資料工程初探
以開放資料清理轉換為例

王學治

【前言】

近年在開放資料趨勢盛行下，普羅大眾能獲得的資料已走向多元，雖然資源豐富，但從資料使用者的角度來看，取得到利用之間，實務上仍存在不少挑戰，例如多樣化檔案格式的處理、檔案內文編碼的轉換、與解讀資料結構的能力與熟悉度等等，對於開放資料的使用者而言，形成一個技術性門檻。 進一步來說，這些門檻是資料工程(Data Engineering)[1]領域嘗試解決的問題之一，為了使資料能進一步被統合使用，包含截取(extract)、轉換(transform)、讀取(load)等傳統資料倉儲整理資料的手段，在資料工程領域中持續地被精進與探討著。

由於針對處理不同檔案格式的需求，衍生出的相關工具五花八門，加上使用不同程式語言處理資料時，還會面臨如何選擇合適的套件(package)或函式庫(library)等問題。有鑑於此，本文淺析利用開放資料時，在處理資料過程中會遇到的狀況，透過一些背景概念的介紹，以及如何藉由資料工程的程序與工具來處理應對，並以一筆開放資料集為範例，示範將檔案格式從「JSON」轉換成「CSV」，以便後續使用統計軟體進行分析的過程，提供讀者作為使用開放資料的參考。

一、開放資料的定義

在介紹資料工程概念之前，首先對開放資料的定義作簡單介紹。根據開放知識基金會（Open Knowledge Foundation，簡稱 OKFN）開放資料手冊[2]中敘述，開放資料 (Open Data)指的是「資料能被任何人自由地使用，重新使用與散佈－我們至多只能要求來源標示，與以相同方式分享 」。

「開放資料的精神」主要是破除取得對象與再使用目的之限制，讓任何使用者都能不受限制的再次加以利用。此議題原本跟資料儲存的媒介(檔案格式)無關，但由於期望資料儲存的方式，能盡可能地降低資料減損的可能性，並擴大資料的獲取性，於是開放資料本身會被要求必須以「機器可讀取」的格式來儲存；此外，為了能夠普及與長久保存，通常會優先考量以文字形式的電子檔案格式來儲存。有興趣了解此背景的讀者，可進一步閱讀「Digital Ice Age 」文章[3] (2009)。

---

1 資料工程 (Data Engineering)參考閱讀
https://gist.github.com/dataengeer/aaaefc8465dcbbf6885b38c82931a9fe。

2 什麼是開放資料？http://opendatahandbook.org/guide/zh_TW/what-is-open-data/。

3 The Digital Ice Age：http://www.popularmechanics.com/technology/gadgets/a1028/4201645/。

在 OKFN 手冊所引用的開放定義[4]中，以開放格式散布被認為是作品具備「開放性」的一個必備條件。於是，兼具「機器可讀取」的格式，又能夠被長久利用、保存等條件下，原本在網際網路中常見用來傳輸分享的電子檔案格式，如 XML、JSON、CSV 等，很自然地就變成「開放資料」傳遞格式的首選。

一、 開放資料應用層面的挑戰

前言已提及，根據資料提供者的選擇，開放資料除了提供不同檔案格式給使用者運用之外，由於儲存資料的內文編碼(encoding)的不同，也會造成使用者讀取開放資料上的困擾。加上文件內儲存的資訊結構，常是以各資料領域專家(Domain Expert)常用的慣例格式或該領域特殊的規範語法。例如地理資訊系統(GIS)領域中，使用 GeoJSON 檔案格式來傳遞地理資訊，使用者解讀時必須先對該領域或語法規範有所理解，方能正確解析出所傳遞的資訊。

綜上所述，使用者可能面臨資料處理的狀況如下：

- 提供者以多樣化的檔案格式傳遞開放資料
- 資料內容所使用的編碼方式(encoding)儲存可能不同
- 內容的資料結構需要判讀，結構化程度也不同
- 不同檔案格式可選用的工具支援豐富程度不一
- 處理資料的程式語言，有不同的函式與套件可供處理特定檔案格式

二、 資料工程是什麼？又如何協助回應上述挑戰？

如果要用一句話來解釋資料工程領域，筆者會推薦參考大數據分析顧問公司 Insight Data 所提出的概念[5]：「資料工程師讓資料科學家更有效率的完成工作(Data engineers enable data scientists to do their jobs more effectively!) 」；也就是說，如果要讓資料科學家更專注在尋求問題解答的過程，則會需要有資料工程背景的人員協助處理「資料蒐集、資料倉儲、資料清理 、資料格式轉換、資料查詢」等工作[6]，以減少資料科學家的負擔，於是這些工作項目組合在一起，便形成了資料工程領域的內涵。

4 OKFN 開放定義 ： Open Definition - Defining Open in Open Data, Open Content and Open Knowledge。

5 資料工程 (Data Engineering)參考閱讀 #4 Data Science vs Data Engineering https://gist.github.com/dataengeer/aaaefc8465dcbbf6885b38c82931a9fe。

6 資料工程 (Data Engineering)參考閱讀 #5 那些大數據書不會教的資料工程 https://gist.github.com/dataengeer/aaaefc8465dcbbf6885b38c82931a9fe

有興趣深入了解的讀者，可以參考 Quora 問答[7] 來進一步理解資料工程領域的相關工作。下表擷取一部分資料領域中的工作，來協助回應上述挑戰。其中，處理使用開放資料「檔案格式轉換」與「資料結構變化」的問題，屬於本文重點，以下將一一介紹。

問題	資料工程領域相關工作	
	擷取	轉換
檔案格式轉換	O	
內文編碼	O	
資料結構變化		O
處理程式語言	O	O
轉換工具	O	O

三、 資料結構化種類與轉換須留意的細節

在進行轉換前，讀者必須先了解「資料結構化」的基礎概念，並留意轉換上的細節。按照資料內容的結構化程度不同，大致可分為下列三種類別：

1. 結構化資料
2. 半結構化資料
3. 非結構化資料

「結構化資料」或者稱為表格化資料(tabular data) [8]，是一種「每列(rows)有相同數量的欄(columns)，且每列中的個別欄所紀錄的資料值(value)都為相同欄位屬性值」的資料種類。結構化資料常見於網頁超文本(HTML)的表格資料，或是關聯式資料庫(relational-databased)的表格資料中；若轉化成純文字檔案，則以 csv(comma separated value)格式最為常見。csv 是以逗號區隔個別欄位值，也稱作分隔符號文字 (delimited text)的一種，其儲存內容規範請參照 W3C 標準 4180 附錄 A[9]。由於 csv 格式內容簡單，容易產生與編輯，是常用的資料交換格式之一。

「半結構化資料」相對於「結構化資料」，是將「欄位定義」與「資料值」混

7 資料工程 (Data Engineering)參考閱讀 #6 What is data engineering? – Quora https://gist.github.com/dataengeer/aaaefc8465dcbbf6885b38c82931a9fe

8 結構化資料：https://www.w3.org/TR/2015/PR-tabular-data-model-20151117/#dfn-tabular-data

9 https://www.w3.org/TR/2015/PR-tabular-data-model-20151117/#standards。

合儲存的一種檔案格式；不同於結構化資料事先嚴格規範特定欄位的值域，半結構化資料能較彈性的紀錄欄位與值，所以也常用於資料交換，特別是在組織之間交換資訊時使用。例如常見的 XML 與 JSON 檔案格式，均屬於半結構化資料的一種。

半結構化資料的另外一個特色是常以「巢狀結構」方式來紀錄資料；由父子層級欄位標籤(tag)與值，複合成巢狀結構片段來組合想傳達的資料。由於巢狀結構彈性極大，對需要驗證 JSON/XML 內容者，可額外指定 schema[10]或 DTD[11]綱要，作為資料收取方自行驗證之用。

「非結構化資料」為可變長度、內文格式沒有明確定義的文本，常見檔案格式為 txt 。表一整理三種結構化資料，並附上範例與檔案格式，提供讀者對照比較。

表一、結構化資料對照表

資料分類	表格tabular	巢狀結構資料	文字資料、圖、二進位檔
屬性	結構化資料	半結構化	非結構化
範例	age, name, place 25, marx, Taipei	<people> <age>25</age> <name>marx</name> <place>Taipei</place> </people>	My name is marx, 25 years old, lived in Taipei City
檔案格式	csv,xls	JSON/XML/HTML	txt、PNG、PDF

值得注意的是，轉換不同結構化屬性檔案時，由於目標格式結構化屬性的不同，而有資料表現能力的落差。例如由「半結構化資料」轉換為「結構化資料」時，由於來源格式可能有多層巢狀(nested)資訊，當目標格式要求「同一欄位」的值必須描述一致的內容時，就會導致資訊轉換對應上的落差。此時，必須以目標檔案格式所能表達資料的能力，做出適當的選擇，例如將一個 JSON 檔存為多個 csv 檔案，或是改由 Excel 分頁的方式儲存，才能保留完整的資訊。

此外，若僅在語法上符合檔案格式要求，而未滿足「結構化資料定義值」的需求(一個欄位內容屬性要一致)，轉換後的結果也將難以再利用，實務上應避免這種情形。例如，將以下內容(半結構化的資料值)視為文字，直接套入 csv 欄位之中：

```
"<age>25</age>"
```

雖然符合此結構化欄位的定義(文字欄位)，內容卻出現半結構化標籤文字

10 W3C XML Schema：https://www.w3.org/XML/Schema；JSON Schema：http://json-schema.org/。

11 DTD (document type definition)：https://www.w3.org/TR/REC-xml/#dt-doctype。

(<age>25</age>)，若要利用此欄位的內容(文字 25)，必須經過消除<age></age>標籤的步驟；也就是說，欄位內容均符合定義，但卻無法直接使用。對此狀況，可透過改變欄位值定義(改為數值)，或請提供者在提供檔案前先刪除 tag，方便後續利用。理解內容結構化的屬性差異與須留意的細節，可避免轉換後資料使用上的困擾。

四、 JSON 格式轉換 CSV 格式範例

為使讀者能夠體會轉換過程，本文以政府資料開放平臺中「各縣市統計區 15 歲以上人口五歲年齡組教育程度統計」資料集[12]作為示範。

【資料介紹與說明】

- 此資料集內按照 22 個縣市行政區劃分 ，各行政區分別提供 JSON 與 XML 檔案兩種版本供下載使用，本文將挑選其中一行政區資料，並採用 JSON 格式來進行示範，此僅為範例操作，由於行政區界線可能變化，讀者於資料處理利用上仍須留意。
- 本例挑選「台中市統計區資料」
- 本資料收集 15 歲以上人口的五歲年齡組之「教育程度」的人口數交叉統計
- 本資料依「二級發布區」[13]加以統計發布

【資料轉換三步驟】

由於原始 JSON 資料檔內文結構屬於半結構化資料，考慮到使用 csv 格式(結構化資料)的通用性，將資料格式轉換為 csv 檔。

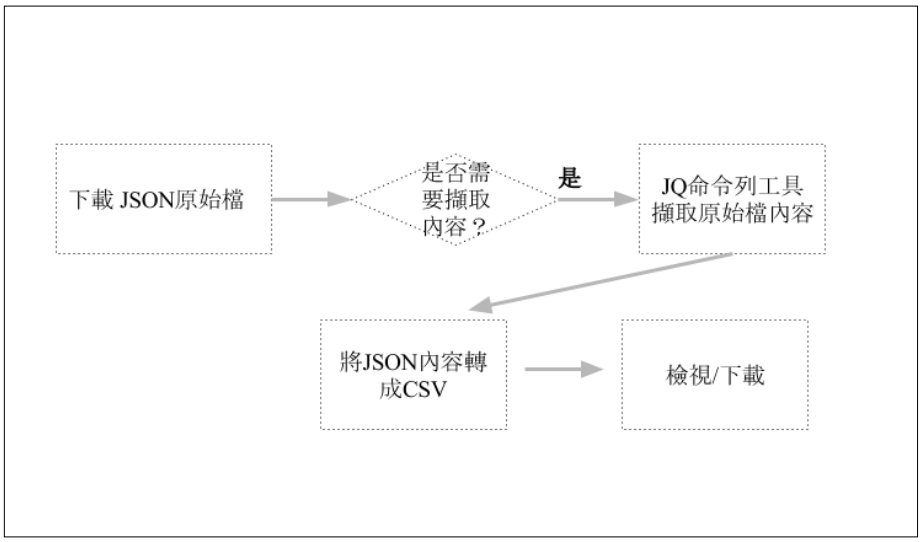
轉換目標：資料格式轉換 (由 JSON → csv)。

轉換步驟如圖一所示，說明如下：

1. 需要從多層半結構化的 JSON 檔中，定位、擷取並輸出所需統計資料部份。
2. 挑選適合的轉換工具，包含命令列或線上工具；這些工具必須支援原始格式的內文編碼。
3. 挑選適當工具來檢視轉換出的 csv 內容 ，方便驗證轉換內容的正確性。

12 資料集網址：http://data.gov.tw/node/18629

13「二級發布區 」的意義可參考「社會經濟資料庫」分組入口網(http://210.65.89.57/stat/Web/Portal/STAT_PortalHome.aspx)常見問題，或至內政部統計處網站，參閱專題分析(http://sowf.moi.gov.tw/stat/topic/list.htm)中有關統計區分類系統文章。



圖一．資料格式轉換步驟

建議讀者在開始轉換之前，可先透過文字編輯器開啓下載的檔案，或利用一般程式語言(R、Python、PHP 等)以相關函式進行讀取動作，檢視原始資料格式，以了解資料原貌。

【資料轉換過程中使用工具】

轉換過程中將用到以下五種工具，取得方式詳見轉換工具[14](①②⑤)。

1. jq (用途 json2json 過濾、挑選轉換)
2. json2csv 命令列工具 (轉換格式)
3. 線上 json2csv 工具(轉換格式工具)
4. csvkit 套裝內的 csvlook 命令列工具 (檢視工具)
5. 文字編輯器 sublime text(檢視工具)

本例透過 linux 作業系統環境執行，相同工作亦可於 windows 環境上完成。表二整理各個轉換步驟所使用的工具及用途。

14 https://gist.github.com/dataengeer/4e6f8ff0abbb7ee787b49645311e9b78

表二．資料轉換步驟與使用工具說明

轉換步驟		使用工具	工具說明	工具用途
1	從多層半結構化的JSON檔中，定位、擷取並輸出所需統計資料部份。	sublime text	GUI檢視工具	檢視json
		jq	命令列工具	擷取json
2	挑選適合的轉換工具，包含命令列或線上工具；這些工具必須支援原始下載格式的內文編碼	json2csv	命令列工具	轉換工具
		http://www.convertcsv.com/json-to-csv.htm https://konklone.io/json/	線上json2csv網站	轉換工具
3	挑選適當工具來檢視轉換出的csv內容 ，方便驗證轉換內容的正確性	csvlook	命令列工具	檢視csv

【步驟一：擷取所需 JSON 內容】

1.1 取得原始資料

進入「政府資料開放平臺」資料集頁面(圖二)，下載「台中市統計區」(方框標示方框處)的 JSON 資料檔，下載並儲存成「taichung.json」檔。

圖二．資料集下載頁面

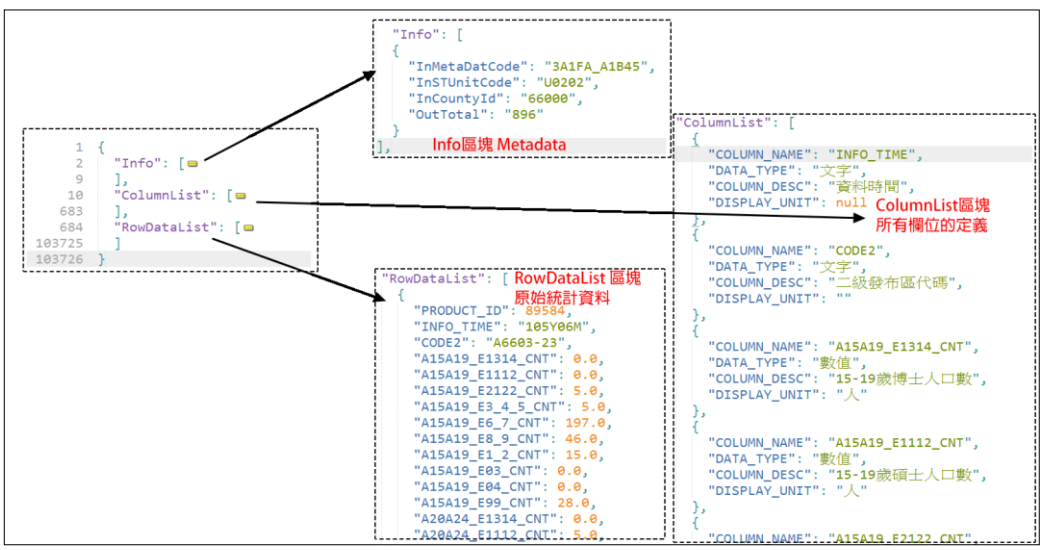
1.2 檢視下載 JSON 檔

打開下載後的原始檔，可發現總行數共有 103,611 行 ，內容可分為三個區塊，分別為紀錄總筆數、資料序號的「Info」區，紀錄資料型態、欄位英文名稱的「ColumnList」區，以及紀錄統計數值的「RowDataList」區（參考表三及圖三）。圖三節錄各區塊部份內容，點擊可展開區塊內容。以 Info 區塊為例，從「OutTotal」欄位可知總筆數為 896 筆（圖四）。

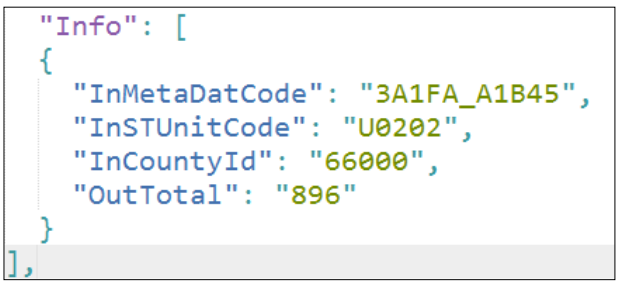
表三．JSON 區塊說明

JSON 區塊	區塊意義	內容說明
Info	Metadata	開放平台序號、縣市代碼、總筆數等
CplumnList	欄位資料	欄位名稱、資料型態、欄位說明等
RowDataList	統計資料	對應統計數值

此外，從圖三的 ColumnList 區塊可知「Code2」欄位的文字是代表某一個二級發布區、「A15A19_E1314_CNT」欄位的數值是紀錄 15-19 歲博士人口數……依此類推，因此對應到圖五的 RowDataList 區塊後，便可從原始檔案內容看出，編號 AA6603-23 這個二級統計區，其區域內 15-19 歲（A15A19_E1314_CNT）、20-24 歲（A20A24_E1314_CNT）、25-29 歲（A25A29_E1314_CNT）、30-34 歲（A30A34_E1314_CNT）年齡層的博士人口數分別為 0、0、2、4 人。



圖三．內文區塊結構



圖四．Info 區塊

圖五「原始統計資料(RowDataList)」的部份，為年齡分組與教育程度分類統計值(二級發布區區內統計)，也是本文後續擷取之目標。


```
  {
    "COLUMN_NAME": "A65UP_E04_CNT",
    "DATA_TYPE": "數值",
    "COLUMN_DESC": "65歲以上不識字人口數",
    "DISPLAY_UNIT": "人"
  },
  {
    "COLUMN_NAME": "A65UP_E99_CNT",
    "DATA_TYPE": "數值",
    "COLUMN_DESC": "65歲以上未詳人口數",
    "DISPLAY_UNIT": "人"
  }
],
"RowDataList": [
  {
    "PRODUCT_ID": 89584,
    "INFO_TIME": "105Y06M",
    "CODE2": "A6603-23",
    "A15A19_E1314_CNT": 0.0,
    "A15A19_E1112_CNT": 0.0,
    "A15A19_E2122_CNT": 5.0,
    "A15A19_E3_4_5_CNT": 5.0,
    "A15A19_E6_7_CNT": 197.0,
    "A15A19_E8_9_CNT": 46.0,
    "A15A19_E1_2_CNT": 15.0,
    "A15A19_E03_CNT": 0.0,
    "A15A19_E04_CNT": 0.0,
    "A15A19_E99_CNT": 28.0,
    "A20A24_E1314_CNT": 0.0,
    "A20A24_E1112_CNT": 5.0,
    "A20A24_E2122_CNT": 227.0,
    "A20A24_E3_4_5_CNT": 12.0,
    "A20A24_E6_7_CNT": 51.0,
    "A20A24_E8_9_CNT": 2.0,
    "A20A24_E1_2_CNT": 0.0,
    "A20A24_E03_CNT": 0.0,
    "A20A24_E04_CNT": 0.0,
    "A20A24_E99_CNT": 0.0,
    "A25A29_E1314_CNT": 2.0,
    "A25A29_E1112_CNT": 63.0,
    "A25A29_E2122_CNT": 187.0,
    "A25A29_E3_4_5_CNT": 10.0,
    "A25A29_E6_7_CNT": 31.0,
    "A25A29_E8_9_CNT": 10.0,
    "A25A29_E1_2_CNT": 0.0,
    "A25A29_E03_CNT": 0.0,
    "A25A29_E04_CNT": 0.0,
    "A25A29_E99_CNT": 0.0,
    "A30A34_E1314_CNT": 4.0,
    "A30A34_E1112_CNT": 48.0,
    "A30A34_E2122_CNT": 170.0,
```


圖五．年齡分組對照教育程度統計原始資料(部分)截圖

1.3 利用命令列工具 jq 來擷取出所需要區塊的 JSON 文字

先簡介 jq 命令列工具：

> ***【jq 命令列工具說明】***
>
> *jq 是一種處理 JSON 文字檔的命令列轉換工具。jq 能讀取 JSON 文字檔內容或文字串流(stream)作為輸入，並且能指定過濾(filter)語法，從輸入(input)中過濾出結果，並產生 JSON 文字檔輸出(output)。*

接著，下指令處理網路上下載的原始檔 (檔名：taichung.json)，並指定擷取 RowDataList 區塊(用 jq 的過濾器 filter 語法來指定)，然後將之輸出為 JSON 文字檔，檔名為 tai-rows.json。

完整的轉換指令與分解說明如下 ：

```
jq -c '.RowDataList | .[] ' taichung.json > tai-rows.json
①          ②                  ③              ④
```

① jq 命令與 -c 參數(產生壓縮格式的輸出)，不加-c 參數則不壓縮。

② jq filter 語法(完整說明請參照[15]) 取出 RowDataList[]的內容。

③ 輸入檔 taichung.json。

④ 輸出檔 tai-rows.json。

> ***【關於 jq 過濾器更多說明】***
>
> *jq 過濾器(filter)是透過語法指定需要過濾的內容，單次執行 jq 命令可指定一個以上的過濾器。過濾器如同篩子，篩選輸入，並將符合指定過濾語法的內容輸出；多個過濾器之間用「|」符號區隔，並且前一個 filter 語法的輸出可以為下一個 filter 語法的輸入。更多過濾器的用法請參考 jq 官網文件。*

【步驟二：執行 json2csv 轉換】

本步驟將 tai-rows.json 轉換成 csv。csv 格式為統計軟體常用輸入格式，透過這樣的轉換，便利讀者使用統計軟體分析資料。

[15] https://stedolan.github.io/jq/manual/。

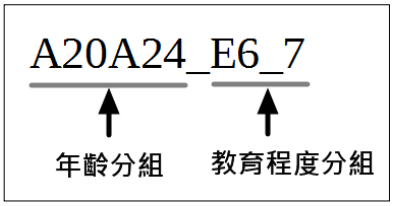
2.1 透過命令列工具 json2csv 來轉換

首先，示範以命令列工具 json2csv，將步驟 1.3 產生的 JSON 文字檔內容，輸出成 csv 文字格式的方式。由於 json2csv 指令必須指定輸出之 JSON 檔案的欄位名稱，表四整理了 JSON 檔案中，用以組成欄位名稱的教育程度和年齡的分組方式及其代碼。

表四．教育程度與年齡分組代碼對照表

組別	代碼	說明	組別	代碼	說明
1	A15A9	15-19 歲	1	E1314	博士
2	A20A24	20-24 歲	2	E1112	碩士
3	A25A29	25-29 歲	3	E2122	大學院校
4	A30A34	30-34 歲	4	E3_4_5	專科
5	A35A39	35-39 歲	5	E6_7	高中職
6	A40A44	40-44 歲	6	E8_9	國中初職
7	A45A49	45-49 歲	7	E1_2	小學
8	A50A54	50-54 歲	8	E03	自修
9	A55A59	55-59 歲	9	E4	不識字
10	A60A64	60-64 歲	10	E99	未詳
11	A65UP	65 歲以上			

完整的欄位名稱規則為「年齡分組_教育程度分組_CNT」，如果想了解「20-24 歲年齡層具備高中職教育程度」分區統計值，應該挑選「A20A24_E6_7」欄位，如圖六所示。同理，要找出 25-29 歲大學與專科層級的分區統計值，則可分別選擇「A25A29_E2122_CNT 」、「A25A29_E3_4_5_CNT」欄位。



圖六．年齡分組對照教育程度分組

如須一次轉出多個 csv 欄位值，在指令中必須使用小寫逗號「,」串接輸出的欄位，並用雙引號(")把全部的欄位包裹在一起。本例中，我們將指定「A20A24_E6_7_CNT、A25A29_E2122_CNT、A25A29_E3_4_5_CNT」三欄位，完整指令與細部說明如下：

```
< tai-rows.json json2csv -p -k "A20A24_E6_7_CNT,A25A29_E2122_CNT,A25A29_E3_4_5_CNT" > result.csv
        ①          ②                        ③                                          ④
```

① tai-rows.json 為輸入檔

② json2csv 指令 (-p 參數不列出 header 列)

③ -k 參數列出指定輸出的欄位。輸出欄位用逗號串接，前後加括號(逗號與欄位名稱中間不可加入空白)

④ 輸出結果儲存為 result.csv 檔

【輸出欄位補充說明】

如果後續要用地理區域呈現各統計值，您需要指定輸出二級發布區代碼 CODE2。另外，根據資料集網頁上的說明，本資料集更新頻率為不定時，請讀者留意統計時間的欄位 INFO_TIME 以避免統計時間上可能的落差。

以上是透過下指令的方式將 JSON 檔轉換成 csv 文字。但如果不想安裝工具，亦可以採用以網站方式進行線上轉換。

2.2 透過線上網站轉換成 csv

本文介紹兩個提供「線上轉換 JSON→csv 檔」的網站並進行比較，分別是：

- http://www.convertcsv.com/json-to-csv.htm
- https://konklone.io/json/

由於本範例筆數(896 筆)不多，線上轉換的速度不算太慢，如果原始來源檔案大小超過 1MB 以上，或讀者操作時發現轉換速度過慢，仍建議讀者使用命令列工具來轉換。

2.2.1 透過網站 convertcsv.com 線上轉換成 csv

轉換工具網址：http://www.convertcsv.com/json-to-csv.htm

【轉換步驟說明】

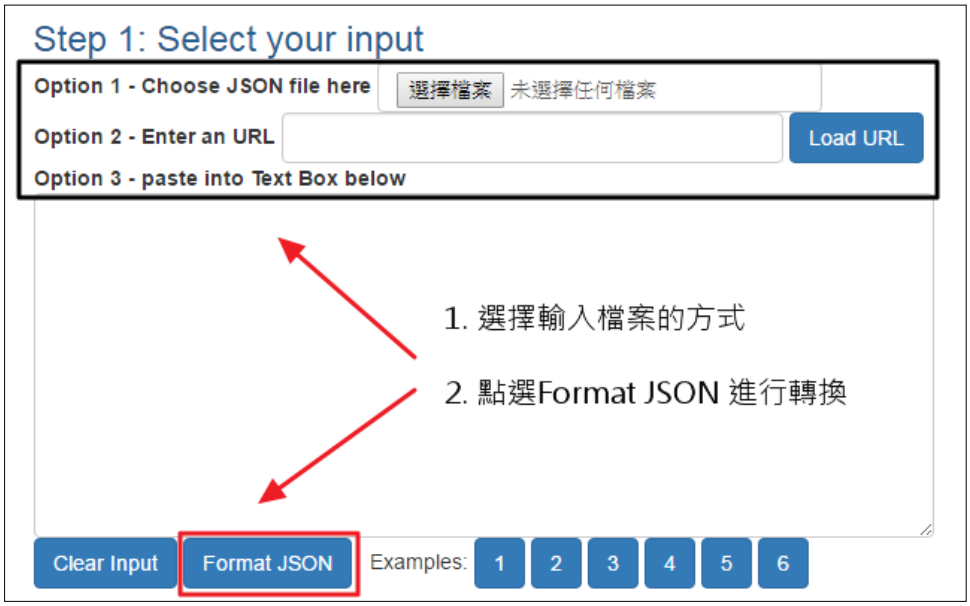
Step 1 ：首先選擇輸入的方式(Select your input)，一共有三種選擇。讀者可選擇其中一種進行轉換：

1. 上傳檔案(Choose JSON file here)
2. 輸入網址(Enter an URL)
3. 貼上 JSON 文字(Paste into Text Box below)

1.與 3.的選項比較直觀，不多作說明。而選項 2.「輸入網址」指的是輸入一串網址(可容許轉換網站線上直接存取 JSON 文件)，至於能不能轉換出結果，端視原始檔是否能被網站可解析而定，通常是 JSON 文字檔內文的結構越簡單，解析成功的可能性越大。此外，若只是想體驗轉換的操作，網站上提供六個範例資料(Examples 1~6)可直接試用。

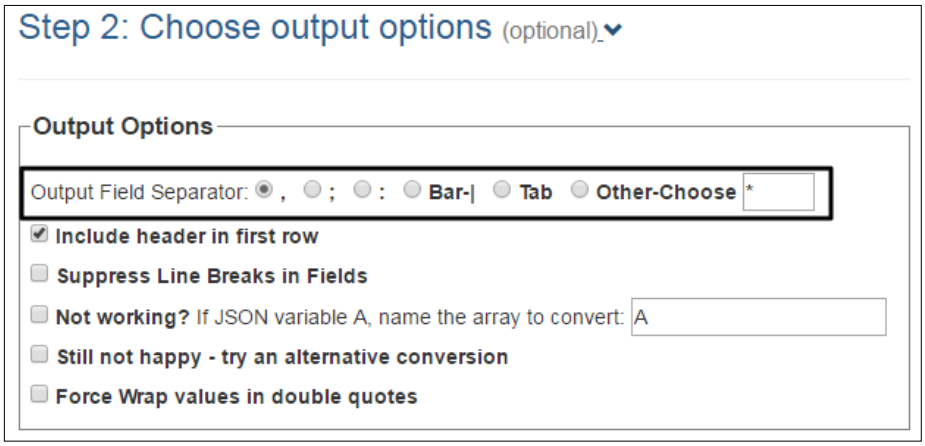
經過實際測試，貼上下載本文範例檔案的網址並按 Load URL ，雖然可以將 JSON 原始檔案的文數字內容載入到下方的文字框中，但無法被該網站解析。因此改用「上傳檔案」的方式，選擇步驟一處理後的 JSON 檔案後上傳。

圖七．網站 convertcsv.com 輸入頁面



Step 2：調整輸出選項

本步驟為可選項目，可忽略不調整，預設為以逗號分隔、並於第一列列出欄位名稱。傳統上，csv 檔案採用逗號為欄位分隔符號；如果需要挑選其他分隔符號，則可於「Output Field Separator」設定分隔符號為分號(;)、冒號(:)、直線(|)、Tab、或者其他(如星號*) ，選項畫面請參考圖八。



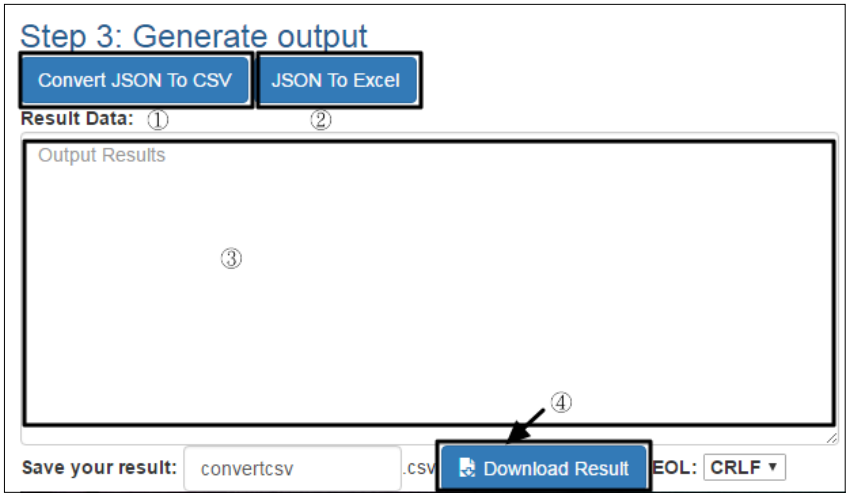
圖八．網站 convertcsv.com 輸出選項頁面

Step 3：將輸出結果另存為 csv 或 Excel 檔案

根據 Step 1 的輸入、Step 2 的輸出參數指定，Step 3 就是輸出結果。

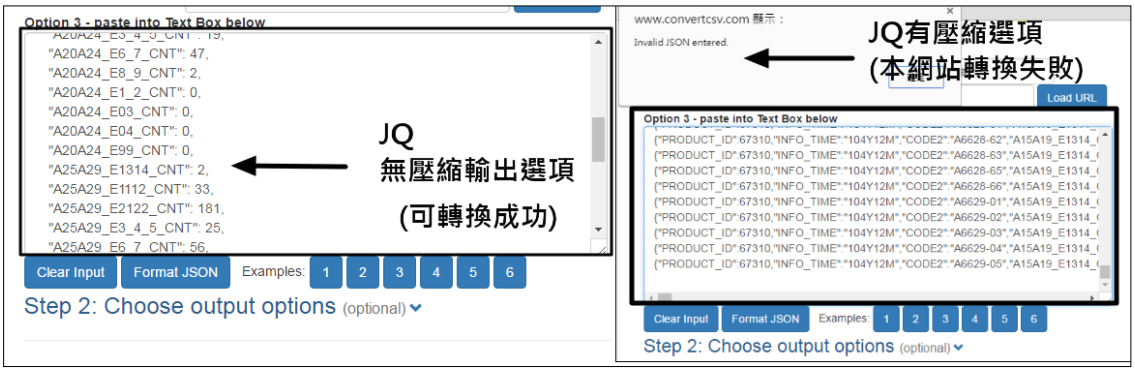
Step 3 的輸出方式有兩種，分別是「Convert JSON to CSV」與「JSON to Excel」，對應到界面(圖九)的① ② 兩個按鈕。如果按下①按鈕，結果會直接出現在下方「Output Results」(下圖③ 區域)。按下② 的結果會是跳出另存新檔的視窗。

如果要將區域③ 的結果輸出，需要按下「Download Result」按鈕(下圖④)另存新檔。可在「Save your result」中指定「另存的檔名」和「換行符號」(預設是 windows 格式 CR＋LF)。



圖九．網站 convertcsv.com 輸出頁面

須提醒讀者注意的是，經過測試，該網站只接受無壓縮格式的 JSON 檔案或文字，如圖十所示，若是上傳經過壓縮的 JSON 檔案，會出現「Invalid JSON entered」的訊息，讀者欲利用該網站的話，務必調整步驟一 1.3 的指令參數(取消 -C 參數)。



圖十．網站 convertcsv.com 壓縮格式測試結果

該網站選項雖然比較多元，但整體的操作動線流程設計還有待改善，無法直覺的操作。雖然乍看之下並不覺得複雜，網站設計者也著實花了不少心力將「輸入、輸出、選項、教學範例」整合成單一界面，但對初次使用網站的人來說，須要一些時間測試與理解界面的正確使用方式。

2.2.2 透過網站 **konklone.io** 線上轉換成 csv

線上工具網址：https://konklone.io/json/

接下來介紹的 JSON to csv 資料轉換工具網站「Konklone.io」，收錄於美國白宮所建置名為「Project Open Data」的專案中[16]。操作上十分簡單直覺，只有「貼上文字」與「輸出下載」兩個步驟，不容易搞錯。另外，本網站可接受輸入壓縮格式與非壓縮格式 JSON 文字。

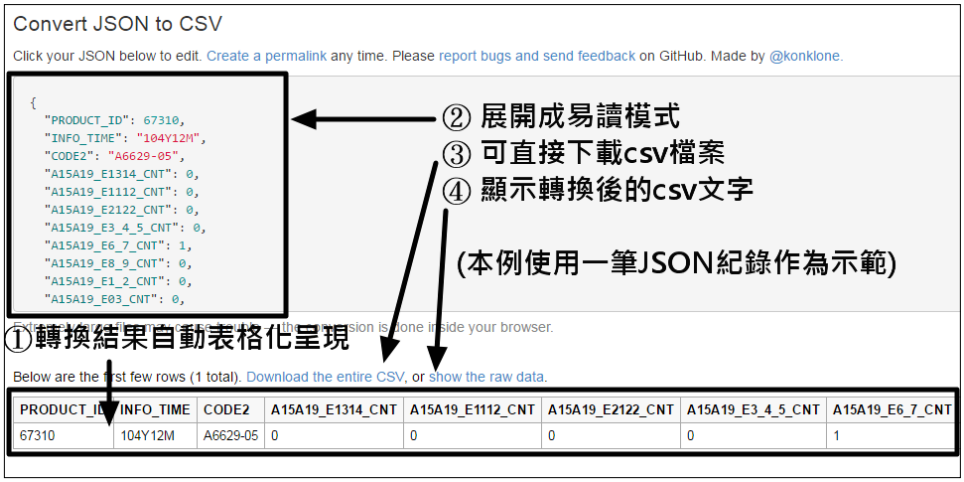
16 Project Open Data，網址 https://project-open-data.cio.gov/，點選 4.Tools 之 4.13 JSON-to-CSV Converter。

【2.2.2.1 輸入】

圖十一．網站 Konklone.io 輸入頁面

於上圖貼上文字後，網頁會自動排版展開成可讀樣式，並將結果輸出在下方，列出產出的筆數，以及提供下載 csv 檔的連結(圖十一為輸入，圖十二為輸出)。相比之下，是不是比前一個線上工具，更容易操作呢？

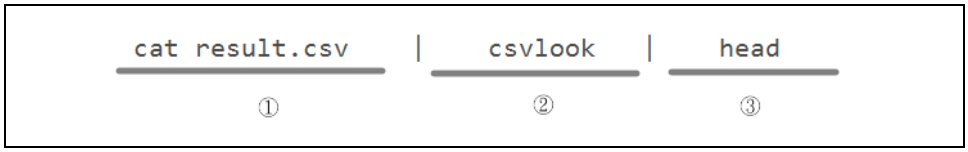
【2.2.2.2 結果與輸出】



圖十二．網站 Konklone.io 輸出頁面

【步驟三：透過命令列工具檢視轉換後的 csv】

在步驟三中，筆者使用 csvkit 套件為工具，示範利用命令列下 csvlook 指令的方式來檢視步驟二中轉換好格式的檔案內容。csvkit 還有其他諸如列出欄位名稱(csvcut)、排序(csvsort)、合併資料(csvjoin)等指令可供讀者應用來檢視或處理 csv 格式的檔案。



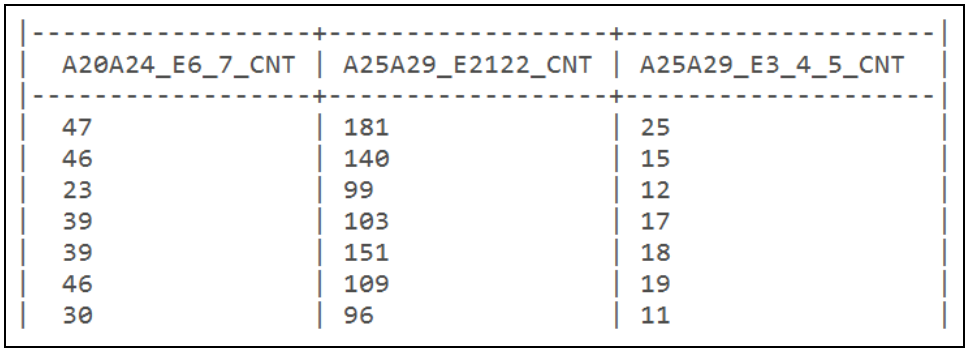
圖十三．用 csvlook 命令檢視步驟二輸出結果

指令說明：

① 讀取步驟二輸出結果 result.csv 內容

② 將內容用 csvlook 命令呈現，csv 檔案內容第一行顯示為欄位名稱

③ 用 head 命令輸出結果前 10 行，如下圖所示



A20A24_E6_7_CNT	A25A29_E2122_CNT	A25A29_E3_4_5_CNT
47	181	25
46	140	15
23	99	12
39	103	17
39	151	18
46	109	19
30	96	11

圖十四．csvlook 命令輸出結果(前十行)

如此，輸出的欄位如果不對或有缺漏，都可透過步驟二、步驟三等提到的工具重新轉換或確認資料的正確性。此三步驟完結之後，轉出的 csv 檔案便可用統計軟體進行後續的分析、處理、運用。

【更多說明】

原始 csv 內容中，分隔符號(逗號)的數量是跟欄位數目有成正比，透過 csvlook 指令直接做編排 ，容易視覺上看出遺漏的欄位。但螢幕大小是有限的，本例資料轉出的 csv 無法透過畫面一目了然。讀者可將其視為資料正確性檢查的一種輔助。

【結語】

資料使用者在「取用到利用」資料過程時，往往面臨不少的考驗。本文介紹資料工程領域的一些概念與工具，透過觀念的分享與實例演練相關工具，來協助讀者初步體會資料整理、轉換處理過程中會遭遇的問題與克服的方法。

電腦科學領域有一句經典名言「Garbage In, Garbage out」(垃圾進，垃圾出)，說明如果輸入電腦的是一堆無用的資訊，即使電腦仍可按使用者指令運算處理，產出的結果也不具任何意義；相同地，包含政府與民間提供的開放資料、巨量資料等資料的處理利用亦是如此。若要避免因資料的謬誤，造成後續分析、運算推導出之結論可能出現「失之毫釐、差之千里」的遺憾，吾人執行資料處理的各個步驟皆應謹慎為之。

展望未來，資料工程領域仍在持續的進化，有兩個趨勢正在影響它的發展：

(1)資料科學團隊分工精細化

(2)聚焦整合資料能力，建構多元化的資料組合供利用

此進階議題的探討，留給有興趣的讀者，自行追蹤最新的發展與變化。

【參考書目】

Janssens, J. (2014). *Data Science at the Command Line*: O'Reilly Media.

Squire, M. (2015). *Clean Data* (pp. 272). Retrieved from http://shop.oreilly.com/product/9781785284014.do