

# HW1

HW1 繳交時間:2019/3/22 23:59 分。

HW1 作業包含「ETF 爬蟲」與「財金指標爬蟲」兩個部分，關於「評比績效介紹」改為下次作業，會擇期公布繳交時間。

## ETF 爬蟲解說

1. 請每組請參考"HW1.rar"中的檔案 HW1 作業分組.xlsx，並根據檔案內「HW1\_爬蟲與指標選定」欄位在"ETF 爬蟲"資料夾中找到你們所屬的資料夾，裡面會有你們要找尋的 ETF 清單。例如在"HW1\_爬蟲與指標選定"欄位中第一組要找的是 BOND，打開路徑為"ETF 爬蟲\BOND (第一組~第五組)\第 1 組"的資料夾後，請參考"Total Bond Market ETF List (82).csv"的檔案。
2. 請各組在所指定的 ETF 資料中，先篩選出於 2015 年年底前既已存在的 ETF(可從 EXCEL 檔中的 W 欄看到時間)。(原先說要特地存成一個 excel 檔案就不用了)

承上，請利用爬蟲方法整理出上述篩選後的 ETF 集合中，其每一檔 ETF 的每日收盤價，時間範圍從該 ETF 於 2015 年年底最後一個交易日起至程式執行的當下(Now()函數)，將每日的資料都抓下來，不需要分月底或三個月(這樣對你們來說比較簡單)。並將整理好的資料彙整在 pandas 的 dataframe 中，如圖 1 顯示，橫軸為 ETF 資料(標頭為「該 ETF 名稱」)，縱軸為年-月-日(格式是 Python 的 Datetime 'yyyy-mm-dd')，標頭為「Date」。如果該 ETF 在該時間沒有資料就空白。如果是用 Jupyter 的人請執行完後搭配執行結果將.ipynb 檔上傳 GitHub 這樣方便助教我檢查是否有完成，如果不是的也請在 Github 上放一個執行後的截圖。

Date	LQD	VCSH	VCIT	.....
2015-12-31	102.51	73.35	75.66	.....
2016-01-04	102.45	73.31	75.76	.....
⋮				
2019-02-21	115.50	78.3	84.59	.....
2019-02-22	116.02	78.82	84.90	.....
⋮	⋮	⋮	⋮	⋮

圖 1.

3. 如果該 ETF 的 Home page 沒有讓你下載每日收盤價的連結的話，請到 Yahoo Finance 輸入你 ETF 的名稱後找尋資料 <https://finance.yahoo.com/>，以「Adj Close\*\*」為準，但希望大家還是先以 Homepage 為主。如果以上方法都不行，請將該 ETF 紀錄在 HW1 資料夾內的 README.md 中，標記為「無法完成的 ETF 名稱」
4. 在 GitHub README.md 上要寫的教學文件必須包含下面內容：
  1. 你選擇用甚麼樣的套件來做網路爬蟲?為什麼要用這個套件
  2. 請用流程圖的方式告訴我們你是怎麼抓到你的目標資料，流程圖的畫法不拘，主要易懂就好
  3. 至少設想並列出 5 種當別人使用你的程式最有可能會遇到的錯誤情況，並提供解決辦法

作業 Output 的項目：

1. GitHub 上面新增一個 HW1 的資料夾，將完整程式碼放在裡面，以及執行結果(或是截圖)
2. README.md 上要寫的教學文件
3. 呈現畫面的內容請參考解說點 3

## 財金指標爬蟲解說

1. 請小組參考"財金指標爬蟲"資料夾中"每組分配到的爬蟲題目.xlsx"，針對分給你們的指標與參考網站，嘗試將該指標的歷史資料爬下來(指標介紹與參考網站在"24 個美國重要經濟指標.docx"中)，時間範圍盡可能從該網站有紀載資料的時間起至程式執行的當下(Now()函數)，將所有的歷史資料都抓下來。並將整理好的資料存在 pandas 的 dataframe 中，橫軸為指標數值，標頭為「Value」，縱軸為年-月-日(格式是 Python 的 Datetime 'yyyy-mm-dd')，標頭為 Date」，並用 df.head(20)呈現出前 20 筆資料(若用 Jupyter 的人請執行完後再上傳 GitHub，若不是的，結果截圖放在 github 內方便我檢查)輸出結果請參考圖 2。

Date	Value
1991-01-01	3.1
1991-04-01	3.16
1991-07-01	3.23
1991-10-01	3.3
1992-01-01	3.13
1992-04-01	3.04
1992-07-01	2.89
1992-10-01	2.79
1993-01-01	2.79
1993-04-01	2.66
1993-07-01	2.57
1993-10-01	2.46
1994-01-01	2.38
1994-04-01	2.25
1994-07-01	2.16
1994-10-01	2.07
1995-01-01	2.08
1995-04-01	2.11
1995-07-01	2.16
1995-10-01	2.18

圖 2.

2. 相關範例程式可以參考我的 github: [https://github.com/kevin21326/web\\_crawler\\_4\\_financial\\_index](https://github.com/kevin21326/web_crawler_4_financial_index)，裡面有做 Delinquency Rate on Single-Family Residential Mortgages 的資料爬蟲範例

作業 Output 的項目：

1. 將完整程式，以及執行結果(或是截圖)放在 GitHub 上的 HW1 的資料夾。
2. 呈現畫面的內容請參考解說點 1