再現性

保持再現性 與 維護性

Reproduceable & Maintainable

技巧 1：

1. 完成人機協作分析後，讓AI撰寫 分析/繪圖 程式
2. 把該分析程式（通常是Python程式）下載保存

下一次要做相同分析的時候，把數據與程式一起上傳給AI

給AI的提示詞 (Prompt)：

以 demo範例程式.py 為藍本，幫我測試該程式能否處理 demo.csv檔案，如果可以，請幫我執行，盡量不要改動原程式，給我執行的結果；如果無法執行，幫我編修程式，編修完後告訴我修改的地方，也給我執行完的結果

demo.csv → demo200.csv

Gemini 再現ChatGPT結果(不同的數據)

PRO

demo200
CSV

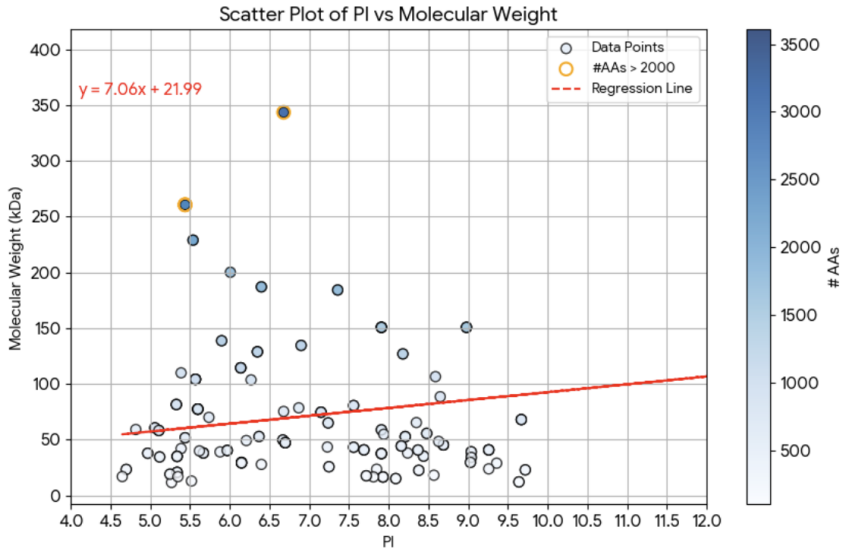
demo範例程式
PY

以 demo範例程式.py 為藍本，幫我測試該程式能否處理 demo.csv檔案，如果可以，請幫我執行，盡量不要改動原程式，給我執行的結果；如果無法執行，幫我編修程式，編修完後告訴我修改的地方，也給我執行完的結果

Examining the Code

問問 Gemini 3

＋ 工具 Pro



1. 執行結果摘要

- **原始資料筆數**：200 筆
- **有效資料筆數**：182 筆（已剔除含有負數或異常值的 18 筆資料）
- **計算出的線性回歸公式**：$y = 7.06x + 21.99$

2. 視覺化圖表

下圖顯示了蛋白質等電點 (PI) 與分子量 (Molecular Weight) 的關係。

- **藍色深淺**：代表氨基酸數量 (# AAs)。
- **橘色圈選**：代表氨基酸數量超過 2000 的特殊點（在此有效數據集中可能沒有，或數量極少）。
- **紅色虛線**：代表線性回歸趨勢線。

demo.csv → demo200.csv

Claude 再現ChatGPT結果 (不同的數據)

測試並執行 demo 程式處理 CSV 檔案

demo200.csv
CSV

demo範例程式.py
54 lines
PY

以 demo範例程式.py 為藍本，幫我測試該程式能否處理 demo.csv檔案，如果可以，請幫我執行，盡量不要改動原程式，給我執行的結果；如果無法執行，幫我編修程式，編修完後告訴我修改的地方，也給我執行完的結果

評估程式相容性，準備測試或修改執行。

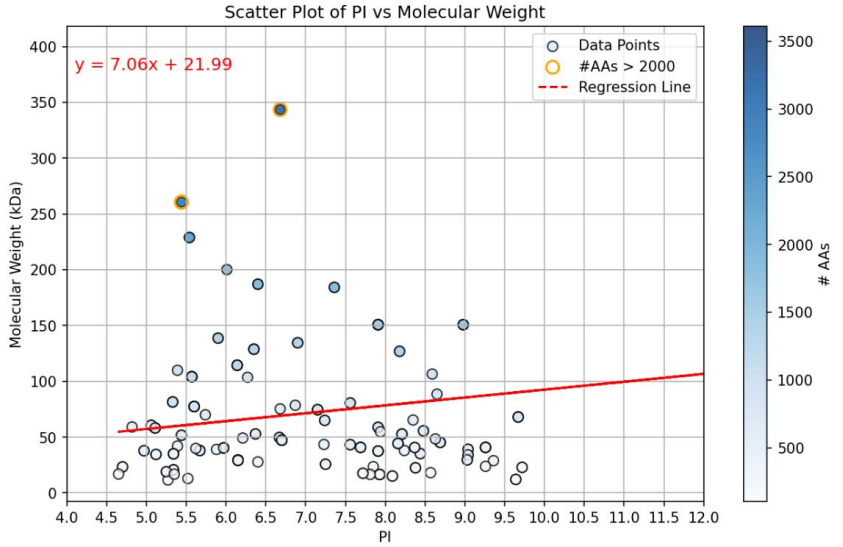
我來先查看上傳的檔案，確認 CSV 檔案的格式和內容，然後測試程式是否能正常執行。

2 steps

Reply...

Opus 4.5

Result plot · PNG



技巧 2：

1. 完成人機協作分析後，讓AI撰寫 分析/繪圖 程式
2. 再由AI撰寫該程式的詳細說明與執行的範例

給AI的提示詞 (Prompt)：

撰寫該程式的詳細說明與執行的範例，幫我輸出成 HTML 格式

demo範例程式.py — 詳細說明與執行範例

本文件說明此程式如何讀取 demo200.csv、清洗資料、建立一元線性回歸（以 calc. pI 預測 MW [kDa]），並繪製散點圖（以 # AAs 作為顏色映射）與回歸線。

輸入檔：demo200.csv　程式：demo範例程式.py　主要套件：pandas / matplotlib / numpy / scikit-learn

1. 程式功能概覽

- 讀取 CSV：demo200.csv
- 資料清洗：移除 MW [kDa]、calc. pI、# AAs 中小於等於 0 的資料列
- 建立模型：使用 LinearRegression 進行一元線性回歸
- 輸出結果：印出回歸公式（y = ax + b）
- 視覺化：繪製散點圖、以 # AAs 做顏色映射、標註 # AAs > 2000 的點、疊加回歸線與公式文字

輸出圖表的視覺重點：

- 點的顏色代表 # AAs（並附 colorbar）
- # AAs > 2000 的點用橘色空心圈特別標示
- 紅色虛線為回歸線，圖上會顯示回歸公式



3.3 終端機輸出（範例）

程式會印出回歸公式，例如：

```
計算出的回歸公式為: y = 7.06x + 21.99
```

注意：實際數值會依 demo200.csv 內容而變動。若你在無 GUI 的環境（例如遠端主機）執行，plt.show() 可能不會彈出視窗，但程式仍可正常完成運算。

4. 常見錯誤與排除

現象 / 錯誤訊息	原因	建議處理方式
FileNotFoundError: ... 'demo200.csv'	工作目錄找不到 CSV（相對路徑讀檔）	先 cd 到 CSV 所在資料夾再執行；或把 CSV 放到執行目錄
圖視窗不顯示（或只有 warning）	無 GUI backend（遠端/無桌面環境常見）	可改成 plt.savefig(...) 存檔；或在支援 GUI 的環境執行
回歸線不合理 / 圖看起來怪	資料含大量極端值或欄位有誤	先檢查 CSV 欄名與單位，或增加更嚴格的資料清洗規則

5. 你可以如何延伸（選讀）

- 改用 CSV 絕對路徑或接受命令列參數，避免工作目錄問題。
- 將圖表改為輸出檔案（PNG/PDF），方便在伺服器環境使用。
- 加入 R^2、MAE 等評估指標，讓回歸品質更可量化。

以上為延伸建議，不影響目前程式的既有功能。

1. 生成式AI vs 代理式AI
2. 生成式AI在研究（數據分析、資料視覺化）的應用
3. 上下文
4. Agentic AI 的程式
5. 上機操作

LLM 知道世界怎麼運作，但不會自動記得現在要做什麼



訓練階段

它先學會世界知識

它有什麼？

- 某個時間點以前的大量資料
- 基本知識、常識與語言能力
- 對世界的一般理解
- 回答問題的基本能力

它沒有什麼？

- 你現在要它做什麼
- 你目前進行到哪裡
- 這次任務哪些資訊最重要

訓練讓它「懂很多」，
但不等於它知道「此刻的任務」。



推理階段

它每次回答，都像重新醒來一次

它需要什麼？

- 現在要完成什麼
- 眼前有哪些重要線索
- 已經完成了哪些部分
- 下一步可能是什麼

這些線索叫什麼？

- Context（這一次醒來時，眼前被放上的線索與文件）
- 沒有這些線索，它只知道世界，不知道當下

推理時能不能做好，
取決於它這次看到了哪些線索。

Slide from 長庚大學AI Innovation & Support Center 朱柏諺老師

Context (上下文) 例子：幫寵物取名字

「幫我的寵物取一個名字。」

➔ AI 知道很多寵物名字，但它不知道你的寵物是什麼樣子，所以只能回答：

「可以叫 Lucky、Mochi、Coco……」

但如果前面的對話是：
你： 我養了一隻橘色的貓。
你： 牠很胖、很愛吃，而且個性很懶。
你： 我希望名字有一點台灣味。
你： 幫牠取一個名字。

➔ 「可以叫『貢丸』，圓滾滾又愛吃，也很有台灣味。」

「**橘色、公貓、胖、愛吃、懶、台灣味**」就是這次對話的 **Context（上下文）**

上下文不是無限的

LLM 面前有一張有限大小的桌子。

只有放在桌面上的資料，它才能一起參考。

當對話內容超過模型的 **Context Window** 時，系統就必須對舊內容做處理，例如捨棄、摘要、壓縮，或只保留較重要的資訊。

主流模型的上下文長度

模型	Context Window	粗估可容納中文字
OpenAI GPT-5.6 Sol	1,050,000 tokens	**約 70 萬字**
Claude Opus 4.6	1,000,000 tokens	**約 67 萬字**
Gemini 3.1 Pro	1,048,576 tokens	**約 70 萬字**
Grok 4.6	500,000 tokens	**約 33 萬字**

上下文越長，不代表 AI 就一定記得越好。

如果把大量無關的對話、文件、搜尋結果全部丟進去，真正重要的資訊反而可能被淹沒。

註：Token 與中文字並非固定比例，實際數量會受到文字內容、標點、英文、數字及模型 tokenizer 影響，僅為概略估算。

壓縮上下文的例子

原始上下文

實驗目標：PCR 擴增特定基因片段
Template DNA：50 ng
Forward primer：0.5 µM
Reverse primer：0.5 µM
$MgCl_2$：1.5 mM
dNTP：200 µM each
Polymerase：1 U
Initial denaturation：95°C，3 min
Denaturation：95°C，30 sec
Annealing：60°C，30 sec
Extension：72°C，30 sec
Cycle number：35 cycles
Final extension：72°C，5 min
……

Context 壓縮後

使用者正在進行特定基因的 PCR 實驗。
Template DNA 為 50 ng，primer concentration 為 0.5 µM，$MgCl_2$ 1.5 mM，dNTP 200 µM each，polymerase 1 U。
PCR 共進行 35 cycles，denaturation 為 95°C，extension 為 72°C。
使用者後續希望整理 PCR protocol、分析 PCR 產物並進行 gel electrophoresis。

使用者：

「請把我們前面討論的條件整理成完整 PCR protocol。」

Initial denaturation：95°C，3 min
Denaturation：95°C，30 sec
Annealing：55°C，30 sec
Extension：72°C，30 sec
Repeat for 35 cycles
Final extension：72°C，5 min

Context 壓縮的風險，不只是「忘記」，而是 LLM 可能用自己的知識把遺失的資訊補回來。

AI 忘掉一個參數並不可怕，可怕的是它忘了之後，還幫你補了一個看起來很合理的參數。

1. 生成式AI vs 代理式AI
2. 生成式AI在研究（數據分析、資料視覺化）的應用
3. 上下文
4. Agentic AI 的程式
5. 上機操作

Agentic AI工具

ChatGPT/Codex https://openai.com/zh-Hant/codex/

Claude Code Desktop https://code.claude.com/docs/zh-TW/desktop

Google Antigravity https://antigravity.google/download/ ←→ Google Antigravity IDE

Opencode https://opencode.ai/zht

……

Opencode


新增工作階段
想問什麼都可以，/ 可使用命令，@ 可加入上下文...
Build
Nemotron 3 Ultra Free
opencode
無 Git


搜尋模型
OpenCode Zen
Big Pickle 免費
Ling 3.0 Flash Fin Free 免費
MiMo V2.5 Free 免費
Muse Spark 1.2 Free 免費
Nemotron 3 Ultra Free 免費
Nemotron 3.5 Lightning Free 免費



模型	開發者 / 國家	簡短介紹
Big Pickle	**未公開**	OpenCode Zen 的「隱身模型（stealth model）」，偏向**程式開發、推理與 Agent 任務**，Context 約 200K。官方刻意不公布真正的模型來源。(Pi Coding Agent)
Ling 3.0 Flash Fin Free	CN **中國大陸** - 螞蟻集團 / InclusionAI（百靈）	124B MoE、實際啟用約 5.1B，主打**速度、低成本、Agent 工作流程、工具呼叫**。屬於螞蟻集團的 Ling / 百靈系列。(Hugging Face)
MiMo V2.5 Free	CN **中國大陸** - 小米 Xiaomi	小米 MiMo 團隊的大模型，主打**推理、Agent、程式能力及多模態**，正式版本支援文字、圖片、影音與聲音，Context 可達 1M。(MiMo)
Muse Spark 1.2 Free	US **美國** - Meta	Meta Superintelligence Labs 的模型，特別強調**Coding、Agent、多代理協作與長時間軟體工程任務**；Muse Spark 1.2 也是 Meta「Muse Code」coding agent 的核心模型。(Reuters)
Nemotron 3 Ultra Free	US **美國** - NVIDIA	NVIDIA Nemotron 系列大型旗艦模型，**550B 總參數、55B active**，偏重**高階推理、Agent、工具使用與複雜任務**，Context 最長約 1M。(NVIDIA)
Nemotron 3.5 Lightning Free	US **美國** - NVIDIA	NVIDIA 新的高效率 Agent 模型，**30B 總參數、3B active**，比 Ultra 小很多，特色是**速度快、成本低、適合 Coding Agent / 子代理與大量工具呼叫**。(NVIDIA NIM)