

# 資料蒐集工具與應用分享

陳文昱

國立臺灣大學  
土木工程學系

Department of Civil Engineering,  
National Taiwan University

中華民國114年7月28日



# Table of Contents

## 1 工具簡介

- GIS
- 程式設計前置作業
- 程式設計工具
- 爬蟲
- 排版語言 $\text{\LaTeX}$ 與MiKTeX
- AI工具

## 2 事故資料彙整

- 道安資訊查詢網
- 原始資料處理
- Accident Analysis.ipynb使用說明
- 報告彙整

## 3 學習資源



# Table of Contents

## 1 工具簡介

- GIS
- 程式設計前置作業
- 程式設計工具
- 爬蟲
- 排版語言 $\text{\LaTeX}$ 與MiKTeX
- AI工具

## 2 事故資料彙整

- 道安資訊查詢網
- 原始資料處理
- Accident Analysis.ipynb使用說明
- 報告彙整

## 3 學習資源



- 尋找圖資、匯入圖資
  - ▶ Quick Map Service使用說明、相關資料
  - ▶ Sentinel-2、ESRI、Google、Bing
- 篩選、環域分析，匯出gpkg或shp檔案
- 匯出地圖png或svg檔案、屬性資料csv檔案



# 資料來源

- 高公局資料庫
- GIS-T交通網路地理資訊倉儲系統
- 國土測繪圖資服務雲
- 政府資料開放平台



# ChromeDriver下載

- 1 Chrome設定>關於Chrome：檢視版本
- 2 進入ChromeDriver網站
- 3 下載對應版本
- 4 將chromedriver.exe放在程式碼要執行的路徑



# 開啓與編輯ipynb 檔案

- 安裝Anaconda
- 使用VS Code或Jupyter Lab寫程式



# 打開CMD.exe Prompt

```
1 pip install selenium
2 pip install requests
3 pip install xmltodict
```





# 程式設計工具概覽

## ● Visual Studio Code:

- ▶ 微軟開發的輕量且功能強大的程式碼編輯器。
- ▶ 支援多種程式語言，並可透過擴充套件擴展功能。
- ▶ 內建終端機、除錯工具和版本控制功能。

## ● GitHub Copilot:

- ▶ 由OpenAI 和GitHub 開發的AI 驅動的程式碼補全工具。
- ▶ 根據上下文提供智慧程式碼建議。
- ▶ 幫助加速編寫程式碼並縮短開發時間。



## ● 功能:

- ▶ **擴充套件市集**: 擁有上千種擴充套件以增加功能。
- ▶ **IntelliSense**: 根據變數類型、函數定義等提供智慧程式碼補全。
- ▶ **Live Share**: 支援開發者之間的即時協作。



## ● 優勢:

- ▶ **生產力**: 提供上下文智慧建議，加速編寫程式碼。
- ▶ **學習輔助**: 幫助開發者學習新的API 和程式語言。
- ▶ **錯誤減少**: 減少常見編碼錯誤，提升程式碼品質。



# 程式設計工具小結

- **VS Code** 和 **GitHub Copilot** 為開發者提供了一套強大的工具。
- 適合新手和有經驗的開發者。
- 顯著提升編碼效率和協作能力。

探索更多：

- Visual Studio Code 官方網站
- GitHub Copilot 官方頁面



# ChromeDriver 概念介紹

ChromeDriver 是一個提供WebDriver 協議的工具，主要用於自動化Chrome 瀏覽器的操作。它允許用戶通過程式碼控制瀏覽器進行各種操作，如打開網頁、填寫表單、點擊按鈕等。ChromeDriver 通常用於測試和網頁自動化的場景。



# ChromeDriver 基本功能介紹

- **Text :**

- ▶ ChromeDriver 允許用戶在網頁上的文本框中輸入文字。這些文字可以是用戶提供的資料或自動生成的測試數據。

- **Click :**

- ▶ 透過ChromeDriver 用戶可以模擬點擊網頁上的按鈕或鏈接。這對於自動化操作流程和測試用戶互動至關重要。

- **SendKeys :**

- ▶ ChromeDriver 支援向網頁元素發送鍵盤輸入，這包括按鍵、刪除鍵、功能鍵等。這使得測試網頁表單和其他輸入功能成為可能。

- **Screenshot :**

- ▶ 用戶可以使用ChromeDriver 擷取瀏覽器視窗的截圖。這對於生成測試報告和確認網頁內容的正確性非常有用。



# ChromeDriver 定位網頁元素的方法(1/2)

ChromeDriver 提供多種方法定位網頁上的元素，以下是幾種常用的方法

- **ID :**

- ▶ 透過元素的唯一識別符(ID) 來定位。這是最直接且效率高的方法，但依賴於元素具有唯一性。

- **Name :**

- ▶ 使用元素的名稱屬性來查找。這在表單元素中特別有用，例如輸入框和選擇框。

- **Class Name :**

- ▶ 根據元素的CSS 類別名稱進行定位。適合用於具有特定樣式的元素群組。

- **Tag Name :**

- ▶ 透過HTML 標籤名稱查找元素。常用於抓取所有某類型的元素，如所有的鏈接或圖片。



# ChromeDriver 定位網頁元素的方法(2/2)

- **XPath :**

- ▶ 利用XPath 表達式來定位元素，支持複雜的查詢條件，能夠在DOM 樹中查找任意位置的元素。

- **CSS Selector :**

- ▶ 使用CSS 選擇器來查找元素。這是定位元素的強大方法，能夠選擇特定樣式或層級結構的元素。

- **Link Text :**

- ▶ 根據鏈接文本查找超鏈接元素，適用於確定文字的鏈接。

- **Partial Link Text :**

- ▶ 使用部分鏈接文本來查找元素，適合用於鏈接文本較長或只記得部分文字的情況。





# ChromeDriver 示例(XPath定位)

```
1 driver.find_element('//*[@id="block"]/div[1]/div  
    [1]/a/div[1]',xpath).click()  
2 print(driver.find_element('//*[@id="block"]/div[1]/  
    div[1]/a/div[1]',xpath).text)  
3 driver.save_screenshot(f'Figure/19260_F01N92K+680  
    _202407110800.png')
```



# 其他瀏覽器的WebDriver 概念介紹

除了ChromeDriver，還有多種其他瀏覽器的WebDriver 可供使用，它們各自支持不同的瀏覽器並提供類似的自動化功能。以下是幾個常見的WebDriver 介紹：

- **GeckoDriver：**

- ▶ 支援Mozilla Firefox 瀏覽器。

- **SafariDriver：**

- ▶ 用於Apple Safari 瀏覽器。

- **EdgeDriver：**

- ▶ 支援Microsoft Edge 瀏覽器。

- **OperaDriver：**

- ▶ 用於Opera 瀏覽器。



# 什麼是 $\text{\LaTeX}$ ？

- $\text{\LaTeX}$  是一種專業排版語言，用於創建高品質的技術和學術文檔。
- 與傳統的文字處理軟體（如Microsoft Word）相比， $\text{\LaTeX}$ 提供：
  - ▶ 更精確的排版控制。
  - ▶ 自動化的參考文獻管理。
  - ▶ 強大的數學公式編輯功能。
- $\text{\LaTeX}$ 的優勢：
  - ▶ 高度可擴展：支援各類模板和套件。
  - ▶ 穩定性：適合處理大型文檔。
  - ▶ 可重現性：使用原始文本文件，可輕鬆協作和版本控制。
  - ▶ 參照功能強大： $\text{\LaTeX}$ 的參照和目錄生成功能自動化，確保準確性，並且能夠自動更新頁碼和編號。
- 易於入門：
  - ▶ 初學者可以透過簡單的範例和在線資源快速上手。
  - ▶ 與Office 軟體相比， $\text{\LaTeX}$ 在格式化和自動化排版上具備優勢。



# MiKTeX 與.tex 檔案簡介

## MiKTeX

- MiKTeX 是一個用於Windows 系統的 $\text{\LaTeX}$ 發行版
- 提供了一套完整的工具集，可編譯.tex 檔案生成高品質的PDF 文件
- 包含 $\text{\TeX}$ 、 $\text{\LaTeX}$ 和相關的宏包(package)，支援自動下載缺失宏包

## .tex 檔案

- .tex 檔案是一種以文本形式儲存的 $\text{\LaTeX}$ 原始碼文件
- 包含了文檔的結構、內容、格式指令等
- 透過編譯，.tex 檔案可生成PDF、DVI 或其他格式的輸出文件



# Python 操作MiKTeX

- Python 可以透過os 模組來呼叫系統指令
- 使用os.system('pdflatex file.tex')指令，可以自動編譯.tex 檔案為PDF
- 這種方法可被用於自動化文檔生成流程，例如批量處理文件  
應用可見第??頁、第??頁



# 使用Beamer 製作簡報

- 本簡報即以Beamer製作。
- **Beamer** 是 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的一個專門用於製作簡報的套件。
- **Beamer** 的特點：
  - ▶ 支援各類主題和色彩方案，便於客製化。
  - ▶ 完全整合 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的排版能力，適合數學公式和圖表的展示。
  - ▶ 自動生成目錄、索引和導航欄。
- **應用場景：**
  - ▶ 學術會議和技術研討會。
  - ▶ 工程項目的報告和演示。



# 使用TikZ Picture 繪圖

- **TikZ** 是一個強大的 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 套件，用於繪製高品質的向量圖。
- **TikZ 的特點：**
  - ▶ 支援創建精確的技術插圖和流程圖。
  - ▶ 能夠繪製複雜的數學圖形和幾何圖形。
  - ▶ 提供多種樣式和效果，以增強視覺吸引力。
- **應用場景：**
  - ▶ 交通工程中的流程圖和網絡圖。
  - ▶ 技術報告中的數據可視化。



# 入門者學習建議

- 使用 Overleaf :

- ▶ 提供方便的線上 $\text{\LaTeX}$ 編輯環境。
- ▶ 擁有豐富的模板庫和範例文件。

- 利用 ChatGPT 學習 :

- ▶ 可以快速解答 $\text{\LaTeX}$ 相關問題。
- ▶ 協助學習者了解語法和功能。

- 學習資源 :

- ▶ 官方文檔和指南。
- ▶ 在線論壇和社群支持。
- ▶ 教學視頻和網課資源。





# 未來應用展望

- **L<sup>A</sup>T<sub>E</sub>X**在交通工程中的應用：

- ▶ 自動化生成交通報告和技術文檔。
- ▶ 創建專業的研究論文和簡報。

- **提升工作效率：**

- ▶ 通過L<sup>A</sup>T<sub>E</sub>X和Beamer 提升團隊的報告製作效率。
- ▶ 使用TikZ 增強圖形和數據的可視化效果。



# 生成式人工智慧概述

- **ChatGPT**：由OpenAI 開發的強大對話生成AI，可應用於多種情境，包括內容創作、技術支援等。
- **Claude**：由Anthropic 開發，專注於生成高質量文本，適合用於正式報告和文書處理。
- **Gamma**：結合大綱生成和內容創作工具，擅長於提供結構化和具體的內容建議。
- 其他工具：Taiwan LLM、TAIDE、Gemeni、Mistral AI、Perplexity AI、Decktopus等



# ChatGPT 的優勢

- **多樣化應用：**

- ▶ 支持多語言交流，涵蓋技術支援、文案創作等多個領域。
- ▶ 擅長生成自然、流暢的對話，提供即時解決方案。

- **智能學習能力：**

- ▶ 能夠根據用戶反饋和數據訓練進行優化和改進。
- ▶ 提供自適應的語言模型，適應不同的使用場景。

- **相對其他AI 的優勢：**

- ▶ 擁有龐大的知識庫，提供快速準確的資訊。
- ▶ 支持插件和API，易於集成到現有系統中。



# Claude 的特點

- **高質量文本生成：**

- ▶ 生成的內容較為可靠，適合正式文檔和報告書。
- ▶ 文字表達洗鍊，提升文件的專業水準。

- **應用場景：**

- ▶ 政府公文撰寫、公司內部報告和學術論文等。
- ▶ 涵蓋複雜技術概念的清晰表達。



- **結構化內容創作：**

- ▶ 結合ChatGPT 生成大綱，提供良好的文檔結構。
- ▶ 適合創建專題報告、策略計劃等。

- **配合AI 工具的最佳效果：**

- ▶ 給予明確的大綱和結構，能夠大幅提升內容質量。
- ▶ 協同AI 工具，共同生成具備深度和廣度的文本。



# 使用生成式AI 的技巧

## ● 指令的明確性：

- ▶ 提供清晰的目標和脈絡(上下文)，有助於獲得精準的輸出。
- ▶ 確保問題和需求的具體性，提高生成效果。

## ● 提示詞的重要性：

- ▶ 設定人設、情境與範例，讓AI 更容易理解目標。
- ▶ 例如：您是大學教授，要審查這份報告，以下是過往的審查意見...

## ● 額外技巧：

- ▶ 善用AI 的批判性思考，讓其提供不同觀點和意見。
- ▶ 透過對話，反覆優化提示詞，逐步提升內容的深度和準確性。
- ▶ 將複雜的任務拆解為簡單的任務，例如：寫一篇文章，可以先生成大綱，再針對各章節生成內容。

應用參見第55頁



# 使用生成式AI 的技巧(1/2)

在使用生成式人工智慧工具時，以下技巧能幫助你提高與AI 的互動效率，並獲得更準確的回應：

## 10 個有效的提示工程技巧

- **給予激勵**：告訴AI 若提供好的答案，將會有獎勵。
- **避免負面指令**：使用正面語句來指示AI，避免使用否定語句。
- **例子提示**：提供範例以引導AI 更好地理解任務要求。
- **雙向提示**：要求AI 向你提問以獲取足夠資訊，增進互動效果。
- **思維鏈提示**：將複雜任務分解成簡單步驟，以幫助AI 梳理思路。



# 使用生成式AI 的技巧(2/2)

## 10 個有效的提示工程技巧

- **結構化格式**：使用適當的格式來撰寫提示，使指令更加清晰明確。
- **輸出引導**：在提示開頭加入所需的輸出內容，以設定AI 回應的方向。
- **任務強調**：使用強調語句來明確任務的重要性，提升AI 的專注度。
- **提及後果**：告知AI 不遵循指示的後果，強化指令的執行力。
- **自我評估提示**：讓AI 自我評估並提出改進建議，促進AI 的學習與改進。

## 參考資料





# AI 在現代工作的角色

- **處理重複性工作：**

- ▶ 自動化常見任務，釋放人力資源以處理更具創意的工作。
- ▶ 減少人工錯誤，提高工作效率。

- **顧問角色：**

- ▶ 協助批判性思考，提供多角度的分析和建議。
- ▶ 辨識潛在問題，助於解決複雜的挑戰。

- **未來展望：**

- ▶ AI 將逐漸成為團隊中的一員，協同工作創造更多價值。
- ▶ 持續創新，為行業帶來新的可能性。



# 小結

會用**AI** 的人，將取代不會用**AI** 的人。

在AI 時代下，學習並掌握生成式人工智慧工具將成為職場中的關鍵競爭力。善用AI 的力量，讓您在工作中更具優勢。



# Table of Contents

## 1 工具簡介

- GIS
- 程式設計前置作業
- 程式設計工具
- 爬蟲
- 排版語言 $\text{\LaTeX}$ 與 $\text{MiKTeX}$
- AI工具

## 2 事故資料彙整

- 道安資訊查詢網
- 原始資料處理
- Accident Analysis.ipynb使用說明
- 報告彙整

## 3 學習資源



# 交通事故的分類

交通事故依照事故嚴重程度區分為三類：

- **A1類事故**：造成人員當場或24小時內死亡。
- **A2類事故**：造成人員受傷，或24小時後死亡。
- **A3類事故**：僅有財物損失，無人傷亡。

理解這三類型有助於資料過濾與分析方向之設定。



利用ChromeDriver從道安資訊查詢網擷取嘉義縣內事故資料

- 彙整「學校周邊500公尺易肇事熱點事故統計表」
- 手動比對國土規劃地理資訊系統，篩選出發生於都會區的事故，彙整為「一般易肇事熱點事故統計表」
- 免去手動點選、輸入的工作，用Excel整理即可
- 該網站缺乏座標資料，無法匯入GIS比對都市計畫區位



# 原始資料取得

- 合併警政署A1、A2類事故資料
- 來源：政府資料開放平臺
  - ▶ 112年傷亡道路交通事故資料
  - ▶ 111年傷亡道路交通事故資料
  - ▶ 即時交通事故資料(A1類)
  - ▶ 即時交通事故資料(A2類)
- 工具：Python
- 格式：csv 檔案



# 程式步驟一：匯出CSV 與JSON

- 原始事故資料中，每筆代表一位涉入者。
- 若欲分析事故事件數（非人數），需將相同事故編號合併。
- **輸出CSV**：方便匯入QGIS 進行地理視覺化。
- **輸出JSON**：便於彙整為「每件事故一筆」，以利後續統計。



# 程式碼：將資料儲存為CSV 檔

```
data = pd.read_csv('事故資料.csv', header=None)
data.columns = ['事故編號', '經度', '緯度', ...]
data.to_csv('output.csv', index=False, encoding='utf-8-sig')
```

- 將欄位命名，利於QGIS 辨識經緯度欄位。
- 設定編碼為utf-8-sig，避免中文亂碼。





## 程式步驟二：使用Feather 格式

- Feather 是快速讀寫的二進位格式。
- 適合大型資料交換與儲存。
- 與CSV 相比，載入速度顯著較快。
- 範例：500萬筆A2 類事故與行政區疊圖分析
  - ▶ QGIS 約需50 分鐘
  - ▶ Python 約50 秒



# 使用Feather 格式儲存

```
import pandas as pd  
df = pd.read_csv('accident.csv')  
df.to_feather('accident.feather')
```

- 載入時使用pd.read\_feather()。
- 適合反覆分析使用，避免每次都要重讀CSV。



## 步驟三：資料疊圖分析

- 使用行政區shp 檔與事故資料進行空間疊圖。
- Python 可大量自動化疊圖，效能佳。
- 步驟說明：
  - ❶ 讀取行政區shapefile。
  - ❷ 將事故資料轉為GeoDataFrame。
  - ❸ 執行空間join，統計各區事故數。



# 程式碼：疊圖與空間Join

```
import geopandas as gpd
accidents = gpd.read_file('accidents.geojson')
districts = gpd.read_file('行政區.shp')
joined = gpd.sjoin(accidents, districts, how='left')
```

- 使用geopandas 執行空間相交。
- 可統計各縣市或鄉鎮的事故數。



# Accident Analysis.ipynb使用說明(1/4)

```
1 file_list=os.listdir('A1')
2 df1=pd.DataFrame()
3 for file in file_list:
4     data=pd.read_csv('A1/'+file,header=None)
5     df1=pd.concat([df1,data],axis=0)
```

```
1 df1.iloc[0,[0,3,7,...,46,47]]
2 d_A1={}
3 for i in range(len(df1)):
4     key=str(df1.iloc[i,2])+...
5     value=[df1.iloc[i,0],...,df1.iloc[i,47]]
6     ...
```



# Accident Analysis.ipynb使用說明(1/4)

- 將csv檔案儲存在A1、A2資料夾
- 0,3,7,...,46,47表示篩選出的欄位，對應：發生年度，發生時間，天候名稱，光線名稱，事故類型及型態大類別名稱，事故類型及型態子類別名稱，肇因研判大類別名稱-主要，肇因研判子類別名稱-主要，死亡受傷人數，當事者區分-類別-大類別名稱-車種，當事者屬-性-別名稱，當事者事故發生時年齡，當事者行動狀態大類別名稱，當事者行動狀態子類別名稱，車輛撞擊部位大類別名稱-最初，車輛撞擊部位子類別名稱-最初，肇因研判大類別名稱-個別，肇因研判子類別名稱-個別
- 以「發生日期\_發生時間\_發生地點」為key，分離出案件主要肇事者



# Accident Analysis.ipynb使用說明(2/4)

```
1 with open('A1.json','w',encoding='utf-8') as f:  
2     json.dump(d_A1,f)  
3     ...
```

```
1 with open('A1.json','r',encoding='utf-8') as f:  
2     d_A1=json.load(f)  
3     ...
```

- 第一次讀取(約數十分鐘)後儲存為json檔
- 第二次讀取json檔即可(約數秒鐘)



# 資料整合與篩選

- 所有資料匯入QGIS 篩選
  - ▶ 期間：108年1月1日起至113年7月18日止
  - ▶ A1類事故9799件，牽涉24929人
  - ▶ A2類事故4541938件，牽涉2017044人
- 確認路段範圍並進行環域分析，導出csv 檔再轉成json檔
- 工具：QGIS、Python





# 資料分析與數據處理

- 計算傷亡人數/件數及各因子事故數
- 以圖表呈現全國/台2線數據
- 統計差異性：卡方檢定
- 工具：Python, Excel



# Accident Analysis.ipynb使用說明(3/4)

```
1 time_analysis(d)
2 year_analysis(d)
3 casulty_analysis(d)
4 casulty_year_analysis(d)
5 factor_death(factor,d)
6 factor_injury(factor,d)
7 age_group_analysis(d)
8 merge_count(*d)
```



# Accident Analysis.ipynb使用說明(3/4)

- `time_analysis`分析事故發生時間(0至23時)件數分布
- `year_analysis`分析各年度事故發生件數
- `casualty_analysis`分析事故死傷人數
- `casualty_year_analysis`分析各年度事故死傷人數
- `factor_death`分析各事故原因死亡人數
- `factor_death`分析各事故原因受傷人數
- `age_group_analysis`分析各事故主要肇事者年齡層
- `merge_count`合併字典計數，用於A1、A2件數或人數合併



# Accident Analysis.ipynb使用說明(4/4)

```
1 one_factor(factor,d)
2 two_factors(factor1,factor2,d)
3 one_factor_person(factor1,factor2,d)
4 two_factors_person(factor1,factor2a,factor2b,d):
5 consolidate_vehicle_types(data)
6 simplify_accident_types(data)
7 consolidate_vehicle_status(data)
8 consolidate_accidents_cause(data)
9 factor_diff_significance(d1,d2,print_oe=False)
```



# Accident Analysis.ipynb使用說明(4/4)

- `one_factor`分析單一屬性，例如光線狀態、第一當事者等
- `two_factors`交叉分析兩個屬性，例如各光線狀態下各車種事故數
- `one_factor_person`分析事故類型、事故肇因或當事者動作
- `two_factors_person`分析單一屬性下事故類型、事故肇因或當事者動作，例如各光線狀態下事故肇因
- `consolidate_vehicle_types`簡化車種
- `simplify_accident_types`簡化事故類型
- `consolidate_vehicle_status`簡化當事者動作
- `consolidate_accidents_cause`簡化肇因
- `factor_diff_significance`比較相同屬性在不同地域之分布是否有顯著差異
  - ▶ 使用卡方(chi-squared)檢定， $\alpha = 0.05$
  - ▶ 卡方統計值( $\chi^2$ )受樣本數大小影響，一律以百分比數比較，並剔除小於5%之項目



# 簡報生成

製作成果：<https://gamma.app/docs/2-m7ml41lffc7rnc1>



# 報告撰寫

- 文字潤飾
- 給定人設、明確標的與補充數據
- 工具：ChatGPT, Claude(可合作，一個生、一個改)
- 範例如第56頁所示，原理如第30頁所示



# 撰寫報告書的指令範例(1/3)

## Claude指令

您是交通安全專家，請協助我撰寫報告書。

## 現有段落

台2線事故中最常見原因為未注意安全，佔46.58%；違規行為次之，佔27.03%。與全國數據比對，卡方檢定顯示差異具顯著性（ $p\text{-value} = 0.0034$ ）。





# 撰寫報告書的指令範例(2/3)

## 數據

未注意安全	667,099	38.98%	138	54.76%	15.79%
違規行為	828,655	48.41%	80	31.75%	-16.67%
危險駕駛	35,596	2.08%	0	0.00%	-2.08%
行人違規	20,960	1.22%	2	0.79%	-0.43%
不當變換車道或方向	68,038	3.98%	4	1.59%	-2.39%
搶快、超速或未依規定減速	55,040	3.22%	16	6.35%	3.13%
車輛問題	5,762	0.34%	3	1.19%	0.85%
逆向行駛	22,655	1.32%	6	2.38%	1.06%
動物	4,760	0.28%	1	0.40%	0.12%
道路、環境、天候因素	1,946	0.11%	2	0.79%	0.68%
光線不佳	1,066	0.06%	0	0.00%	-0.06%



# 撰寫報告書的指令範例(3/3)

## 需求

更新段落中的數據，並檢視數據是否有可以再闡述的地方，以公文用語改寫為一段文字。無須提出改善建議，並避免"進一步分析發現，台2線範圍內"之類的多餘字句，也不用講件數；不須談低於全國平均的部分，除非該項占的比例不小。p-value的呈現方式維持（p-value = 原數據）



# 撰寫報告書的輸出範例

- 依據最新統計資料，茲將事故原因分析結果修正如下：  
台2線事故中最常見原因為未注意安全，佔54.76%，高於全國平均15.79個百分點；違規行為次之，佔31.75%。搶快、超速或未依規定減速佔6.35%，高於全國平均3.13個百分點。逆向行駛佔2.38%，高於全國平均1.06個百分點。車輛問題佔1.19%，高於全國平均0.85個百分點。道路、環境、天候因素佔0.79%，高於全國平均0.68個百分點。動物因素佔0.40%，略高於全國平均0.12個百分點。與全國數據比對，卡方檢定顯示差異具顯著性（ $p\text{-value} = 0.0034$ ）。本分析結果顯示，台2線事故原因分布與全國趨勢有明顯差異，尤以未注意安全、搶快超速及逆向行駛等原因比例較高為甚。



# 工程師的角色

- 清楚界定問題
- 判斷資料合理性
- 決定分析方法（統計學方法、須分析之因子等）



# Table of Contents

## 1 工具簡介

- GIS
- 程式設計前置作業
- 程式設計工具
- 爬蟲
- 排版語言 $\text{\LaTeX}$ 與MiKTeX
- AI工具

## 2 事故資料彙整

- 道安資訊查詢網
- 原始資料處理
- Accident Analysis.ipynb使用說明
- 報告彙整

## 3 學習資源



# 電子圖書資源與課程資源

可至各公立圖書館查閱相關電子書：

- 新北市立圖書館
- 臺北市立圖書館
- 桃園市立圖書館
- 新竹市立圖書館
- 國家圖書館
- 國立公共資訊圖書館
- 國立臺灣圖書館

課程資源

- 生成式人工智慧導論

電子書平臺：

- HyRead
- udn數位讀書館
- 華藝線上圖書館
- 台灣雲端書庫

其他

- 語言學習
- 期刊、雜誌、報紙
- 電影欣賞



# 感謝聆聽，敬請指教



龍天立老師：「問題認識的深度就是解決問題的程度！」

圖片來源：賴勇成老師

