大數據分析概觀與實務研討

影響空氣品質的因素之分析

四工管三乙B10601104 許喬雅

四電機二甲B10707032 楊政憲

四資工二甲B10715029 陳彥瑋

四資工一B10815034 吳承翰

**目錄**

[一、 動機 9](#_Toc28632462)

[二、 資料蒐集 10](#_Toc28632466)

[三、 分析方法 13](#_Toc28632473)

[（一） 數據分析 13](#_Toc28632474)

[（二） 機器學習 13](#_Toc28632475)

[四、 預期結果 13](#_Toc28632476)

[五、 分析過程 14](#_Toc28632480)

[（一） 資料採集 14](#_Toc28632481)

[（二） 資料整理 16](#_Toc28632483)

[（三） 數據分析 18](#_Toc28632484)

[（四） 機器學習 29](#_Toc28632485)

[六、 分析結果 32](#_Toc28632488)

[七、 分析結論 35](#_Toc28632495)

[八、 未來展望 36](#_Toc28632500)

[九、 我們的看法 36](#_Toc28632502)

[十、 參考資料 37](#_Toc28632504)

[十一、 附錄 38](#_Toc28632519)

[（一） 網頁爬蟲程式碼 38](#_Toc28632520)

[（二） 資料整理程式碼 41](#_Toc28632521)

[（三） 機器學習程式碼 44](#_Toc28632589)

1. 動機

近年來環保意識提升，不只有民眾自發性的減塑、節電等活動，政府單位也提出了相應的政策與口號，例如：台中市政府核定台中火力發電廠全廠全年生煤使用量上限為1104萬噸，另要求中火在空污季減少5部機組用量，希望能因此改善台中的空氣品質[[1]](#footnote-1)[1]。

另外，2018年中華民國全國性公民投票(1124公投)中有幾項與環境有關的題目，其中一項提案為：「您是否同意以平均每年至少降低1%之方式逐年降低火力發電廠發電量?」提案的理由大致是因為火力發電造成了PM2.5的上升，進而影響了空氣的品質。其結果為：同意票73.8%、不同意票19.6%、無效票6.6%[[2]](#footnote-2)[2]，因此此提案通過並送交相關單位審查。

從上述兩例當中，可以發現空氣品質好像深受火力發電的影響，但是我們認為影響空氣品質的因素眾多，不可能藉由停止火力發電而能全面改善空氣品質。影響空氣品質的主因真的是火力發電嗎？如果不是的話，那又會是什麼？因此，我們想要找出真正影響空氣品質的元兇。

1. 資料蒐集

我們想探討影響空氣品質的關鍵變數，經過資料調查後，我們整理出幾項可能與空氣品質相關的變因並設定為自變量，包含：空氣、火力發電發電量、交通量、工業區數目與面積。另外，因變量我們設定為空氣品質指數(AQI)，以下將簡介各變量之介紹、選擇原因及資料來源。

1. 天氣
   1. 簡介：蒐集全台22縣市之歷年報表，並統計期每月資料，由於氣象站的監測值眾多，經評估後我們選擇五項可能與空氣品質相關的監測值來做分析，包含：氣溫、降水量、相對溼度、氣壓、風速。
   2. 選擇原因：猜測汙染源濃度可能與空氣對流或境外擴散(季風)有關，期望找到各地之天氣評估數值與空氣品質間的相關性。
   3. 資料來源：交通部中央氣象局 觀測資訊查詢系統 (網址：https://[e-service.cwb.gov.tw/HistoryDataQuery/index.jsp](https://e-service.cwb.gov.tw/HistoryDataQuery/index.jsp))
2. 火力發電發電量
   1. 簡介：台灣電力公司火力發電各機組歷年發電量，統計其每月總量，因為只有10個縣市設有台電的火力發電廠，因此我們統計此10個縣市的資料，其他縣市不做分析。
   2. 選擇原因：空氣品質似乎與火力發電的比例息息相關，期望找到火力發電各機組發電量與其所在位置之空氣品質的相關性。
   3. 資料來源：臺灣電力公司 過去電力供需資訊 (網址：<https://www.taipower.com.tw/tc/page.aspx?mid=210>)
3. 交通量
   1. 簡介：各縣市公路交通量統計表，以縣市9年之每日平均車流量評估(臺北市、金門縣、連江縣資料缺失)。
   2. 選擇原因：猜測空氣品質可能與汽機車廢氣有關係，期望找到各縣市不同的交通量對與空氣品質，有沒有直接的影響性。
   3. 資料來源：交通部公路總局 公路交通量調查統計 (網址：<https://www.thb.gov.tw/sites/ch/modules/download/download_list?node=bcc520be-3e03-4e28-b4cb-7e338ed6d9bd&c=83baff80-2d7f-4a66-9285-d989f48effb4>)
4. 工業區數目與面積
   1. 簡介：以各縣市當前工業區數目、面積及佔各縣市面積比例評估
   2. 選擇原因：猜測空氣品質可能與工業區排放之廢氣有關聯，期望找到各縣市不同的工業區面積與比例對空氣品質的影響。
   3. 資料來源：台灣工業用地供給與服務資訊網 工業區統計 (網址：<https://idbpark.moeaidb.gov.tw/Environ/statistical>)
5. 空氣品質指數
   1. 簡介：空氣品質指數(Air Quality Index，AQI)是政府從2016年12月開始實施的空氣品質監測標準，測量的污染項目有六種，分別是臭氧()、細懸浮微粒()、懸浮微粒()、一氧化碳(CO)、二氧化硫()及二氧化氮()，其能更全面地監測多項污染物。
   2. 計算方式：將不同污染物對人體健康的影響程度分別換算出副指標值，再以當日各副指標之最大值為該測站當日之AQI，以確保不會因為只關注單一指標而忽視了其他污染物造成的影響。
   3. 舉例：如果今天的小於15.4，落在「良好」的綠色區間，但是的小時平均值是0.166，落在「對所有族群不健康」的紅色區間，那AQI會取最高值，顯示為「對所有族群不健康」的紅色區間。下圖為**污染物濃度及污染副指標對照表** (圖一)，可以以此為對照。

**污染物濃度及污染副指標對照表** (圖一)

* 1. 資料來源：行政院環境保護署 環保統計查詢網 (網址：<https://stat.epa.gov.tw/>)

1. 分析方法
2. 數據分析：蒐集各變因的資料，並做整理把無效資料刪除，使用excel得到各自變量與AQI的相關矩陣，並從相關係數判斷各項自變量與AQI的相關性。另外，也可以使用自變量對比AQI的數據繪製直方圖或折線圖，觀察其中的關係與趨勢。
3. 機器學習：使用邏輯回歸（Logistic Regression）預測變因在特定情況下時，會有怎樣的結果，並且使用準確度（Accuracy）與召回率(Recall)評估機器學習的成效。
4. 預期結果
5. 火力發電與AQI無關聯：氣候也是影響空氣品質的因素之一，我們認為有許多汙染物是隨著東北季風而來，把源自於中國的黃土與污染粒子帶來台灣，因此需考慮境外的污染物排放量與氣候擴散因素。
6. 汙染源大多來自境內，而電力帶來的汙染只佔其中的一小部分：我國的汙染源可分為境外與境內來源，其中境外來源約佔34 ~ 40%，而境內污染源約佔60 ~ 66%。在境內污染源中，移動源影響約為30 ~ 37%、工業源約為27 ~ 31%、其他約為32 ~ 43%，而電力業僅佔境內污染源10%! 因此我們認為製電過程中對空氣造成的汙染應該只佔一小部分而已。
7. 空氣汙染有季節性：我國空污應該有季節性，且多集中在10~3月，因為東北季風的季節大致為秋季到春季之間，因此我們推論空氣汙染應在10~3月較為嚴重。
8. 分析過程
9. 資料採集

在取得資料的部分我們使用了兩種方式，分別是直接下載以及網頁爬蟲，以下是細節的部分。

* 1. 直接下載

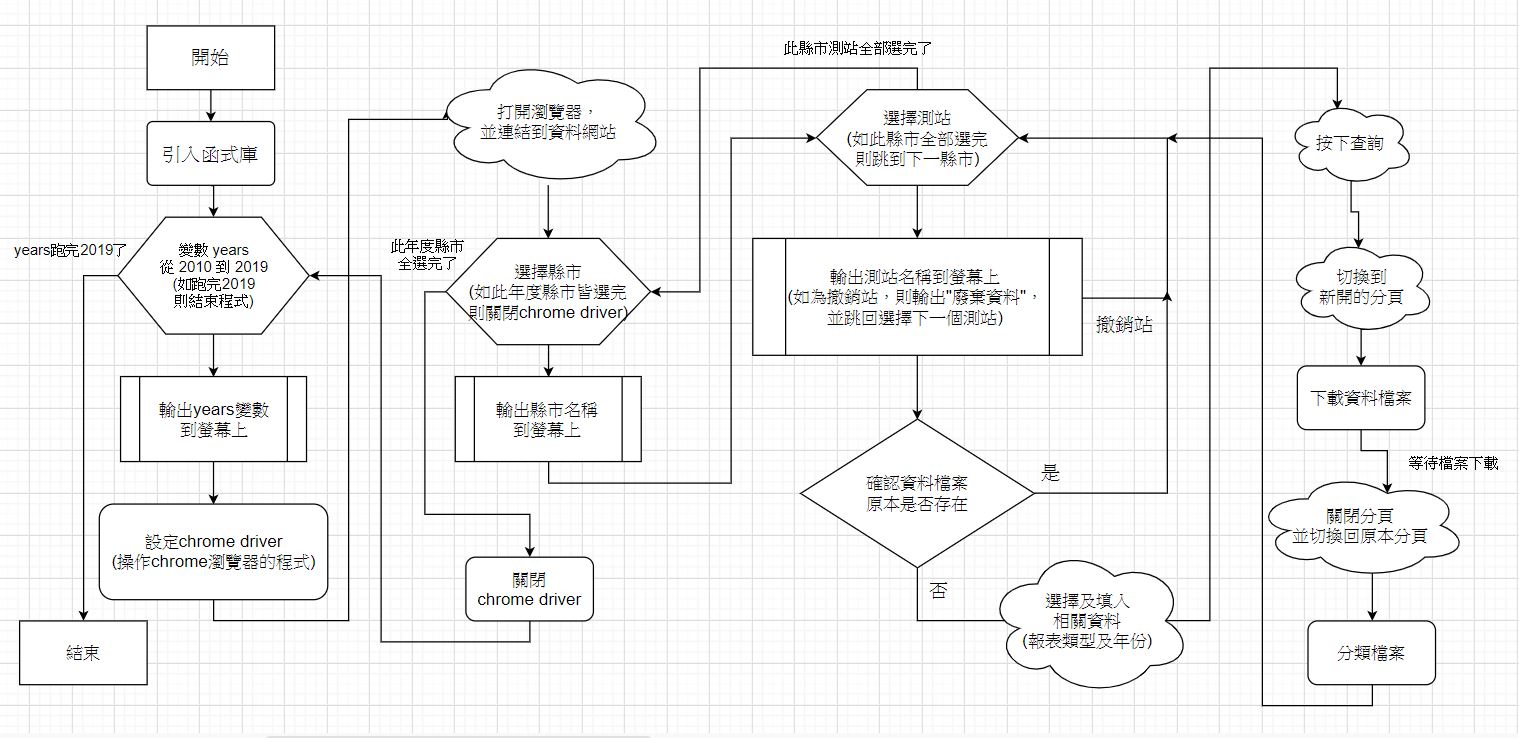
除了天氣資料之外，其餘的資料都只需要進行少數的點擊就足夠滿足我們的需求，因此這部分的資料非常容易取得。

* 1. 網頁爬蟲

對於天氣資料而言，因為我們取得資料的網頁一次只能取得特定年份且特定天氣測站的資料，如果以直接下載的方式會變得非常的繁瑣，因此我們透過Python語言使用網頁爬蟲的技術自動地進行資料的下載，以下是詳細的說明。

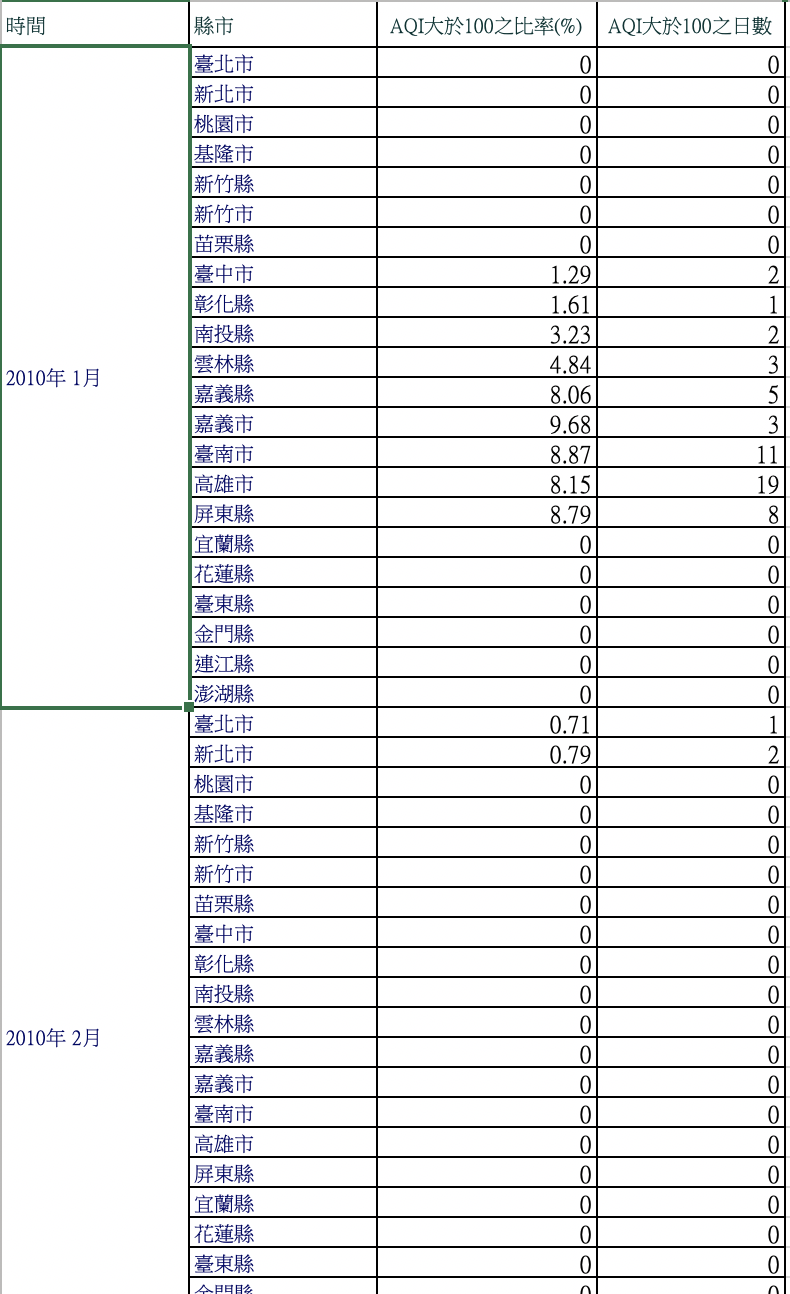
在 (十一) 附錄、（一）網頁爬蟲程式碼中，我們使用了os, time, selenium函式庫。selenium是用來自動化操作網頁的一個函式庫，我們主要是用來做資料檔案的自動下載。而time函式庫是用來取得系統時間，我們是用在等待檔案下載的部分。os函式庫是一個用來控制系統指令的函式庫，我們是做檔案下載後的自動分類部分。

以下是程式的執行流程[[3]](#footnote-3)[1]，詳細的程式碼在附錄(十一)、（一）網頁爬蟲程式碼中。

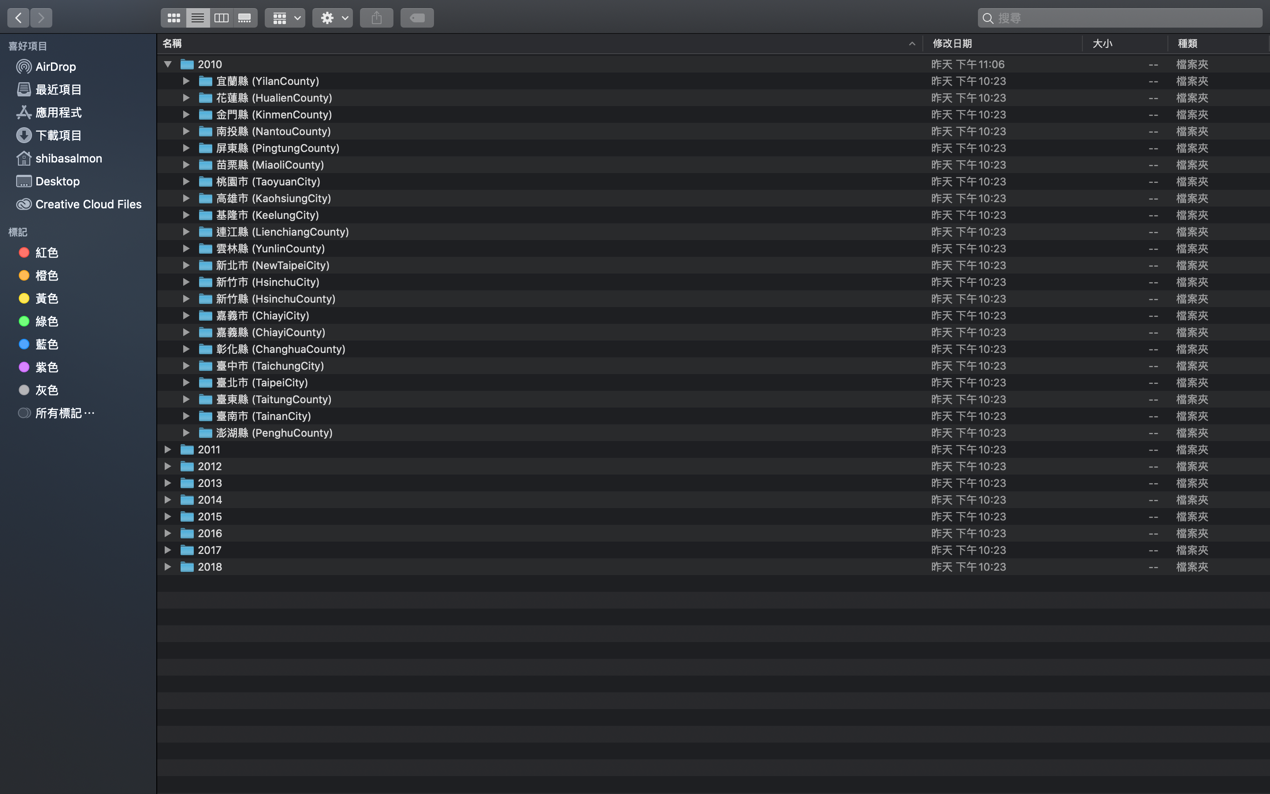


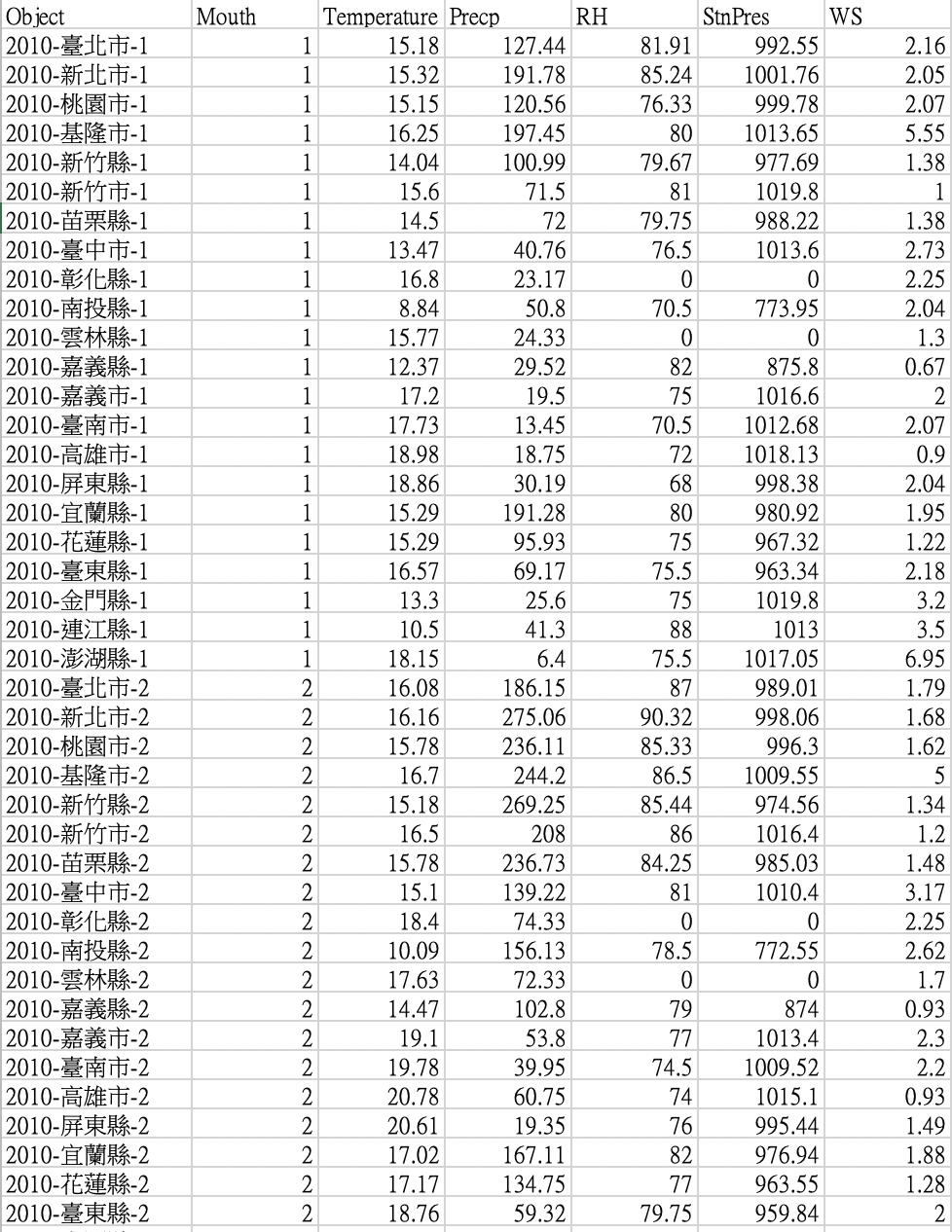
流程圖中的方框是一般的程式處理，六角形框是迴圈的部分，左右有雙線的方框是程式的輸出，雲形狀框是selenium操作瀏覽器的部分，菱形框是條件處理。

1. 資料整理
   1. 資料整理選擇：我們用於機器學習的資料，必須將其做整理。於是我們將「天氣資訊」利用Python整理到跟AQI資訊的格式一模一樣，如下圖



不難看出AQI的資訊是以「年份-月份-縣市」按順序排列，我們必須將現有的天氣資訊合併處理成一個檔案，並按照上述排列，如下圖



* 1. 整理結果：最後可得整理結果如下

1. 數據分析
   1. 天氣

在分析的過程中，我們首先將天氣資料以及AQI資料做成表格，以方便進行資料的排序及篩選。

再來，我們將做分析用的表格（以下簡稱為分析表格）的資料編排格式訂為先將單一年月份的22縣市排完再排下一月份22縣市的資料欄位格式[[4]](#footnote-4)[1]，如下圖所示。

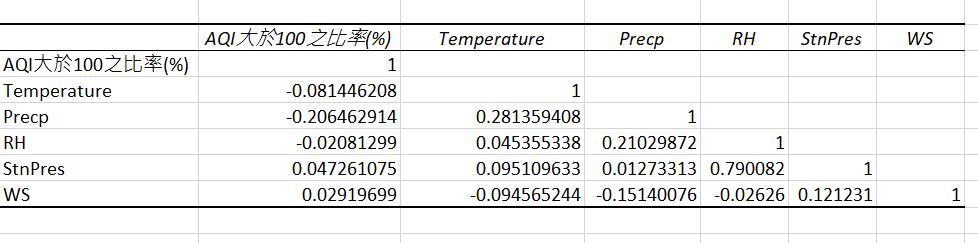


在決定完分析表格格式之後，我們將資料整理進此格式，下圖是部分的資料，圖中資料欄位的AQI大於100之比例(%)是指當月此縣市所有測站測出AQI大於100 (不良)的總日數加總後，再除以當月此縣市所有測站的總運作日數並取百分比表示[[5]](#footnote-5)[2]。

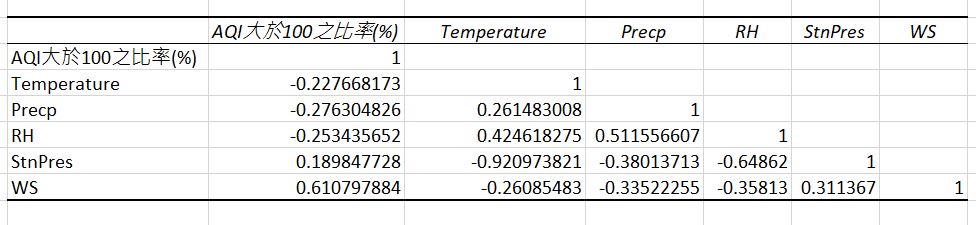
其他欄位如Temperature是溫度、Precp是降水量、RH是濕度、StnPress是測站氣壓、WS是風速。



在填完資料之後，我們使用相關矩陣來分析AQI大於100之比例與其他天氣指標的關聯性，下圖是運算出來的結果。



在分析完整體的資料後，我們也分別對單一縣市做同樣的分析，以下是其中一縣市的結果。



* 1. 交通量

對於交通量來說，因為原始資料形式的緣故，我們將分析表格的資料順序訂為以各縣市來區分，如下圖所示，此圖已將缺失交通量的縣市排除。

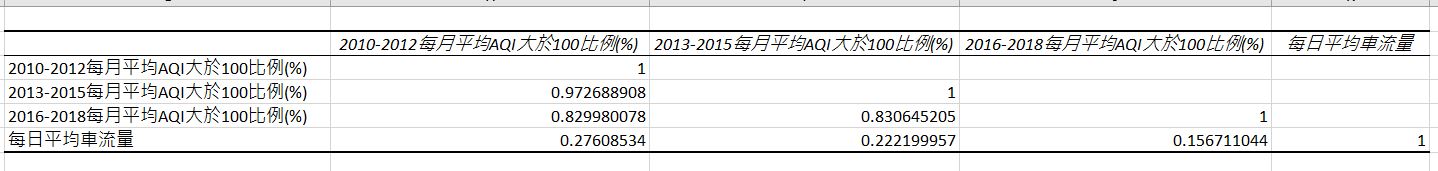


決定了分析表格排序方式之後，我們將資料的AQI欄位又另外分為2010~2012、2013~2015、2016~2018這三個時間區間的平均每月AQI大於100比例(%)，也就是將「每月AQI大於100比例(%)」這一個AQI的原始項目再依據時間的範圍做一個平均[[6]](#footnote-6)[1]，原因是因為我們要避免掉直接將此AQI項目以9年的時間做平均可能造成出來的數值受單一年度的極端值影響的風險。

決定完了資料的格式之後，我們將資料整理之後填入分析表格，下圖是填完後的結果，圖中下方的部分是缺乏平均每日車流量的縣市。



填完資料後，我們一樣使用相關矩陣來分析他們的關聯性，下圖是運算後的結果。



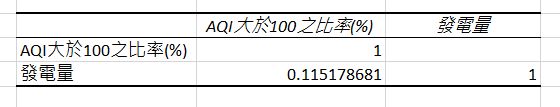
* 1. 火力發電量

我們定義AQI與火力發電量分析表格的順序和AQI與天氣的順序是相同的，在此不再贅述。

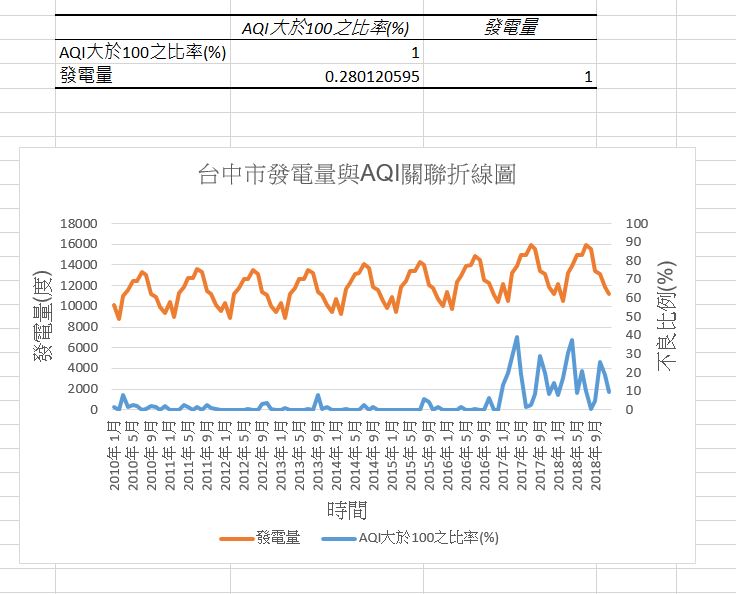
定義完之後，我們將資料填入，下圖是填入後的結果。



填入資料後，我們一樣使用相關矩陣進行分析，以下是相關矩陣的內容。



做完整體的分析之後，我們也使用同樣的方法對單一縣市做分析，並且也將火力發電量及AQI大於100比例(不良比例)依據時間軸製作出折線圖，以下是其中一縣市的相關矩陣及折線圖。



* 1. 工業區數目與面積

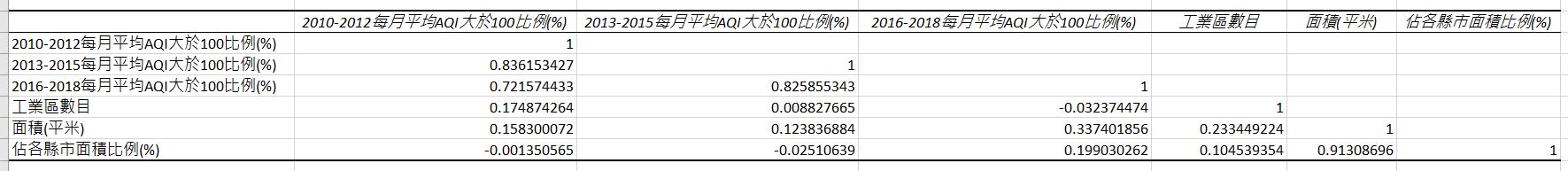
我們定義此項目分析表格的排序方式和AQI與交通量的表格順序是相同的，皆是以縣市做分類。

和交通量一樣，在工業區這一個比較項目中我們也將AQI分成3個時間區段，而我們也將工業區在各縣市的數量、面積、占縣市面積比例列入分析表格的欄位中。

下圖是填入資料後的結果。



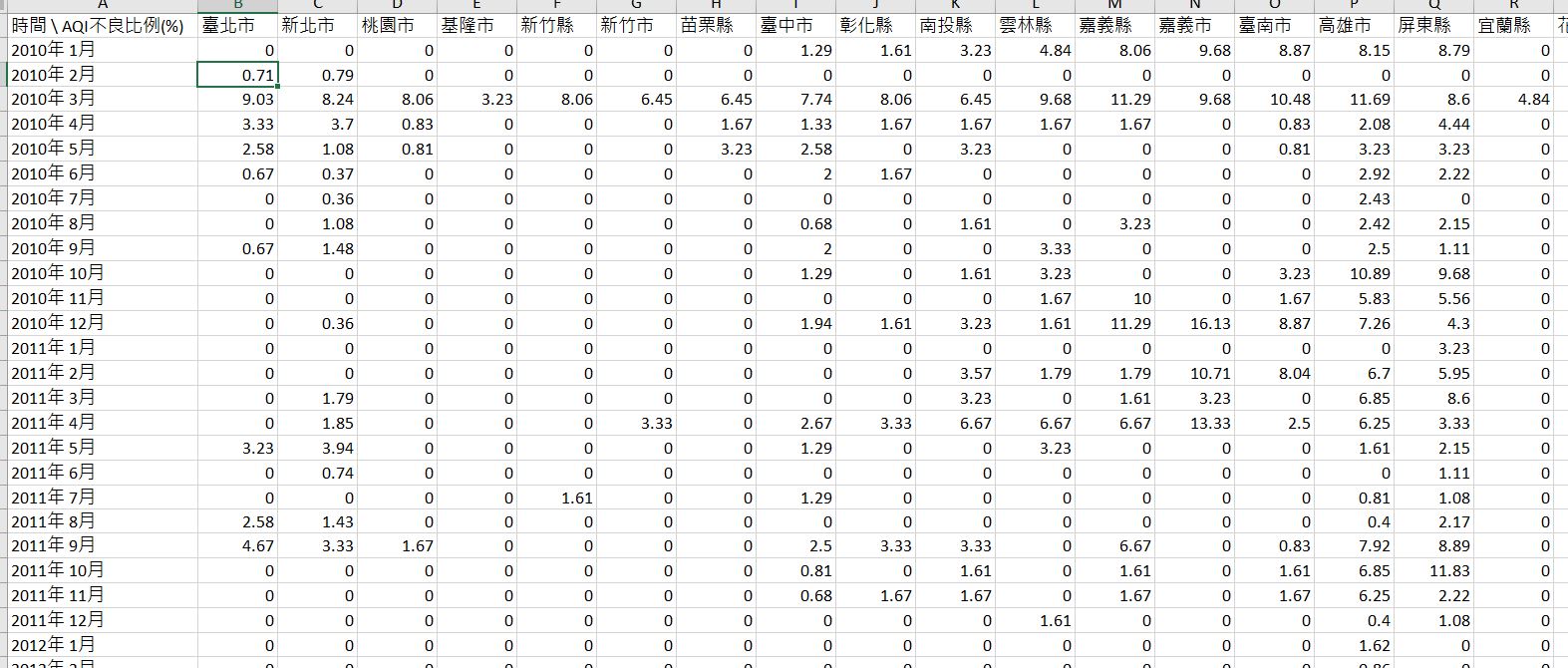
填入資料後，與之前相同，我們也使用相關矩陣進行關聯性的分析，以下是相關矩陣內容。



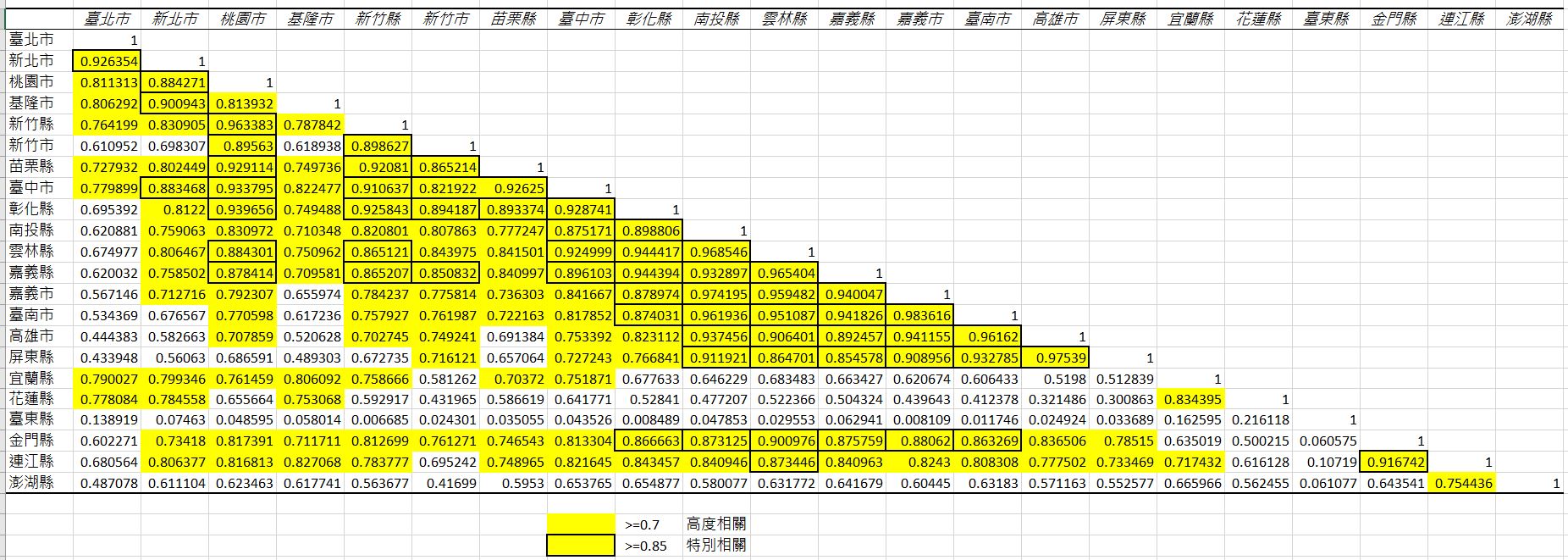
* 1. 地理區域

對於地理區域的資料，我們是將AQI的資料整理成在相同的月份各縣市的AQI大於100比例，每一行是2010~2018的每個月份，每一列是各個縣市的AQI大於100比例。

以下是部分的資料。



整理了資料之後，我們仍然是使用相關矩陣進行關聯性分析，以下是相關矩陣的內容。

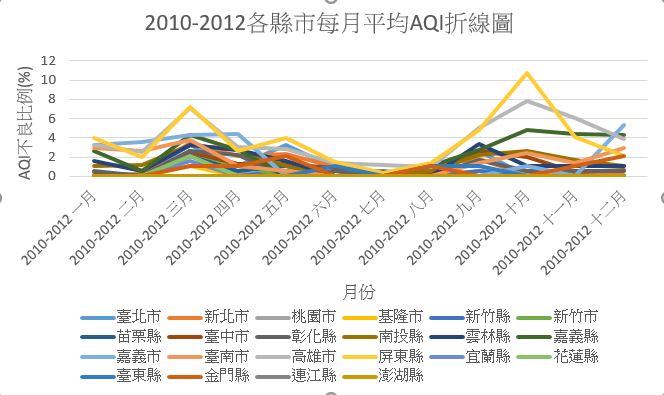


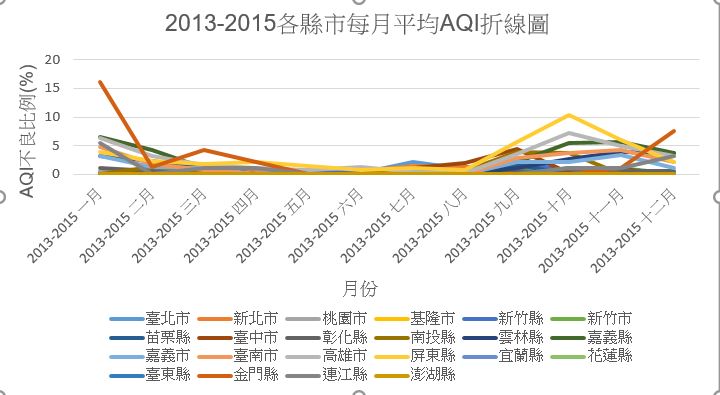
* 1. 月份

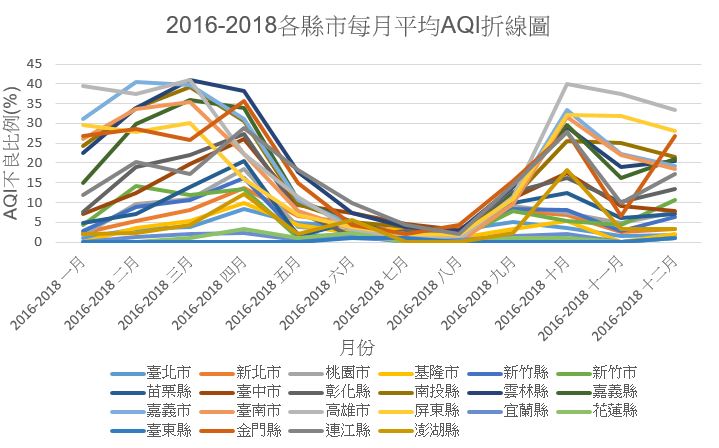
我們整理AQI與月份分析表格的方式也是用AQI與地理區域的方式進行，不一樣的是，我們每一列代表的是2010~2012、2013~2015、2016~2018年這三個時間區段它們在每個月算出它們這區段3年同一個月份的平均AQI大於100比例[[7]](#footnote-7)[1]，如下圖所示。



在月份的部分，我們分析的方式並不是使用相關矩陣，而是直接對三個時間區段畫出每個縣市在1~12月的平均AQI大於100比例折線圖，以下是折線圖的內容。



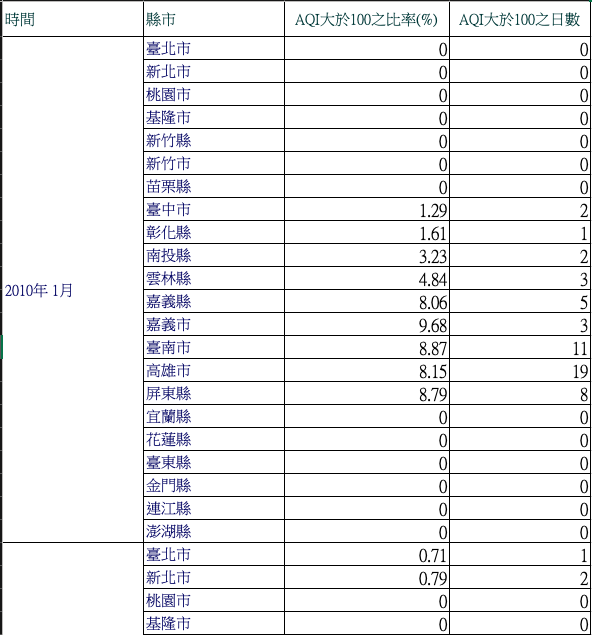




1. 機器學習
   1. 資料選擇：

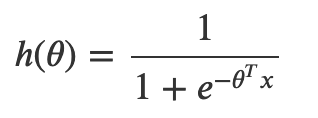
基於資料處理的「方便性」及資料的「完整性」，我們最終只選擇了天氣資料及AQI作為機器學習的學習資料（Training Set）。

* 1. 學習方法：

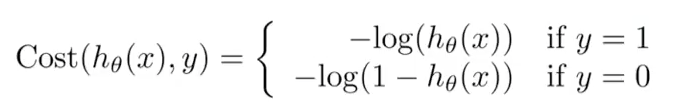
我們試過利用線性迴歸（Linear Regression），作為資料分析的方法，但基於AQI數據呈現「許多資料為0」的狀態，我們發現並不適合用線性迴歸。如下圖所示

故我們決定以AQI大於100的天數比例是否大於0%為分界點，使用了邏輯迴歸（Logistic Regression）來進行「二元分類」。

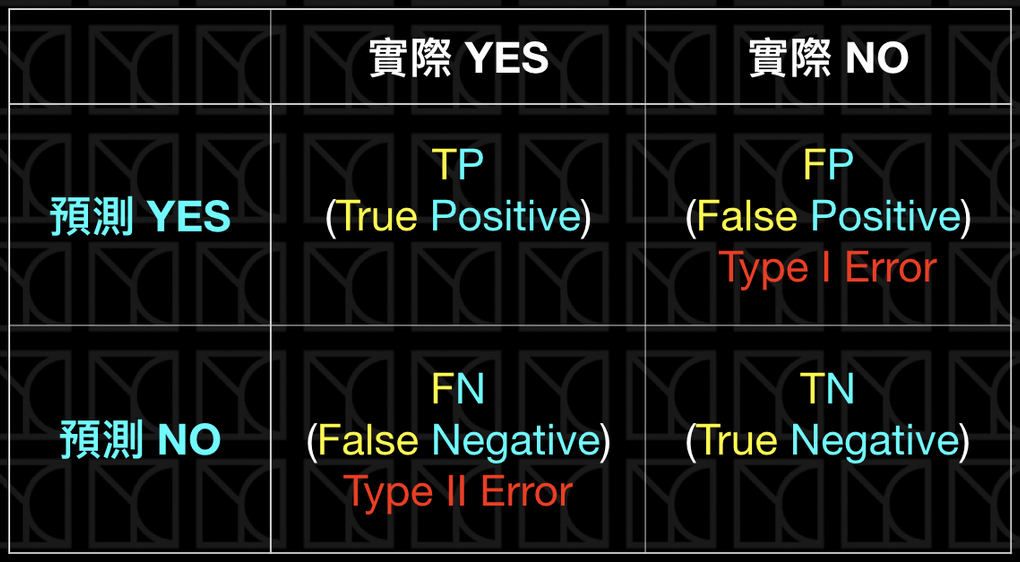
* 1. 邏輯迴歸：
     1. 利用Sigmoid函數「對參數的變化很敏感」的特性可將資料進行「二元分類」。
     2. 使用的假設函數（Hypothesis），用來表示的數學模型，如下圖



* + 1. 使用成本函數（Cost Function），用來表示每個參數的誤差量，判斷學習是否收斂，如下圖



* + 1. 利用梯度下降法（Gradient Descent）進行訓練。
  1. 使用混淆矩陣（Confusion Matrix）的「召回率（Recall）」進行分析：



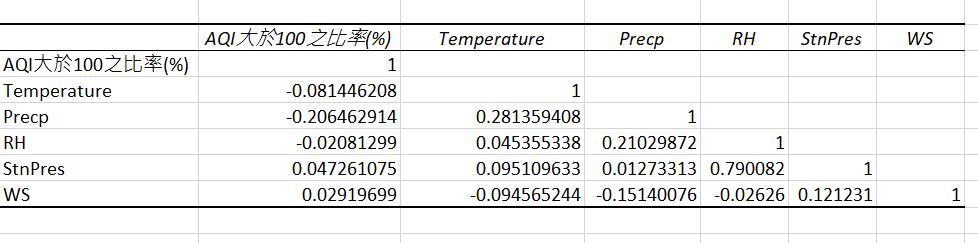
* + 1. TP（True Positive）：預測是真，實際也是真（預測對）
    2. TN（True Negative）：預測是假，實際也是假（預測對）
    3. FP（False Positive）：預測是真，實際是假（預測錯）
    4. FN（False Negative）：預測是假，實際是真（預測錯）

Recall：TP / (TP+FN)。

意義：（預測是真，實際也是真） / （所有真實發生真的情況），代表實際是真的情況，能夠被預測多少比例，因為我們很不希望壞空氣的天數沒有被偵測到，所以要比較看重「實際是壞天氣，能夠被預測出的比例」。

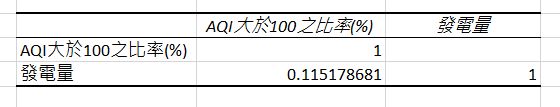
* 1. 結果：
     1. 準確度（Accuracy）： 65.99%
     2. 召回率（Recall）： 33.04%
     3. 以「二元分類」來說，此結果差強人意，表示出天氣資料與AQI的值並沒什麼太大關係。

1. 分析結果
2. 天氣
   1. 就總體而言，AQI與天氣幾乎沒有關連（0.04左右），只有與降水量可能比較有相關(-0.2左右)，如下圖。



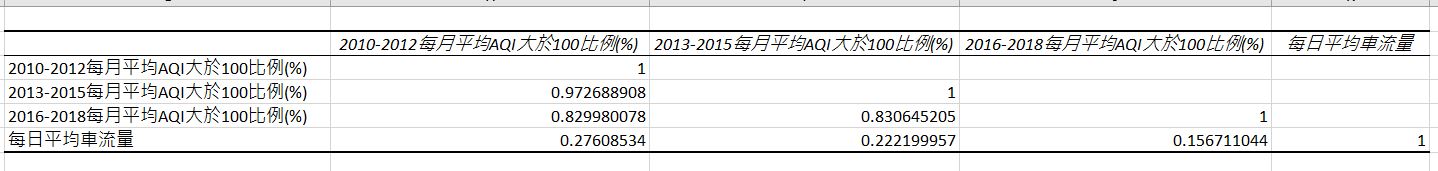
* 1. 就區域來說，北部(新北)、南部(高雄)、東部(花蓮)部分AQI與天氣較無關聯；中部(台中)部分AQI與測站氣壓有中度相關(-0.58)；離島(金門)部分AQI與風速有中度相關(0.61)

1. 火力發電量
   1. 就總體而言，AQI與火力發電量幾乎沒有關聯（0.11左右）

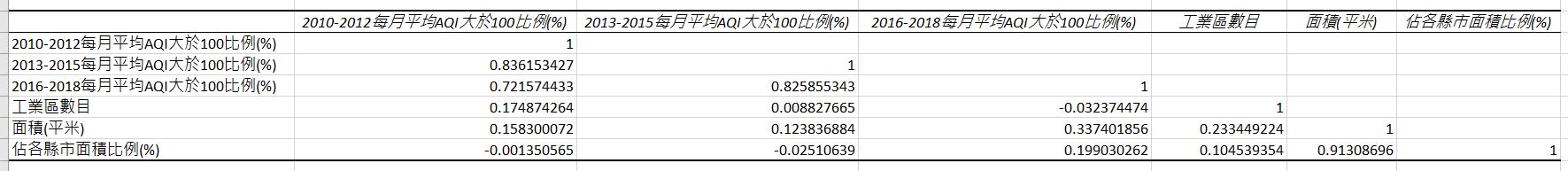


* 1. 從單一縣市來看，雖然大多城市的發電量與AQI皆沒有太大的關連性，但是有些縣市會比其他縣市有更高的相關係數，像是台中的相關係數有達到0.28，但是新北的相關係數卻只有0.01

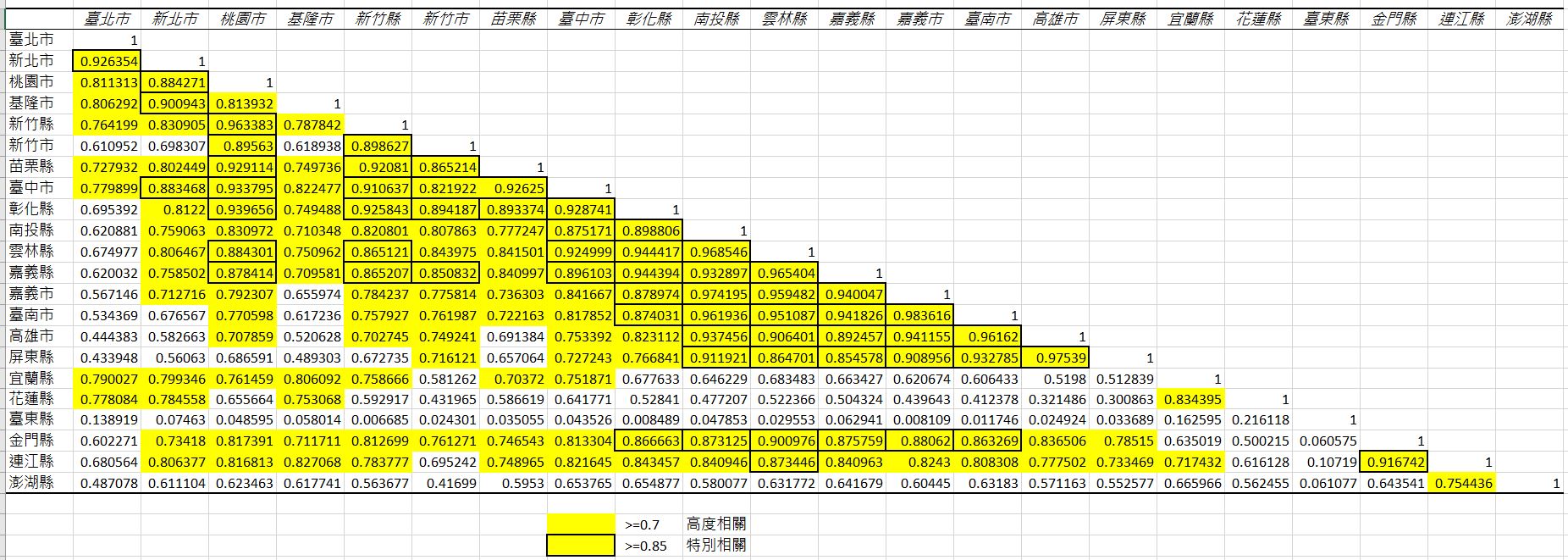
1. 交通量
   1. 總的來說，AQI與交通流量關聯不大，數值都落在0.27左右



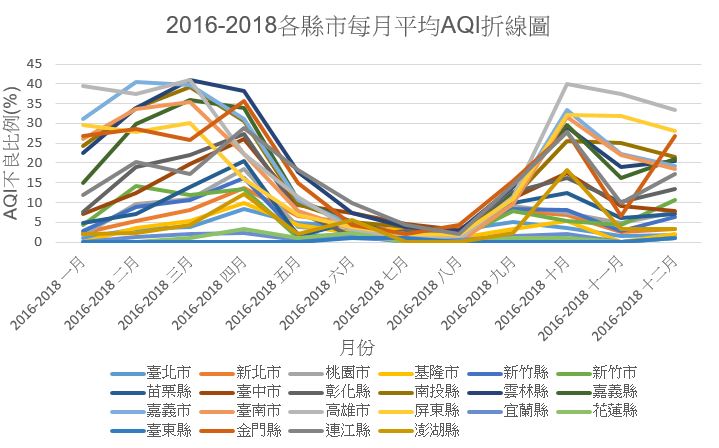
1. 工業區
   1. 從數據來看，工業區的面積與AQI也沒有什麼關聯(0.12~0.33)



1. 地理區域
   1. 從相關矩陣中可以看到，鄰近的區域在相同的時間點對於AQI的值會有非常高度的相關，但是東半部卻沒有這樣的現象



1. 月份
   1. 從圖中可以看到AQI與月份有顯著的關係，在大約10月到隔年4月會有較高的AQI不良比例，而在6~8月時比例較低



1. 分析結論
2. 各縣市之AQI受不同因素影響
   1. 台中市-氣壓和火力發電量
   2. 苗栗縣-火力發電量
   3. 嘉義市-相對溼度
   4. 金門縣和連江縣-風速
   5. 其他縣市-其他待探討因素
3. 天氣
   1. 依據各地區氣候地形條件的不同，對於AQI會有不同的影響。在某些特定氣候地形情況下AQI會與天氣的一些參數有顯著的關聯，但在其他的情況下可能就不會有太大的關聯性。**例如：金門與風速有相關性，可能是因為離島地形平坦易受中國大陸汙染物的擴散影響**
   2. 在特定的情況下，AQI確實會與天氣有關聯，與預期結果相符，但在其他情況下則不一定
4. 火力發電

AQI與火力發電量確實沒有什麼關聯，與預期結果相符

1. 月份

與我們的預期相符，AQI有季節性

1. 未來展望

我們希望我們做得這份研究能為台灣空氣品質的改善盡一份心力，讓政府、民間的團體或媒體注意到我們分析出來的結果，讓以後在討論這個話題時能把重點放在更準確的方向，而不是不斷的處理些並不是那麼重要的問題，期望有更準確的研究團隊對我們的題目做更深入的研究並發表，讓政府能對不同縣市不同地區的空氣品質問題的根本原因去改進。

1. 我們的看法

依據分析的結果，其實我們沒有找到一個影響AQI的決定性因素，我們推測是因為影響AQI的因素實在是太多了，導致最多只能找到中度相關的因素，但還是有比較明顯的因素那就是月份，跟月份最相關的推測是季風的因素，再將資料與季風比對發現冬季吹東北季風時AQI特別高空氣品質特別差，東北季風帶來的汙染源可追朔到中國內陸，所以我們覺得台灣最大的空氣問題還是從大陸隨著季風帶來的汙染物。

1. 參考資料
2. 盧市府防空污 擬卡台中火力電廠生煤許可證、用量 (<https://news.ltn.com.tw/news/politics/breakingnews/2840027>)
3. 107年公民投票第7至16案投票結果 (<https://web.cec.gov.tw/central/cms/RefResults7to16>)
4. 交通部中央氣象局 觀測資訊查詢系統

(https://[e-service.cwb.gov.tw/HistoryDataQuery/index.jsp](https://e-service.cwb.gov.tw/HistoryDataQuery/index.jsp))

1. 臺灣電力公司 過去電力供需資訊 (<https://www.taipower.com.tw/tc/page.aspx?mid=210>)
2. 交通部公路總局 公路交通量調查統計 (<https://www.thb.gov.tw/sites/ch/modules/download/download_list?node=bcc520be-3e03-4e28-b4cb-7e338ed6d9bd&c=83baff80-2d7f-4a66-9285-d989f48effb4>)
3. 台灣工業用地供給與服務資訊網 工業區統計 (<https://idbpark.moeaidb.gov.tw/Environ/statistical>)
4. 行政院環境保護署 環保統計查詢網

(<https://stat.epa.gov.tw/>)

1. Vol.5\_空氣品質指標AQI是什麼？跟常聽見的PM2.5又有什麼關係？(<https://www.xpure-tw.com/blog/vol5_aqipm25>)
2. Machine Learning學習日記 — Coursera篇 (Week 3.2):Cost Function, Simplified Cost Function and Gradient Descent, Advanced Optimization

([https://medium.com/@ken90242/machine-learning學習日記-coursera篇-week-3-2-cost-2e7aa00e8c4a](https://medium.com/@ken90242/machine-learning%E5%AD%B8%E7%BF%92%E6%97%A5%E8%A8%98-coursera%E7%AF%87-week-3-2-cost-2e7aa00e8c4a))

1. 如何辨別機器學習模型的好壞？秒懂Confusion Matrix

(<https://www.ycc.idv.tw/confusion-matrix.html>)

1. 附錄
2. 網頁爬蟲程式碼

import os

import time

from selenium import webdriver

from selenium.webdriver.support.ui import Select

yearlist = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

#yearlist = [2016, 2017, 2018, 2019]

for Years in yearlist:

print(Years)

options = webdriver.ChromeOptions()

prefs = {'profile.default\_content\_settings.popups': 0,

'download.default\_directory': 'D:\\Main\\NTUST\\bigData\_finalTeamProj\\weather\\'+str(Years)}

options.add\_experimental\_option('prefs', prefs)

driver = webdriver.Chrome(

executable\_path="D:\\Main\\NTUST\\bigData\_finalTeamProj\\chromedriver79.exe",

chrome\_options=options)

driver.get("https://e-service.cwb.gov.tw/HistoryDataQuery/index.jsp")

assert "觀測資料查詢系統" in driver.title

driver.implicitly\_wait(30)

selectCountry = Select(driver.find\_element\_by\_id('stationCounty'))

for city in range(len(selectCountry.options)):

currCity = selectCountry.options[city].text

print(currCity)

selectCountry.select\_by\_index(city)

driver.implicitly\_wait(30)

selectStation = Select(driver.find\_element\_by\_id('station'))

for station in range(len(selectStation.options)):

currStation = selectStation.options[station].text

print("\t"+currStation)

if "撤銷站" in currStation:

print("廢棄資料")

continue

fileExist = False

dir\_link = 'D:\\Main\\NTUST\\bigData\_finalTeamProj\\weather\\'+str(Years)

newName = currCity.split(" ")[0]+"\_"+currStation.split(" ")[0]+"\_"+str(Years)+".csv"

if "?" in newName:

newName = newName.replace("?", "[E]")

for file in os.listdir(dir\_link+"\\"+currCity+"\\"):

if newName in file:

print(currCity.split(" ")[0]+"\_"+currStation.split(" ")[0]+"\_"+str(Years)+".csv"+" exist!")

fileExist = True

break

if fileExist:

continue

selectStation.select\_by\_index(station)

driver.implicitly\_wait(30)

selectClass = Select(driver.find\_element\_by\_id('dataclass'))

selectClass.select\_by\_value("inquire")

driver.implicitly\_wait(30)

selectType = Select(driver.find\_element\_by\_id('datatype'))

selectType.select\_by\_value("year")

driver.implicitly\_wait(30)

year = driver.find\_element\_by\_id("datepicker")

year.clear()

year.send\_keys(str(Years))

driver.find\_element\_by\_id("doquery").click()

driver.implicitly\_wait(30)

time.sleep(1)

windows=driver.window\_handles

driver.switch\_to.window(windows[-1])

driver.find\_element\_by\_css\_selector('a#downloadCSV>input').click()

time.sleep(2)

driver.close()

driver.switch\_to\_window(windows[0])

dir\_link = 'D:\\Main\\NTUST\\bigData\_finalTeamProj\\weather\\'+str(Years)

dir\_lists = os.listdir(dir\_link)

dir\_lists.sort(key=lambda fn: os.path.getmtime(dir\_link + '\\' + fn))

os.chdir(dir\_link)

newName = currCity.split(" ")[0]+"\_"+currStation.split(" ")[0]+"\_"+str(Years)+".csv"

if "?" in newName:

newName = newName.replace("?", "[E]")

os.rename(dir\_lists[-1], newName)

os.replace(dir\_link+"\\"+newName, dir\_link+"\\"+currCity+"\\"+newName)

print()

driver.quit()

print()

1. 資料整理程式碼

import pandas as pd

import os

def step1():

for year in os.listdir():

if os.path.isdir(year):

os.chdir(year)

for city in os.listdir():

if os.path.isdir(city):

os.chdir(city)

all\_data = pd.DataFrame()

for csvFile in os.listdir():

if '.csv' in csvFile and 'all' not in csvFile:

df = pd.read\_csv(csvFile)

all\_data = pd.concat([all\_data, df])

print(os.getcwd())

all\_data.to\_csv('all.csv')

os.chdir('../')

os.chdir('../')

def step2():

for year in os.listdir():

if os.path.isdir(year):

os.chdir(year)

for city in os.listdir():

if os.path.isdir(city):

os.chdir(city)

try:

os.remove('result.csv')

except:

print()

os.chdir('../')

os.chdir('../')

def step3():

import pandas as pd

import os

pd.set\_option('mode.chained\_assignment', None)

for year in os.listdir():

if os.path.isdir(year):

os.chdir(year)

for city in os.listdir():

if os.path.isdir(city):

os.chdir(city)

items = ['Temperature', 'Precp', 'RH', 'StnPres', 'WS']

df = pd.read\_csv('all.csv')

new\_df = pd.DataFrame()

new\_df['Mouth'] = range(1, 13)

for item in items:

new\_df[item] = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]

length = df.shape[0]

l = [0,0,0,0,0,0,0,0,0,0,0,0]

for i in range(length):

try:

float(df[item][i])

mouth = df['ObsTime'][i]

l[mouth-1] += 1

new\_df[item][mouth-1] += float(df[item][i])

except ValueError:

a=0

for i in range(12):

if l[i] != 0:

new\_df[item][i] = float(new\_df[item][i]) / l[i]

new\_df[item][i] = round(new\_df[item][i], 2)

new\_df.to\_csv('result.csv')

os.chdir('../')

os.chdir('../')

step1()

step2()

step3()

1. 機器學習程式碼

import numpy as np

import pandas as pd

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import confusion\_matrix

LR = LogisticRegression()

df1 = pd.read\_csv('天氣總表.csv')

yi = pd.read\_csv('for\_MC.csv')['AQI大於100之比率(%)'][:2376]

xi = df1[['Temperature','Precp','RH','StnPres','WS']]

xi.insert(0, 'Constant', 1)

# 用來增加參數

xi['T^2'] = df1['Temperature'] \* df1['Temperature']

xi['P^2'] = df1['Precp'] \* df1['Precp']

xi['R^2'] = df1['RH'] \* df1['RH']

xi['S^2'] = df1['StnPres'] \* df1['StnPres']

xi['W^2'] = df1['WS'] \* df1['WS']

xi['TP'] = df1['Temperature'] \* df1['Precp']

xi['TR'] = df1['Temperature'] \* df1['RH']

xi['TS'] = df1['Temperature'] \* df1['StnPres']

xi['TW'] = df1['Temperature'] \* df1['WS']

xi['PR'] = df1['Precp'] \* df1['RH']

xi['PS'] = df1['Precp'] \* df1['StnPres']

xi['PW'] = df1['Precp'] \* df1['WS']

xi['RS'] = df1['RH'] \* df1['StnPres']

xi['RW'] = df1['RH'] \* df1['WS']

xi['SW'] = df1['StnPres'] \* df1['WS']

# ====================================

x = xi.to\_numpy()

y = yi.to\_numpy(dtype=int)

sc = StandardScaler()

sc.fit(x)

x = sc.fit(x).transform(x)

LR.fit(x, y)

# =============

xt = x

yt = LR.predict(xt)

accuracy = LR.score(x, y)

tn, fp, fn, tp = confusion\_matrix(y, yt).ravel()

recall = tp / (tp + fn)

print(str(accuracy) + ' ' + str(recall))

df = pd.DataFrame(y)

df.to\_csv('result.csv')

1. [1] 資料來源: <https://news.ltn.com.tw/news/politics/breakingnews/2840027> [↑](#footnote-ref-1)
2. [2] 公投結果來源為以下網址之「107年全國性公民投票案第7-16案結果.pdf」: <https://web.cec.gov.tw/central/cms/RefResults7to16>，此處之數據為同意票、不同意票、無效票人數分別占總實際投票人數之比例，與資料來源之同意票、不同意票分別占有效票數之比例的計算方式不同，並非數據之誤植。 [↑](#footnote-ref-2)
3. [1] 完整大圖: <https://franktaiwan123.github.io/getWeatherFlowChart/Flow.html> [↑](#footnote-ref-3)
4. [1] 舉例來說，就是先排西元2010年1月的所有縣市之後再排西元2010年2月的所有縣市，依序下去 [↑](#footnote-ref-4)
5. [2] 舉例來說，西元2016年5月台北市5個測站中的中山、古亭、萬華在當月分別有1天、2天、2天的AQI超過100，另外的2個測站在當月的AQI皆沒有任何一日超過100，且所有測站在當月的31天內皆有運作，所以2016年5月台北市AQI大於100之比例為 (1 + 2 + 2) / (31 \* 5) ≈ 0.0323 = 3.23%，該欄位之數值為3.23。 [↑](#footnote-ref-5)
6. [1] 舉例來說，2013~2015台北市的平均每月AQI大於100比例(%)計算的方式就是將台北市在2013~2015期間的「每月AQI大於100比例(%)」這一個項目做算術平均 [↑](#footnote-ref-6)
7. [1] 舉例來說，2010~2012這三年1月的「AQI大於100比例」做算術平均就會變成其中一列，列名為「2010-2012 一月」 [↑](#footnote-ref-7)