

# 資料的探索與處理

Sung-Chien Lin

2018 年 8 月 26 日

## 課程簡介

### 課程簡介

- 本次課程的目的為運用上單元學習到的若干概念，進行資料探索與處理
- 包括以下的內容：
  - 讀入資料，儲存為 **data frame** 資料形態
  - 瀏覽與檢視資料內容
  - 整理資料成適合分析的形式
  - 進行簡單分析
- 本次課程將以環境資源資料開放平臺上的空氣品質指標為例，進行上述的資料探索與處理

### 學習目標

- 能夠在 R 語言中讀入 csv 格式的資料檔案
- 能夠利用 R 語言進行資料的探索性分析
- 能夠將分析過程的程式寫入檔案，便於日後使用在同樣或類似的工作上

# 預備工作

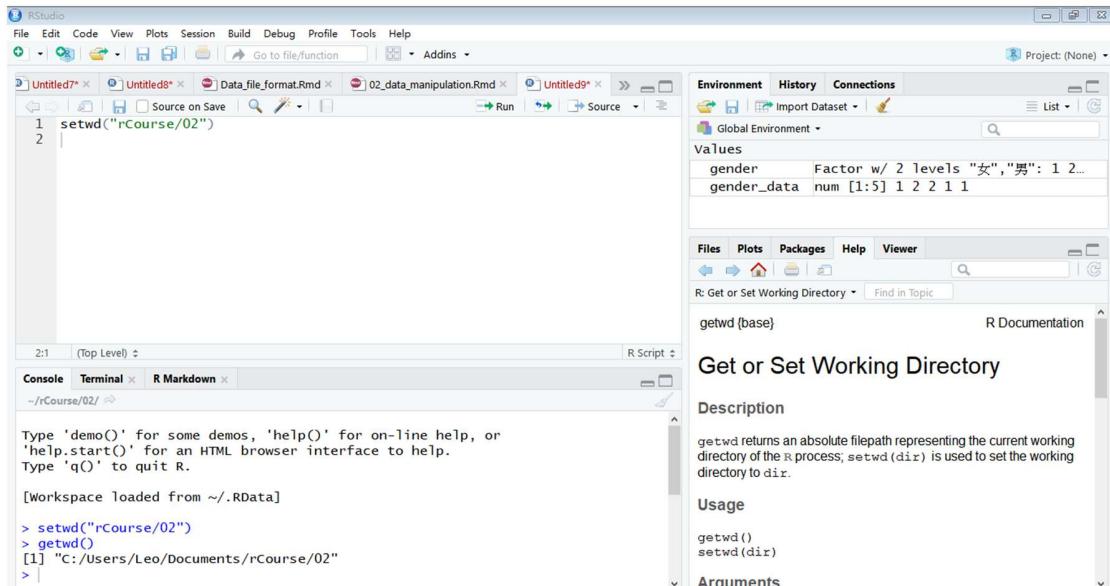
## 設定工作目錄

- 首先利用 Windows 的檔案總管，在"我的文件下"建立新的目錄 rCourse，然後在 rCourse 下建立此次工作目錄 02
- 在 Source 中增加新的 Script
- 在 Script 上設定這個任務的工作目錄

```
setwd("rCourse/02")
```

- 執行可點選 Source 右上方的 Run
- 可以在 Console 上，輸入 getwd() 檢查設定工作目錄是否成功

```
getwd()
```



- 注意：為什麼 setwd("rCourse/02")寫在 Source 上，而 getwd()卻寫在 Console？Source 與 Console 上寫入的指令是否不同？

# 讀入檔案資料

## 空氣品質指標資料來源

- 進入環境資源資料開放平臺(<https://opendata.epa.gov.tw/>)

The screenshot shows the main interface of the OpenData.epa website. At the top, there's a search bar with the placeholder "哈囉！找什麼資料嗎？" and a magnifying glass icon. Below the search bar are five key statistics: 1,236項環境資料 (1,236 environmental datasets), 46個協力單位 (46 partner units), 使用範例 (Usage examples), 17項應用服務 (17 application services), and 9,276萬次瀏覽下載 (9,276 million views/downloads). The central part of the page features a section titled "資料集類別查詢" (Dataset Category Search) with a sub-section "月熱門資料集" (Most Popular Datasets This Month). Below this are three large buttons labeled "大氣" (Atmosphere), "水" (Water), and "地" (Earth).

- 查詢空氣品質指標
- 點選空氣品質指標(AQI) (<https://opendata.epa.gov.tw/Data/Contents/AQI/>)

This screenshot shows the search results for the AQI dataset. On the left, there's a sidebar with a search form for adding filters. The main area displays a table with AQI data for various locations. The table has columns for AQI, CO, CO\_8hr, and County. The data is as follows:

AQI	CO	CO_8hr	County
44	0.7	0.5	基隆市
49	0.75	0.5	新北市
58	0.68	0.5	新北市
38	0.66	0.5	新北市
38	0.71	0.6	新北市
49	0.68	0.5	新北市
33	0.7	0.6	新北市
32	0.67	0.5	新北市

## 3 種常見的資料格式

- 環境資源資料開放平臺提供三種資料科學最常見的資料格式
  - CSV
  - XML
  - JSON

### CSV

- Comma Separated Values
- 利用逗點(,)分隔各資料
- 第一列為資料名稱
- 其他列為資料值

### XML

- eXtensible Markup Language
- 將資料表示成 xml 元素的集合，例如  
`<DATA><AQI>32</AQI><CO>0.26</CO></DATA>`
- 每一個元素由標籤(tag)構成，例如`<AQI>32</AQI>`
- 標籤分為開始標籤與結束標籤，開始標籤`<AQI>`，結束標籤`</AQI>`
- 標籤裡是資料名稱
- 開始標籤與結束標籤之間便是資料值

### JSON

- JavaScript Object Notation
- 以物件(object)的方式表示資料
- 例如：`{"AQI":"32", "CO":"0.26"}`
- 其中的 AQI 為資料名稱，32 為資料值

## 取得資料

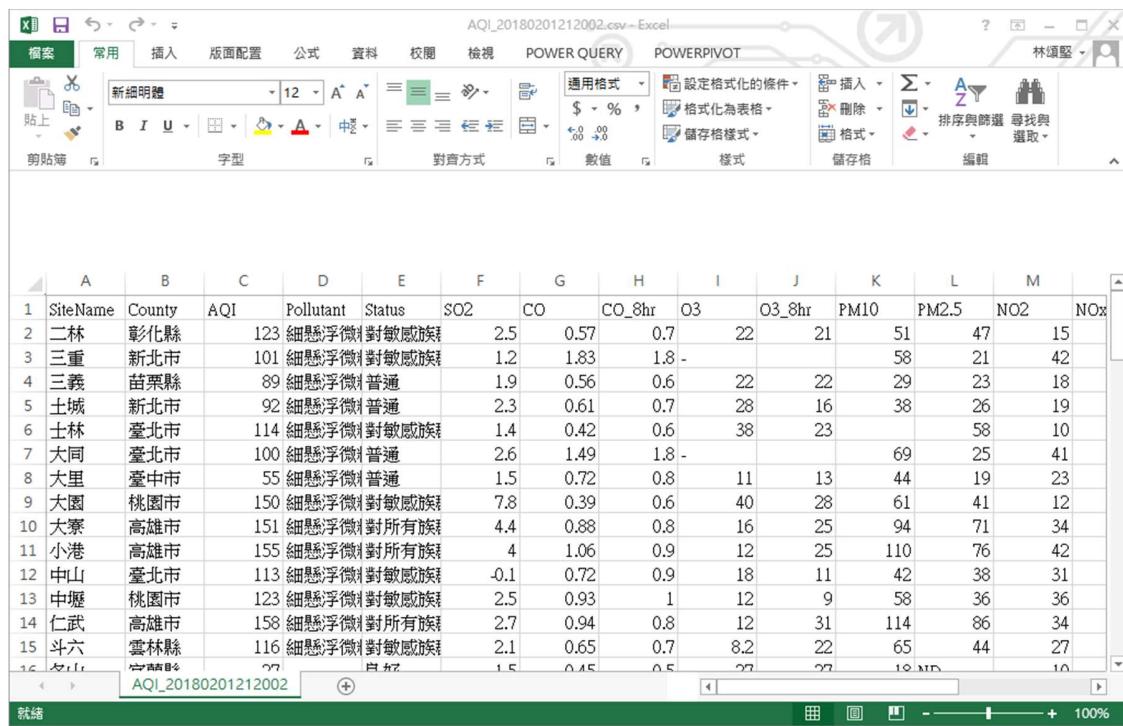
- 在本次課程中，先以最簡易的方式下載與讀取空氣品質指標的 CSV 格式檔案
- 未來可利用 R 語言的套件直接讀取 XML 和 JSON 格式資料
- 首先點選 CSV 下載，將空氣品質指標資料的 CSV 檔案移到工作目錄下



The screenshot shows the OpenData.epa website interface. At the top, there's a search bar with a magnifying glass icon and the text 'OpenData.epa'. Below the search bar, there are links for '首頁', '關於我們', '開發指南', '權限登記', and '問卷調查'. On the left, there's a sidebar with a '資料集目錄' section and a '資料查詢' section. The main content area is titled '關於資料集' and '資料檢視'. It displays a table of AQI data for various locations. The table has columns for AQI, CO, CO\_8hr, and County. The data includes rows for cities like Keelung, New Taipei City, and Taipei.

AQI	CO	CO_8hr	County
85	0.37	0.5	基隆市
118	0.53	0.6	新北市
110	0.35	0.5	新北市
93	0.79	0.7	新北市
96	0.83	0.8	新北市
121	0.82	0.9	新北市
112	0.93	0.9	新北市
108	0.84	0.9	新北市

- 用 Excel 開啟 CSV 檔



The screenshot shows an Excel spreadsheet titled 'AQI\_20180201212002.csv - Excel'. The data is presented in a table with 15 columns. The columns are labeled: SiteName, County, AQI, Pollutant, Status, SO2, CO, CO\_8hr, O3, O3\_8hr, PM10, PM2.5, NO2, and NOx. The data consists of 15 rows, each representing a different location and its environmental parameters. The table is styled with a light blue header row and white data rows.

SiteName	County	AQI	Pollutant	Status	SO2	CO	CO_8hr	O3	O3_8hr	PM10	PM2.5	NO2	NOx
2 二林	彰化縣	123	細懸浮微粒	對敏感族群	2.5	0.57	0.7	22	21	51	47	15	
3 三重	新北市	101	細懸浮微粒	對敏感族群	1.2	1.83	1.8	-		58	21	42	
4 三義	苗栗縣	89	細懸浮微粒	普通	1.9	0.56	0.6	22	22	29	23	18	
5 土城	新北市	92	細懸浮微粒	普通	2.3	0.61	0.7	28	16	38	26	19	
6 士林	臺北市	114	細懸浮微粒	對敏感族群	1.4	0.42	0.6	38	23		58	10	
7 大同	臺北市	100	細懸浮微粒	普通	2.6	1.49	1.8	-		69	25	41	
8 大里	臺中市	55	細懸浮微粒	普通	1.5	0.72	0.8	11	13	44	19	23	
9 大園	桃園市	150	細懸浮微粒	對敏感族群	7.8	0.39	0.6	40	28	61	41	12	
10 大寮	高雄市	151	細懸浮微粒	對所有族群	4.4	0.88	0.8	16	25	94	71	34	
11 小港	高雄市	155	細懸浮微粒	對所有族群	4	1.06	0.9	12	25	110	76	42	
12 中山	臺北市	113	細懸浮微粒	對敏感族群	-0.1	0.72	0.9	18	11	42	38	31	
13 中壢	桃園市	123	細懸浮微粒	對敏感族群	2.5	0.93	1	12	9	58	36	36	
14 仁武	高雄市	158	細懸浮微粒	對所有族群	2.7	0.94	0.8	12	31	114	86	34	
15 斗六	雲林縣	116	細懸浮微粒	對敏感族群	2.1	0.65	0.7	8.2	22	65	44	27	
16 大肚	台東縣	???	自???	自???	1.5	0.15	0.5	???	???	10.110	10.110	10	

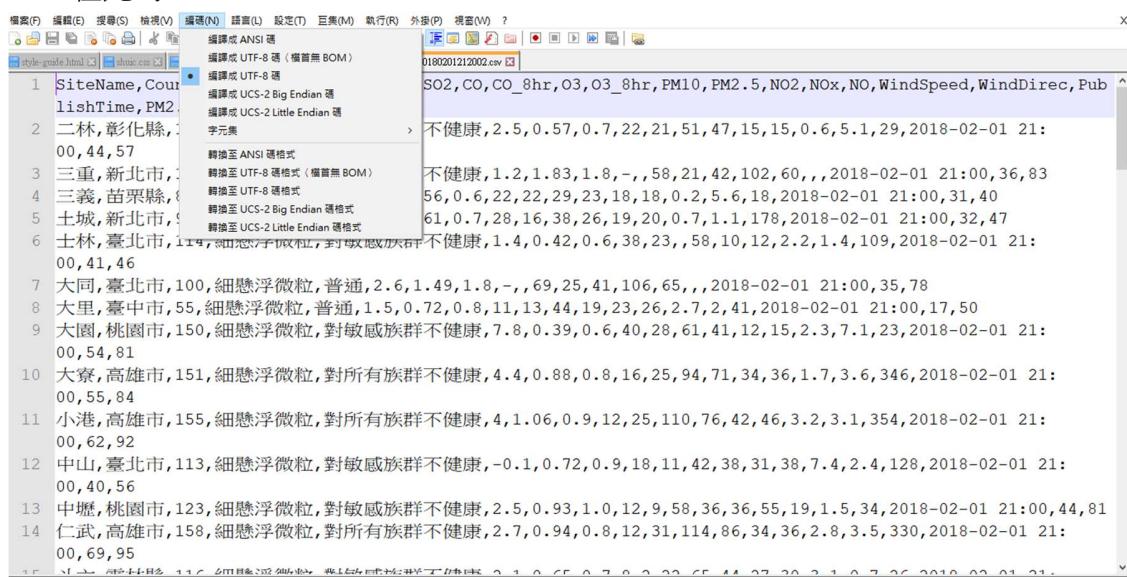
## 查閱空氣品質指標資料

- 先看看資料形式
  - 包含哪些欄位
  - 每一個欄位內的資料呈現形式
- 提出可能的問題

# 資料瀏覽

## 預先準備

- 利用文字編輯軟體 Notepad++，開啟空氣品質指標資料
- 從編碼確認 CSV 的編碼方式
- 在此為 UTF-8



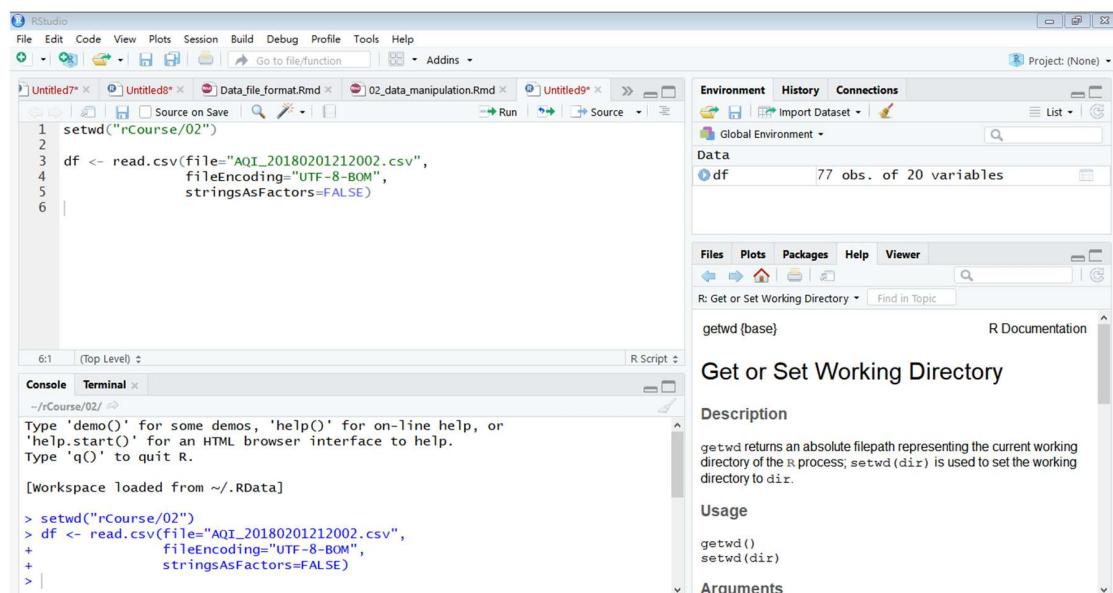
- 注意：文字檔案有許多編碼格式，在以 R 語言讀入檔案時須先知道目前處理的文字檔案使用哪一種編碼格式，才不會讀成亂碼。因此，在收到文字檔案後，可以先用 Notepad++查看，確定檔案的編碼格式。
- Notepad++可在 <https://notepad-plus-plus.org/zh/> 下載。

## 在 R 中讀入 csv 資料

- 在 Script 上輸入下面的敘述

```
# 讀入空氣品質指標資料 CSV 檔案

df <- read.csv(file="AQI_20180128061645.csv",
               fileEncoding="UTF-8-BOM",
               stringsAsFactors=FALSE)
```



- **注意：每次下載的檔名皆不同 (跟下載時間有關)**
- 點選敘述的第一行後，將滑鼠指標移到 Source 右上的 Run，點擊執行
- 查看 Environment 可發現新增一個名稱為 `df` 的 Data frame
  - 讀入的資料形成行與列為基礎的資料結構：data frame
  - 77 個 observations (列)
  - 20 個 variables (行)
- `read.csv()`的參數
  - `file` 參數為檔案名稱
  - 因為檔案編碼為 UTF-8，所以 `read.csv()`的 `fileEncoding` 參數設為"UTF-8-BOM"
  - 希望讀入的文字資料不要存成 Factor，所以將 `stringsAsFactors` 參數設為 FALSE

- 注意：**Factor** 是 R 語言中用來表示名目(nominal)與次序(ordinal)尺度的方式，例如：測站名稱(SiteName)、縣市(County)、空氣汙染指標物(Pollutant)、狀態(Status)等等。在通常的狀況下，建議開始時先以 character 的方式讀入檔案，若有需要再轉換成 Factor 的方式。

## 瀏覽讀入的資料

- 大抵來說，在拿到資料後，往往不會馬上知道如何使用
- 所以會先對資料進行探索性分析
  - 有哪些資料？
  - 資料的數量？
  - 資料的型態？
  - 資料的分布？

## 檢視讀入的資料

- 直接點選 Environment 上的 df
- 或在 Console 輸入

The screenshot shows the RStudio interface. In the center, a modal dialog titled "View(df)" displays a table of data. The table has columns: SiteName, County, AQI, Pollutant, Status, SO2, CO, CO\_8hr, O3, and O3\_8hr. The data consists of 77 rows, showing various locations like Yilan, New Taipei City, and Taichung, along with their respective AQI values and pollutant levels. To the right of the dialog, the "Environment" pane is visible, showing the variable "df" with a value of "77 obs. of 20 variables". Below the Environment pane, the "Help" pane is open, specifically for the "getwd" function, providing its description, usage, and arguments.

SiteName	County	AQI	Pollutant	Status	SO2	CO	CO_8hr	O3	O3_8hr
1 二林	彰化縣	123	細懸浮微粒	對敏感族群不健康	2.5	0.57	0.7	22	2:
2 三重	新北市	101	細懸浮微粒	對敏感族群不健康	1.2	1.83	1.8	-	1:
3 三義	苗栗縣	89	細懸浮微粒	普通	1.9	0.56	0.6	22	2:
4 土城	新北市	92	細懸浮微粒	普通	2.3	0.61	0.7	28	2:
5 士林	臺北市	114	細懸浮微粒	對敏感族群不健康	1.4	0.42	0.6	38	2:
6 大同	臺北市	100	細懸浮微粒	普通	2.6	1.49	1.8	-	1:
7 大里	臺中市	55	細懸浮微粒	普通	1.5	0.72	0.8	11	1:
8 大園	桃園市	150	細懸浮微粒	對敏感族群不健康	7.8	0.39	0.6	40	21:

- 若是資料數量太多，可以先檢視頭尾幾筆資料

## 檢視前後六筆資料

- `head(x, n)`：檢視 data frame x 的前 n 筆資料，n 預設為 6
- `tail(x, n)`：檢視 data frame x 的後 n 筆資料，n 預設為 6
- 在 Console 輸入

```
head(df)
```

```
tail(df)
```

The screenshot shows the RStudio interface with two code snippets in the console tab:

```
1 setwd("rCourse/02")
2
3 df <- read.csv(file="AQI_20180201212002.csv")
4
5 head(df)
6
7 tail(df)
```

The output of the `head(df)` command is displayed in the console, showing the first six rows of the dataset. The output of the `tail(df)` command is also shown, displaying the last six rows.

SiteName	County	AQI	Pollutant	Status	SO2	CO	CO_8hr	O3	O3_8hr	PM10	PM2.5	NO2	NOx
二林	彰化縣	123	細懸浮微粒	對敏感族群不健康	2.5	0.57	0.7	22	21	51	47	15	15
三重	新北市	101	細懸浮微粒	對敏感族群不健康	1.2	1.83	1.8	-	NA	58	21	42	102
三義	苗栗縣	89	細懸浮微粒	普通	1.9	0.56	0.6	22	22	29	29	18	18
士城	新北市	92	細懸浮微粒	普通	2.3	0.61	0.7	28	16	38	20	19	20
士林	臺北市	114	細懸浮微粒	對敏感族群不健康	1.4	0.42	0.6	38	23	NA	58	10	12
大同	臺北市	100	細懸浮微粒	普通	2.6	1.49	1.8	-	NA	69	25	41	106
					NO	WindSpeed	WindDirec	PublishTime	PM2.5_AVG	PM10_AVG			
1	0.6	5.1	29	2018-02-01 21:00		44		57					
2	60.0	NA	NA	2018-02-01 21:00		36		83					
3	0.2	5.6	18	2018-02-01 21:00		31		40					
4	0.7	1.1	178	2018-02-01 21:00		32		47					
5	2.2	1.4	109	2018-02-01 21:00		41		46					
6	65.0	NA	NA	2018-02-01 21:00		35		78					

SiteName	County	AQI	Pollutant	Status	SO2	CO	CO_8hr	O3	O3_8hr	PM10	PM2.5	NO2	NOx
橋頭	高雄市	154	細懸浮微粒	對所有族群不健康	2.9	0.89	0.8	12	29	99	7	30.0	3
頭份	苗栗縣	127	細懸浮微粒	對敏感族群不健康	2.9	0.53	0.6	34	24	58	3	13.0	7
龍潭	桃園市	102	細懸浮微粒	對敏感族群不健康	1.7	0.55	0.6	34	24	36	2	12.0	2
豐原	臺中市	56	細懸浮微粒	普通	2.1	0.56	0.6	19	23	12	12	14.0	8
關山	臺東縣	37		良好	2.4	NA	NA	29	33	32	12	8.1	7
觀音	桃園市	152	細懸浮微粒	對所有族群不健康	3.0	0.42	0.6	38	28	NA	3	7.5	1
					NO	WindSpeed	WindDirec	PublishTime	PM2.5_AVG	PM10_AVG			
72	33.0	2.2	2.9	14 2018-02-01 21:00		60		84					
73	14.0	1.2	2.7	52 2018-02-01 21:00		46		69					
74	14.0	1.8	4.6	69 2018-02-01 21:00		36		48					
75	14.0	0.3	1.8	42 2018-02-01 21:00		18		26					
76	8.8	0.7	1.7	49 2018-02-01 21:00		12		30					
77	8.0	0.5	7.0	17 2018-02-01 21:00		56		NA					

## data frame 的資料數量

- `nrow(x)` : x 上有多少列 (observations)
- `ncol(x)` : x 上有多少行 (variables)
- `dim(x)` : x 上有多少列、行
- 在 Console 輸入下面的敘述

```
nrow(df)
```

```
ncol(df)
```

```
dim(df)
```

The screenshot shows the RStudio interface. In the top-left, the code editor displays R code for reading a CSV file and setting the working directory. In the bottom-left, the console window shows the execution of `nrow(df)`, `ncol(df)`, and `dim(df)`, returning the values 77, 20, and 77 20 respectively. To the right, the Global Environment pane shows a data frame named `df` with 77 observations and 20 variables. The Help pane is open, displaying the documentation for the `getwd` function.

```
1 setwd("rCourse/02")
2
3 df <- read.csv(file="AQI_20180201212002.csv",
4   fileEncoding="UTF-8-BOM",
5   stringsAsFactors=FALSE)
6

~/rCourse/02/ >
- NOX NO WindSpeed WindDirec PublishTime PM2.5_AVG PM10_AVG
72 33.0 2.2 2.9 14 2018-02-01 21:00 60 84
73 14.0 1.2 2.7 52 2018-02-01 21:00 46 69
74 14.0 1.8 4.6 69 2018-02-01 21:00 36 48
75 14.0 0.3 1.8 42 2018-02-01 21:00 18 26
76 8.8 0.7 1.7 49 2018-02-01 21:00 12 30
77 8.0 0.5 7.0 17 2018-02-01 21:00 56 NA

> nrow(df)
[1] 77
> ncol(df)
[1] 20
> dim(df)
[1] 77 20
>
```

Get or Set Working Directory

Description

`getwd` returns an absolute filepath representing the current working directory of the R process; `setwd(dir)` is used to set the working directory to `dir`.

Usage

`getwd()`  
`setwd(dir)`

Arguments

## data frame 上各欄位的資料型態

- `str(x)` : x 上各行(variable)的資料型態
- 在 Console 輸入下面的敘述

```
str(df)
```

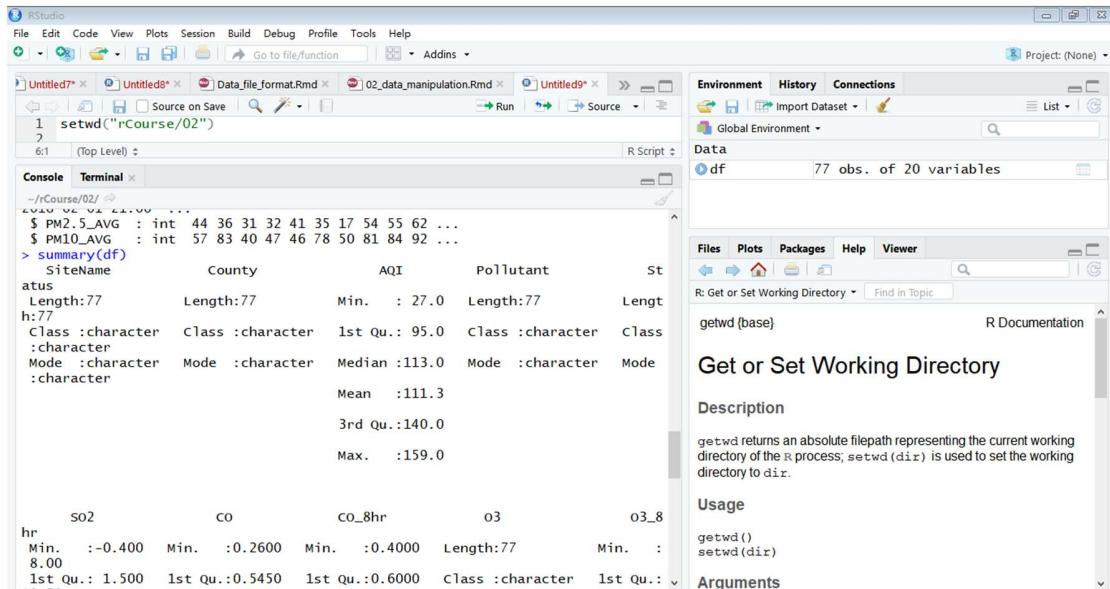
```
str(df)
'data.frame': 77 obs. of 20 variables:
 $ SiteName : chr "二林" "三重" "三義" "土城" ...
 $ County   : chr "彰化縣" "新北市" "苗栗縣" "新北市" ...
 $ AQI      : int 123 101 89 92 114 100 55 150 151 155 ...
 $ Pollutant: chr "細懸浮微粒" "細懸浮微粒" "細懸浮微粒" "細懸浮微粒" ...
 $ Status   : chr "對敏感族群不健康" "對敏感族群不健康" "普通" "普通" ...
 $ SO2      : num 2.5 1.2 1.9 2.3 1.4 2.6 1.5 7.8 4.4 4 ...
 $ CO       : num 0.57 1.83 0.56 0.61 0.42 1.49 0.72 0.39 0.88 1.06 ...
 $ CO_8hr   : num 0.7 1.8 0.6 0.7 0.6 1.8 0.8 0.6 0.8 0.9 ...
 $ O3       : chr "22" "22" "28" ...
 $ O3_8hr   : int 21 NA 22 16 23 NA 13 28 25 25 ...
 $ PM10     : int 51 58 29 38 NA 69 44 61 94 110 ...
 $ PM2.5    : chr "47" "21" "23" "26" ...
 $ NO2     : num 15 42 18 19 10 41 23 12 34 42 ...
 $ NOX     : num 15 102 18 20 12 106 26 15 36 46 ...
 $ NO      : num 0.6 60 0.2 0.7 2.2 65 2.7 2.3 1.7 3.2 ...
 $ Windspeed: num 5.1 NA 5.6 1.1 1.4 NA 2 7.1 3.6 3.1 ...
 $ Winddirec: num 29 NA 18 178 109 NA 41 23 346 354 ...
 $ PublishTime: chr "2018-02-01 21:00" "2018-02-01 21:00" "2018-02-01 21:00" ...
 2018-02-01 21:00" ...
 $ PM2.5_AVG: int 44 36 31 32 41 35 17 54 55 62 ...
 $ PM10_AVG: int 57 83 40 47 46 78 50 81 84 92 ...
```

- 目前 df 上各行的資料型態可以看到有 `int(integer)`、`num(numeric)` 和 `chr(character)`

## 資料摘要

- `summary(x)` : x 上各行(variable)的資料分布情形
- 在 Console 輸入下面的敘述

```
summary(df)
```



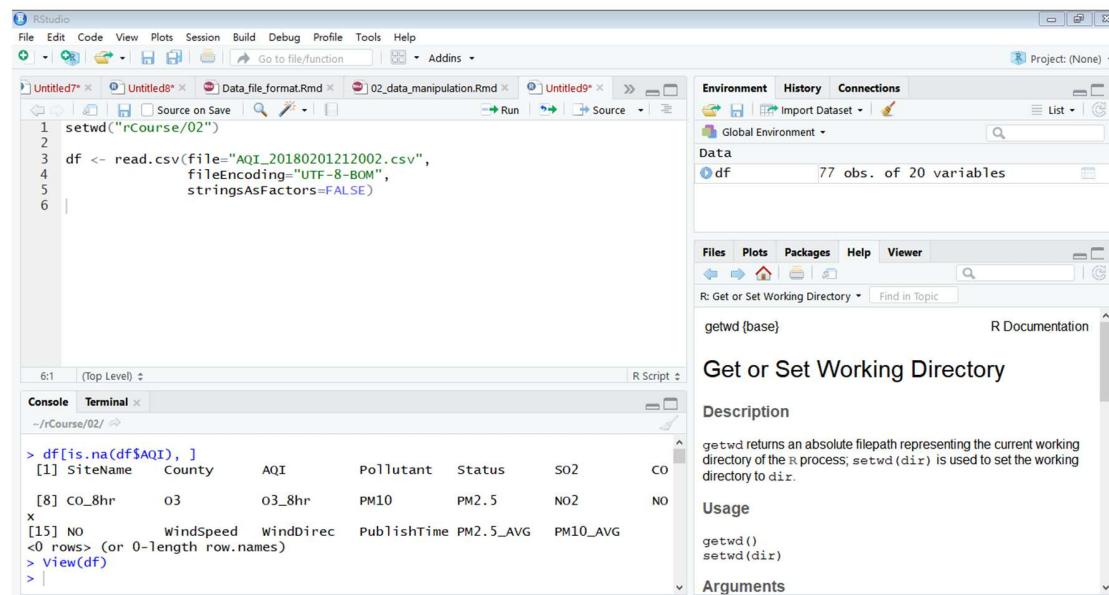
- 若是欄位資料型態為 `integer` 或 `numeric`，會呈現最小值、第一四分位數、中位數、平均數、第三四分位數、最大值等
- 若是欄位資料型態為 `character`，只會呈現它的型態
- **注意：**原本檔案中如果有遺漏的資料時，讀入資料後會以 `NA(not-available)` 方式呈現，檢查資料摘要時需要特別留意。

# 資料整理

## 去除 AQI 為 NA 的資料紀錄

- 先在 Console 查看 AQI 為 NA 的資料紀錄
- `is.na(x)`：如果  $x$  是 NA 的話，結果為 TRUE，否則為 FALSE

```
df[is.na(df$AQI), ]
```



- 這個資料中，所有的偵測站都有提供 AQI
- 如果有任一偵測站的 AQI 為 NA 的話，可以利用程式刪除該筆紀錄
- 在 Script 上輸入下面的敘述，並且執行
- `!is.na(x)`：如果  $x$  是 NA 的話，結果為 FALSE，否則為 TRUE

```
df <- df[ !is.na(df$AQI), ]
```

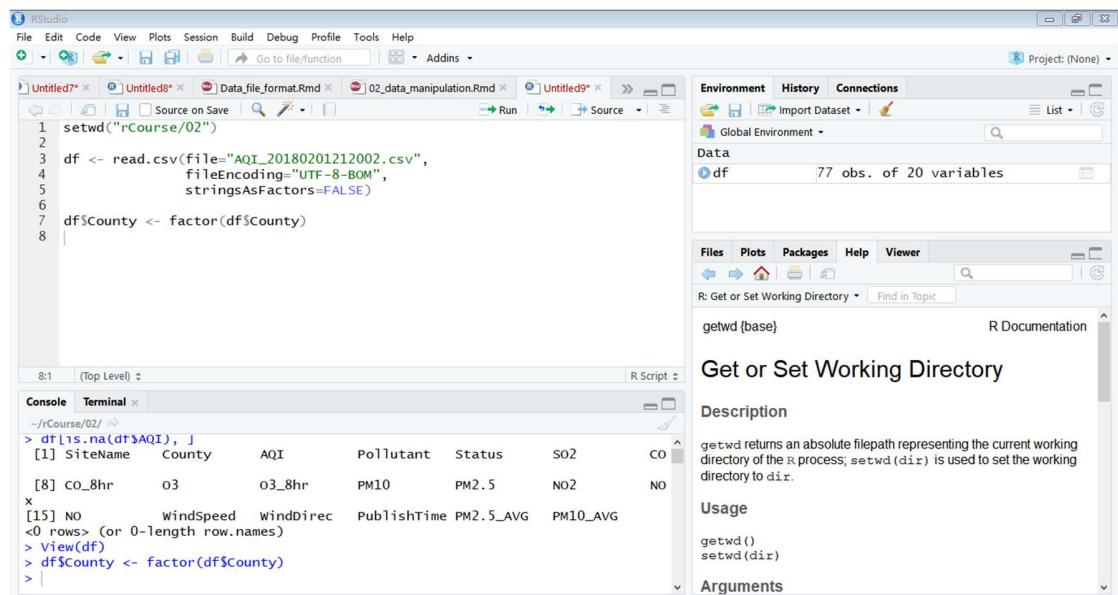
## 練習

- 請在 Console 上再次查看 `df` 的資料數量、資料型態、資料摘要，看看與先前執行有何不同

## 將 character 型態的欄位資料改為 factor

- 在 Script 上輸入下面的敘述，並且執行

```
df$County <- factor(df$County)
```



## 練習

- 在執行上面的敘述後，在 Console 再次查看資料型態與資料摘要，注意 County 欄位的結果
- 欄位資料型態為 factor，會呈現各種值出現次數。(請參考下頁的結果)

The screenshot shows the RStudio interface. In the top-left pane, there are several tabs: Untitled7\*, Untitled8\*, Data\_file\_format.Rmd, 02\_data\_manipulation.Rmd, and Untitled9\*. The Untitled9 tab is active, displaying the following R code:

```
1 setwd("rCourse/02")
2
3 df <- read.csv(file="AQI_20180201212002.csv",
4                 fileEncoding="UTF-8-BOM",
5                 stringsAsFactors=FALSE)
6
7 df$County <- factor(df$County)
8
```

In the bottom-left pane, the Console tab is selected, showing the output of the code:

```
~/rCourse/02/ > df$County <- factor(df$County)
> summary(df)
  SiteName      County       AQI     Pollutant      Status
Length:77    高雄市 :12   Min.   : 27.0  Length:77
Class :character 新北市 :12   1st Qu.: 95.0  Class :character
cter          臺北市 : 7   Median :113.0  Mode  :character
Mode :character 桃園市 : 6   Mean   :111.3   Mode  :character
cter          臺中市 : 5   3rd Qu.:140.0
雲林縣 : 4   Max.   :159.0
(oother):31
```

The right side of the interface features the Global Environment panel, which lists the dataset 'df' with 77 observations and 20 variables. Below it is the Help panel, specifically the 'Get or Set Working Directory' page. The page includes sections for Description, Usage, and Arguments.

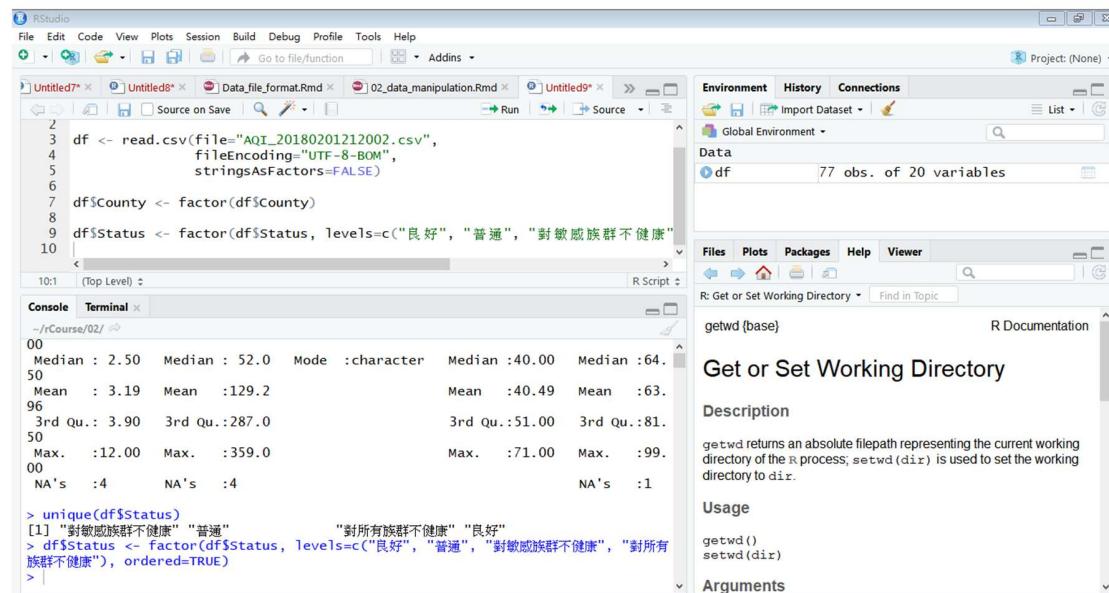
## 有順序的 factor

- 空氣品質狀態(Status)上有哪些值
- unique(x) : 列出 vector x 上的不同的資料

```
unique(df>Status)
```

- 空氣品質狀態的四種不同值，可以依據 AQI 值的大小排序，"良好" < "普通" < "對敏感族群不健康" < "對所有族群不健康"
- 在 Script 上輸入，並執行

```
df$status <- factor(df$status, levels=c("良好", "普通", "對敏感族群不健康", "對所有族群不健康"), ordered=TRUE)
```



## 練習

- 在執行上面的敘述後，在 Console 再次查看資料型態與資料摘要，注意 Status 欄位的結果

# 資料分析

## 回答以下幾個問題

- 針對某一縣市，找出它有哪幾個偵測站？
  - 關鍵點：選擇偵測站所在縣市
- 空氣品質最糟糕的偵測站？
  - 關鍵點：空氣品質指數最高的紀錄是第幾筆
- 計算汙染物是細懸浮微粒的偵測站數量？
  - 關鍵點：選擇汙染物是細懸浮微粒的偵測站，然後計算它的數量
- 找出空氣品質狀態對敏感性族群或一般族群不健康的偵測站數量？
  - 關鍵點：對空氣品質狀態統計

## 雲林縣有哪幾個偵測站

- `df$County` 是偵測站所在的縣市
- 當偵測站所在縣市是雲林縣時，`df$County=="雲林縣"`為 TRUE，否則為 FALSE
- `df$SiteName[df$County=="雲林縣"]`：傳回位於雲林縣的偵測站

```
df$County=="雲林縣"
```

```
df$SiteName[df$County=="雲林縣"]
```

The screenshot shows the RStudio interface. In the top-left pane, there are several tabs: 'Untitled7\*', 'Untitled8\*', 'Data\_file\_format.Rmd', '02\_data\_manipulation.Rmd', and 'Untitled9\*'. The 'Console' tab is active, displaying R code and its output. The code reads a CSV file 'AQI\_20180201212002.csv' into 'df', sets 'County' as a factor, and 'Status' as a factor with levels '良好', '普通', and '族群不健康'. It then filters the data for 'County == "雲林縣"' and prints the resulting 'SiteName' column. The output shows three entries: '斗六', '崙背', '麥寮', and '臺西'. The right side of the interface shows the 'Environment' and 'Global Environment' panes, which list the 'df' object. Below these is the 'Help' pane, specifically the 'Get or Set Working Directory' page, which provides information on the `getwd()` and `setwd()` functions.

## 練習

- 新北市各偵測站的空氣品質指數分別是多少

## 哪個偵測站偵測到的空氣品質最糟糕

- 空氣品質最糟糕 -> AQI 最高
- 第幾筆記錄的空氣品質最糟糕 -> which.max(df\$AQI)
- 哪個偵測站偵測到的空氣品質最糟糕 -> df\$SiteName[which.max(df\$AQI)]

The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Project:** Project: (None).
- Console:** Shows R code and its execution results. The code reads a CSV file, creates factors for County and Status, and then finds the site name with the highest AQI. The output shows the site name "復興" with an AQI of 159.
- Environment:** Shows the global environment with a dataset named "df".
- Help:** Shows the help page for the "getwd" function, which describes it as returning the current working directory.

```
2
3 df <- read.csv(file="AQI_20180201212002.csv",
4                 fileEncoding="UTF-8-BOM",
5                 stringsAsFactors=FALSE)
6
7 df$County <- factor(df$County)
8
9 df$status <- factor(df$status, levels=c("良好", "普通", "對敏感族群不健康"))
10
11 df$SiteName <- ifelse(df$County=="雲林縣", "斗六", "嘉義", "臺西")
12
13 which.max(df$AQI)
14
15 df$SiteName[which.max(df$AQI)]
16
17 df[which.max(df$AQI), c("SiteName", "AQI")]
18
19 SiteName AQI
20     復興 159
21
```

## 練習

- 空氣品質最糟糕的偵測站偵測到的 PM10 和 PM2.5 指數各是多少？

## 哪幾個偵測站偵測到的空氣品質超過 100

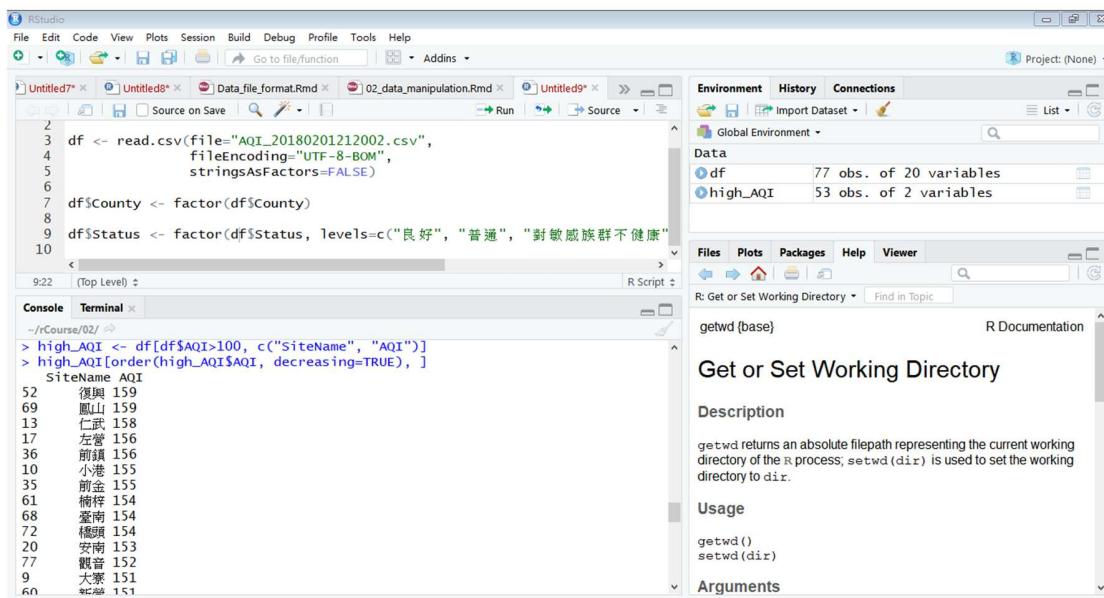
- 當偵測站的空氣品質指標超過 100 時， $\text{df}\$AQI > 100$  為 TRUE，否則為 FALSE
- $\text{df[, c("SiteName", "AQI")]}$ 列出偵測站名與它的 AQI 值

```
df[df$AQI>100, c("SiteName", "AQI")]
```

- 將這些偵測站依照他們的空氣品質指標由大到小排序

```
high_AQI <- df[df$AQI>100, c("SiteName", "AQI")]
```

```
high_AQI[order(high_AQI$AQI, decreasing=TRUE), ]
```



## 練習

- 空氣品質指標超過 100 的偵測站都在哪幾個縣市？

## 多少個偵測站的汙染物是細懸浮微粒

- `length(x)` : x 的長度，x 上的資料數量

```
df$SiteName[df$Pollutant=="細懸浮微粒"]
```

```
length(df$SiteName[df$Pollutant=="細懸浮微粒"])
```

The screenshot shows the RStudio interface with the following details:

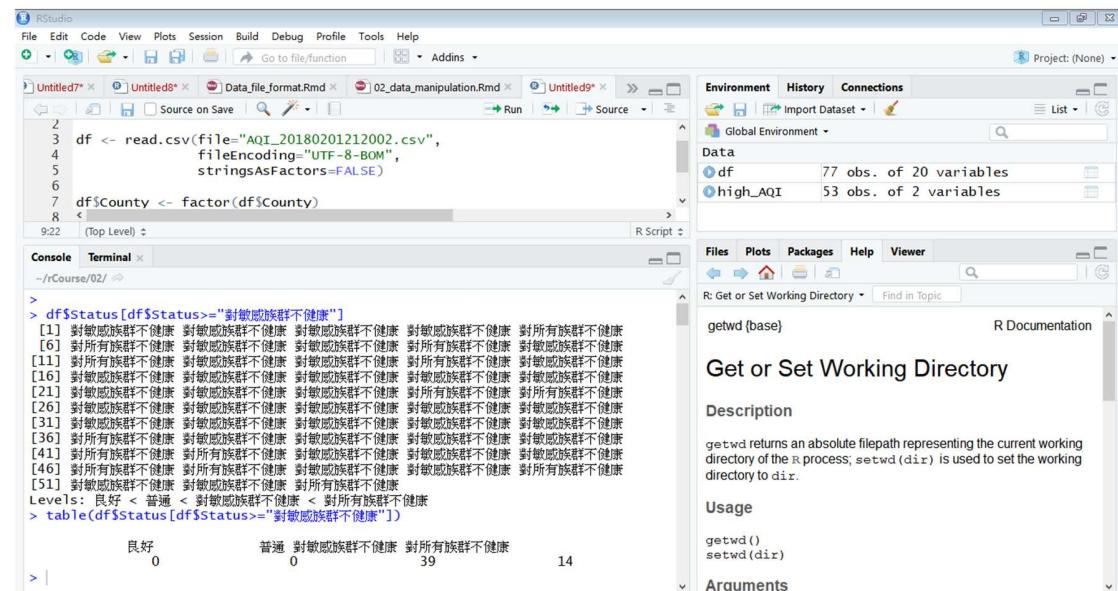
- Console:** Displays the R code and its output. The code filters the 'df' dataset to find sites with "細懸浮微粒" (PM2.5) and then counts them. The output shows 71 such sites.
- Environment:** Shows the global environment with two datasets: 'df' (77 observations, 20 variables) and 'high\_AQI' (53 observations, 2 variables).
- Help:** The right pane shows the help documentation for the `getwd` function, which returns the current working directory.

```
df$SiteName[df$Pollutant=="細懸浮微粒"]
[1] "二林"  "三重"  "三義"  "土城"  "士林"  "大同"  "大里"  "大園"  "大寮"
"小港"
[11] "中山"  "中壢"  "仁武"  "斗六"  "古亭"  "左營"  "平鎮"  "永和"  "安南"
"朴子"
[21] "汐止"  "竹山"  "竹東"  "西屯"  "沙鹿"  "宜蘭"  "忠明"  "松山"  "板橋"
"林口"
[31] "林園"  "金門"  "前金"  "前鎮"  "南投"  "屏東"  "美濃"  "苗栗"  "埔里"
"桃園"
[41] "馬公"  "馬祖"  "基隆"  "嵩背"  "淡水"  "臺東"  "善化"  "富貴角" "復興"
"湖口"
[51] "菜寮"  "新竹"  "新店"  "新莊"  "新港"  "新營"  "楠梓"  "萬里"  "萬華"
"嘉義"
[61] "彰化"  "臺西"  "臺南"  "鳳山"  "潮州"  "線西"  "橋頭"  "頭份"  "龍潭"
"豐原"
[71] "觀音"
> length(df$SiteName[df$Pollutant=="細懸浮微粒"])
[1] 71
>
```

## 依據指標等級，對敏感性族群或一般族群不健康的偵測站分別有多少？

- `table(x)`：統計 x 上各種資料的數量

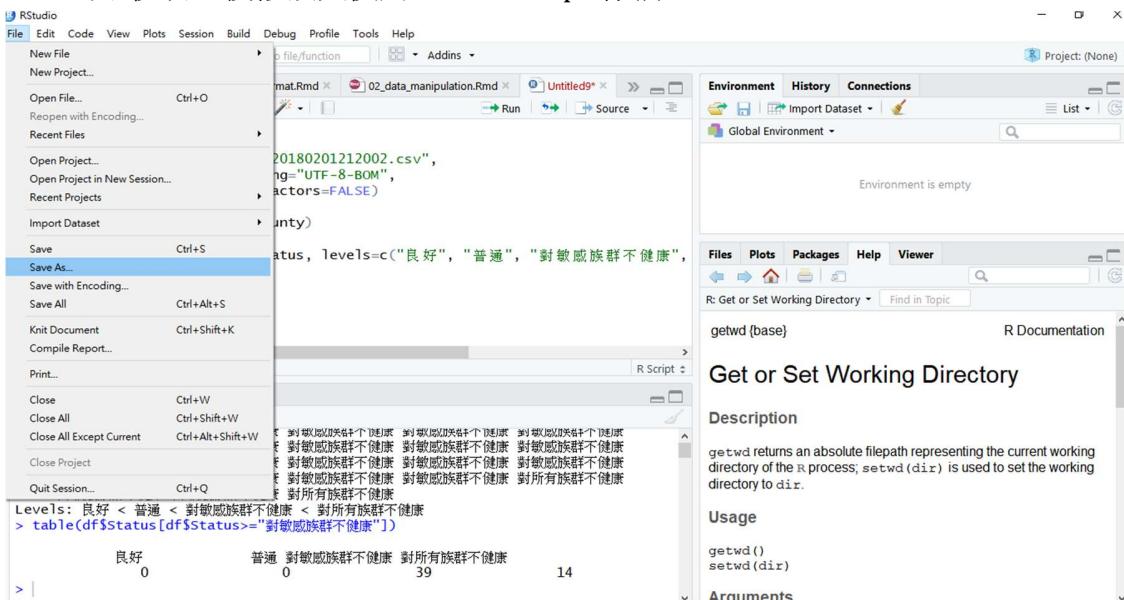
```
df$status>="對敏感族群不健康"  
  
df$status[df$status>="對敏感族群不健康"]  
  
table(df$status[df$status>="對敏感族群不健康"])
```



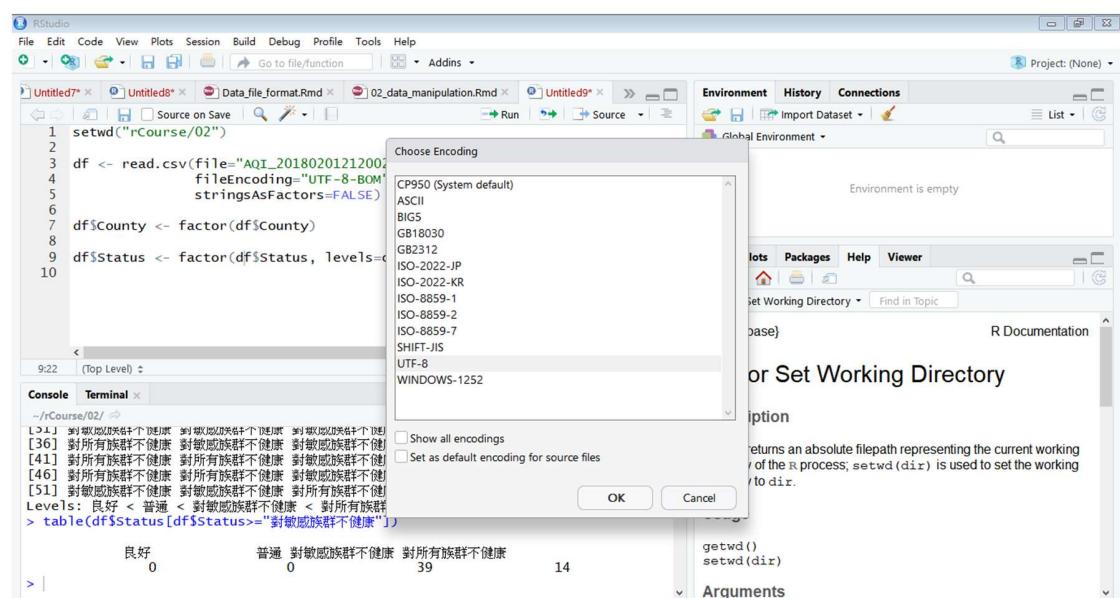
# 程式存檔

## 程式存檔

- 為了便於日後修改與使用，可將 Script 存檔



- 建議利用工具列 File 下面的 Save with Encoding 存檔



- Choose Encoding 可選擇 UTF-8，避免日後傳輸時中文編碼出問題。

- 未來有需要再開啟這個 Script 的話，可以利用 File 下的 Reopen with Encoding，再選擇 UTF-8，重新開啟。

# 本次課程小結

## 小結

- 資料處理與分析的步驟
  - 問題擬定
  - 資料取得
  - 資料清理
  - 資料分析
  - 結果解釋
- 參考資料：Shafique, U., & Qaiser, H. (2014). A comparative study of data mining process models (KDD, CRISP-DM and SEMMA). *International Journal of Innovation and Scientific Research*, 12(1), 217-222.
- 參考資料：盧安邦, & 鄭宇君. (2017). 用方法說故事: 探析電腦輔助文本分析工具在框架研究之應用. 傳播研究與實踐, 7(2), 145-178.

## 小結

- 程式的敘述在 Script 上或是在 Console 上寫？
- 如果未來還會使用或是敘述較長，可以在 Script 上編寫，便於日後執行與修改
- 如果是暫時性(一次性)而且敘述較短，在 Console 上寫，較為省事簡便

## 小結

- 本次課程主要運用 R 語言中 vector、factor 和 data frame 等的進階資料型態以及相關函數，進行資料探索與處理。
  - which.max()
  - order()
  - table()
  - ...

## 小結

- data frame 中什麼時候用 character？什麼時候用 factor？
  - 需要進行字串運算、文本處理等文字為主的運用時，可用 character
  - 需要進行統計時，可用 factor

## 延伸思考

1. 當收到一個新的資料檔案時，如何了解這些資料以便利用？
2. 在你的工作或研究中，有哪些資料可以拿來進行分析？