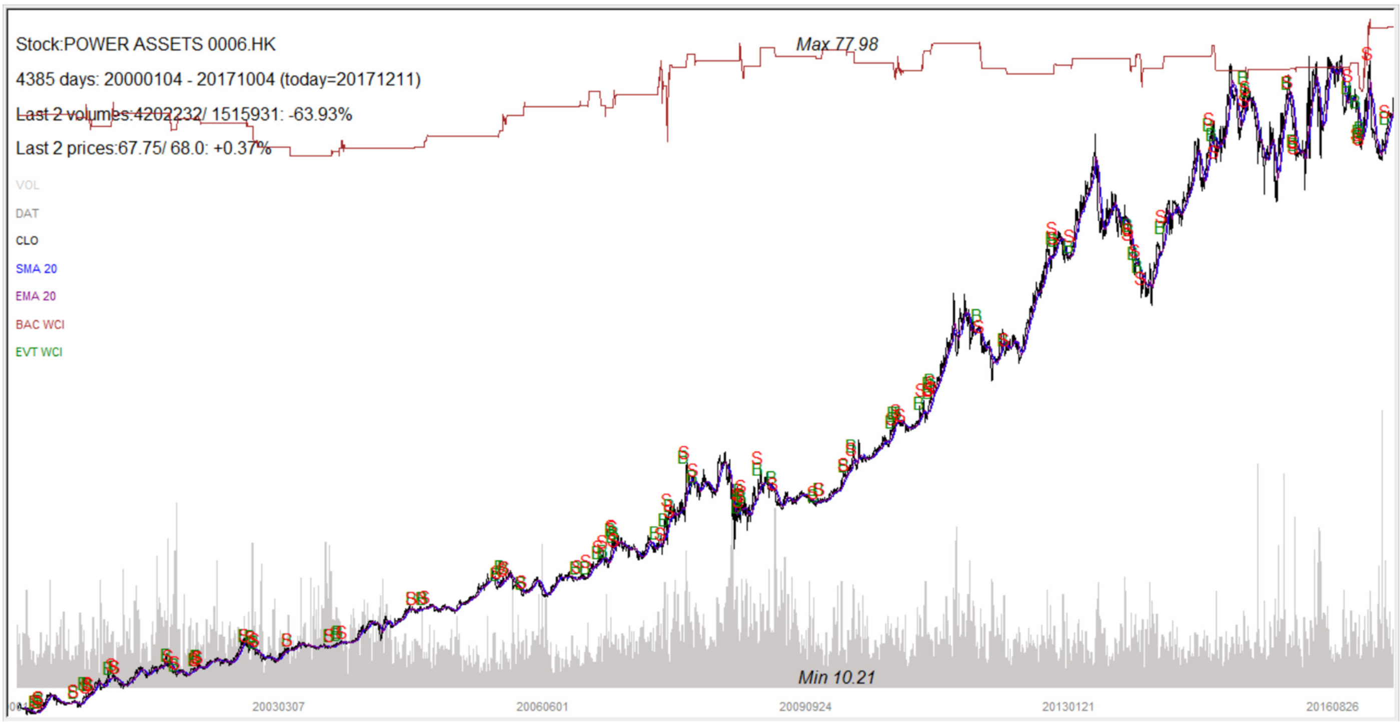


Figure 15: The program display of the back-test result with 0006.HK. (+5.14% net worth.)

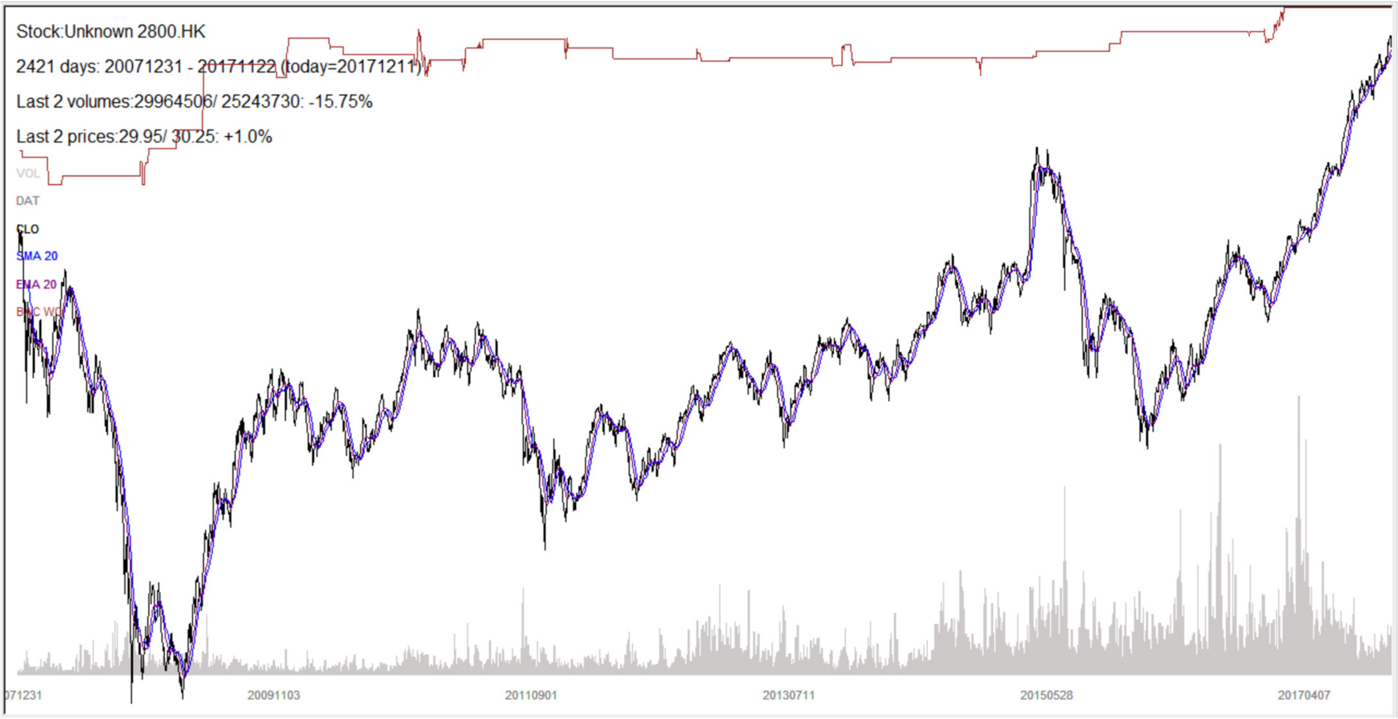


Figure 16: The program display of the back-test result with 2800.HK. (+26.03% net worth.)

Part I

"Buy and Hold" Result Summary

Stock	duration	Net worth % change
0006.HK	4385 days	+527.51%
2800.HK	2421 days	+26.47%

WCI (weather only) Result Summary

Stock	duration	Net worth % change
0006.HK	4385 days	+5.14%
2800.HK	2421 days	+26.03%

For Power Asset (0006.HK), it generates around 5.14% revenue in net worth. It is not a good strategy as the growth of the stock is much more than 5% (527.51%) as shown in the graph.

For tracker fund (2800.HK), it generates 26.03% revenue in net worth. It might be a good strategy, since it avoids the significant fall in 2008 and gives a decent profit. However, it is unclear that how it could avoid the financial crisis in 2008.

Compared to Power Asset, tracker fund is a more general representation of the stock market in Hong Kong (closely related to Hang Sang Index), it might suggest that the weather can be a factor to consider for the overall market performance, rather than the single performance of a stock. The buying pattern for Hong Kong stocks with market representation will be the same as such trading decision depends on the WCI only. It means that similar result will be achieved. Therefore, it would be the same for those stock. But it is unclear that if such a strategy gives better result without the economic downturn.

4.3.2 Back-Test Part II

As mentioned, this part will include "day of the week" factor (additional 5%) to the WCI. The trading settings are identical. It buys when WCI-A/WCI-B is larger than or equal to 70 and sells when it falls below 40.

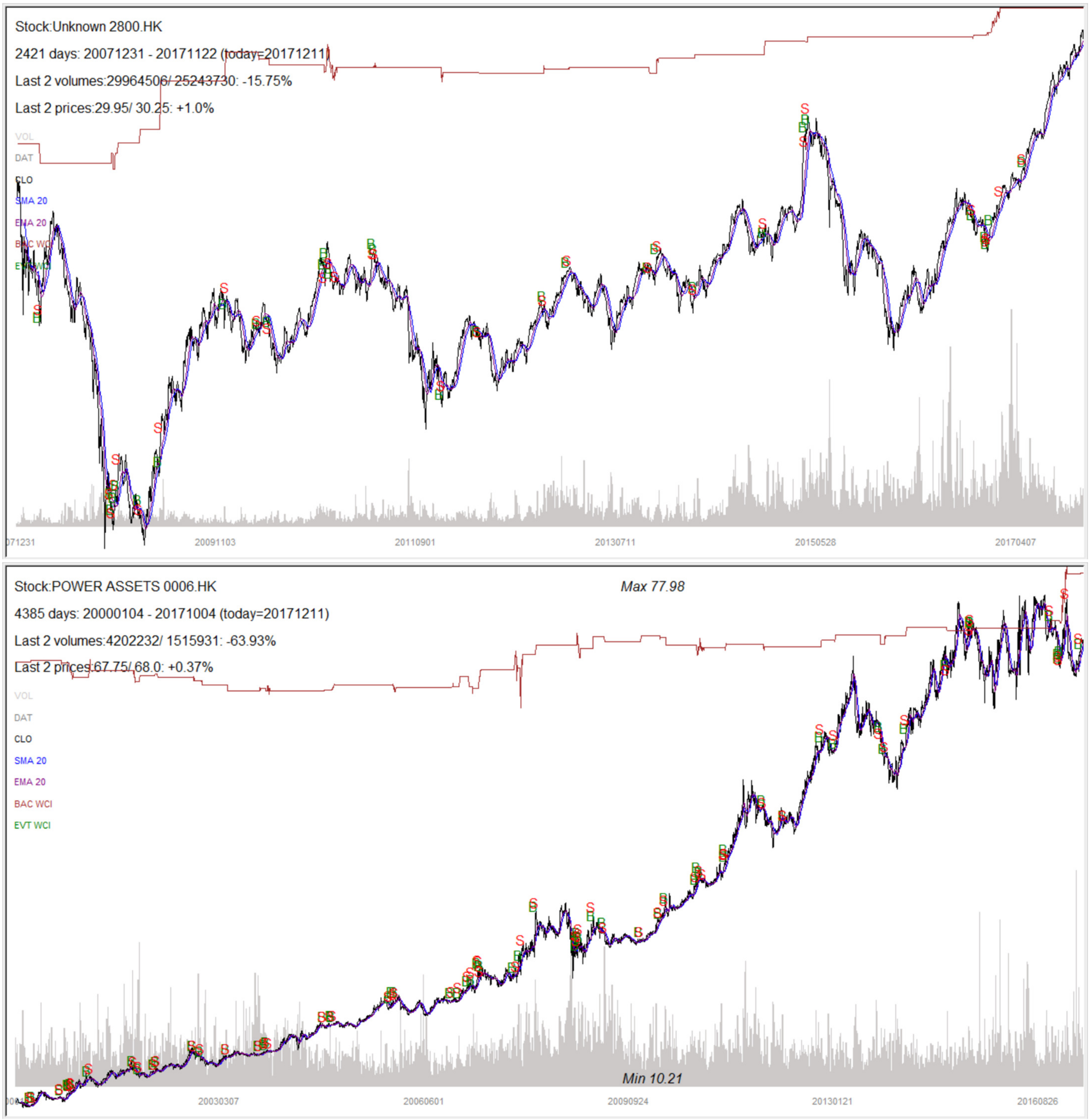


Figure 17 & 18 : Gain in net worth using WCI-A strategy. (+33.62% & +20.05% net worth respectively)

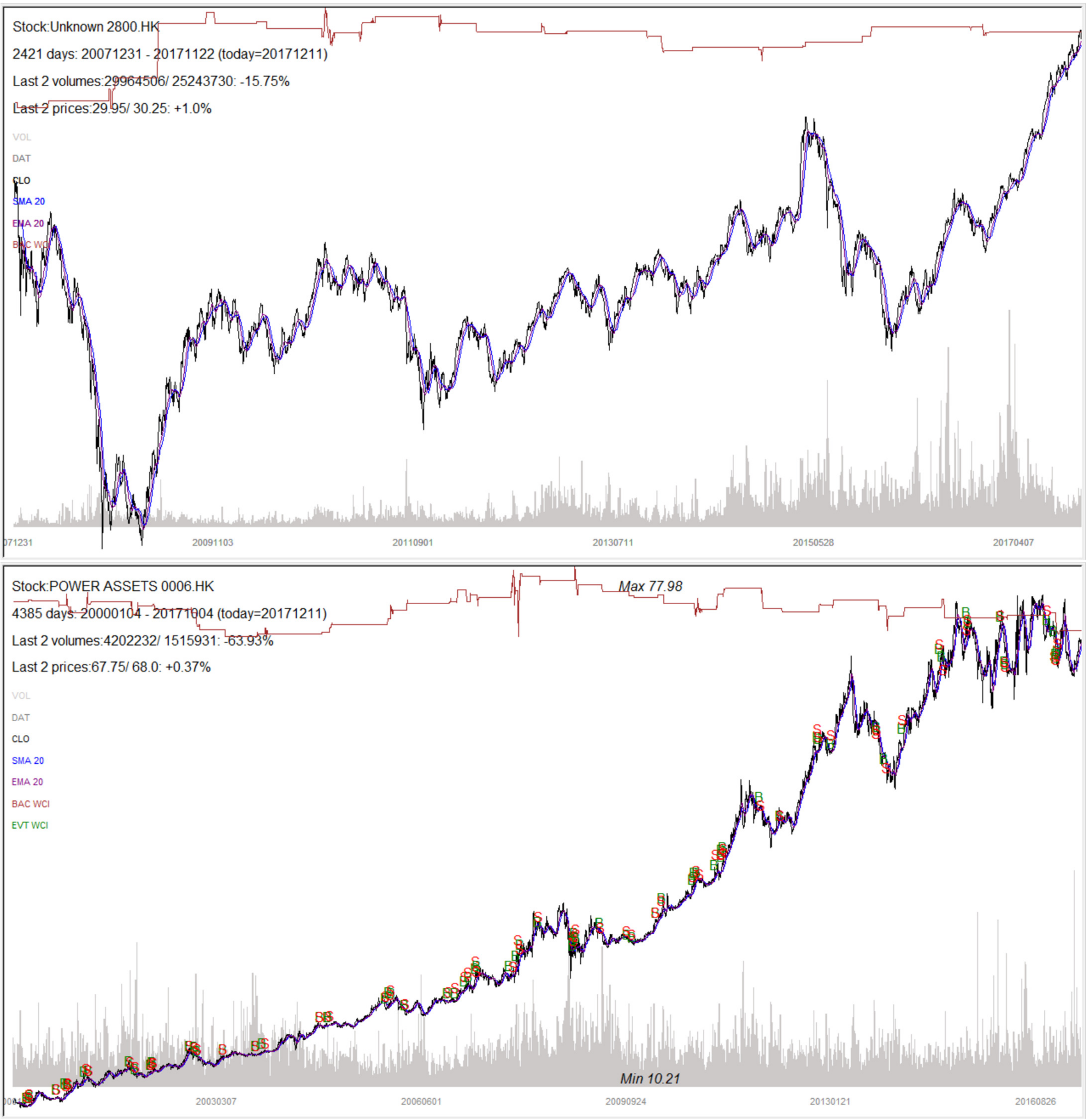


Figure 19 & 20: Gain in net worth using WCI-B strategy. (+15.85% & -5.71% net worth respectively.)

WCI-A (weather and day of the week) Result Summary

Stock	duration	Net worth % change
0006.HK	4385 days	+20.05%
2800.HK	2421 days	+33.62%

WCI-B (weather and day of the week) Result Summary

Stock	duration	Net worth % change
0006.HK	4385 days	-5.71%
2800.HK	2421 days	+15.85%

WCI-A strategy, which preferably buy on Friday (closing price) and sell on Monday and Wednesday (closing price) has a roughly 15% and 7 % increase on respective stocks, comparing to WCI strategy. While WCI-B, which preferably sell on Friday (closing price) and buy on Monday and Wednesday (closing price) has a 10% decrease on both stocks, in comparison with WCI. This shows that the trading day may have an effect on the trading and when it is combined with the WCI strategy, it may generate even more net worth. In this case, a preference of buying on Friday (closing price) and selling on Monday and Wednesday (closing price) would generate more profit, as compared to the reverse. It might also mean that in the WCI strategy conditions, stock prices on Friday are generally lower than Monday and Wednesday. Also, the performance of WCI-A is better than the "Buy and Hold" strategy on 2800.HK(+33.62% vs +26.47%), which represents the general market movement. It might be a useful strategy for trading the general market trend, instead of specific stocks (eg 0006.HK). Nonetheless, the rationale behind such result is unknown.

5. Conclusion

Including external factors like week of the day and the comfort level of weather may help to improve the performance of a trading strategy. In week of the day, Monday and Wednesday has a higher variation, while Friday has a relatively stable market. For the comfort level, it may help to avoid the market downfall to some extent. Combining both factors may be beneficial in general stock, as they can generate a considerable amount of profit comparing to specific stocks. However, it is important to note that the data size (stock with market representation) might not be enough to prove its correlation and the reason behind is uncertain.

6. References

[1] M.H. Melin, "How Systematic, Algorithmic Trading Impacts Stocks, And How To Benefit From This Understanding," *Seeking Alpha*, 22-Aug-2017. [Online]. Available: https://seekingalpha.com/article/4101073-systematic-algorithmic-trading-impacts-stocks-benefit-understanding. [Accessed: 27-Nov-2017].

[2] *Direct Energy*, "What is the Right Temperature for My Air Conditioner in Summer?" 20-Nov-2017. [Online]. Available: http://www.directenergy.com/blog/right-temperature-air-conditioner-summer. [Accessed: 27-Nov-2017].

[3] NWS Heat Safety Home Page and US Department of Commerce, NOAA, National Weather Service, “NWS Heat Safety Home Page,” *National Weather Service*, 01-Jan-2001. [Online]. Available: http://www.nws.noaa.gov/om/heat/heat_index.shtml. [Accessed: 27-Nov-2017].

[4] S. Horstmeyer, “Relative Humidity....Relative to What?” [Online]. Available: http://www.shorstmeyer.com/wxfaqs/humidity/humidity.html. [Accessed: 27-Nov-2017].

[5] M. Bell, “Dew Point- Basic Information,” [Online]. Available: https://iridl.ldeo.columbia.edu/dochelp/QA/Basic/dewpoint.html. [Accessed: 27-Nov-2017]
.
[6] *Environmental Protection Department*, “Inquire and Download Air Quality Monitoring Data,” 31-Oct-2017. [Online]. Available: http://cd.epic.epd.gov.hk/EPICDI/air/station/?lang=en. [Accessed: 27-Nov-2017]