

113年度衛生福利資料科學中心統計軟體推廣課程

2024.05.03 (五) 09:00-16:00 @ 國立成功大學 75201電腦教室

健康資料研究與R軟體

《資料管理基礎班》

劉品崧 統計諮詢暨分析師

花蓮慈濟醫院健康長壽中心

課程大綱

- 課程導覽
 - 課程目的、設計思維
- R軟體入門
 - 名詞定義、下載及安裝、RStudio環境設定
 - R軟體的互動模式、使用套件與函數
- R軟體資料管理實作
 - 資料的來源
 - 資料管理程式撰寫
 - `data.table`套件
 - 讀取、寫出、型態、篩選、排序、修改、歸戶、合併

課程導覽

- 課程目的
- 設計思維

健康資料管理與研究的基石

- 知識
 - 機率分布、資料特性、統計分析、報表解讀 ...
- 觀念
 - 研究設計、流行病學、公共衛生、樣本估算 ...
- 技術
 - 撰寫指令、程式管理、除錯偵測、維護編修 ...
- 實務
 - 獨立研究、團隊合作、工作規劃、執行計畫 ...

ChatGPT

持續學習

今日主軸

個人造化

理解，邏輯，執行

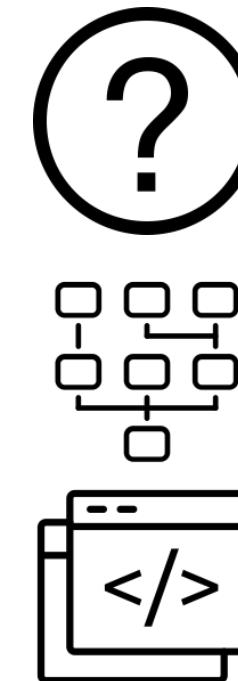
- Object-oriented programming

- 物件導向程式設計
- 目的導向程式設計

- 思考核心理念

- 建構工作流程

- 應用技術實踐



上課原則

- 每50分鐘會休息10分鐘
 - Intake / output
 - Error / warning
 - 補齊程式碼沒跟上的部分
- 隨時可以打斷我問問題
 - 不要太害羞
 - 問題留給我
 - 收穫你帶走

R軟體入門

- 名詞定義
- 下載及安裝
- RStudio環境設定
- R軟體的互動模式
- 使用套件與函數

關於R軟體

1. R是一個軟體（ **software** ）, 具資料管理、統計運算與視覺化等功能
2. R語言（ **R language** ）是泛指與R軟體溝通的代碼（ **code** ）
又稱為指令（ **command** ）或函數（ **function** ）
3. 一支程式（ **program** ）由許多代碼及註解（ **annotation** ）所組成
4. 代碼之間依循語法（ **syntax** ）撰寫組成
5. 物件（ **object** ）會以不同類型（ **class** ）存於工作環境（ **environment** ）
6. 套件（ **package** ）包含一群相關功能的函數（ **function** ）
讓使用者完成目的, 使用者依據需求可以自己安裝（ **install** ）套件

關於RStudio軟體

1. RStudio是整合開發環境
(integrated development environment, IDE)
2. 核心仍然是R，所以必須先安裝R軟體，之後RStudio才可以運作
3. 圖形化使用者介面 (graphical user interface, GUI)
讓初學者比較好上手

關於RTools軟體

1. 只有Windows使用者需要安裝RTools軟體
2. 在R套件的安裝中擔任一個編譯（compile）的角色

各類電腦作業系統下載需求

- R系列皆為免費開源軟體，Google搜尋各個軟體名稱即可找到下載點
- Windows使用者名稱，不可以是中文
- 依據所使用的作業系統不同，你需要安裝以下軟體

作業系統	安裝軟體	R	RTools	Xcode	RStudio
Windows 務必以系統管理員身分執行安裝		 V	 V		 V
Mac 請先確認iOS夠新可以安裝		V		V	V
Linux		V			V

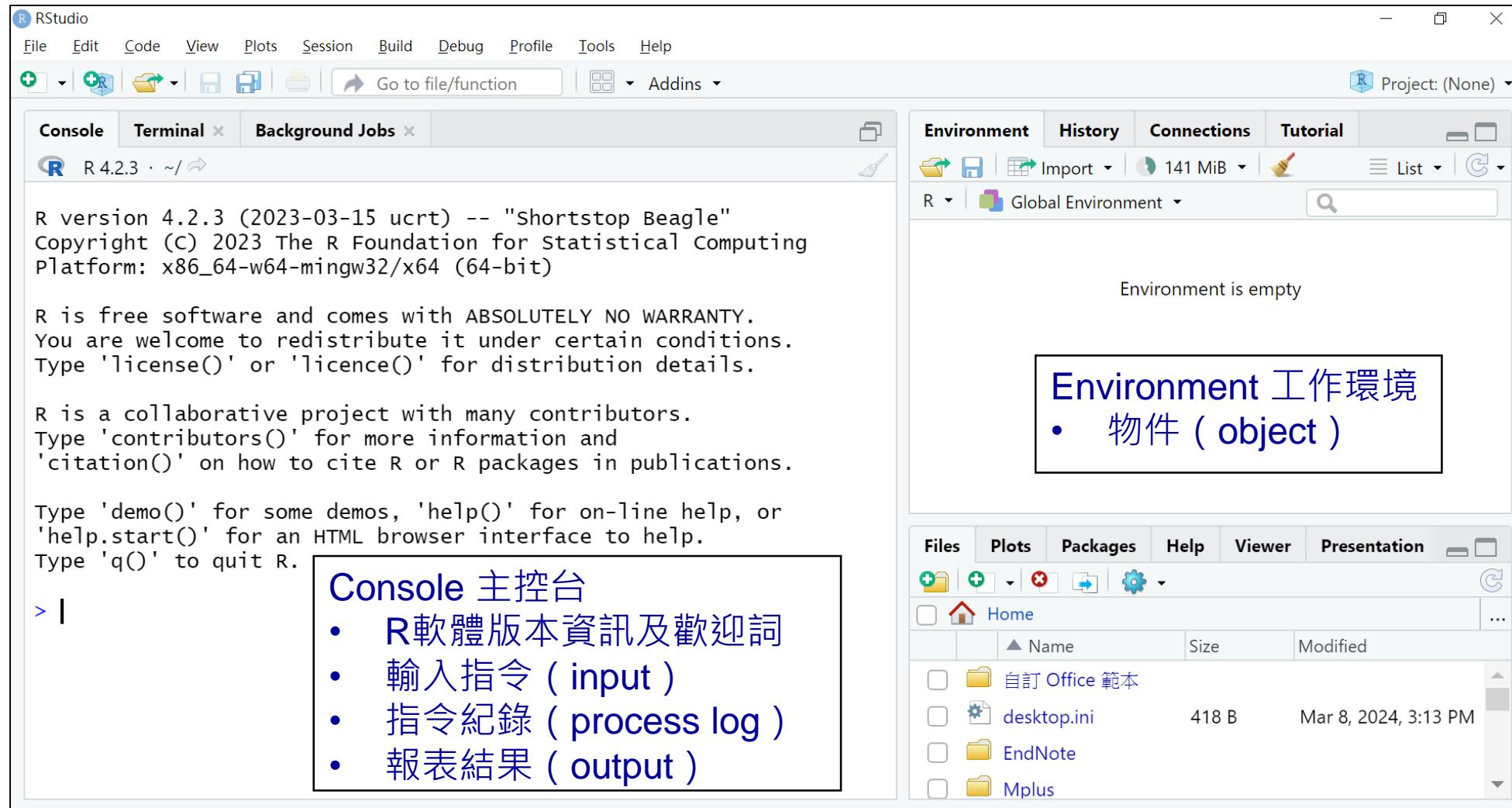
課程使用軟體版本

- R : 4.2.3 版本
 - 進入 <https://cran.r-project.org/bin/windows/base.old/4.2.3/>
 - 點選 R-4.2.3-win.exe
- Rtools : 42 版本
 - 進入 <https://cran.r-project.org/bin/windows/Rtools/rtools42/rtools.html>
 - 點選 Rtools42 installer
- RStudio
 - 進入
https://dailies.rstudio.com/version/2022.07.2+576.pro12/?_gl=1*1bhvt7d*_ga*MTkyNTM4MDg1OS4xNzA5NTM2NzA3*_ga_2C0WZ1JHG0*MTcwOTUzNjcwNy4xLjEuMTcwOTUzNjc0My4wLjAuMA
 - 點選 RStudio-2022.07.2-576.exe

軟體安裝

- Windows使用者全部都要對安裝檔點右鍵「以系統管理員身分執行」
- R
 - 全部使用預設並點選「[下一步](#)」
- RTools
 - 全部使用預設並點選「[Next](#)」
- RStudio
 - 全部使用預設並點選「[下一步](#)」

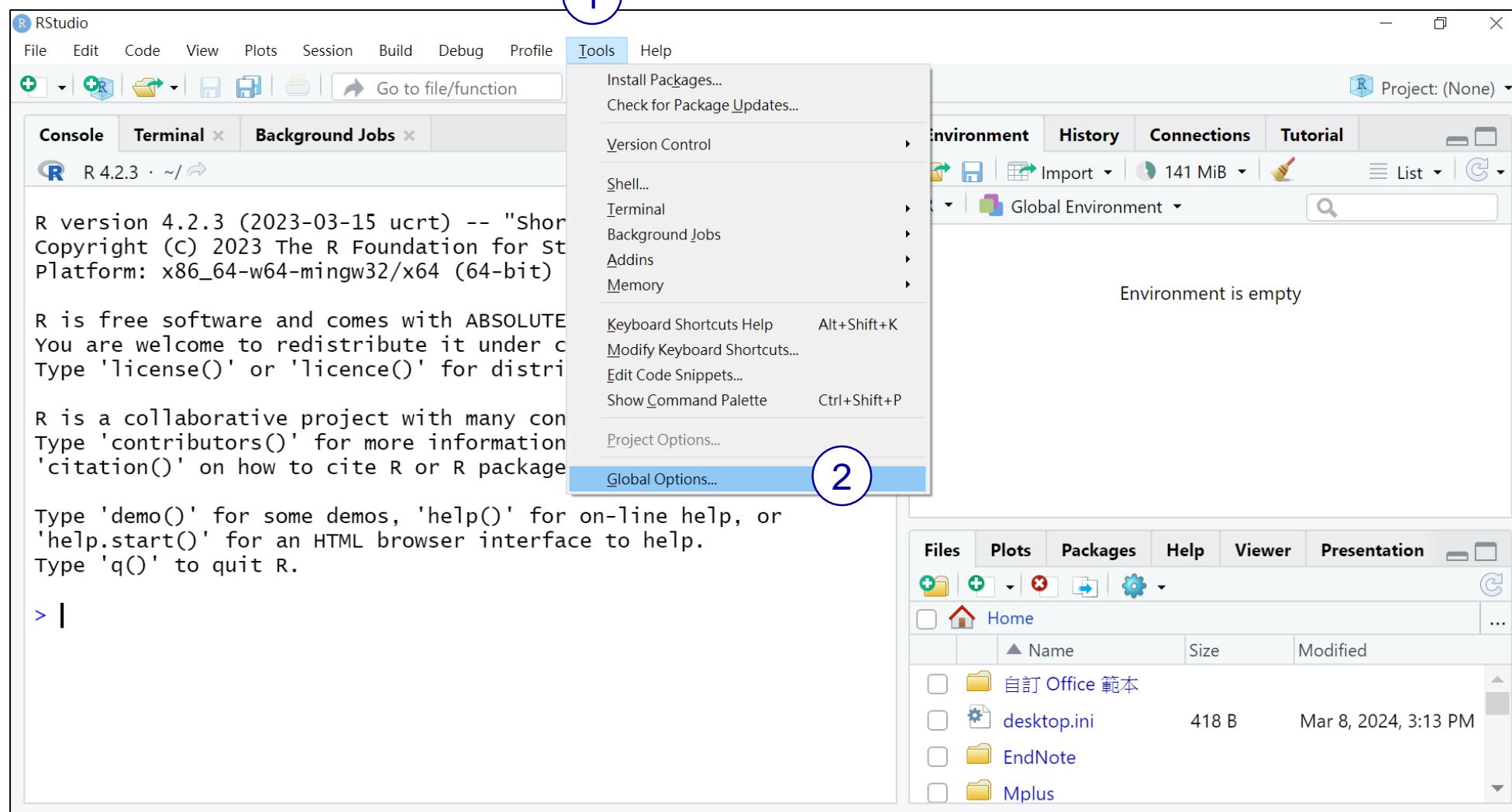
開啟 RStudio



Console 主控台

- R軟體版本資訊及歡迎詞
- 輸入指令 (input)
- 指令紀錄 (process log)
- 報表結果 (output)

大部分的設定都在這裡



R目前使用版本以及變更版本

The screenshot shows the RStudio interface with the 'Session' tab selected in the 'Options' dialog. The R console output is displayed on the left, showing the R version 4.2.3 startup message. The 'General' section of the 'Session' tab is active, displaying settings for R Sessions, Workspace, History, and Other. The 'Basic' tab is selected. The 'Project' sidebar on the right shows an empty 'Tutorial' project.

General

- R version 顯示目前版本
- 按Change 變更版本
- 按「OK」完成設定

調整程式檔案文字編碼

The screenshot shows the RStudio interface with the 'Options' dialog box open. The 'Saving' tab is selected. Under the 'General' section, the 'Default text encoding' is set to 'UTF-8'. A callout box highlights this setting with the text: '→ 按Change改為UTF-8'.

Code
→ Saving
→ 按Change改為UTF-8
按「OK」完成設定

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Console Terminal Backg R 4.2.3 ~/
R version 4.2.3 (2023 Copyright (c) 2023 Th Platform: x86_64-w64-
R is free software an You are welcome to re Type 'license()' or '
R is a collaborative Type 'contributors()' 'citation()' on how t
Type 'demo()' for som 'help.start()' for an Type 'q()' to quit R.
General
Code
Console
Appearance
Pane Layout
Packages
R Markdown
Python
Editing Display Saving Completion Diagnostics
General
Ensure that source files end with newline
Strip trailing horizontal whitespace when saving
Restore last cursor position when opening file
Serialization
Line ending conversion: Platform Native
Default text encoding: UTF-8 Change...
Auto-save
Always save R scripts before sourcing
Automatically save when editor loses focus
When editor is idle: Backup unsaved changes
Idle period: 1000ms
Project: (None)
Tutorial
List C
empty
Power Presentation C
Modified
Mar 8, 2024, 3:13 PM
Mplus

調整外觀，舒適為主

The screenshot shows the RStudio interface with the 'Options' dialog open. The 'Appearance' tab is selected in the left sidebar. The 'RStudio theme:' dropdown is set to 'Modern'. The 'Zoom:' dropdown is set to '100%'. The 'Editor font:' dropdown is set to 'Lucida Console'. The 'Editor font size:' dropdown is set to '10'. The 'Editor theme:' dropdown lists various themes: Ambiance, Chaos, Chrome, Clouds, Clouds Midnight, Cobalt, Crimson Editor, Dawn, Dracula, Dreamweaver, Eclipse, Gob, and Idle Fingers. The main pane displays R code for plotting objects, and the right pane shows the 'Tutorial' and 'Presentation' tabs.

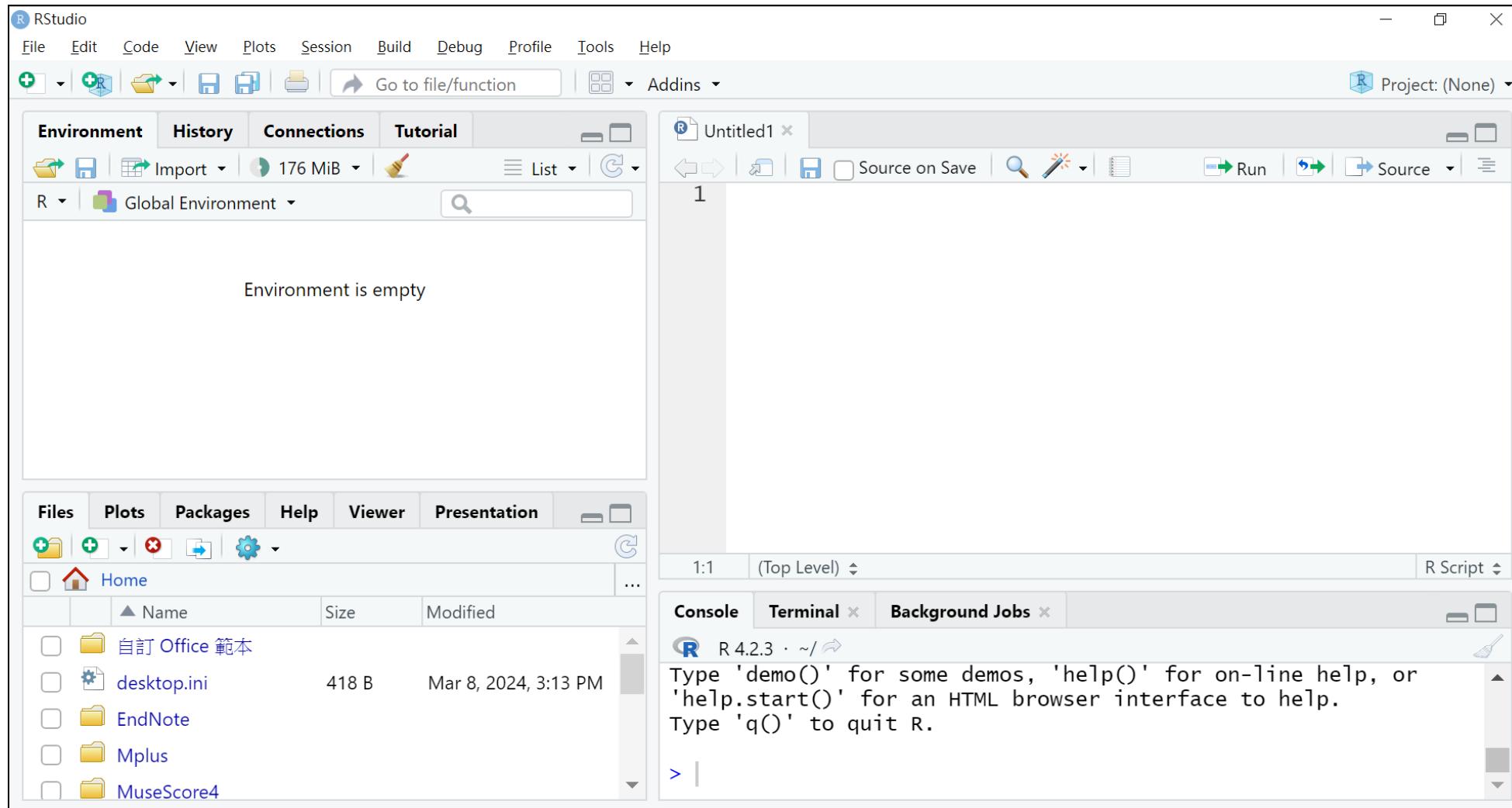
Appearance

- Zoom : 縮放比例
- Font & size : 字體及大小
- Theme : 配色主題

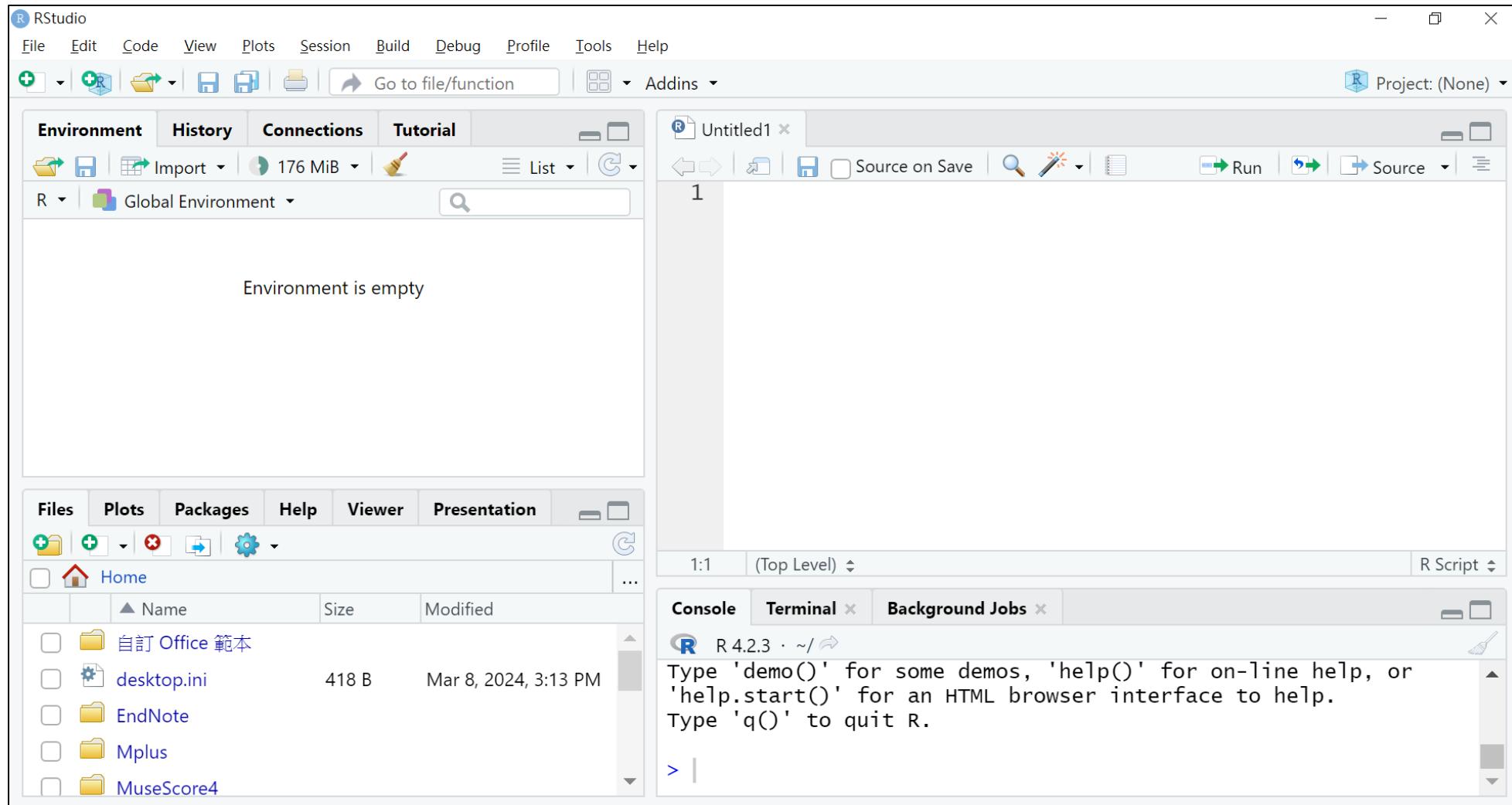
按「OK」完成設定

調整版面配置

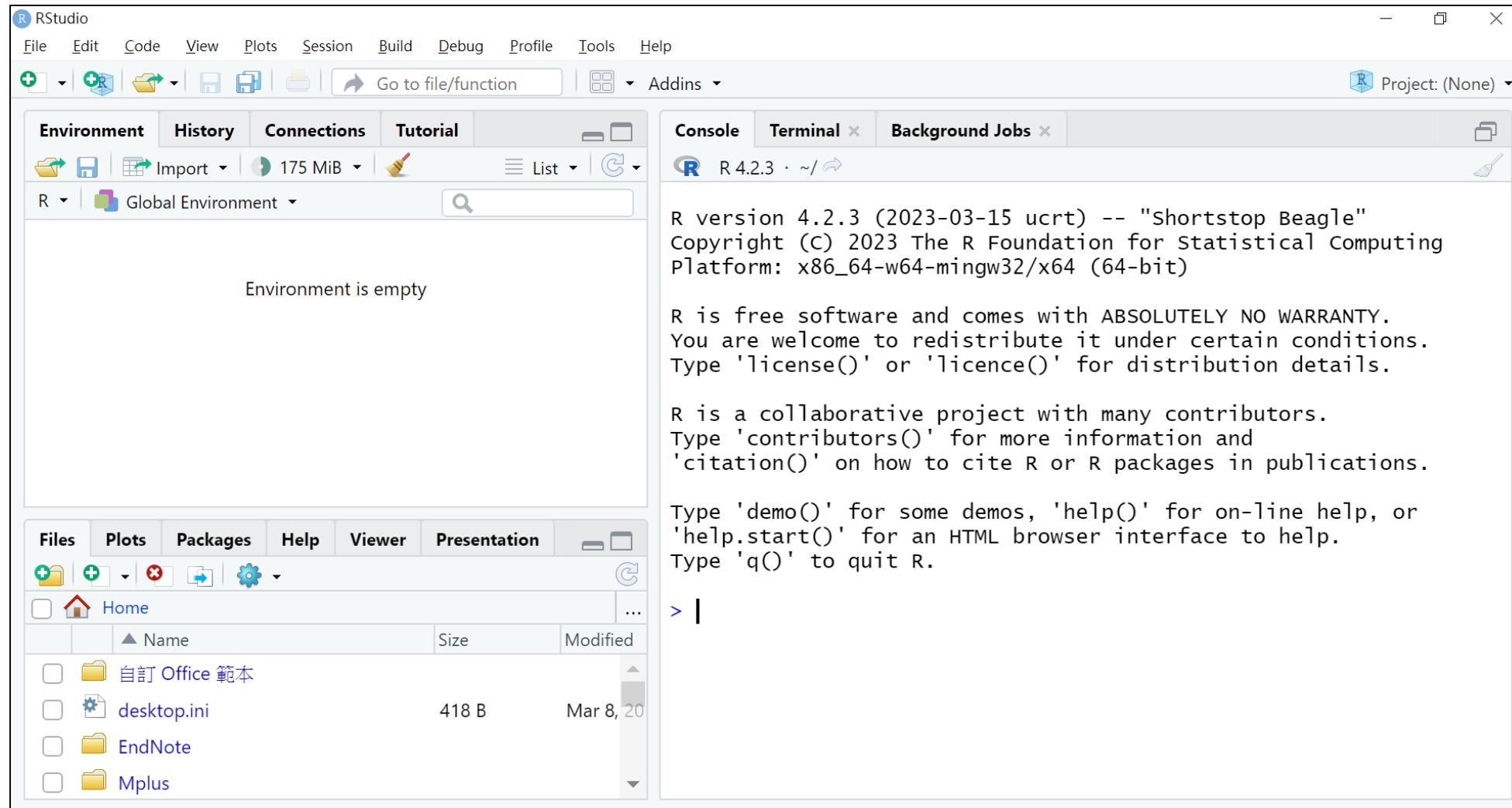
Panel Layout
→ 下拉式選單決定配置
按「OK」完成設定



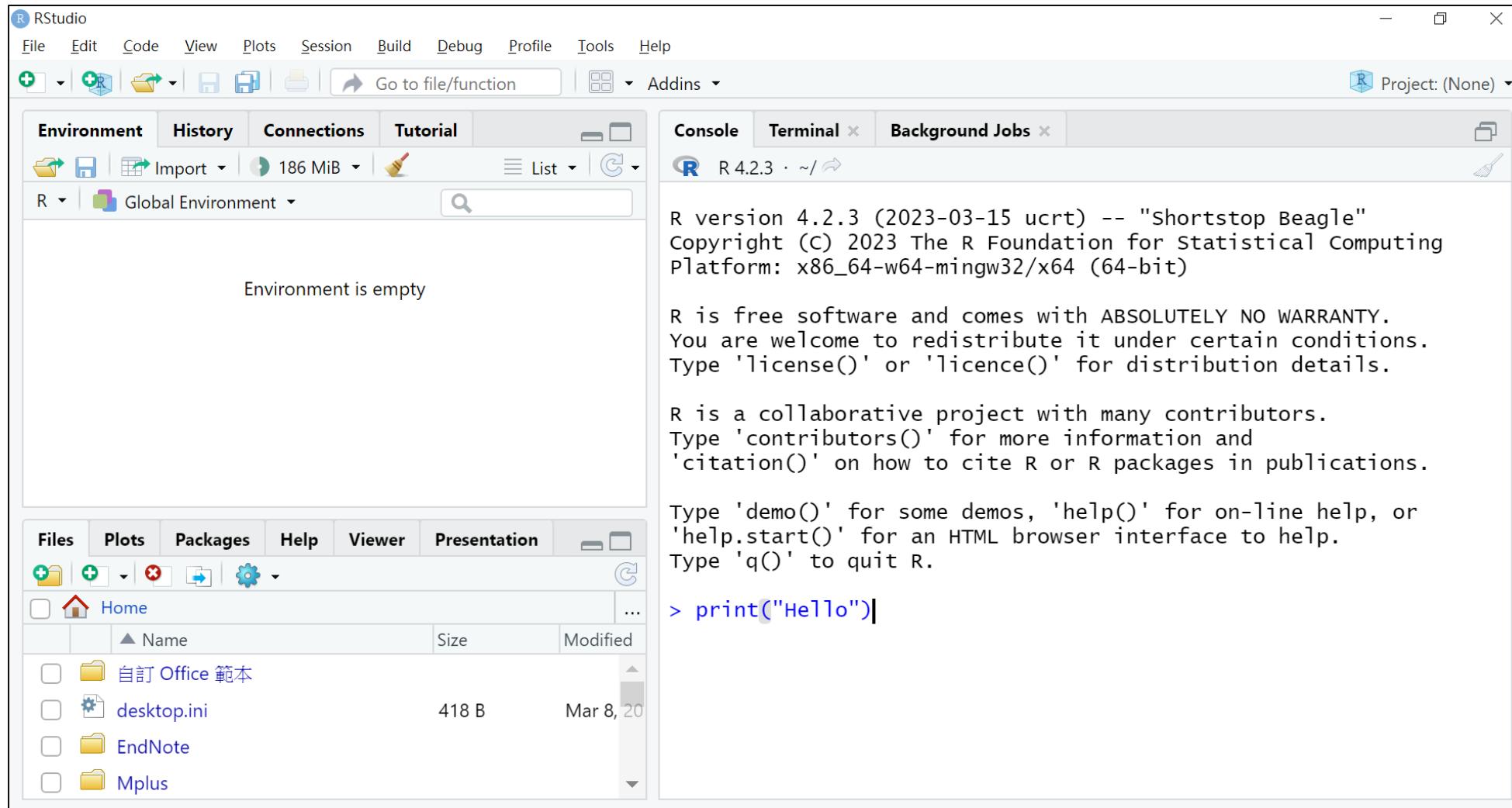
完成設定（隨時可以再次修改）



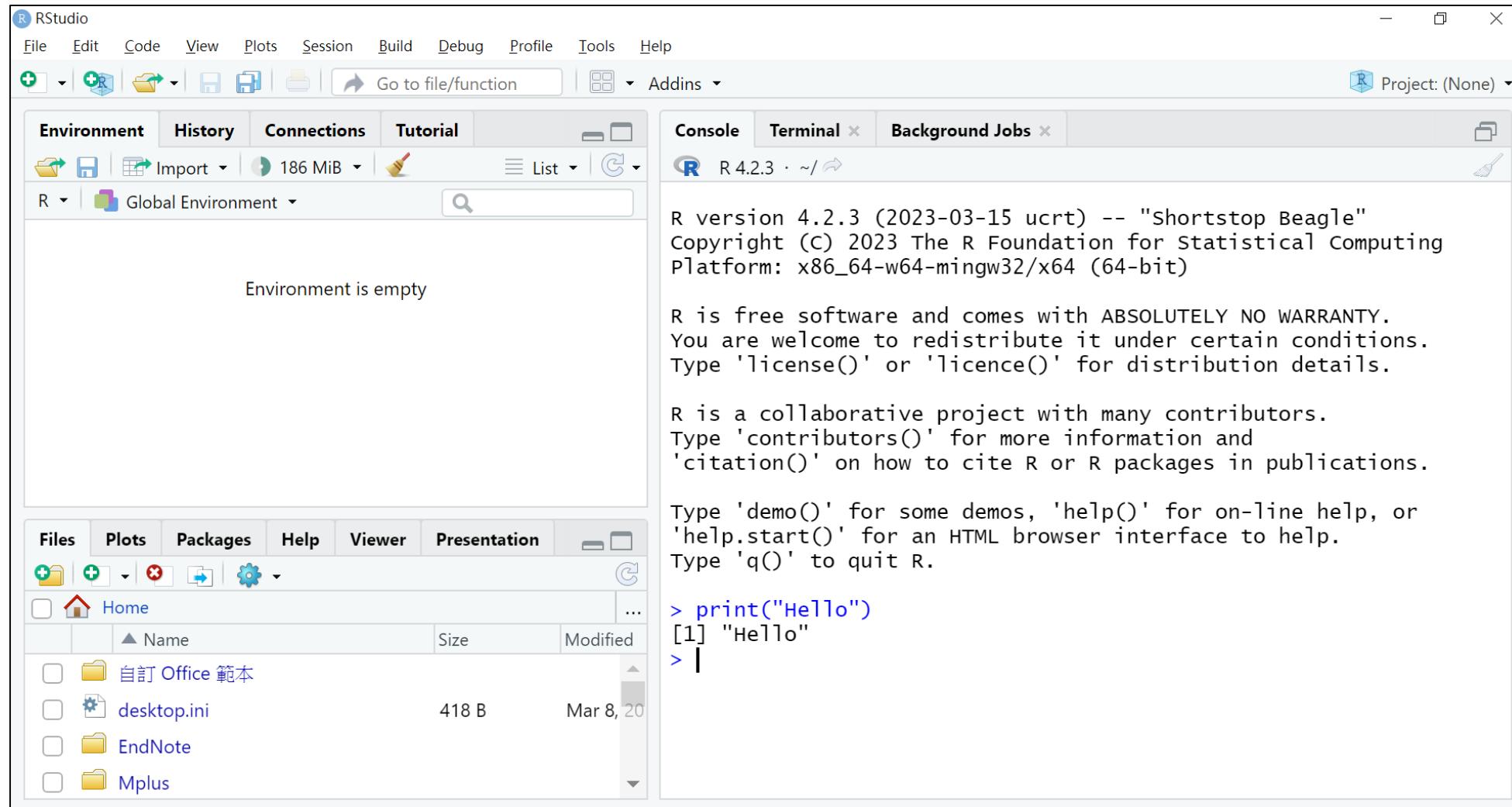
平常直接打開RStudio的樣子



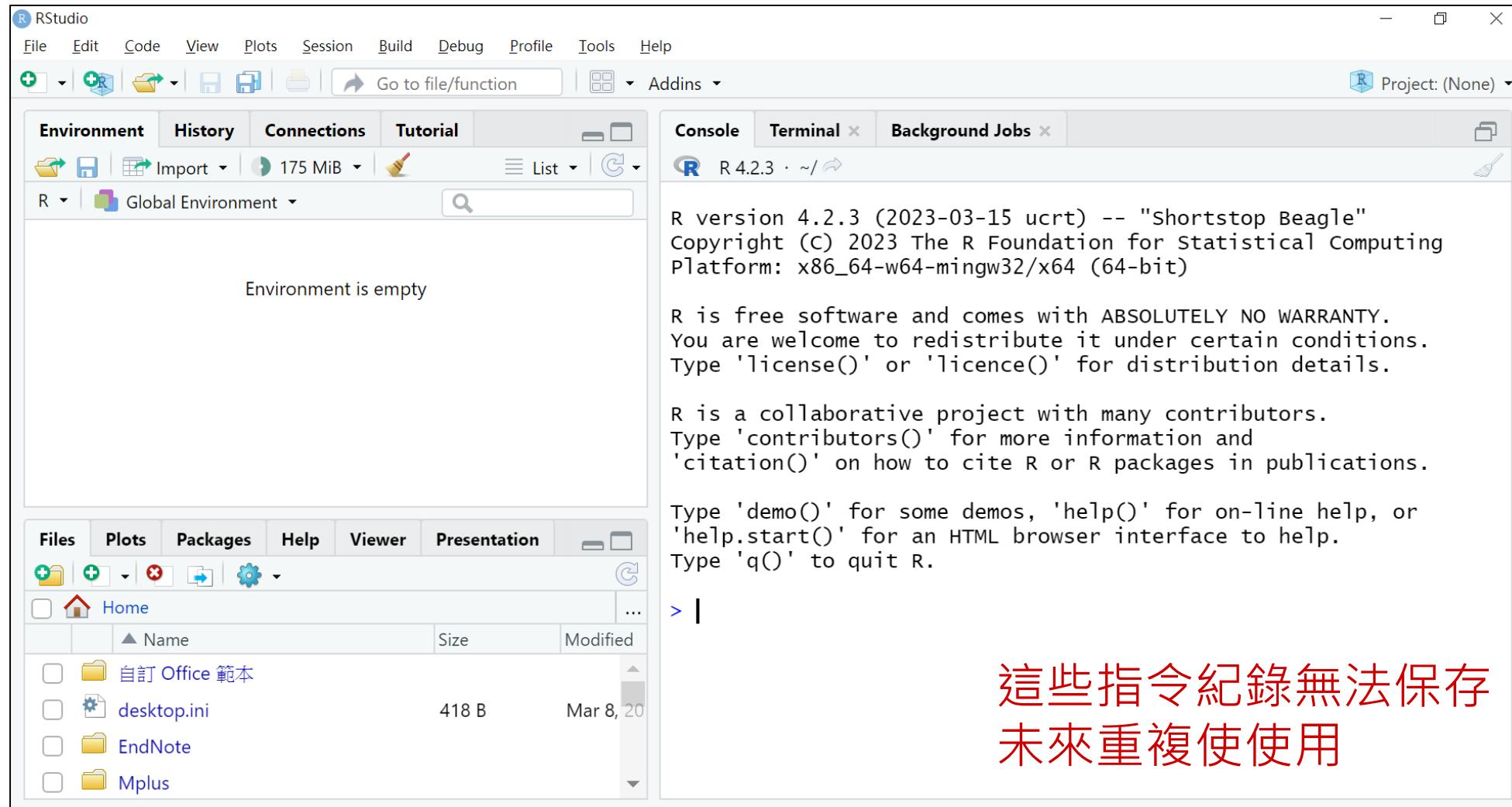
在Console輸入指令 print("Hello") 按Enter



在Console區域得到輸入指令的執行 / 回傳結果

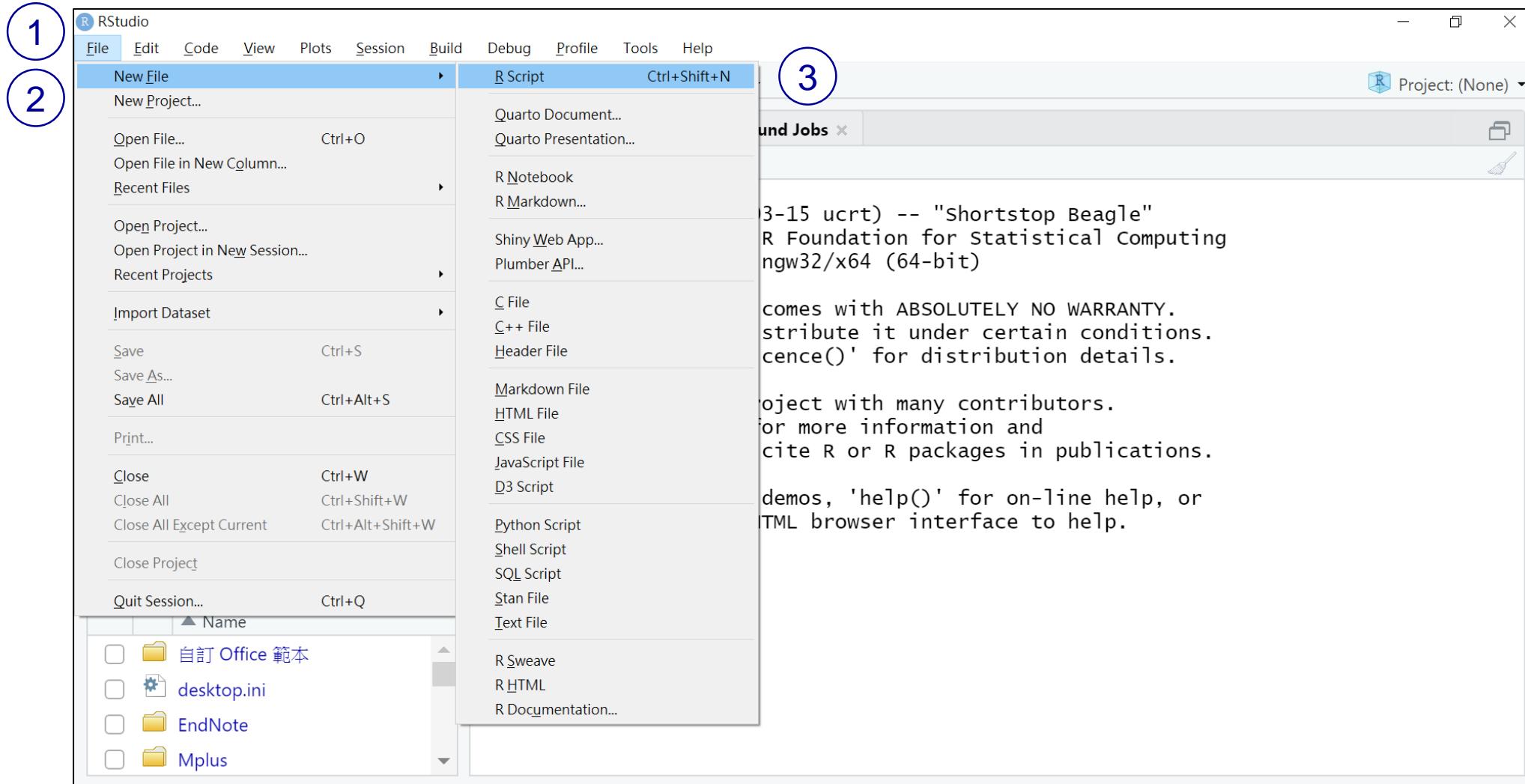


重新開啟RStudio會全部都不見

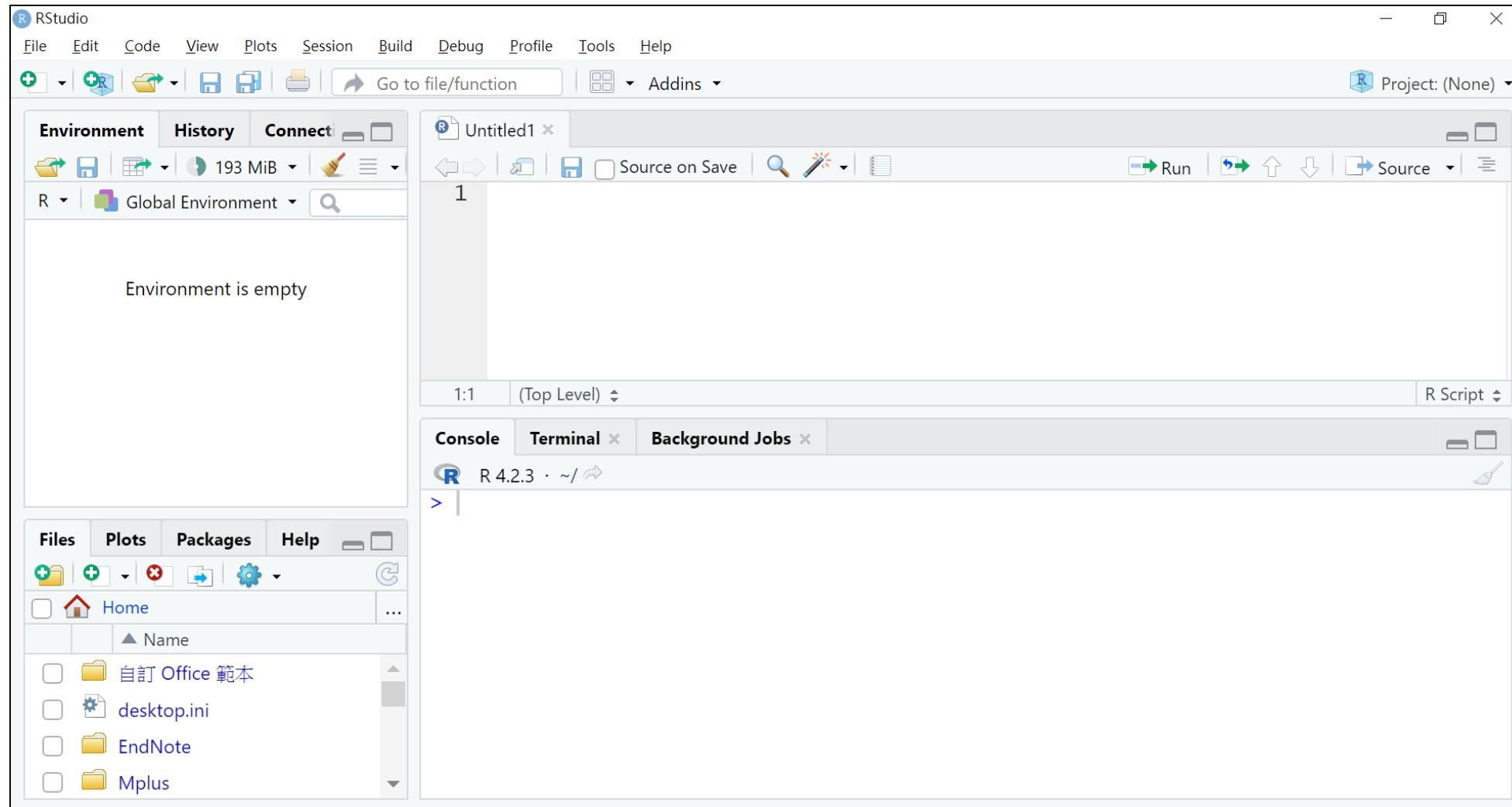


開啟指令稿

滑鼠 : File → New File → R Script
鍵盤 : Ctrl + Shift + N



開啟指令稿完成，開始在指令稿輸入指令



請在 R Script 裡面輸入以下指令

```
# Practice
```

```
# 建立物件 -----
```

```
my_company <- "Hualien Tzu Chi Hospital"
```

```
# 使用函數列印物件 -----
```

```
print(my_company)
```

編輯指令稿內容的原則

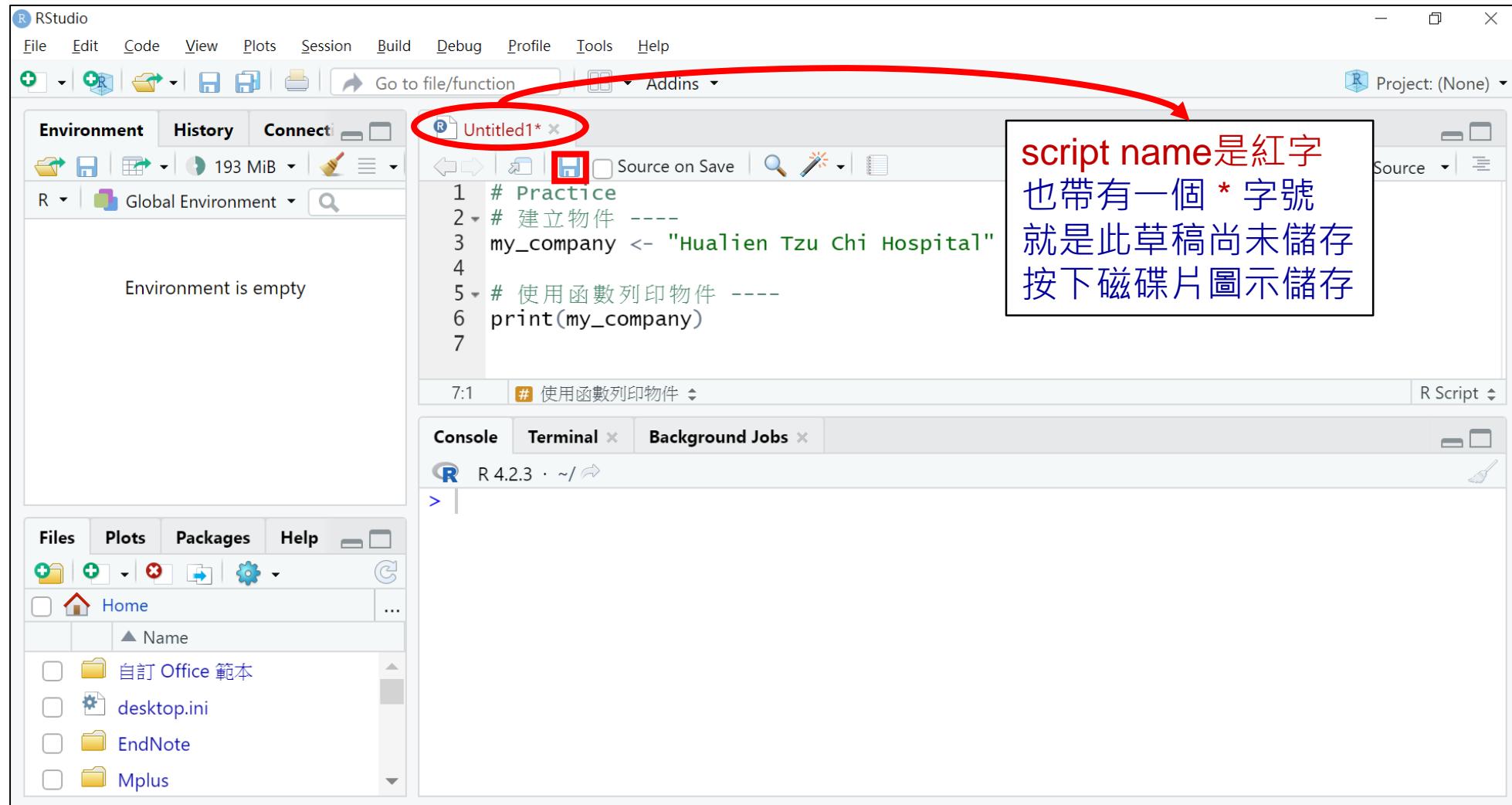
The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons. The main workspace on the left is titled 'Environment' and shows 'Environment is empty'. The central area contains an R script named 'Untitled1.R' with the following content:

```
1 # Practice
2 # 建立物件 ----
3 my_company <- "Hualien Tzu Chi Hospital"
4
5 # 使用函數列印物件 ----
6 print(my_company)
7
```

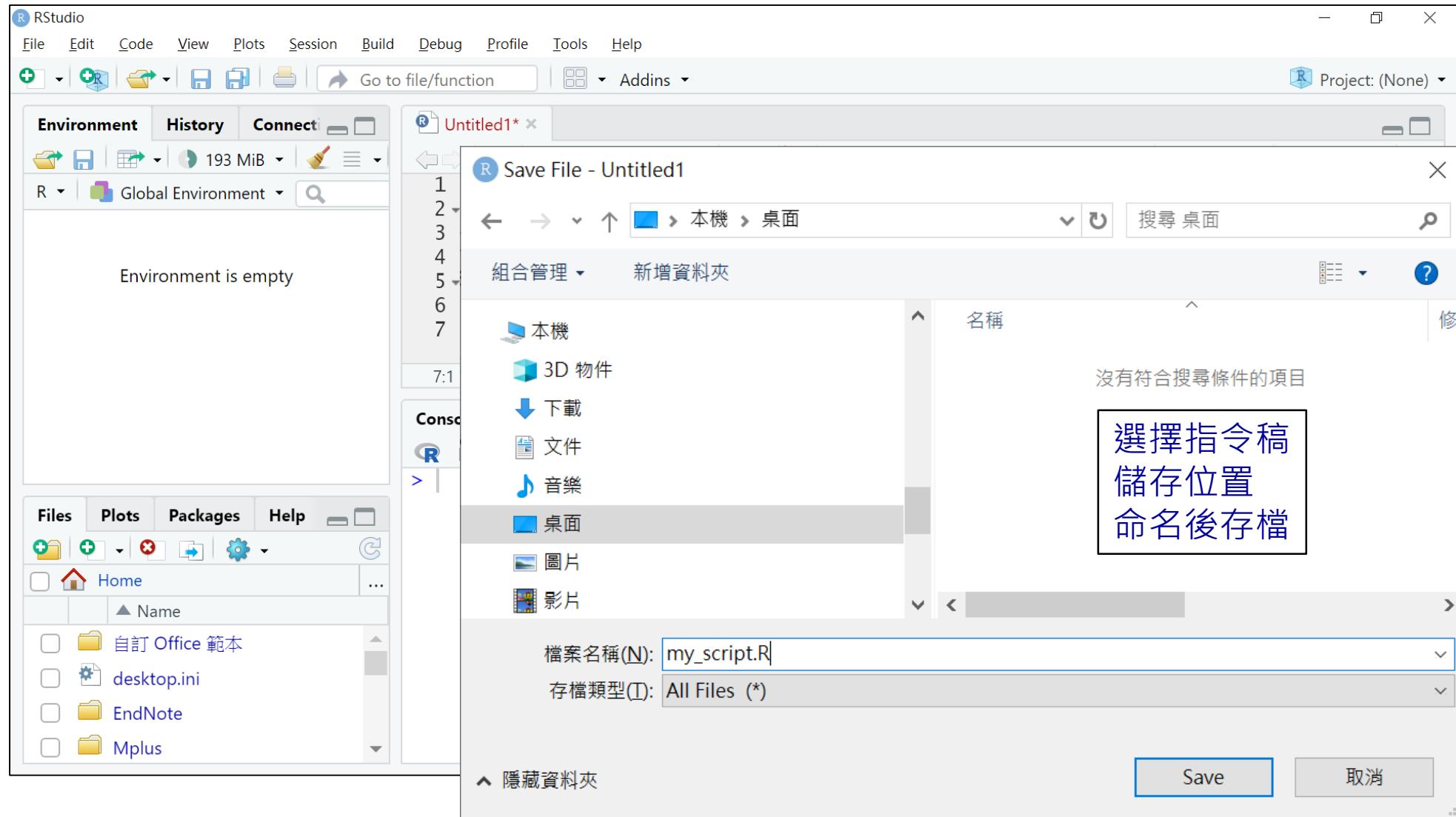
The status bar at the bottom indicates '7:1' and '使用函數列印物件'.

1. 指令使用大小寫英文字是有區別的（**case sensitive**），誤用會造成錯誤
2. 以空白鍵和逗號區隔指令，多個空白鍵將會被視為1個
3. 不同行的程式會分開執行，除非被括號包含，如：
 1. 小括號 () : `function(...)`，用於指定函數的參數內容
 2. 中括號 [] : `object[...]`，用於運用向量處理資料
 3. 大括號 {} : `for (...) { ... }`、`function (...) { ... }`，用來撰寫迴圈或是函數
4. 基本物件操作指令形式
 - 物件名稱 <- 物件內容：指派 (`assign`) 指令為小於 (<) 和減號 (-) 連接組成
快捷鍵為 **Alt + -**

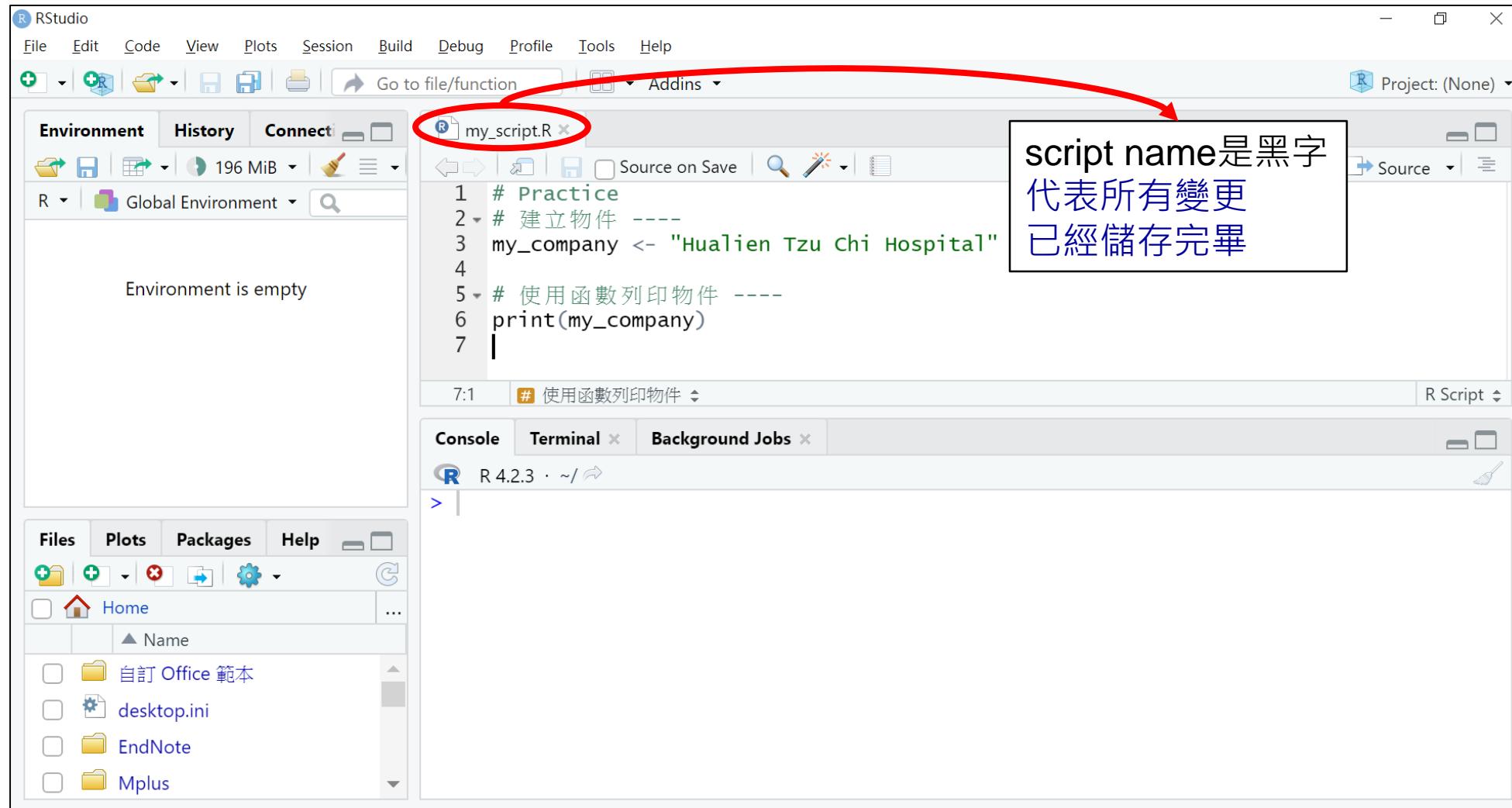
儲存指令稿內容



儲存指令稿內容



儲存指令稿內容



儲存指令稿內容

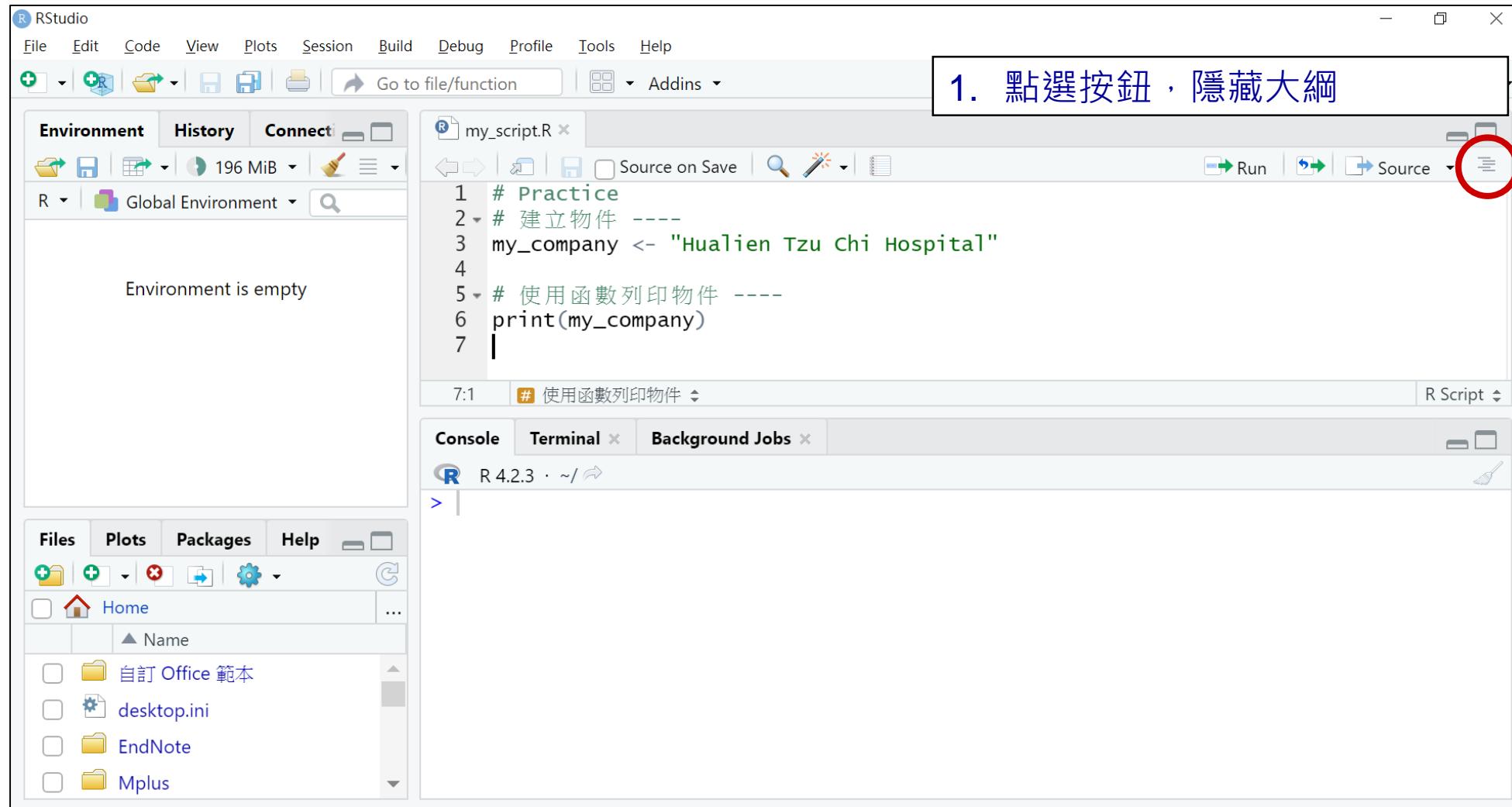
The screenshot shows the RStudio interface. On the left is the Environment pane, which is currently empty. In the center is the Source editor pane, titled "my_script.R", containing the following R code:

```
1 # Practice
2 # 建立物件 ----
3 my_company <- "Hualien Tzu Chi Hospital"
4
5 # 使用函數列印物件 ----
6 print(my_company)
7
```

The code uses both single-line comments (prefixed with '#') and multi-line comments (prefixed with '#-' and followed by '-')). A red circle highlights the "Table of Contents" icon in the top right corner of the Source editor toolbar. To the right of the Source editor is a sidebar with two sections: "建立物件" and "使用函數列印物件". At the bottom of the Source editor is a status bar showing "7:1" and "# 使用函數列印物件". Below the Source editor are the Console, Terminal, and Background Jobs panes.

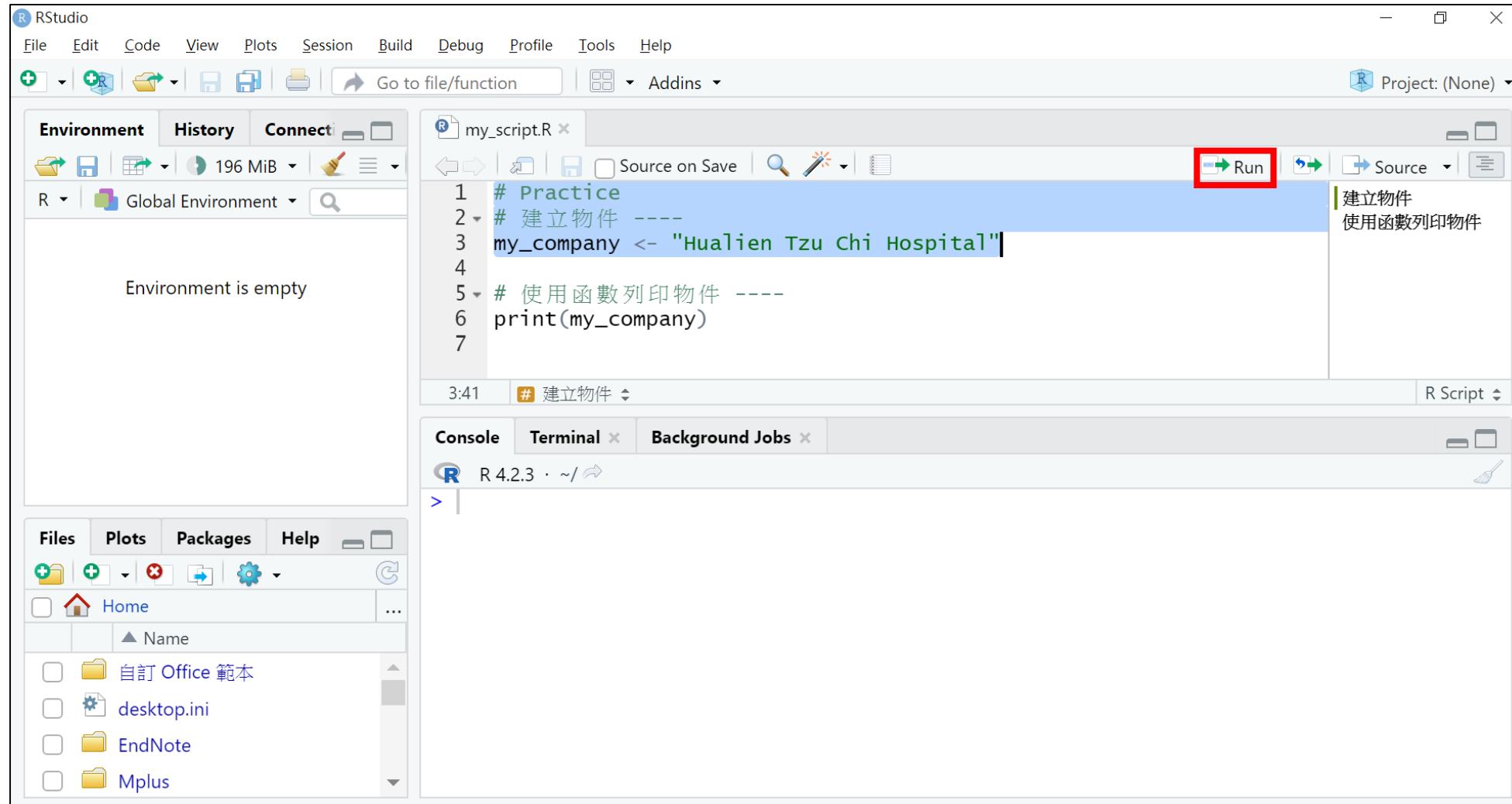
1. 點選按鈕，顯示大綱
2. 點選大綱文字，直接跳到段落
3. 只有 # 開頭是註解
4. 組合 # 和 ---- 結尾才是大綱

儲存指令稿內容



執行指令稿內容

1. 可以按Run按鈕或是Ctrl + Enter執行選所選取指令
2. 如果不反白直接按Run按鈕或是Ctrl + Enter，則執行游標所在位置的該行指令



執行指令稿內容

The screenshot shows the RStudio interface. In the top-left corner, there's a small icon of a blue circle with an 'R' inside. The main window has a title bar 'RStudio' and a menu bar with options like File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons for file operations. A search bar says 'Go to file/function'. On the right, there's a 'Project: (None)' dropdown. The left sidebar has tabs for Environment, History, and Connect, with 'Environment' currently selected. It shows a 'values' section containing 'my_com...' and 'Hualien Tzu Chi...'. The main workspace shows an R script named 'my_script.R' with the following code:

```
1 # Practice
2 # 建立物件 ----
3 my_company <- "Hualien Tzu Chi Hospital"
4
5 # 使用函數列印物件 ----
6 print(my_company)
7
```

To the right of the script, there are two explanatory boxes: '建立物件' (Create object) and '使用函數列印物件' (Use a function to print objects). The bottom part of the interface is the 'Console' tab, which displays the R session history:

```
R 4.2.3 · ~/ ↵
> # Practice
> # 建立物件 ----
> my_company <- "Hualien Tzu Chi Hospital"
>
```

A callout box points to the first two lines in the console with the text '← 註解：以#為開頭的註解' (Annotation: a comment starting with '#'). Another callout box points to the third line with the text '← 記錄：剛才請R執行的指令' (Record: the command executed by R).

執行指令稿內容

The screenshot shows the RStudio interface with the following components:

- Environment pane:** Shows the variable `my_company` assigned to "Hualien Tzu Chi Hospital".
- Code editor pane:** Displays the script `my_script.R` containing the following code:

```
1 # Practice
2 # 建立物件 ----
3 my_company <- "Hualien Tzu Chi Hospital"
4
5 # 使用函數列印物件 ----
6 print(my_company)
```
- Console pane:** Shows the R session output:

```
R 4.2.3 · ~/R
> # Practice
> # 建立物件 ----
> my_company <- "Hualien Tzu Chi Hospital"
>
```
- Annotations:** A callout box highlights the first two lines of the script with the text "← 註解：以#為開頭的註解".
- Annotations:** A callout box highlights the last line of the script with the text "← 記錄：剛才請R執行的指令".

執行指令稿內容

The screenshot shows the RStudio interface. In the top-left corner, there's a small icon of a blue circle with an 'R'. The main window has a title bar 'RStudio' with tabs for 'File', 'Edit', 'Code', 'View', 'Plots', 'Session', 'Build', 'Debug', 'Profile', 'Tools', and 'Help'. Below the title bar is a toolbar with various icons for file operations like 'New', 'Open', 'Save', etc., followed by a 'Go to file/function' search bar and a 'Project: (None)' dropdown.

The left sidebar contains tabs for 'Environment', 'History', and 'Connect'. The 'Environment' tab is selected, showing a list of objects in the 'Global Environment': 'values' and 'my_com...', with a tooltip 'Hualien Tzu Chi...'. The 'History' tab shows a list of previous commands. The 'Connect' tab is also present.

The central workspace is divided into two panes. The top pane is an 'R Script' editor for 'my_script.R' containing the following code:

```
1 # Practice
2 # 建立物件 ----
3 my_company <- "Hualien Tzu Chi Hospital"
4
5 # 使用函數列印物件 ----
6 print(my_company)
7
```

The bottom pane is a 'Console' window showing the output of the script:

```
R 4.2.3 · ~/R
> # Practice
> # 建立物件 ----
> my_company <- "Hualien Tzu Chi Hospital"
> # 使用函數列印物件 ----
> print(my_company)
[1] "Hualien Tzu Chi Hospital"
>
```

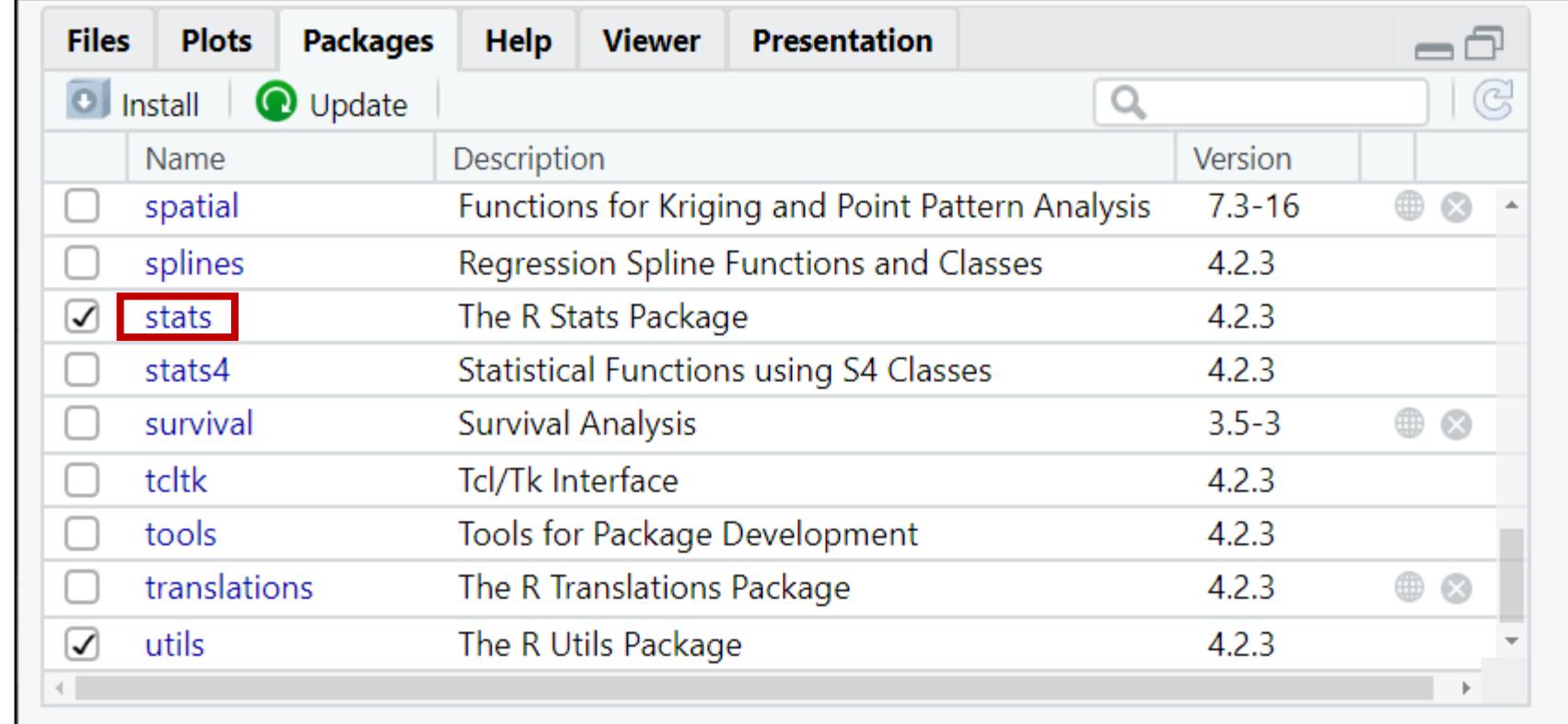
On the right side of the console, there is a callout box with three entries:

- ← 註解：以#為開頭的註解
- ← 記錄：剛才請R執行的指令
- ← 輸出：產出且會列印的結果

套件和函數是預先建立好的功能指令（集）

- 套件 (package) 包含很多函數 (function) 成為一組工具包
- 套件必須在載入 (Loading / Attaching) 的狀態下才可以使用 (左邊打勾)

點選套件名稱
可以看到所有函數



The screenshot shows the RStudio interface with the 'Packages' tab selected in the top menu bar. Below the menu, there are two buttons: 'Install' and 'Update'. A search bar is located to the right of the buttons. The main area displays a table of installed packages. The columns are 'Name', 'Description', and 'Version'. The 'Name' column contains links to each package's documentation. Two packages are highlighted with red boxes around their names: 'stats' and 'utils'. Both 'stats' and 'utils' have a checked checkbox in the leftmost column, indicating they are currently loaded.

	Name	Description	Version		
<input type="checkbox"/>	spatial	Functions for Kriging and Point Pattern Analysis	7.3-16		
<input type="checkbox"/>	splines	Regression Spline Functions and Classes	4.2.3		
<input checked="" type="checkbox"/>	stats	The R Stats Package	4.2.3		
<input type="checkbox"/>	stats4	Statistical Functions using S4 Classes	4.2.3		
<input type="checkbox"/>	survival	Survival Analysis	3.5-3		
<input type="checkbox"/>	tcltk	Tcl/Tk Interface	4.2.3		
<input type="checkbox"/>	tools	Tools for Package Development	4.2.3		
<input type="checkbox"/>	translations	The R Translations Package	4.2.3		
<input checked="" type="checkbox"/>	utils	The R Utils Package	4.2.3		

套件內所有函數列表

Documentation for package ‘stats’ version 4.2.3

- [DESCRIPTION file](#).
- [Code demos](#). Use `demo()` to run them.

Help Pages

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [misc](#)

[stats-package](#)

The R Stats Package

-- R --

點選函數名稱
可以看到說明文件
或使用指令查詢
?rnorm

[rlnorm](#)

The Log Normal Distribution

[rlogis](#)

The Logistic Distribution

[rmultinom](#)

The Multinomial Distribution

[rnbinom](#)

The Negative Binomial Distribution

[rnorm](#)

The Normal Distribution

[rpois](#)

The Poisson Distribution

[rsignrank](#)

Distribution of the Wilcoxon Signed Rank Statistic

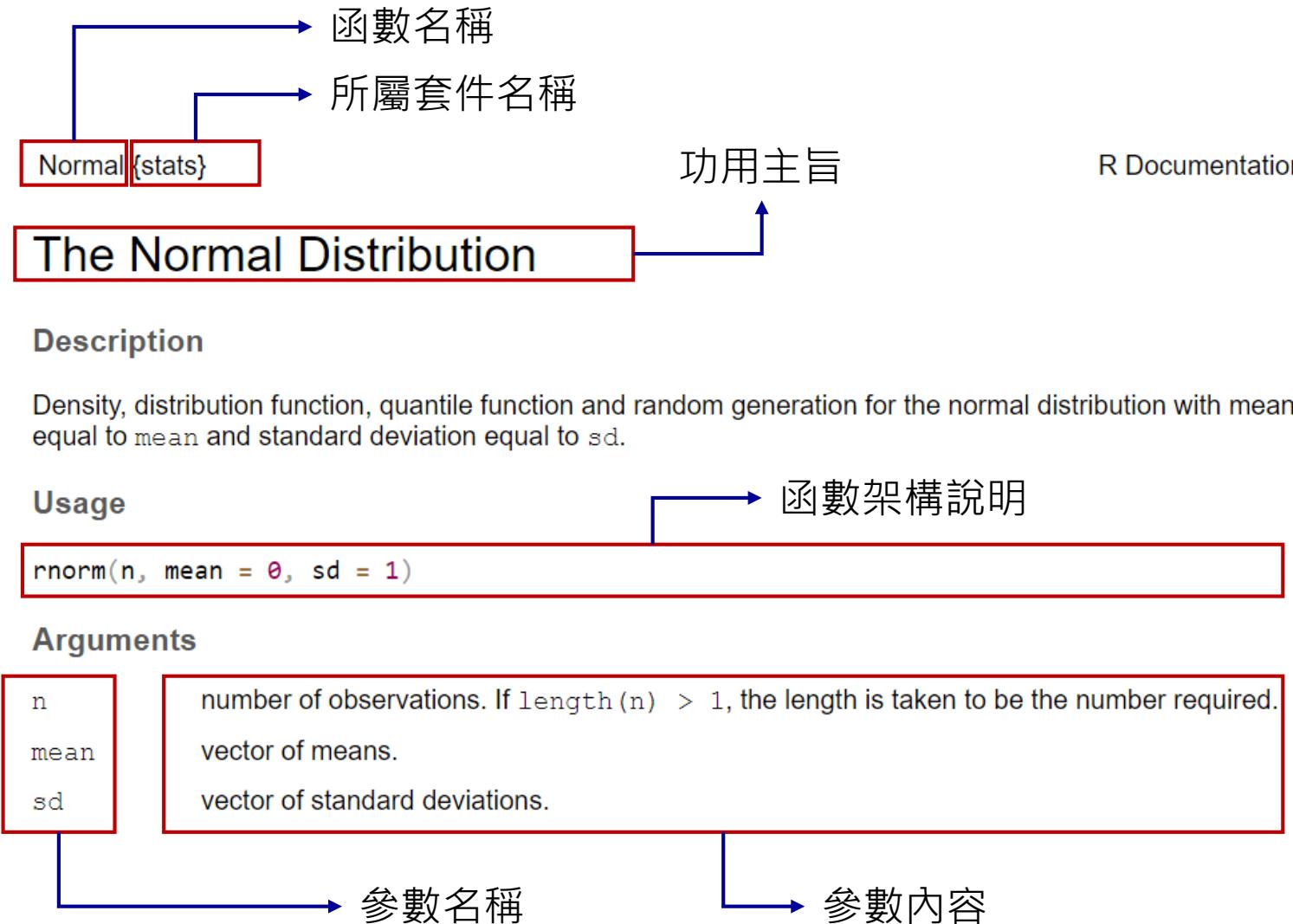
[rsmirnov](#)

Distribution of the Smirnov Statistic

[rstandard](#)

Regression Deletion Diagnostics

閱讀函數說明文件的方式



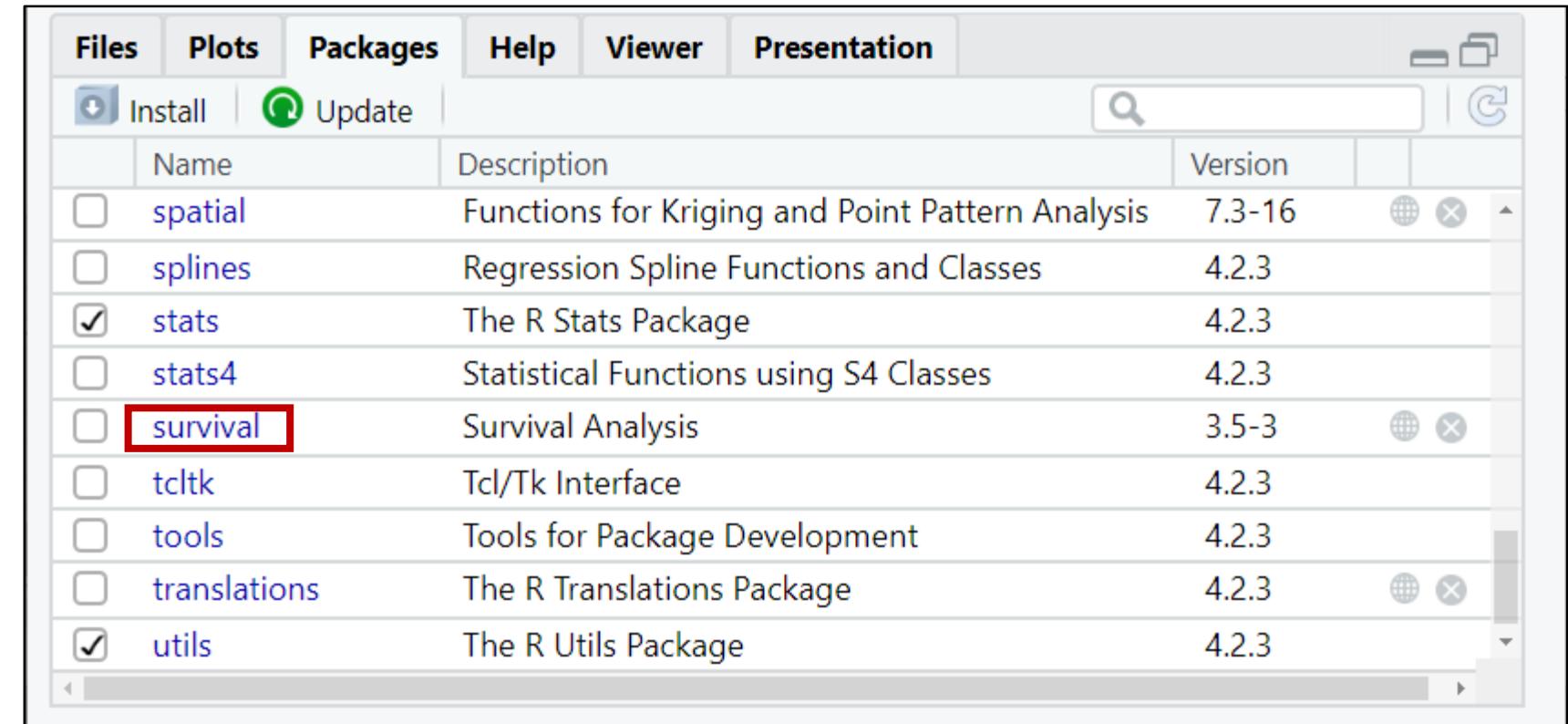
- 依照 **函數架構** 說明
在正確的 **參數名稱** 裡面
寫正確的 **參數內容**
- **rnorm()** : 函數名稱
 - 常態分佈隨機抽樣
- **n** : 抽樣個數
 - 必須指定，否則無法執行
- **mean** : 抽樣平均值
 - 預設 (default) 是0
- **sd** : 抽樣標準差
 - 預設 (default) 是1

動手實際將指令在Console執行

- `rnorm(n = 50)`
 - 依照預設值 (`mean = 0, sd = 1`) 抽樣50個常態分佈亂數
- `rnorm(n = 50, mean = 1)`
 - 依照指定值 (`mean = 1`) 和預設值 (`sd = 1`) 抽樣50個常態分佈亂數
- `rnorm(n = 50, mean = 1, sd = 0.23)`
 - 依照指定值 (`mean = 1, sd = 0.23`) 抽樣50個常態分佈亂數
- `rnorm(50, 1, 0.23)`
 - 前者的簡化版，照順序填寫參數內容的話，可忽略參數名稱不寫

非預設載入的套件使用library()指令載入

- library(survival)
 - 每次開啟RStudio都要執行



打勾就可以使用咯

用install.packages()下載非預設安裝套件

- 先確認電腦有連上網路
 - 套件名稱要放在雙引號內
 - 安裝過一次就可以了，除非你換電腦 / 重灌R / 更換為第一次使用的R版本
-
- # 下載套件
 - `install.packages("data.table")`
 - `install.packages("lubridate")`

套件與函數的基本觀念複習

- 套件
 - 非預設載入的套件，每次開啟RStudio都要用`library()`載入
 - 工具拿在手上才可以用

- 非預設安裝的套件，使用`install.packages()`安裝過一次就可以了
 - 除非你換電腦 / 重灌R / 更換為第一次使用的R版本
 - 指定的套件名稱要用雙引號包含

- 函數
 - 函數名稱(參數名稱 = 參數內容)，這才是完整可執行的指令
 - 有些參數必須給參數內容才可以執行
 - 有些參數有預設值、指定形式

R軟體資料管理實作

- 資料的來源
- 資料管理程式撰寫
- `data.table`套件
- 讀取、寫出、型態、篩選
- 排序、修改、歸戶、合併

資料的產生：真實世界

- 2月28日深夜，一名68歲男性由親屬帶入急診

時 人 地

- 主訴胸悶胸痛，經診斷為急性心肌梗塞 (acute myocardial infarction, AMI)

事

- 在90分鐘內完成心導管介入處置 (Percutaneous coronary intervention, PCI)

物

資料的儲存：資料與編碼

- 結構化資料表 (data table)
- 譯碼簿 (codebook)

欄 / column / 變項 / variable

列 / row / 觀察值 / observation

id	date	male	age	diagnosis	procedure
S1911	02-28	1	68	I21	PCI
...	
...	
...	
...	
...	

變項名稱	中文意義	資料類型	編碼方式
id	身分證號	文字	S+四位數字
date	就醫日期	日期	mm-dd
male	男性	數值	1 = 男性 ; 0 = 女性
age	年齡	數值	單位：歲
diagnosis	主診斷	文字	ICD-10-CM編碼
procedure	處置項目	文字	項目名稱

範例資料與譯碼簿（ coding book ）說明

- data-management-basic → base → 門診檔（ data_opd.csv ）

id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
S00000741	20140923	Hospital_A	家醫科	S00135841	389	273
S00000895	20140101	Hospital_A	家醫科	S00135841	924	228
S00005005	20140408	Hospital_A	家醫科	S00127301	250	681
S00005005	20140423	Hospital_A	家醫科	S00127301	250	208
S00005005	20140624	Hospital_A	家醫科	S00127301	250	611
S00005005	20140917	Hospital_A	家醫科	S00127301	250	428
S00005005	20141231	Hospital_A	家醫科	S00127301	250	511
S00006411	20140312	Hospital_A	家醫科	S00135841	729	3508
S00006411	20140429	Hospital_A	家醫科	S00135841	729	178
S00010305	20140714	Hospital_A	家醫科	S00121987	141	26628
S00010305	20140806	Hospital_A	家醫科	S00182860	141	21778
S00017542	20140129	Hospital_A	家醫科	S00135841	401	594
S00017542	20140409	Hospital_A	家醫科	S00135841	401	566
S00017542	20140618	Hospital_A	家醫科	S00135841	401	656

變項名稱	中文意義	型態	編碼方式
id	身分證號	文字	S+8位數字
opd_date	門診日期	文字	yyyymmdd
hosp_id	醫院代號	文字	Hospital_A
func_name	科別名稱	文字	中文科別名稱
prsn_id	醫師代號	文字	S+8位數字
icd_code	主診斷	文字	ICD-9-CM編碼
fee	費用	文字	就醫費用

範例資料與譯碼簿（ coding book ）說明

- data-management-basic → base → 住院檔（ data_ipd.csv ）

id	ipd_date	hosp_id	func_name	prsn_id	icd_code	fee	bed_day
S00055891	20140313	Hospital_A	內科	S00062643	008	19560	5
S00133035	20140206	Hospital_A	內科	S00119860	485	25946	7
S00143075	20140410	Hospital_A	內科	S00186014	560	117527	21
S00143075	20140428	Hospital_A	內科	S00186014	537	65512	19
S00221869	20140918	Hospital_A	內科	S00119860	682	14517	6
S00000569	20140508	Hospital_A	外科	S00056705	151	9737	2
S00002042	20140116	Hospital_A	外科	S00198036	574	8540	3
S00014318	20140522	Hospital_A	外科	S00174106	532	34650	7
S00031212	20141216	Hospital_A	外科	S00154796	193	47452	3
S00035354	20141031	Hospital_A	外科	S00169025	174	49109	4
S00051894	20141015	Hospital_A	外科	S00089540	155	5695	1
S00057157	20140302	Hospital_A	外科	S00157090	174	60706	5
S00057157	20140303	Hospital_A	外科	S00157090	174	36867	5
S00059690	20140529	Hospital_A	外科	S00047401	576	136033	27

變項名稱	中文意義	型態	編碼方式
id	身分證號	文字	S+8位數字
ipd_date	住院日期	文字	yyyymmdd
hosp_id	醫院代號	文字	Hospital_A
func_name	科別名稱	文字	中文科別名稱
prsn_id	醫師代號	文字	S+8位數字
icd_code	主診斷	文字	ICD-9-CM編碼
fee	費用	文字	就醫費用
bed_day	住院天數	文字	住院天數

資料處理和資料管理的差別？

- 資料處理 (processing)
 - 有能力完成一個行為
 - 這個行為基本上是對的 (運算)
- 資料管理 (management)
 - 知道行為背後的目的
 - 知道為何做以及如何做更好 (效果)
- 永遠要問自己「為什麼」
- 計算醫療費用的平均值
 - `mean(醫療費用)`
- 為什麼要計算？
 - 醫務管理 / 品質管理
- 為了這個目的怎麼樣可以更好？
 - 費用通常為偏態分布，只看平均會失準
 - 四分位數 `quantile(醫療費用)`
 - 直方圖 `hist(醫療費用)`
- 這些更可以讓資料回答你的目的

寫代碼與寫程式的差別是什麼？

- 寫代碼 (coding)
 - 只有片段性動作
- 物件命名前後不一
- 資料使用無法連貫
- 函數執行雜亂無序
- 寫程式 (programming)
 - 具有整體性規劃
- 物件命名具有系統
- 資料使用導向明確
- 函數執行模式統一

data.table是非常好用的資料管理套件

- 請用 `class(dt)` 確認物件 `dt` 具有 `data.table` 屬性
 - 讀取、寫出、型態、篩選、排序、修改、歸戶、合併
- 對物件的操作方式寫在`dt`後面的中括號裡
 - `dt[i, j, by]`
 - `i`：選擇哪一些觀察值子集？
 - `j`：操作哪一些欄位(提取/建立/更新/整合/計算)？
 - `by`：前述的處理動作要依照什麼欄位進行分組處理？
- 還有很多很好用的函數
 - `merge, melt, dcast, shift, rbindlist`

打開指令稿一起實作！
`data-management-basic-ST.R`

從硬碟讀取資料

```
# 載入套件
library(data.table)
```

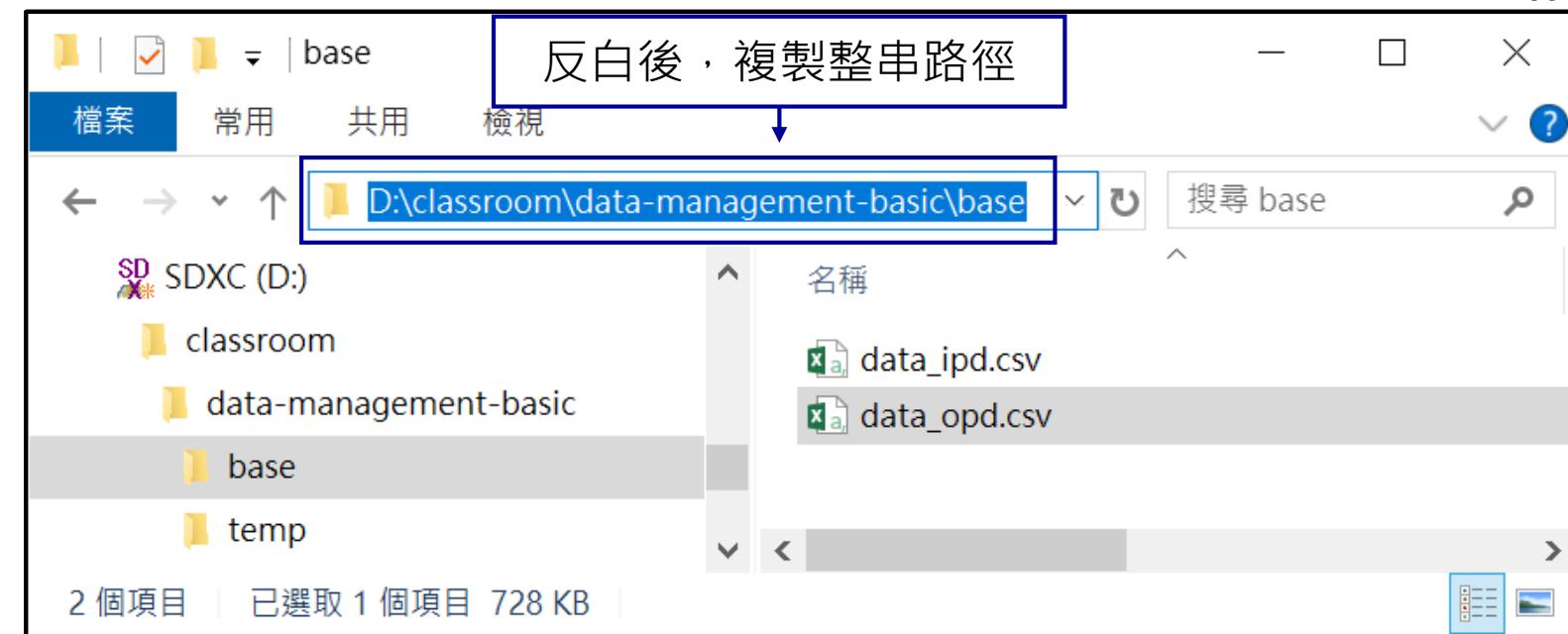
路徑設定

```
path_base <- "D:/classroom/data-management-basic/base"
```

指定資料處理的硬碟路徑

```
setwd(path_base)
```

反白後，複製整串路徑



把所有分隔斜線 \ 改成 /

fread是data.table套件用來讀取csv檔案的函數

```
dt_opd <- fread("data_opd.csv", colClasses = "character")
```

常用資料讀取與寫出的函數指令

- 檔案存於電腦硬碟中，RStudio只是暫時空間，關掉後資料立即消失
- 讀取才可以操作，寫出才可以保存

資料格式	讀取資料		寫出資料	
	套件	函數	套件	函數
Excel	xlsx	read.xlsx	xlsx	write.xlsx
CSV	data.table	fread	data.table	fwrite
SAS	haven	read_sas	haven	write_sas
Stata	haven	read_dta	haven	write_dta
SPSS	haven	read_sav	haven	write_sav

確認大致樣貌及描述性統計

```
# head 印出資料的前6列(預設)
```

```
head(dt_opd)
```

	id <char>	opd_date <char>	hosp_id <char>	func_name <char>	prsn_id <char>	icd_code <char>	fee <char>
1:	S00000741	20140923	Hospital_A	家醫科	S00135841	389	273
2:	S00000895	20140101	Hospital_A	家醫科	S00135841	924	228
3:	S00005005	20140408	Hospital_A	家醫科	S00127301	250	681
4:	S00005005	20140423	Hospital_A	家醫科	S00127301	250	208
5:	S00005005	20140624	Hospital_A	家醫科	S00127301	250	611
6:	S00005005	20140917	Hospital_A	家醫科	S00127301	250	428

確認大致樣貌及描述性統計

```
# summary 摘要統計
```

```
summary(dt_opd)
```

id	opd_date	hosp_id	func_name
Length:12155	Length:12155	Length:12155	Length:12155
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character
prsn_id	icd_code	fee	
Length:12155	Length:12155	Length:12155	
Class :character	Class :character	Class :character	
Mode :character	Mode :character	Mode :character	

確認大致樣貌及描述性統計

```
# nrow 資料集觀察值筆數
```

```
# ncol 資料集欄位數量
```

```
# dim 資料集列數和欄數
```

```
nrow(dt_opd)
```

12155

```
ncol(dt_opd)
```

7

```
dim(dt_opd)
```

12155 7

文字轉日期或數值

```
# class 確認資料類型
```

```
# dt$v1 代表dt資料集裡面的v1變項
```

```
class(dt_opd$opd_date)
```

```
"character"
```

```
class(dt_opd$fee)
```

```
"character"
```

```
# ymd 將資料轉日期格式
```

```
# as.numeric 將資料轉數值格式
```

```
dt_opd$opd_date <- ymd(dt_opd$opd_date)
```

```
dt_opd$fee <- as.numeric(dt_opd$fee)
```

```
class(dt_opd$opd_date)
```

```
"Date"
```

```
class(dt_opd$fee)
```

```
"numeric"
```

文字轉日期或數值

```
head(dt_opd)
```

1:	S00000741	家醫科	2014-09-23	273
2:	S00000895	家醫科	2014-01-01	228
3:	S00005005	家醫科	2014-04-08	681
4:	S00005005	家醫科	2014-04-23	208
5:	S00005005	家醫科	2014-06-24	611
6:	S00005005	家醫科	2014-09-17	428

文字轉日期或數值

```
summary(dt_opd)
```

id	func_name	opd_date	fee
Length:12155	Length:12155	Min. :2014-01-01	Min. : 100
Class :character	Class :character	1st Qu.:2014-03-29	1st Qu.: 423
Mode :character	Mode :character	Median :2014-06-26	Median : 860
		Mean :2014-06-27	Mean : 2360
		3rd Qu.:2014-09-25	3rd Qu.: 1912
		Max. :2014-12-31	Max. :164426
		NA's :2	

取得想要的觀察值 《遺漏值》

```
# 文字的空值是 ""  
# 找出空值的文字欄位
```

```
nrow(dt_opd[icd_code == ""])
```

```
2
```

```
head(dt_opd[icd_code == ""])
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
	<char>	<Date>	<char>	<char>	<char>	<char>	<num>
1:	S00015092	2014-12-13	Hospital_A	小兒科	S00026172	[REDACTED]	312
2:	S00015092	2014-12-15	Hospital_A	小兒科	S00026172	[REDACTED]	1092

取得想要的觀察值 《遺漏值》

```
# 數值的空值是 NA
```

```
# 找出空值的數值欄位
```

```
nrow(dt_opd[is.na(fee)])
```

```
2
```

```
head(dt_opd[is.na(fee)])
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
	<char>	<Date>	<char>	<char>	<char>	<char>	<num>
1:	S00222548	2014-02-12	Hospital_A	神經科	S00093632	781	NA
2:	S00164810	2014-09-23	Hospital_A	直腸外科	S00088611	154	NA

取得想要的觀察值《指定條件》

對單一欄位(向量)進行次數分析

```
table(dt_opd$func_name)
```

小兒外科	小兒科	中醫科	內分泌科	內科	...
24	728	659	679	111	...

比較運算子(operator)

== 是將左右兩者進行全等比較

%in% 是比對左邊物件是否在右邊的清單當中

a != b 或 !(a == b)都表示a不等於b

!(a %in% b)表示a不屬於b清單當中

文字內容都要被雙引號包住

取得想要的觀察值《指定條件》

func_name變項的內容為內科

```
ot_select_func_name_1 <- dt_opd[func_name == "內科"]
```

func_name變項的內容為心臟血管內科或是心臟血管外科

```
ot_select_func_name_2 <- dt_opd[func_name %in% c("心臟血管內科", "心臟血管外科")]
```

func_name變項的內容"不為"復健科

```
ot_select_func_name_3 <- dt_opd[!(func_name == "復健科")]
```

```
ot_select_func_name_4 <- dt_opd[func_name != "復健科"]
```

func_name變項的內容"不為"中醫科或是牙科

```
ot_select_func_name_5 <- dt_opd[!(func_name %in% c("中醫科", "牙科"))]
```

取得想要的觀察值《比大小》

對單一欄位(向量)進行描述性統計分析

```
summary(dt_opd$fee)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
100	423	860	2360	1912	164426	2

a > b a大於b

a < b a小於b

a >= b a大於等於b

a <= b a小於等於b

(a <= b) & (b <= c) a小於等於b 而且(AND) b小於等於c

(a <= b) & (c == d) a小於等於b 而且(AND)) c等於d

(a <= b) | (c == d) a小於等於b 或者且(OR) c等於d

多個邏輯條件連接時請善用小括號()

取得想要的觀察值 《比大小》

fee變項的內容為大於等於100,000

```
ot_select_fee_100k <- dt_opd[100000 <= fee]
```

fee介於50,000到100,000之間的觀察值, "&" = AND 代表條件交集, "|" = OR 代表條件聯集

```
ot_select_fee_range_1 <- dt_opd[(50000 <= fee) & (fee <= 100000)]
```

fee介於50,000到100,000之間的觀察值

```
ot_select_fee_range_2 <- dt_opd[inrange(x = fee, lower = 50000, upper = 100000)]
```

取得想要的觀察值《比大小》

```
# fee介於50,000到100,000之間而且是放射腫瘤科觀察值  
ot_select_fee_range_3 <- dt_opd[  
  inrange(x = fee, lower = 50000, upper = 100000) & func_name == "放射腫瘤科"]
```

```
# fee介於50,000到100,000之間或是放射腫瘤科觀察值  
ot_select_fee_range_4 <- dt_opd[  
  inrange(x = fee, lower = 50000, upper = 100000) | func_name == "放射腫瘤科"]
```

取得想要的觀察值《字串模式》

```
table(dt_opd$func_name)
```

小兒外科	小兒科	中醫科	內分泌科	內科	...
24	728	659	679	111	...

```
# grep
```

用於判斷哪些位置(1, 2, 3, ... n)的觀察值符合指定文字條件

回傳的值為位置序號向量(1, 2, 62, 359 ...)

只有符合的才會回傳

找出門診檔func_name字串裡面含有"內科"的觀察值回傳位置序號向量

```
ot_select_string_1_vector <- grep(pattern = "內科", x = dt_opd$func_name)
```

```
head(ot_select_string_1_vector)
```

89 90 91 92 93 94

取得想要的觀察值《字串模式》

找出門診檔func_name字串裡面含有"內科"的觀察值

```
ot_select_string_1 <- dt_opd[grep(pattern = "內科", x = func_name)]
```

```
table(ot_select_string_1$func_name)
```

內科	心臟血管內科	胸腔內科	腎臟內科	腸胃內科
111	812	420	258	466

取得想要的觀察值《字串模式》

```
# grep1  
# 用於判斷哪些位置(1, 2, 3, ... n)的觀察值符合指定文字條件  
# 回傳的值為邏輯向量(T, T, F, T, ... T)  
# 所有判斷邏輯值都會回傳  
  
# 找出門診檔func_name字串裡面含有"腫瘤"的觀察值回傳位置序號向量  
ot_select_string_2_vector <- grep1(pattern = "腫瘤", x = dt_opd$func_name)  
  
table(ot_select_string_2_vector)  
FALSE   TRUE  
11582    573
```

取得想要的觀察值《字串模式》

```
# 找出門診檔func_name字串裡面含有"腫瘤"的觀察值  
ot_select_string_2 <- dt_opd[grep1(pattern = "腫瘤", x = func_name)]  
head(ot_select_string_2)  
  
      id    opd_date    hosp_id   func_name    prsn_id icd_code    fee  
      <char>     <Date>     <char>     <char>     <char>    <char> <num>  
1: S00000026 2014-07-17 Hospital_A 血液腫瘤科 S00191804        285    768  
2: S00003597 2014-06-04 Hospital_A 血液腫瘤科 S00018815        142    391
```

```
table(ot_select_string_2$func_name)
```

血液腫瘤科 放射腫瘤科

419

154

取得想要的觀察值《字串模式》

```
# regexp  
# regular expression 正規表達式  
# grep1 + regexp 基本型態 ^ 代表"開頭"  
  
# 找出門診檔func_name字串裡面"開頭"為心臟的觀察值  
ot_select_string_3 <- dt_opd[grep1(pattern = "^\u5e02\u5e02", x = func_name)]
```

```
table(ot_select_string_3$func_name)
```

心臟血管內科 心臟血管外科

812

141

取得想要的觀察值《字串模式》

```
# regexp  
# regular expression 正規表達式  
# grepl + regexp 基本型態 $ 代表"結尾"  
  
# 找出門診檔func_name字串裡面"結尾"為外科的觀察值  
ot_select_string_4 <- dt_opd[grepl(pattern = "外科$", x = func_name)]  
  
table(ot_select_string_4$func_name)  
小兒外科 心臟血管外科 外科 直腸外科 神經外科 整形外科  
24 141 348 188 189 181
```

取得想要的觀察值《字串模式》

```
# 找出icd_code字串裡面為433開頭或是434開頭的觀察值  
ot_select_string_5 <- dt_opd[grep1(pattern = "^433|^434", x = icd_code)]  
  
# 找出icd_code字串裡面為43開頭，後面接著3或4的觀察值  
ot_select_string_6 <- dt_opd[grep1(pattern = "^43[34]", x = icd_code)]
```

```
head(ot_select_string_5)  
  id      opd_date      hosp_id func_name prsn_id      icd_code    fee  
S00012020 2014-03-06 Hospital_A 神經科 S00128445        433 1276
```

```
table(ot_select_string_5$icd_code)  
433 434
```

13 63

取得想要的觀察值 《字串模式》

找出icd_code字串裡面為43開頭，後面接著0 ~ 8的觀察值

```
ot_select_string_7 <- dt_opd[grep1(pattern = "43[0-8]", icd_code)]
```

```
table(ot_select_string_7$icd_code)
```

```
430 431 433 434 435 436 437 438
```

```
11 23 13 63 11 34 14 25
```

取得想要的變項

```
# i索引處空白，寫個逗號分隔i和j，在j索引處寫一個句點，一對小括號，裡面寫入想要留下的變項名稱  
# 從dt_opd中留下三個變項id, opd_date, hosp_id  
ot_variable_select <- dt_opd[, .(id, opd_date, hosp_id)]  
  
head(ot_variable_select)  
      id    opd_date    hosp_id  
1: S00000741 2014-09-23 Hospital_A  
2: S00000895 2014-01-01 Hospital_A  
3: S00005005 2014-04-08 Hospital_A  
4: S00005005 2014-04-23 Hospital_A  
5: S00005005 2014-06-24 Hospital_A  
6: S00005005 2014-09-17 Hospital_A
```

讓資料成為想要的順序

```
# 依照就醫日期排序，預設為遞增（ ascending ）排序
```

```
ot_sort <- ot_sort[order(opd_date)]
```

```
head(ot_sort)
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
	<char>	<Date>	<char>	<char>	<char>	<char>	<num>
1:	S00000895	2014-01-01	Hospital_A	家醫科	S00135841	924	228
2:	S00185013	2014-01-01	Hospital_A	外科	S00032959	174	267
3:	S00040287	2014-01-01	Hospital_A	小兒科	S00000261	271	312
4:	S00212915	2014-01-01	Hospital_A	婦產科	S00166168	V70	430
5:	S00008761	2014-01-01	Hospital_A	骨科	S00155825	358	787
6:	S00048277	2014-01-01	Hospital_A	骨科	S00134227	715	217

讓資料成為想要的順序

```
# 依照多個排序欄位(病患編號,就醫醫院,就醫日期)之間以逗點隔開
```

```
ot_sort <- ot_sort[order(id, hosp_id, opd_date)]
```

```
head(ot_sort)
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
	<char>	<Date>	<char>	<char>	<char>	<char>	<num>
1:	S00000026	2014-07-17	Hospital_A	血液腫瘤科	S00191804	285	768
2:	S00000147	2014-08-12	Hospital_A	耳鼻喉科	S00202514	V76	130
3:	S00000200	2014-01-15	Hospital_A	內科	S00186014	154	636
4:	S00000200	2014-07-16	Hospital_A	內科	S00186014	154	571

讓資料成為想要的順序

```
# 若要改為遞減 ( descending ) 排序，在變項前面加上負號 ( - )
```

```
ot_sort <- ot_sort[order(id, hosp_id, -fee)]
```

```
head(ot_sort)
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
	<char>	<Date>	<char>	<char>	<char>	<char>	<num>
1:	S00000026	2014-07-17	Hospital_A	血液腫瘤科	S00191804	285	768
2:	S00000147	2014-08-12	Hospital_A	耳鼻喉科	S00202514	V76	130
3:	S00000200	2014-01-15	Hospital_A	內科	S00186014	154	636
4:	S00000200	2014-01-15	Hospital_A	內科	S00186014	154	571

依據排序將觀察值選出

```
# SD = sub-data within BY variable  
# 取出每人在門診就醫的第1筆  
ot_obs_order_1 <- ot_obs_order_1[order(id, opd_date)]  
ot_obs_order_1 <- ot_obs_order_1[, .SD[1], by = .(id)]
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
1:	S00000026	2014-07-17	Hospital_A	血液腫瘤科	S00191804	285	768
2:	S00000147	2014-08-12	Hospital_A	耳鼻喉科	S00202514	V76	130
3:	S00000200	2014-01-15	Hospital_A	內科	S00186014	154	636
4:	S00000326	2014-02-04	Hospital_A	腸胃內科	S00088572	571	320
5:	S00000330	2014-04-12	Hospital_A	婦產科	S00060863	V22	845
6:	S00000470	2014-01-03	Hospital_A	急診醫學科	S00098094	487	598

依據排序將觀察值選出

```
# SD = sub-data within BY variable  
# 取出每人在門診就醫的第k筆，例如說K = 3  
ot_obs_order_3 <- ot_obs_order_3[order(id, opd_date)]  
ot_obs_order_3 <- ot_obs_order_3[, .SD[3], by = .(id)]
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
1:	S00000026	<NA>	<NA>	<NA>	<NA>	<NA>	NA
2:	S00000147	<NA>	<NA>	<NA>	<NA>	<NA>	NA
3:	S00000200	2014-07-16	Hospital_A	內科	S00186014	154	571
4:	S00000326	2014-06-19	Hospital_A	腸胃內科	S00088572	571	320
5:	S00000330	2014-06-14	Hospital_A	婦產科	S00060863	V22	600
6:	S00000470	<NA>	<NA>	<NA>	<NA>	<NA>	NA

依據排序將觀察值選出

```
# SD = sub-data within BY variable  
# 取出每人在門診就醫的第m:n筆  
ot_obs_order_m <- ot_obs_order_m[order(id, opd_date)]  
ot_obs_order_m <- ot_obs_order_m[, .SD[1:3], by = .(id)]
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
1:	S00000026	2014-07-17	Hospital_A	血液腫瘤科	S00191804	285	768
2:	S00000026	<NA>	<NA>	<NA>	<NA>	<NA>	NA
3:	S00000026	<NA>	<NA>	<NA>	<NA>	<NA>	NA
4:	S00000147	2014-08-12	Hospital_A	耳鼻喉科	S00202514	V76	130
5:	S00000147	<NA>	<NA>	<NA>	<NA>	<NA>	NA
6:	S00000147	<NA>	<NA>	<NA>	<NA>	<NA>	NA

依據排序將觀察值選出

```
# SD = sub-data within BY variable  
# 取出每人在門診就醫的第N筆（最後一筆）  
ot_obs_order_n <- ot_obs_order_n[order(id, opd_date)]  
ot_obs_order_n <- ot_obs_order_n[, .SD[ .N], by = .(id)]
```

	id	opd_date	hosp_id	func_name	prsn_id	icd_code	fee
1:	S00000026	2014-07-17	Hospital_A	血液腫瘤科	S00191804	285	768
2:	S00000147	2014-08-12	Hospital_A	耳鼻喉科	S00202514	V76	130
3:	S00000200	2014-12-02	Hospital_A	內科	S00186014	154	571
4:	S00000326	2014-09-11	Hospital_A	腸胃內科	S00088572	571	320
5:	S00000330	2014-08-06	Hospital_A	婦產科	S00060863	V22	450
6:	S00000470	2014-01-03	Hospital_A	急診醫學科	S00098094	487	598

產生變數

```
# 新增一個變項名為long_los，內容為數值9
```

```
ot_variable_generate$long_los <- 9
```

```
head(ot_variable_generate)
```

							long_los
S00055891	2014-03-13	Hospital_A		內科	008	5	9
S00133035	2014-02-06	Hospital_A		內科	485	95	9

```
table(ot_variable_generate$long_los)
```

```
9
```

```
479
```

改變變數

```
# 把既有的變項內容修改(更新)為0
```

```
ot_variable_generate <- ot_variable_generate[, `:=` (long_los = 0)]
```

```
head(ot_variable_generate)
```

S00055891	2014-03-13	Hospital_A		内科	008	5	0
S00133035	2014-02-06	Hospital_A		内科	485	95	0

```
table(ot_variable_generate$long_los)
```

```
0
```

```
479
```

有條件改變變數

```
# 如果是住院天數大於90天，判斷為超長住院天數(long LOS = 1)
ot_variable_generate <- ot_variable_generate[90 <= bed_day, `:=`(long_los = 1)]  
  
head(ot_variable_generate)  
  
      id    ipd_date    hosp_id func_name icd_code bed_day long_los  
S00055891 2014-03-13 Hospital_A     內科      008       5       0  
S00133035 2014-02-06 Hospital_A     內科      485      95       1  
  
table(ot_variable_generate$long_los)  
  0 1  
478 1
```

計算變數

把總費用扣掉掛號費500元

```
ot_variable_caculate <- ot_variable_generate[, .(id, fee, fee_appl = fee - 500)]
```

```
head(ot_variable_caculate)
```

	id	fee	fee_appl
	<char>	<num>	<num>
1:	S00055891	19560	19060
2:	S00133035	25946	25446
3:	S00143075	117527	117027
4:	S00143075	65512	65012

其他加減乘除相同操作概念

去除重複

```
# unique 依據所有資料內的變項進行比對和去重複  
# 依照data.table內所有的變數進行比對，刪除重複的觀察值  
ot_unique_1 <- unique(dt_ipd[, .(id, func_name)])  
  
head(ot_unique_1)  
          id func_name  
1: S00055891    內科  
2: S00133035    內科  
3: S00143075    內科  
4: S00221869    內科  
5: S00000569    外科  
6: S00002042    外科
```

去除重複

```
# unique 依據所有資料內的變項進行比對和去重複  
# 依照data.table內所有的變數進行比對，刪除重複的觀察值  
ot_unique_2 <- unique(dt_ipd[, .(id)])  
  
head(ot_unique_2)  
      id  
1: S00055891  
2: S00133035  
3: S00143075  
4: S00221869  
5: S00000569  
6: S00002042
```

依據變項內容相同者歸納資訊

```
# 計算每人平均看診費用  
# 將分群統計資料存為新的變項，觀察值總數變少(aggregate to a summary dataset)  
ot_groupby_aggregate <- dt_ipd[, .(fee_m = mean(fee)), by = .(id)]
```

	id	fee_m
1:	S00055891	19560.0
2:	S00133035	115347.2
3:	S00143075	91519.5
4:	S00221869	14517.0
5:	S00000569	9737.0
6:	S00002042	8540.0

依據變項內容相同者統計資訊

```
# 計算每人平均看診費用  
# 將分群統計資料存為新的變項，觀察值總數不變(re-merge to original dataset)  
ot_groupby_remerge <- dt_ipd[, `:=`(fee_m = mean(fee)), by = .(id)]
```

	id	ipd_date	hosp_id	icd_code	fee	fee_m
1:	S00055891	2014-03-13	Hospital_A	008	19560	19560.0
2:	S00133035	2014-02-06	Hospital_A	485	25946	115347.2
3:	S00143075	2014-04-10	Hospital_A	560	117527	91519.5
4:	S00143075	2014-04-28	Hospital_A	537	65512	91519.5
5:	S00221869	2014-09-18	Hospital_A	682	14517	14517.0
6:	S00000569	2014-05-08	Hospital_	51	9737	9737.0

依據變項內容相同者串聯不同資料集

```
# 將門診和住院的部分變項留下
```

```
# 想要看同一醫病之間(id, hosp_id, prsn_id)在住院前後的門診就醫紀錄
```

```
ot_x <- dt_ipd[, .(id, hosp_id, prsn_id, ipd_date)]
```

```
ot_y <- dt_opd[, .(id, hosp_id, prsn_id, opd_date, opd_fee = fee)]
```

```
ot_merge <- merge.data.table(x = ot_x, y = ot_y, by = c("id", "hosp_id", "prsn_id"))
```

1:	S00001370	Hospital_A	S00008185	2014-01-24	2014-01-07	567
2:	S00001370	Hospital_A	S00008185	2014-01-24	2014-02-27	277
3:	S00001370	Hospital_A	S00008185	2014-01-24	2014-04-30	336
4:	S00001370	Hospital_A	S00008185	2014-01-24	2014-08-16	321

資料合併：不同水平合併的比較

```
# 交集 inner join(merge.data.table預設模式)
```

```
ot <- merge(x = dt_1, y = dt_2, by = c("id"))
```

```
# 左聯結 left join
```

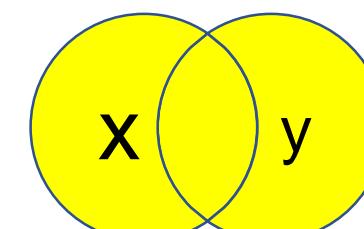
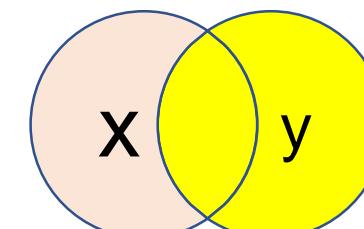
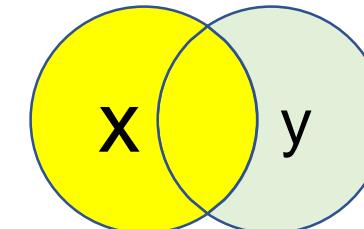
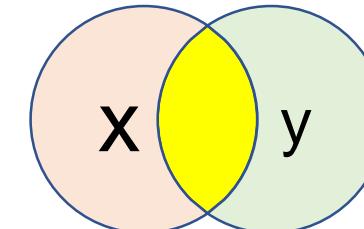
```
ot <- merge(x = dt_1, y = dt_2, by = c("id"), all.x = T)
```

```
# 右聯結 right join
```

```
ot <- merge(x = dt_1, y = dt_2, by = c("id"), all.y = T)
```

```
# 聯集 full join
```

```
ot <- merge(x = dt_1, y = dt_2, by = c("id"), all = T)
```



依據變項名稱相同者堆疊不同資料集

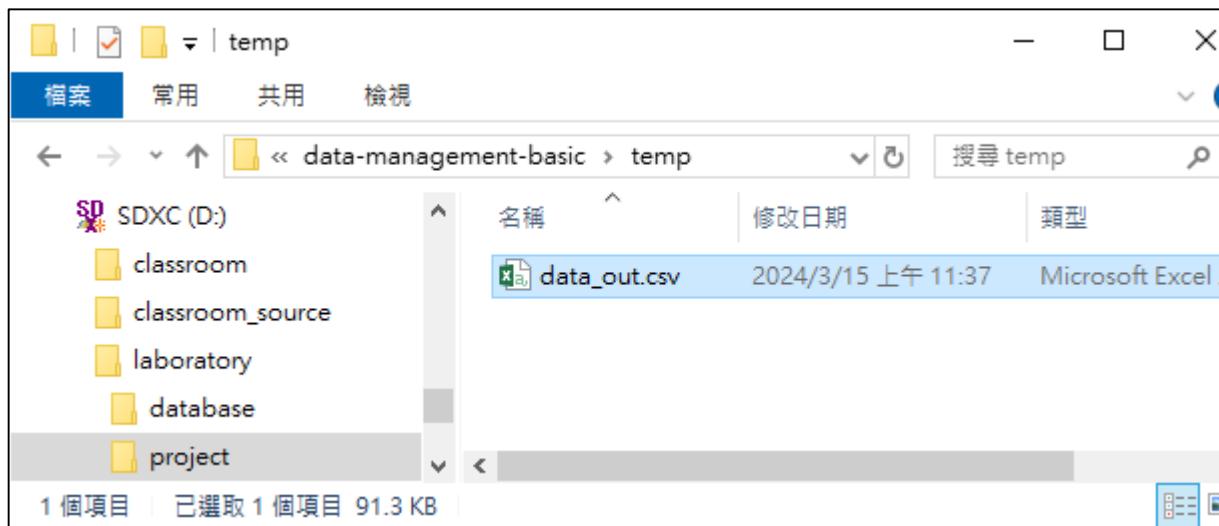
```
# 將門診和住院的部分變項留下  
ot_b1 <- dt_ipd[, .(id, func_name, prsn_id, icd_code, visit_date, dep = "ipd")]  
ot_b2 <- dt_opd[, .(id, func_name, prsn_id, icd_code, visit_date, dep = "opd")]  
ot_bind <- rbind(ot_bind_1, ot_bind_2)
```

						dep
1:	S00000200	內科	S00186014	154	2014-01-15	opd
2:	S00000200	內科	S00186014	154	2014-04-01	opd
3:	S00000200	內科	S00186014	154	2014-07-16	opd
4:	S00000200	內科	S00186014	154	2014-09-09	opd
5:	S00000200	內科	S00186014	154	2014-12-02	opd
6:	S00000200	直腸外科	S00140819	154	2014-09-02	opd

將資料寫出（儲存）到硬碟

```
# 指定寫出路徑  
setwd(path_temp)
```

```
# 寫出csv檔案  
fwrite(ot_merge, "data_out.csv")
```



工作環境整理

```
# 刪除指定名稱的物件
```

```
rm(ot_bind)
```

```
# 用ls()函數找出環境中名稱符合指定模式的物件，刪除
```

```
rm(list = ls(pattern = "^\$ot_bind"))
```

```
# 用ls()函數找出環境中全部的物件名稱，刪除
```

```
rm(list = ls())
```

知道目的，正確操作

操作處理

- 讀取
- 寫出
- 型態
- 篩選
- 排序
- 修改
- 歸戶
- 合併

管理目的

- 開啟想要處理的資料
- 保存處理完成的資料
- 正確理解輪廓及對應處理方式
- 取得想要的資料
- 賦予資料有意義順序配合操作
- 賦予資料更有意義的內容
- 統計或集結資訊
- 串聯或堆疊不同資料集

Summary

- R軟體入門
 - 名詞定義、下載安裝、RStudio設定
 - R軟體的互動模式
 - 使用套件與函數
- R軟體資料管理實作
 - 資料的來源
 - 資料管理程式撰寫
 - `data.table`套件
 - 讀取、寫出、型態、篩選
 - 排序、修改、歸戶、合併
- 開放提問時間
 - FB : 劉品崧
 - Peter Pin-Sung Liu
 - psliu520@gmail.com
 - <https://github.com/PSLiu/>



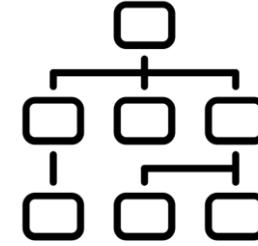
109年度R基礎課程-劉品崧老師



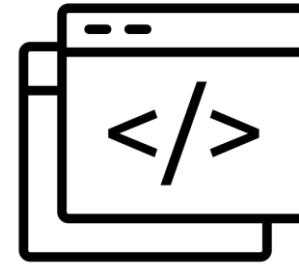
使用創用CC圖片宣告



Created by John Chapman
from the Noun Project



Created by QualityIcons
from the Noun Project



Created by SBTS
from the Noun Project