

RStudio × 生成式 AI

在統計程式設計中使用 ChatGPT、Claude、Gemini 與 Posit Assistant

115 學年度第 1 學期 | 程式設計與統計軟體

第一週補充講義

課程教師：吳漢銘

更新日期：2026 年 9 月 10 日

核心觀念： AI 是需要驗證的協作助理，不是答案機器。學生必須能閱讀、解釋、修改並驗證 AI 產生的程式。

學習目標

- 熟悉 RStudio 的 Source、Console、Environment、Help、Terminal 與 Project。
- 區分瀏覽器聊天、R 套件 API、Posit Assistant、Terminal coding agent 四種模式。
- 使用 ellmer 連接 OpenAI、Anthropic Claude 與 Google Gemini。
- 使用 chattr 在 RStudio 中聊天，並理解 context engineering。
- 在 RStudio Terminal 使用 Claude Code、Gemini CLI 與 OpenAI Codex CLI。
- 理解 API key、成本、隱私、權限、hallucination、可重現性與 AI 使用揭露。

RStudio 中使用 AI 的四條路

模式	資料流	適合用途
A. 瀏覽器聊天	RStudio ↔ copy/paste ↔ ChatGPT / Claude / Gemini	初學、概念、單段除錯
B. R 套件 API	R → ellmer / chattr → LLM API	程式化、批次、可重現
C. IDE 內建助理	RStudio → Posit Assistant → LLM	利用 live R session
D. Terminal Agent	Terminal → Claude Code / Gemini CLI / Codex	讀專案、改檔、執行命令、測試

教學順序：建議先 A → B → C，再示範 D。先理解 prompt、context、verification，再進入 agentic coding。

1. RStudio 快速導覽

區域	功能	AI 協作重點
Source	撰寫 .R/.Rmd/.qmd	AI 程式先閱讀再執行
Console	互動執行 R	快速驗證小片段
Environment	查看物件/資料框	提供 str() 等 context
Help / Packages	查官方文件	AI 回答仍需核對
Terminal	shell / Git / CLI	AI coding agent 入口
Project	界定工作目錄	限制 agent 的專案範圍

重要：不要把 API key 寫進 .R 程式，也不要將含金鑰的 .Renvron 上傳 GitHub。

2. 方法 A：瀏覽器直接使用 ChatGPT、Claude、Gemini

這是最簡單的方式，但 AI 不會自動知道 RStudio 目前有哪些物件、套件版本或檔案；必須主動提供 context。

優點	限制
----	----

不需安裝 R 套件；介面直覺	copy/paste 容易遺漏 context
適合概念解釋、程式閱讀、逐步除錯	AI 不知道 live R session
可上傳程式或資料（依服務/方案）	不可上傳密碼、API key 或敏感資料

2.1 好的統計程式 prompt

你是我的 R 程式助教。請使用 base R 與 ggplot2。

資料：mtcars

任務：

1. 以 mpg 為 y、wt 為 x 畫散佈圖。
2. 加入線性迴歸線與 95% confidence band。
3. cyl 視為 factor。
4. 每段程式加簡短註解。
5. 先說明分析步驟，再給程式。
6. 最後列出 3 個我應自行驗證的地方。

2.2 除錯時應提供的資訊

- 完整錯誤訊息，而不是只說「不能跑」。
- 出錯的程式片段。
- 預期結果與實際結果。
- 必要時提供 sessionInfo()、str(data)、head(data)。
- 敏感資料改用模擬資料或去識別化 toy example。

```
sessionInfo()
str(mtcars)
head(mtcars)
```

把出錯那一行 + 完整 error message 一起交給 AI

3. 方法 B：用 R 套件直接呼叫 LLM

若希望 AI 成為 R workflow 的一部分，可透過 API 從 R 呼叫模型。2026 年建議優先認識 ellmer；它提供一致的 R 介面連接多家 LLM，並支援 streaming、tool/function calling 與 structured data extraction。

3.1 ellmer：一套介面連接三大模型

```
install.packages("ellmer")
library(ellmer)
```

服務	ellmer 函數	常見憑證
OpenAI	chat_openai()	OPENAI_API_KEY
Anthropic Claude	chat_anthropic()	ANTHROPIC_API_KEY
Google Gemini	chat_google_gemini()	GEMINI_API_KEY / GOOGLE_API_KEY

Posit AI	chat_posit()	Posit AI 設定
本機 Ollama	chat_ollama()	本機服務

3.2 API key：放在環境變數，不要硬編碼

```
install.packages("usethis")
usethis::edit_r_environ()

# 在 .Renviron 中加入；不要把真實 key 放到作業或 GitHub
OPENAI_API_KEY=你的_OpenAI_API_key
ANTHROPIC_API_KEY=你的_Anthropic_API_key
GEMINI_API_KEY=你的_Gemini_API_key

# 重新啟動 RStudio 後，只確認是否存在，不要印出 key
nzchar(Sys.getenv("OPENAI_API_KEY"))
nzchar(Sys.getenv("ANTHROPIC_API_KEY"))
nzchar(Sys.getenv("GEMINI_API_KEY"))
```

3.3 三家模型的最小範例

```
library(ellmer)

chat <- chat_openai()
chat$chat("請用 R 解釋 sample mean 的 sampling distribution, 並給 n=30 的模擬範例。")

chat <- chat_anthropic()
chat$chat("請檢查這段 R 程式的邏輯；先指出 bug, 不要直接重寫全部。")

chat <- chat_google_gemini()
chat$chat("請用初學者能理解的方式比較 data.frame、matrix 與 tibble。")
```

API 與聊天網站訂閱不同： ChatGPT、Claude 或 Gemini 的網頁方案，不必然等於已有 API 額度；API 常有獨立的憑證與計費。

4. chattr：在 RStudio 裡開一個 LLM 對話介面

chattr 是 RStudio / Positron 的 LLM 介面，可從 R console 使用，也可開啟 Shiny Gadget。新版 chattr 透過 ellmer 連接模型，並可加入目前 R 環境中的資料框名稱/結構與工作目錄資訊。

```
install.packages("chattr")
library(chattr)
chattr_app()

# 例如把 Claude 設為 chattr 預設連線
options(.chattr_chat = ellmer::chat_anthropic())

# 若希望每次啟動 R 都保留設定
usethis::edit_r_profile()
```

可在 RStudio : Tools → Modify Keyboard Shortcuts → 搜尋「Open Chat」，替 chattr Addin 指定快捷鍵。

隱私提醒： 工具自動加入 R session context 後通常更精準，但也可能把更多內容送往外部模型；使用前先判斷資料是否允許外傳。

5. btw : context engineering 的觀念

btw 是 Posit 開發的 R/LLM context 工具箱，可協助蒐集 R session、資料結構、套件與文件資訊，再交給 ChatGPT、Claude 或其他 LLM。對初學者最重要的是學會「只提供足夠而相關的 context」。

- 先提供資料結構，再決定是否需要資料內容。
- 把套件版本、錯誤訊息與函數文件納入 context。
- 避免把 token、密碼、個資、未公開研究資料送出。
- AI 回答品質 = 模型能力 + prompt 品質 + context 品質 + 驗證品質。

6. 方法 C : RStudio 內建 Posit Assistant

截至 2026 年，Posit Assistant 已進入新版 RStudio。它是針對資料科學工作流程設計的 AI 助理，可在 IDE 中聊天並取得 live R session context；Next Edit Suggestions (NES) 則依目前程式與 session context 建議下一個編輯動作。

功能	用途
Chat / Agent	寫程式、改程式、除錯、EDA、理解專案
Live R session context	理解已載入套件、資料框與分析狀態
Next Edit Suggestions	依上下文建議下一個修改
Project context	理解多檔案分析專案
Model provider	依 RStudio / Assistant 版本設定 Posit AI 或支援的 provider

6.1 啟用步驟

- 使用支援 Posit Assistant 的新版 RStudio；官方文件目前要求 RStudio Desktop/Server 2026.04.0 以上。
- 在 RStudio 工具列按 Posit Assistant 圖示，依提示安裝 Assistant 元件。
- 設定模型 provider。Posit AI 是官方建議的簡易方式；新版文件亦列出 Anthropic、OpenAI、Google Gemini、Amazon Bedrock 等 provider。
- 完成後可直接在 Assistant pane 對話。

6.2 為什麼它適合統計課？

範例提問：

「請查看目前 R session 中的 mtcars。」

不要先寫完整分析。
先告訴我每個變數的型態、可能的資料品質問題，
再建議三個適合初學者的 EDA 問題。」

瀏覽器 ChatGPT 若沒有收到資料內容，無法知道目前 Environment；Posit Assistant 的優勢正是 IDE / session context。

不要混淆： Posit Assistant 是 AI coding/data-science assistant；Posit AI 是可為 Assistant 提供模型存取的服務。兩者不是同一個名稱。

7. 方法 D：在 RStudio Terminal 使用 AI coding agents

RStudio 的 Terminal 不只可執行 Git，也可直接啟動 AI coding agent。這類工具比一般 chat 權限高：可能讀取多個檔案、修改程式、建立檔案、執行 Rscript 或測試。因此必須先建立 Project、使用 Git，並確認每一步權限。

安全原則： 初學者先讓 agent『讀取與解釋』，再讓它『提出 patch』，最後才允許『執行命令』。不要一開始就給最高自動執行權限。

7.1 Claude Code

Claude Code 是 Anthropic 的 terminal coding agent。官方文件支援 Windows 10+（可透過 WSL 或 Git for Windows 等環境）、macOS 與 Linux。

```
# 典型 npm 安裝
npm install -g @anthropic-ai/claude-code

# 確認/診斷
claude doctor

# 在 RStudio Project 的 Terminal 啟動
claude

# 常用 CLI 形式
claude "explain this R project"
claude -p "review analysis.R for possible bugs"
claude -c

# 課堂 prompt
請先閱讀本專案，但不要修改任何檔案。
找出所有 .R 檔，說明每個檔案的用途，
再指出你需要我確認的三個問題。
```

Claude Code 適合多檔案專案、重構、測試與長一點的 coding workflow。後續課程可再介紹 CLAUDE.md、plan-review-code-test 與 Skills。

7.2 Gemini CLI

Google 的 Gemini CLI 也可直接從 RStudio Terminal 執行。官方教學目前要求 Node.js 20+。

```
npm install -g @google/gemini-cli
gemini --version
gemini
```

首次啟動可依畫面選擇 Google 帳號登入、Gemini API key 或 Vertex AI 等驗證方式。Gemini CLI 可使用 GEMINI.md 提供專案層級指引。

```
# GEMINI.md 的簡單概念
# Project instructions
- This is an R statistics course project.
- Prefer base R and ggplot2.
- Do not modify data/raw/.
- Explain statistical assumptions before changing analysis code.
- Ask before installing packages.
```

7.3 OpenAI Codex CLI : ChatGPT 生態的 Terminal coding agent

若要在 RStudio Terminal 使用 OpenAI 的 coding agent，可使用 Codex CLI。新版 Codex 可使用 ChatGPT 帳號登入；也可依環境採 API 憑證。

```
npm install -g @openai/codex

# 啟動
codex

# 診斷（新版 Windows 支援亦提供）
codex doctor
```

Codex CLI 適合在 IDE 的 Terminal 中讀取專案、修改程式、執行測試。它與『在瀏覽器 ChatGPT 貼一段 R code』是不同層級的工作模式。

8. 三種 Terminal Agent 怎麼選？

工具	主要入口	優勢	本課程定位
Claude Code	claude	成熟的 terminal agent、專案工作流	後半學期重點
Gemini CLI	gemini	Google 生態、CLI、GEMINI.md	比較與延伸
OpenAI Codex CLI	codex	ChatGPT/OpenAI coding agent 生態	比較與延伸

第一週不需要學生熟練三種 CLI；目標是知道 RStudio Terminal 可以成為 AI agent 的入口。真正的 agentic workflow 建議等學生具備基本 R 程式能力、Git 與除錯能力後再深入。

9. 同一題，四種 AI 工作模式的比較

任務：找出 mtcars 中 mpg 與 wt 的關係，畫圖並建立簡單線性迴歸。

方法	學生要做什麼	AI 知道什麼	風險
瀏覽器 chat	貼 prompt / 程式 / 資料摘要	只知道你提供的內容	context 不足
ellmer	由 R 程式送 prompt	你程式明確送出的內容	API 成本/資料外傳
Posit Assistant	在 IDE 中詢問	可取得較豐富 session context	需留意 context 權限
Terminal agent	在 project 中交辦任務	可讀專案檔並可能執行命令	誤改檔案/執行高權限命令

10. 統計課最重要的 AI 驗證流程

AI 可以產生語法正確但統計上錯誤的程式。統計軟體課的驗證不能只看『程式有沒有跑完』。

層次	檢查問題	例子
語法	程式能否執行？	函數名、括號、套件
資料	資料型態/缺失/編碼正確嗎？	factor 被當 numeric
統計	方法與假設合理嗎？	把 paired data 當 independent
數值	結果是否可用另一方法驗證？	手算、summary、模擬
圖形	圖是否誤導？	座標軸、群組、尺度
可重現	別人能否重跑？	set.seed、package version、資料來源
課程口訣：先自己想 → 再問 AI → 自己驗證 → 能向別人解釋。若不能解釋 AI 的程式，就還不能把它當成自己的答案。		

11. 常見錯誤用法

- 把整份作業貼給 AI，直接交回輸出而不閱讀。
- AI 說某個 R 函數存在就相信，沒有用 ?function 或 package 文件核對。
- 把 ChatGPT/Claude/Gemini 網頁訂閱與 API key/費用混為一談。
- 把 API key 寫進 script、Quarto 或 GitHub repo。
- 讓 coding agent 在不相關的大資料夾啟動，而不是在乾淨的 RStudio Project。
- 允許 agent 執行安裝、刪除、覆寫等命令前沒有看 command。
- 把真實個資、學生資料、未公開研究資料直接傳給外部模型。
- 只驗證 p-value 或圖表外觀，沒有檢查資料處理與統計假設。

12. 課堂實作：30–40 分鐘

實作 A：瀏覽器 AI

- 在 RStudio 建立新的 Project：week01-ai-lab。
- 建立 lab01.R，載入 mtcars。
- 自行先寫出 summary(mtcars) 與 plot(mpg ~ wt)。
- 再把『想改成 ggplot2 並加入 regression line』交給任一聊天 AI。
- 逐行解釋 AI 程式；至少修改一個圖形設定。

實作 B：ellmer（有 API 憑證者）

```
library(ellmer)
chat <- chat_openai() # 或 chat_anthropic() / chat_google_gemini()
chat$chat("請出一題 mtcars 的 R 練習題，只給題目，不要給答案。")
```

實作 C：Posit Assistant

- 開啟 Posit Assistant。
- 要求它描述目前 R session。
- 要求它只提出 EDA 計畫，不先產生完整 code。
- 比較它與瀏覽器 AI 回答的差異：哪一個更知道你的工作環境？

實作 D：Terminal 示範（教師操作）

```
# 在 Project 的 Terminal 中擇一示範
claude
gemini
codex
```

第一個 prompt 一律採 read-only 思維：

請先閱讀目前專案，不要修改檔案，也不要執行任何命令。

告訴我你看到哪些 R 檔案，以及你認為這個專案要做什麼。

13. 建議的 AI 使用自我規範

可作為第一週作業「AI 使用自我規範」的參考框架：

- 我會先自行定義問題，再使用 AI。
- 我不會提交自己無法解釋的 AI 程式。
- 我會核對統計假設、函數文件與重要數值。
- 我會保留重要 prompt / AI 協作紀錄。
- 我不會把 API key、密碼、個資或未授權資料提供給 AI。
- 我會遵守課程對 AI 使用的規定；考試禁用 AI 時不使用任何 AI 工具。

- 若作業或專題大量使用 AI，我會適當揭露工具、用途與人工驗證方式。

14. 教師可強調的三個觀念

觀念	一句話
AI literacy	不是會問 ChatGPT，而是會定義問題、提供 context、驗證結果。
Statistical literacy	程式跑得動 ≠ 統計分析正確。
Agent literacy	Agent 能改檔與跑命令；能力越強，權限管理越重要。

15. 官方參考資料（更新至 2026-09-10）

- [RStudio IDE User Guide](#)
- [Posit Assistant – RStudio User Guide](#)
- [Posit Assistant Documentation](#)
- [ellmer – Chat with Large Language Models](#)
- [chattr – Interact with LLMs in RStudio](#)
- [btw – toolkit for connecting R and LLMs](#)
- [Anthropic Claude Code – Getting Started](#)
- [Anthropic Claude Code – CLI Reference](#)
- [Google Gemini CLI – Quick Guide](#)
- [OpenAI – Using Codex with your ChatGPT plan](#)
- [課程網站：115-1 程式設計與統計軟體](#)

版本提醒： AI 工具更新很快。本講義以 2026-09-10 可查得的官方文件為準；實際按鈕名稱、模型名稱、登入方式與 CLI 安裝命令，應以授課當日官方文件為準。