

交通數據處理與流量分析

國立台灣師範大學資訊工程系

學生：40947012S 黃至瑜、40947064S 蔡少芸，指導教授：王科植

(三) 摘要

本次專題旨在進行交通數據處理與流量分析。我們採用了大量的交通數據，並運用互動式儀表板來呈現數據的特徵，目標是幫助決策者（目標受眾）解決交通相關的問題和優化交通流量，透過操作儀表板深入了解交通狀況以及不同變數之間的相關性，如車流量是否會影響車速。

我們所使用的資料包含三種類型：基本資訊、車速與車流量、其他資訊。基本資訊包括道路座標、名稱和分類等。車速與車流量則提供有關交通流量和車輛速度的數據。其他資訊則包括道路走向和更新時間等相關資訊。其涵蓋的時間範圍從 2020 年 1 月至 2022 年 3 月，地點主要集中在台灣北部，特別是台北市（ $121.62\sim 121.46^{\circ}\text{E}$ ， $25.17\sim 24.9^{\circ}\text{N}$ ）。這些數據將為我們的專題提供有關交通狀況和相關特徵的重要信息。

在上學期的工作中，我們主要專注於儀表板的結構設計和互動圖表的實現。然而，由於算法的時間複雜度過高，導致儀表板的載入時間過長。因此，在本學期中，我們的主要目標是進行算法優化，以提高數據載入和處理的效率。同時，我們也將根據使用者的需求和實用性考量，進行功能的修改，包括圖表刪減或增加。最後，我們還將致力於完善儀表板的互動體驗，以確保操作的流暢性，提升整體儀表板的性能和使用者滿意度。

(四) 研究動機與背景

交通是日常生活中必不可少的元素，不論是外出旅遊或是日常外出，交通的便利都讓我們的生活水平上升，但隨之也會引起各種各樣的問題，如交通事故的發生、車輛造成的噪音影響住宅品質。因此我們希望可以把近年來的交通數據以視覺化的形式呈現出來，讓目標受眾能使用它做一些交通相關的規劃：

1. 交通部

可以預測車輛壅塞狀況。因為在連假可能會對部分高速公路路段進行高乘載管制，所以需要平面替代道路，如省道等。除此之外，交通部需要各種車的车速來進行道路的新建與改

善，而道路規劃需要根據此路段車輛需求。

2. 一般使用者

若使用者需要購置新房屋，也可以利用儀表板所呈現的數據，利用車流量預估新住宅的道路噪音狀況或交通便利性，找到更心儀的住所。若使用者需要編排旅遊行程，可以依據車流量選定旅行月份和預估車程往返的時間。若目的地路線的車流量較多，就可以預留更長的時間，也可以選擇流量較少的路線作為第一選擇，以便他們規劃旅行路線及景點數目。

製作儀表板的初衷是為了可以讓目標受眾能使用更便利的工具來協助他們解決上述問題。雖然交通部獲得交通數據是非常容易的事，但在沒有任何工具的協助的情況下，一般使用者其實很難憑自己的能力去做相關的分析，想做好旅遊的規劃也只能逐一在網上查找各種各樣的資料，而且進行交通數據分析也需要擁有一定程度的知識。或者說選擇住所，噪音狀況或交通便利性除了網上搜尋資料外，就只能依靠自己實地勘察，這樣花費的時間成本就更長，但儀表板是一個整合各種數據的工具，因此可以幫助目標受眾更快得到預期的結果。

（五） 相關文獻/既有系統探討

交通數據處理已經應用於相當多案例中，並且以不同的互動形式讓使用者可以探索和比較結果。例如：視覺化地查詢計程車路線，除了支持標準分析，也支持使用者查詢起點與目的地。[1] 交通數據處理主要是分析不同時間與空間下的交通狀況，並描述數值與類型屬性的方法，最重要的是數據視覺化可以有效地顯示數據中隱藏的資訊[2]，因此視覺化分析方法也影響著儀表板的呈現情形。[3]

（六） 開發工具與軟/硬體需求

本次專題使用 Python 和它的一些外部的模組來實作儀表板：

Plotly、shapely—繪製圖表

Dash、json—製作儀表板界面及實現圖表間的互動

Pandas、numpy 等—資料的處理

除了撰寫程式外，我們也使用了 gunicorn 和 render.com 來架設網站，讓儀表板可以在 server 中運作，這樣使用者只需透過網址即可進入儀表板的介面。對於電腦硬體無任何要求。

(七) 系統結構與操作流程

圖表繪製使用了 Python 中的 plotly.express 套件去繪製圖表，我們使用到的函式有 `px.scatter_mapbox`、`px.bar`、`px.imshow`、`px.pie`、`px.line` 及 `px.choropleth`。圖表互動使用 Dash 的 Callback function 實現，除了 Callback function 的定義外，在 source code 中我們也建立了三個函式，分別是 `time_def`、`week_def` 及 `get_text_fig`，主要是處理熱力圖的資料時會用到，由於原資料中並沒有提供星期的數據，因此我們有定義 `week_def` 函式去做星期的計算。然後還有區分時段，我們有定義 `time_def` 把時間作分類。`get_text_fig` 則是根據參數回傳一個文字方塊的圖表，主要是為了讓程式可讀性提高。並且在繪製圖表前我們先對資料做一些處理，抽出需要的欄位建立 Data Frame，並先做好 Mapbox 的等級劃分，方便繪圖時使用。

使用者可根據自身的需求對圖表進行操作，除了面量圖和熱力圖不可作互動操作外，其他圖表皆可通過選取或點擊事件對自己在內的圖表做互動，以下為使用案例說明（想規劃旅遊路線的一般使用者）：

首先透過地圖和面量圖可以知道台北地區的交通狀況如何，比如我的行程中打算在以下區域逛街（看 Figure 1），選取後的界面（Figure 2）可看到熱力圖顯示該區域在星期六的中上及中午流量較為密集，假如我想再多一個操作，因為我決定在 2 月時來旅遊，那使用者可以再從折線圖中選取 2 月的資料，這樣篩選條件為 Figure 1 選取的範圍和 Figure 3 中選取的 2 月份的資料的交集。

(八) 系統功能之分項說明

系統功能以互動圖表為主：

- 1.地圖（Mapbox）：以 Bubble 形式表示每個感測器所取得之車流量。鼠標移動到 Bubble 上能看到對應感測器的詳細資訊，如路段名稱、坐標點及感測器偵測到的總流量。圈選地圖上某一個範圍，該範圍內的資訊將會作為篩選條件套用至其他圖表上，並更新界面。實作的方法為使用 plotly.express 套件的 `scatter_mapbox` 繪製圖表，由於預設的地圖並沒有顯示詳細的道路資訊，故 Mapbox 的樣式有更改為 "carto-positron"。

- 2.長條圖（Bar Chart）：統計 Mapbox 地圖上的所有 Color Bubble 的數量。點選長條圖上某一顏色的方塊，該方塊內的 Color Bubble 的資訊會作為篩選條件套用至其他圖表上，並更

新界面。

3.熱力圖 (Heat Map)：顯示資料在每周不同日、早午晚三田人十口時段的車流量它集程度，其中密集程度以連續性的顏色條表示，高度密集則會更接近紅色，反之更接近黃色。由於資料的不完整性，導致某些日期並無資料內容，故該圖表無法作為篩選條件控制其他圖表。熱力圖是用 plotly.express 套件的 imshow 繪製，顏色的設定為 color_continuous_scale 中提供的 Pinkyl。

4.圓餅圖 (Pie Chart)：顯示各車型佔資料中總流量的比例。可通過點選相應顏色的區塊進行互動，點選色塊後會篩選出資料中該車型的資訊作為其他圖表的輸入值，並更新界面。使用 plotly.express 套件的 pie 繪製，其中顏色的設定為離散的。

5.折線圖 (Line Chart)：顯示各車型佔每日平均的速度。可圈選折線圖中的範圍，該範圍內的資訊將會作為篩選條件套用至其他圖表上，並更新界面。使用 plotly.express 套件的 line 繪製，顏色的設定為離散的。

6.面量圖 (Choropleth Map)：按照台灣行政區分割成明的區域，並顯示各行政區車輛的總流量。該圖表可受其他圖表控制，但不可作為主控制（即不可操作）。面量圖是用 plotly.express 套件的 choropleth 繪製，其中行政區的劃分有引用開源的 Geojson 檔案。

7.文字方塊：僅顯示資料的總流量及平均速度，可受其他圖表控制，但不可作為主控制（即不可操作），文字方塊的繪製使用 plotly 的 scatter。因為資料已經做了預處理，故可直接使用 pandas 提供的.sum 取得總車流量，而平均速度也是可直接使用 pandas 提供的.mean 取得。

以上提到的互動皆可交叉操作，若同時有多個圖表的內容作為篩選條件，則會取其交集作為最終的篩選條件。

（九）系統特色

1. 根據使用者需求可實現不同圖表間互動，包含以下四種圖表的任意交集

- a. Mapbox（地圖）圈選特定路段
- b. Bar Chart（長條圖）點選任一車流量大或小的道路
- c. Pie Chart（圓餅圖）點選任一車輛種類

d. Line Chart（折線圖）圈選連續特定日期

交集後的所有圖表與數字均受其選擇的影響，因此可準確的提供相關資訊。

2. 連續顏色分布

Heat Map（熱力圖）中可看到以連續的顏色表示相對的車流量大小，以及每天早中晚那些時段的車輛較多。Choropleth Map（面量圖）中以連續顏色表示台北市各行政區車流量比較。使用者可直觀地找出車輛較密集之區域與時段，避免塞車等情況。

3. 同顏色對應到不同圖表

Bar Chart（長條圖）為 Mapbox（地圖）上所有 Color Bubble 的數量統計，所以使用者可以依據 Bar Chart（長條圖）直接了解 Mapbox（地圖）的分布狀況。

4. 清晰的地圖與面量圖

原本 Mapbox（地圖）使用的是 Open Street Map 的預設地圖，由於 OSM 預設地圖須表示不同道路種類與地形，所以色彩非常繽紛。但是此地圖使 Bubble 不容易可見並造成使用體驗差，所以改成灰階地圖 Carto Positron。Choropleth Map（面量圖）使用台北市各行政區的 Geojson file，可對照 Mapbox（地圖）的行政區車流量。

5. 數字與圖表的對照

介面新增的兩項數字欄位，分別為 Total Volume 與 Average Speed，雖然記錄的也是車流量與平均速度，但是與其他圖表的分項資料有所不同，因為這兩者都是統整的資料，所以令使用者能直接掌握的整體 datasets 的大小。

（十） 各組員之貢獻

以下編號為時間順序

黃至瑜：

1. 撰寫 Line Chart（折線圖）內容、撰寫 Heat Map（熱力圖）內容

2. 減少運行時間及提升效率、增加資料量與時間範圍、新增 Pie Chart（圓餅圖）

點選互動、新增 Line Chart（折線圖）點選互動

3. 撰寫 Choropleth Map（面量圖）內容、增加所有圖表交集的互動模式、增加文字欄位

蔡少芸：

1. 整合圖表到 dashboard、撰寫 Mapbox（地圖）內容、撰寫 Bar Chart（長條圖）內容、撰寫 Pie Chart（圓餅圖）內容、新增 Mapbox（地圖）圈選互動
2. 新增 Bar Chart（長條圖）圈選互動、更換 Mapbox（地圖）的預設地圖、更換 Heat Map（熱力圖）為同顏色漸變、更新 dashboard 排版
3. 部署 dashboard 到網頁

（十一）後續研究/實作方向

目前儀表板的互動介面已全數完成，包含各圖表間的交集互動，也提升效率、改善錯誤與美化界面。另外，我們的儀表板已經部署到網站上，然而我們所使用的工具是 plotly 官方推薦的網站 render，免費的方案有資料大小與速度限制，由於後臺數據相當龐大，因此只能呈現部分數據且造成互動速度變慢。

未來我們可能會改善部署網站的問題，運用 Synology NAS 或是其他工具來解決現有的大小與速度困境，並且致力於將完整的儀表板部署到網站上提供線上使用者使用。我們也會根據使用者給我們的回饋，持續改善儀表板可能潛在的問題。

（十二）參考文獻

- [1] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. IEEE Transactions on Visualization and Computer Graphics, 19(12):2149–2158, 2013.
- [2] W. Chen, F. Guo, and F.-Y. Wang. A survey of traffic data visualization. IEEE Transactions on Intelligent Transportation Systems, 16(6):2970–2984, 2015.
- [3] N. Andrienko and G. Andrienko. A visual analytics framework for spatiotemporal analysis and modelling. Data Mining and Knowledge Discovery, pp. 1–29, 2013.