

# 資料探索與分析中的 Tidy 格式

Sung-Chien Lin

2018 年 8 月 27 日

## 課程簡介

### 課程簡介

- 本次課程為介紹近來利用 R 語言進行資料分析時常見的 tidy data format
- 本課程將首先介紹 tidy data format 與其套件
- 然後以上次的空氣品質指標資料為例，說明一些常用的資料分析方法

### 學習目標

- 能夠安裝套件，並且在需要時載入。
- 能夠說明使用 tidy data format 的優點，並將輸入資料表示成 tidy data format。
- 能夠運用 tidyverse 套件中常用的指令進行資料分析

# tidy data format

## data frame 與資料探索與分析

- 在上一單元，我們已經知道在 R 語言的資料探索與分析中，data frame 是相當重要的角色
- 但仍有許多處理，無法在 data frame 上完成，需要利用 vector 及 list 等資料型態
- 為此緣故，近來 R 語言的資料科學家發展出一套方式，可以在 data frame 類似的格式上完成絕大多數的資料探索與分析 --> tidy data format

## tidy data format

- 較容易進行處理
- 較容易進行視覺化
- 較容易應用資料模型

## package: tidyverse

- 目前在 R 語言已經有許多套件可以支援 tidy data format
- 所以有人將比較常用的套件集合起來，成為一個套件 tidyverse
- 注意：如果第一次使用 tidyverse 套件，需要在 Console 上先安裝該套件
- 在 Console 上輸入

```
install.packages("tidyverse")
```

- 完成安裝後，載入套件
- 注意：以後每次開啟 RStudio，便需要重新載入

## wide data format vs. long data format

- data frame 依據其上的資料欄位形式分為：
- wide data format :
  - 例如：每一年各城市的平均氣溫

##	year	Taipei	Taichung	Kaohsiung	Hualien
## 1	2015	26	31	28	22
## 2	2016	27	34	24	33
## 3	2017	24	21	27	23

- long data format

##	year	city	temp
## 1	2015	Taipei	26
## 2	2016	Taipei	27
## 3	2017	Taipei	24
## 4	2015	Taichung	31
## 5	2016	Taichung	34
## 6	2017	Taichung	21
## 7	2015	Kaohsiung	28
## 8	2016	Kaohsiung	24
## 9	2017	Kaohsiung	27
## 10	2015	Hualien	22
## 11	2016	Hualien	33
## 12	2017	Hualien	23

- 在 long data format 上，每一行上只有一個觀測結果，在上面的例子中是每一年各城市的平均氣溫(temp)

## R 語言中使用 tidy data format

- 盡量以 long data format 為主
- 若是原本的資料為 wide format，可以改為 long format

## tidy data format 使用的資料型態：tibble

- tibble 類似 data frame，但兩者間仍有不同
- 在運用 tidyverse 套件時，data frame 自動轉換為 tibble

## 預備工作

### 設定工作目錄

- 以檔案總管在「我的文件」下的「rCourse」目錄內新增工作目錄「03」
- 開啟新的 Script
- 在 Script 上，設定工作目錄

```
setwd("rCourse/03")
```

### 載入套件

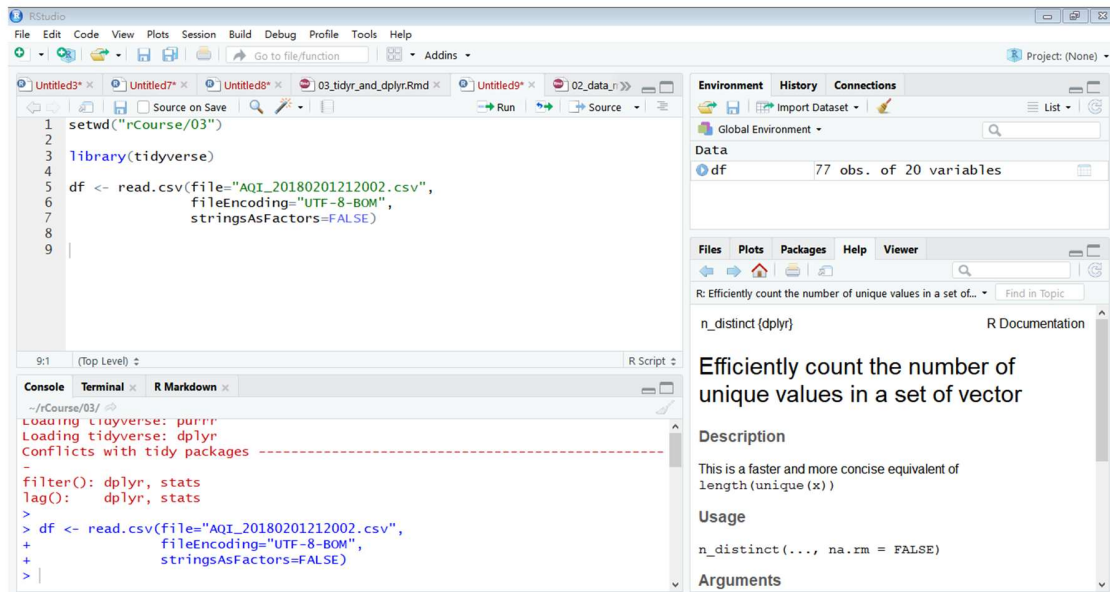
- 載入 tidyverse 套件

```
library(tidyverse)
```

### 讀取空氣品質指標資料進行資料分析

- 將上次課程的空氣品質指標資料複製到新工作目錄中
- 讀取空氣品質指標資料的 CSV 檔案
- 在 Script 上輸入

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



- 執行後，可先在 Console 或 Environment 上查看 df

## tidy data format 的資料分析方法

### 幾個常用的 tidyverse 方法

- 選取 tibble 中的幾個欄位：select()
- 依照紀錄位置選取 tibble 中的紀錄：slice()
- 根據條件選取 tibble 中的紀錄：filter()
- 增加或修改 tibble 的欄位：mutate()

### 幾個常用的 tidyverse 方法

- 依照某個欄位的資料數值排列紀錄：arrange()
- 依照某個欄位的資料數值選出前幾筆：top\_n()
- 依照某個欄位的資料數值將紀錄分群：group\_by()
- 對紀錄進行彙整(加總、平均、...)：summarise()

## 應用 tidyverse 進行分析

### 範例練習

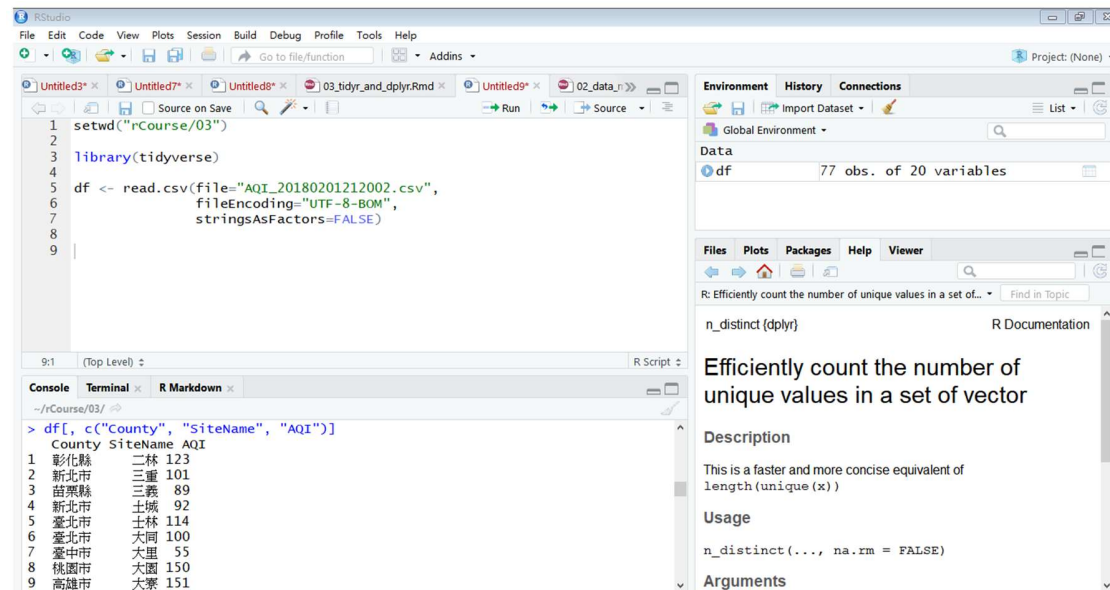
- 回答以下幾個問題
  - 選取空氣品質指標資料中的縣市、偵測站與空氣品質指標
  - 選取空氣品質指標資料中前五筆及後五筆
  - 選取空氣品質指標資料中 AQI 值大於或等於 120 的紀錄
  - 按照 AQI 的值，由大到小排序空氣品質指標資料
  - 選取空氣品質指標資料中 AQI 值最大的五筆紀錄
  - 計算各縣市的偵測站數目



## 選取空氣品質指標資料中的縣市、偵測站與空氣品質指標

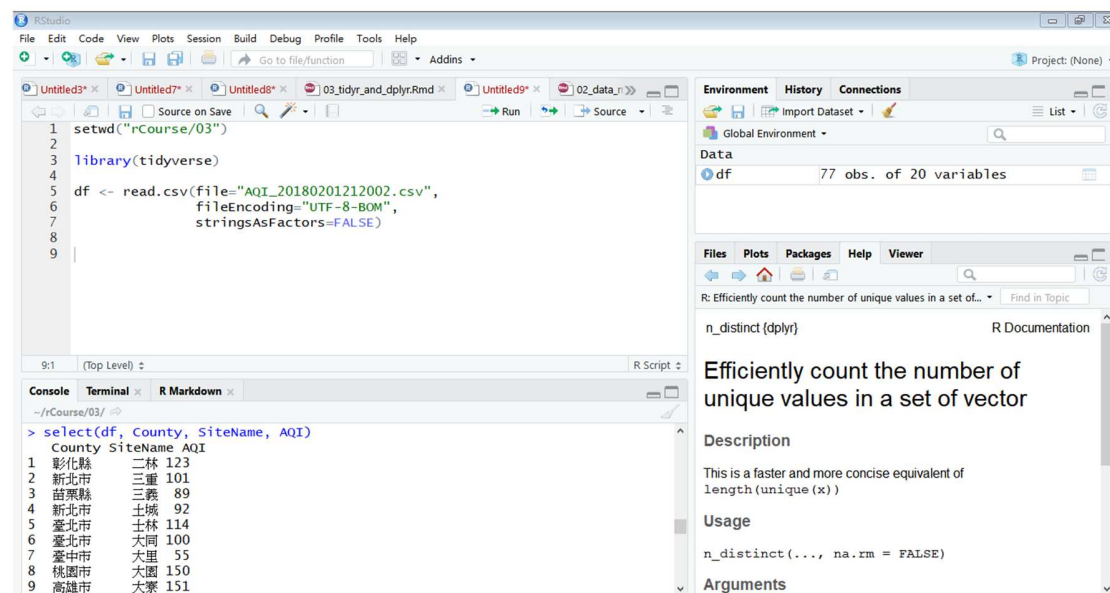
- 提示：選取縣市、偵測站與空氣品質指標等三個欄位
- 先前的做法(利用 data frame 上的欄位名稱)

```
df[, c("County", "SiteName", "AQI")]
```



- 在 tidy 的做法(利用 select()方法，加上欄位名稱)

```
select(df, County, SiteName, AQI)
```



## 選取空氣品質指標資料中前五筆及後五筆

- 提示：依照前五筆及後五筆等位置選取紀錄
- 先前的做法(根據 data frame 上的位置)

```
df[c(1:5, (nrow(df)-4):nrow(df)),]
```

The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 setwd("rCourse/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

The console shows the result of the command `df[c(1:5, (nrow(df)-4):nrow(df)),]`:

```
> df[c(1:5, (nrow(df)-4):nrow(df)), ]
  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 21
2 三義 新北市 101 細懸浮微粒 對敏感族群不健康 1.2 1.83 1.8 - 22
3 三義 苗栗縣 89 細懸浮微粒 普通 1.9 0.56 0.6 22 22
4 土城 新北市 92 細懸浮微粒 普通 2.3 0.61 0.7 28 16
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 23
73 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
74 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
75 豐原 臺中市 56 細懸浮微粒 普通 2.1 0.56 0.6 19 23
76 關山 臺東縣 37 良好 2.4 NA NA 29 33
77 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
PM10 PM2.5 NO2 NOx NO WindSpeed WindDirec PublishTime PM2.5_AVG
1 51 47 15.0 15.0 0.6 5.1 29 2018-02-01 21:00 44
```

The Environment pane shows the data frame 'df' with 77 observations and 20 variables.

- 在 tidy 的做法(利用 slice() 方法，加上紀錄的位置)

```
slice(df, c(1:5, (nrow(df)-4):nrow(df)))
```

The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 setwd("rCourse/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

The console shows the result of the command `slice(df, c(1:5, (nrow(df)-4):nrow(df)))`:

```
> slice(df, c(1:5, (nrow(df)-4):nrow(df)))
# A tibble: 10 x 20
  SiteName County AQI Pollutant Status SO2 CO CO_8hr O3 O3_8hr
  <chr> <chr> <int> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
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 21
2 三義 新北市 101 細懸浮微粒 對敏感族群不健康 1.2 1.83 1.8 - 22
3 三義 苗栗縣 89 細懸浮微粒 普通 1.9 0.56 0.6 22 22
4 土城 新北市 92 細懸浮微粒 普通 2.3 0.61 0.7 28 16
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 23
6 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
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
8 豐原 臺中市 56 細懸浮微粒 普通 2.1 0.56 0.6 19 23
9 關山 臺東縣 37 良好 2.4 NA NA 29 33
10 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
```

The Environment pane shows the data frame 'df' with 77 observations and 20 variables.

## 選取空氣品質指標資料中 AQI 值大於或等於 120 的紀錄

- 提示：根據條件「AQI 值大於或等於 120」選取紀錄
- 先前的做法(利用條件索引)

```
df[df$AQI>=120,]
```

The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 setwd("~/Course/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

The console output shows the result of the indexing operation:

```
# PM2.5_AVG <1nt>, PM10_AVG <1nt>
> df[df$AQI>=120, ]
  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 21
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 28
9 大寮 高雄市 151 細懸浮微粒 對所有族群不健康 4.4 0.88 0.8 16 25
10 小港 高雄市 155 細懸浮微粒 對所有族群不健康 4.0 1.06 0.9 12 25
12 中壢 桃園市 123 細懸浮微粒 對敏感族群不健康 2.5 0.93 1.0 12 9
13 仁武 高雄市 158 細懸浮微粒 對所有族群不健康 2.7 0.94 0.8 12 31
17 左營 高雄市 156 細懸浮微粒 對所有族群不健康 3.3 0.97 0.8 13 36
18 平鎮 桃園市 136 細懸浮微粒 對敏感族群不健康 1.7 0.64 0.8 26 16
20 安南 臺南市 153 細懸浮微粒 對所有族群不健康 1.9 0.68 0.8 14 35
21 朴子 嘉義縣 142 細懸浮微粒 對敏感族群不健康 4.7 0.63 0.7 16 30
23 竹山 南投縣 131 細懸浮微粒 對敏感族群不健康 0.6 0.67 0.7 6.7 20
25 龍崎 嘉義縣 155 細懸浮微粒 對敏感族群不健康 2.3 0.00 0.8 14 35
```

The Environment pane shows the data frame 'df' with 77 observations and 20 variables.

- 在 tidy 的做法(利用 filter() 方法與條件)

```
filter(df, AQI>=120)
```

The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 setwd("~/Course/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

The console output shows the result of the filter operation:

```
77 NA
> filter(df, AQI>=120)
  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 21
2 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 28
3 大寮 高雄市 151 細懸浮微粒 對所有族群不健康 4.4 0.88 0.8 16 25
4 小港 高雄市 155 細懸浮微粒 對所有族群不健康 4.0 1.06 0.9 12 25
5 中壢 桃園市 123 細懸浮微粒 對敏感族群不健康 2.5 0.93 1.0 12 9
6 仁武 高雄市 158 細懸浮微粒 對所有族群不健康 2.7 0.94 0.8 12 31
7 左營 高雄市 156 細懸浮微粒 對所有族群不健康 3.3 0.97 0.8 13 36
8 平鎮 桃園市 136 細懸浮微粒 對敏感族群不健康 1.7 0.64 0.8 26 16
9 安南 臺南市 153 細懸浮微粒 對所有族群不健康 1.9 0.68 0.8 14 35
10 朴子 嘉義縣 142 細懸浮微粒 對敏感族群不健康 4.7 0.63 0.7 16 30
11 竹山 南投縣 131 細懸浮微粒 對敏感族群不健康 0.6 0.67 0.7 6.7 20
```

The Environment pane shows the data frame 'df' with 77 observations and 20 variables.

## 按照 AQI 的值，由大到小排序空氣品質指標資料

- 提示：依照 AQI 欄位的數值排列紀錄
- 先前的做法
  - `order()`：資料的大小順序，`decreasing=TRUE` 表示由大到小

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

The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 setwd("~/rCourse/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

The console shows the execution of the command `df[order(df$AQI, decreasing=TRUE), ]` and the resulting data frame:

```
32      69
33      NA
> df[order(df$AQI, decreasing=TRUE), ]
  SiteName County AQI Pollutant Status SO2 CO CO_8hr O3 O3_8hr
52 復興 高雄市 159 細懸浮微粒 對所有族群不健康 3.2 1.14 1.0 9 27
69 鳳山 高雄市 159 細懸浮微粒 對所有族群不健康 4.1 1.01 1.0 9.3 22
13 仁武 高雄市 158 細懸浮微粒 對所有族群不健康 2.7 0.94 0.8 12 31
17 左營 高雄市 156 細懸浮微粒 對所有族群不健康 3.3 0.97 0.8 13 36
36 前鎮 高雄市 156 細懸浮微粒 對所有族群不健康 2.5 1.07 0.9 11 29
10 小港 高雄市 155 細懸浮微粒 對所有族群不健康 4.0 1.06 0.9 12 25
35 前金 高雄市 155 細懸浮微粒 對所有族群不健康 2.3 0.99 0.8 14 35
61 楠梓 高雄市 154 細懸浮微粒 對所有族群不健康 2.3 0.86 0.8 13 33
68 臺南 臺南市 154 細懸浮微粒 對所有族群不健康 2.3 0.71 0.8 14 34
72 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
```

The Environment pane shows the data frame `df` with 77 observations and 20 variables. The right pane shows the documentation for `n_distinct(dplyr)`.

- 在 tidy 的做法

`arrange(df, desc(AQI))`

The screenshot shows the RStudio interface. The script editor on the left contains the following code:

```
1 setwd("rCourse/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

The console on the bottom left shows the output of the command `arrange(df, desc(AQI))`, displaying a table with 12 rows and 10 columns:

	SiteName	County	AQI	Pollutant	Status	SO2	CO	CO_8hr	O3	O3_8hr
1	復興	高雄市	159	細懸浮微粒	對所有族群不健康	3.2	1.14	1.0	9	27
2	鳳山	高雄市	159	細懸浮微粒	對所有族群不健康	4.1	1.01	1.0	9.3	22
3	仁武	高雄市	158	細懸浮微粒	對所有族群不健康	2.7	0.94	0.8	12	31
4	左營	高雄市	156	細懸浮微粒	對所有族群不健康	3.3	0.97	0.8	13	36
5	前鎮	高雄市	156	細懸浮微粒	對所有族群不健康	2.5	1.07	0.9	11	29
6	小港	高雄市	155	細懸浮微粒	對所有族群不健康	4.0	1.06	0.9	12	25
7	前金	高雄市	155	細懸浮微粒	對所有族群不健康	2.3	0.99	0.8	14	35
8	楠梓	高雄市	154	細懸浮微粒	對所有族群不健康	2.3	0.86	0.8	13	33
9	臺南	臺南市	154	細懸浮微粒	對所有族群不健康	2.3	0.71	0.8	14	34
10	楠梓	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
11	安南	臺南市	153	細懸浮微粒	對所有族群不健康	1.9	0.68	0.8	14	35
12	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

The environment pane on the right shows the variable `df` with 77 observations and 20 variables. The R Documentation pane on the bottom right shows the documentation for `n_distinct(dplyr)`, which is used to efficiently count the number of unique values in a set of vector.



## 選取空氣品質指標資料中 AQI 值最大的五筆紀錄

- 提示：依照 AQI 欄位的數值選出前幾筆
- 先前的做法 (利用 `order()` 找出資料大小順序，選出最前面的五筆)

```
df[order(df$AQI, decreasing=TRUE)[1:5], ]
```

```
1 setwd("~/Course/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

	SiteName	County	AQI	Pollutant	Status	SO2	CO	CO_8hr	O3	O3_8hr
52	復興	高雄市	159	細懸浮微粒	對所有族群不健康	3.2	1.14	1.0	9	27
69	鳳山	高雄市	159	細懸浮微粒	對所有族群不健康	4.1	1.01	1.0	9.3	22
13	仁武	高雄市	158	細懸浮微粒	對所有族群不健康	2.7	0.94	0.8	12	31
17	左營	高雄市	156	細懸浮微粒	對所有族群不健康	3.3	0.97	0.8	13	36
36	前鎮	高雄市	156	細懸浮微粒	對所有族群不健康	2.5	1.07	0.9	11	29

- 在 tidy 的做法 (利用 `top_n()` 方法)

```
top_n(df, 5, AQI)
```

```
1 setwd("~/Course/03")
2
3 library(tidyverse)
4
5 df <- read.csv(file="AQI_20180201212002.csv",
6               fileEncoding="UTF-8-BOM",
7               stringsAsFactors=FALSE)
8
9
```

	PM10	PM2.5	NO2	NOX	NO	WindSpeed	WindDirec	PublishTime	PM2.5_AVG
1	114	86	34	36	2.8	3.5	330.0	2018-02-01 21:00	69
2	117	83	33	34	1.5	3.7	1.6	2018-02-01 21:00	65
3	115	74	38	42	4.1	1.9	335.0	2018-02-01 21:00	65
4	109	79	39	44	4.8	1.5	144.0	2018-02-01 21:00	69
5	120	94	39	40	1.4	2.0	354.0	2018-02-01 21:00	71

## 組合多個函數

### %>%(pipe)

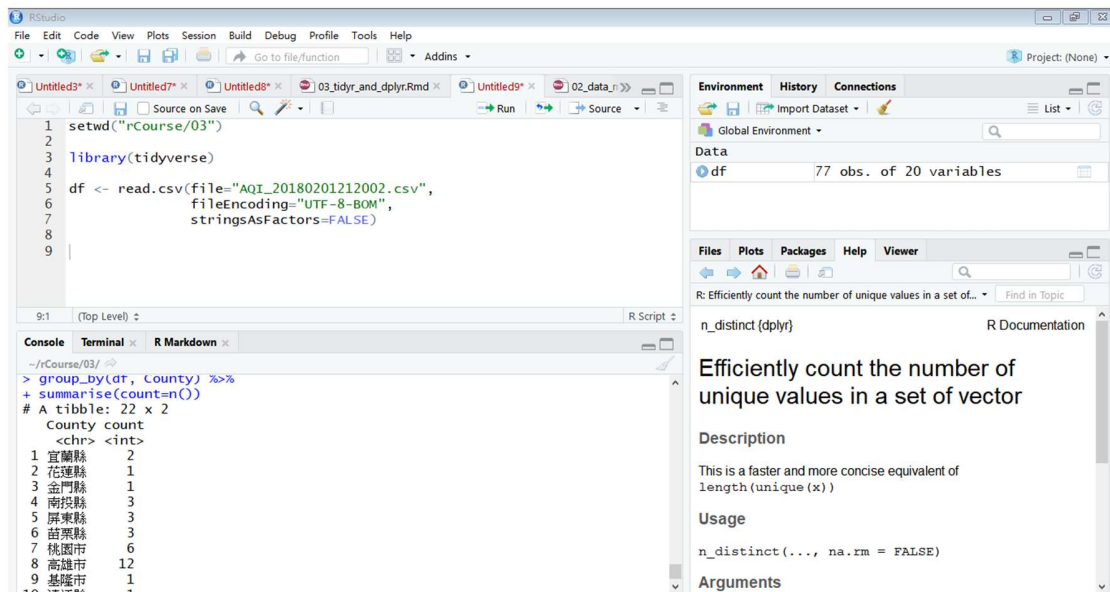
- 當較複雜的運算，無法只用單一函數完成時，**tidyverse** 提供了一個方法可以組合多個函數
- **%>%** (稱為 **pipe**) 是 **tidy** 中相當重要的方法，可以將多個函數串接使用
- **%** 會將函數的運算結果導入下一個函數，做為下一個函數的第一個參數
- 例如以上面的「選取空氣品質指標資料中 AQI 值最大的五筆紀錄」為例
  - 先依照 AQI 欄位的數值排列紀錄
  - 再依據前五筆的位置取出紀錄

```
arrange(df, desc(AQI)) %>%  
  slice(5)
```

## 計算各縣市的偵測站數目

- 提示：先依照縣市欄位的資料將紀錄分群，再彙整統計每一分群的資料數量

```
group_by(df, County) %>%  
summarise(count=n())
```



- 說明：計算各縣市的偵測站數目時，首先將 **tibble** 資料依照各縣市分群，然後統計每一個分群上的資料數量。
  - `group_by(df, County)`：依照各縣市將資料分群
  - `%>%`：將資料分群的結果導入下一個函數
  - `n()`：統計資料數量
  - `summarise(count=n())`：統計每一個分群上的資料數量
  - `summarise()`的第一個參數是 `group_by()`的計算結果



## 對各縣市偵測站數目的結果依大小排序

- 先依照縣市欄位的資料將紀錄分群，其次彙整統計每一分群的資料數量，最後依據資料數量大小排序
- 將前面各縣市的偵測站數目的計算結果導入排序
  - `arrange(desc(count))`：依據 `count` 的大小，由大到小排序
  - 注意：`arrange()` 的第一個參數是各縣市的偵測站數目的計算結果
  - `desc(count)`：由大到小排序

```
group_by(df, County) %>%  
  summarise(count=n()) %>%  
  arrange(desc(count))
```

The screenshot shows the RStudio interface. The script in the editor is as follows:

```
1 setwd("~/rCourse/03")  
2  
3 library(tidyverse)  
4  
5 df <- read.csv(file="AQI_20180201212002.csv",  
6               fileEncoding="UTF-8-BOM",  
7               stringsAsFactors=FALSE)  
8  
9  
10  
11  
12
```

The console output shows the execution of the following commands:

```
> group_by(df, County) %>%  
+ summarise(count=n()) %>%  
+ arrange(desc(count))  
# A tibble: 22 x 2  
#   County count  
#   <chr> <int>  
1 高雄縣 12  
2 新北市 12  
3 臺北市 7  
4 桃園市 6  
5 臺中市 5  
6 雲林縣 4
```

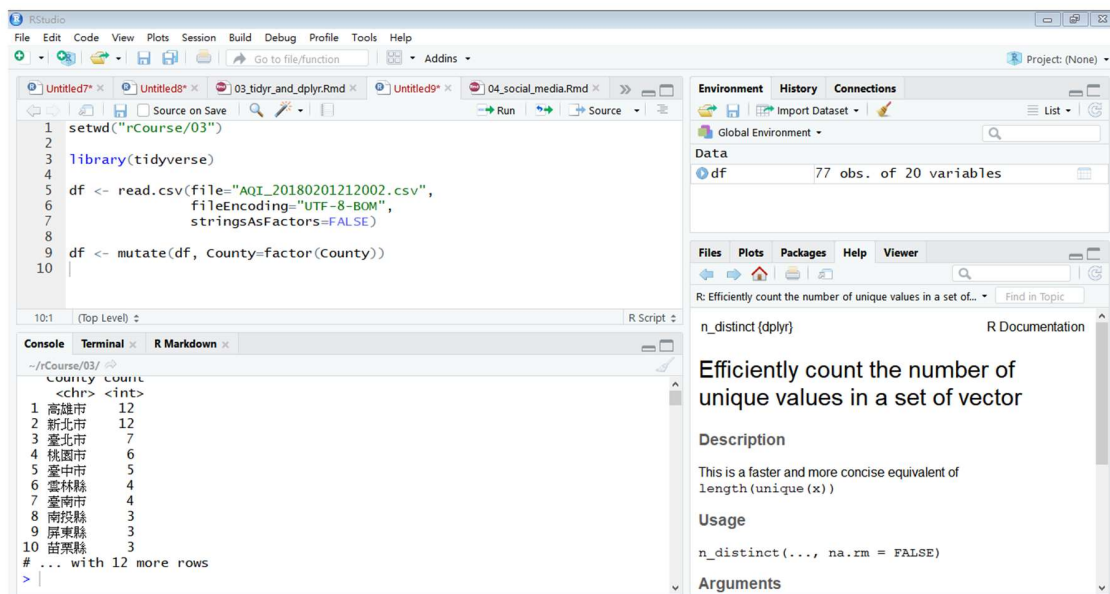
The right sidebar shows the Environment pane with 'df' having 77 observations and 20 variables. The Viewer pane displays the documentation for `n_distinct()` from the dplyr package.

## 編輯 tibble 資料

### 將縣市欄位改為 factor 型態

- 提示：修改 tibble 上的縣市欄位
- 在 Script 上輸入並執行

```
df <- mutate(df, County=factor(County))
```

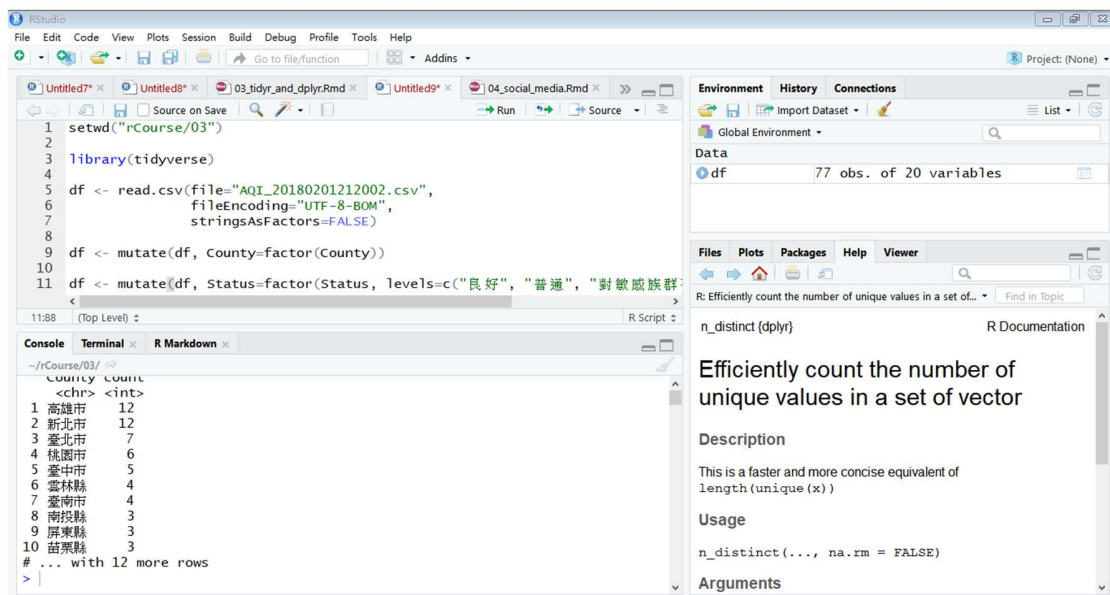


- `mutate(df, County=factor(County))`：將 County 上的資料改為 factor 型態

## 將空氣品質狀態欄位改為有順序的 factor 型態

- 提示：修改 tibble 上的空氣品質狀態欄位
- 在 Script 上輸入並執行

```
df <- mutate(df, Status=factor(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))
```



- `factor(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)`：將 Status 上的資料，設為 factor 型態，並且依照 levels 上的次序("良好", "普通", "對敏感族群不健康", "對所有族群不健康")設定其順序。

## 資料分析

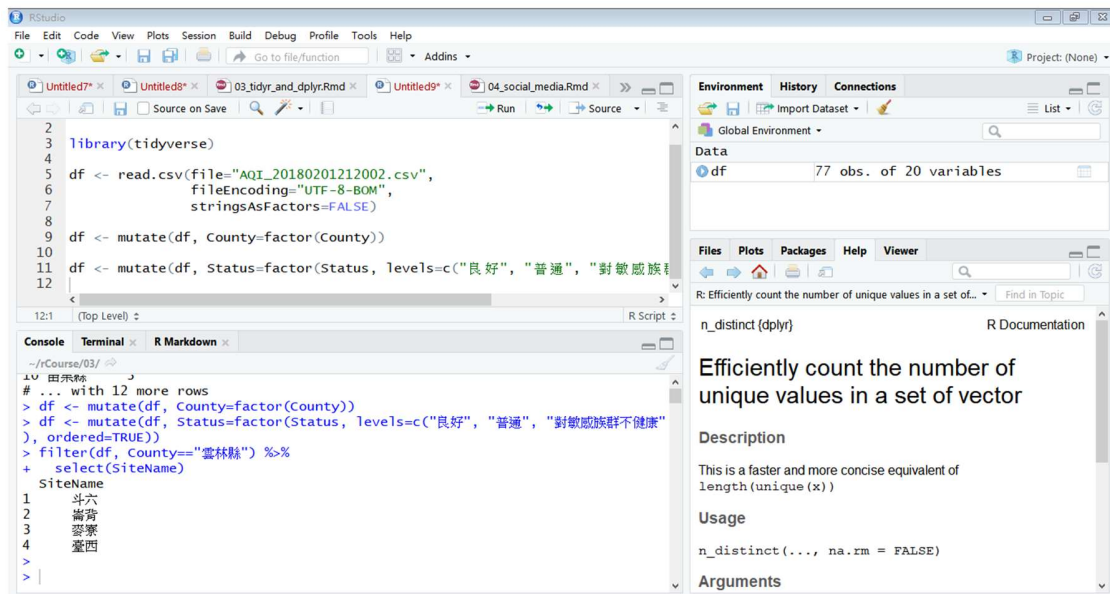
### 回答以下幾個問題

- 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
  - 提示：1. 根據 XXX 條件選取紀錄 2. 選取欄位
- 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
  - 提示：1. 依照某個欄位的資料數值選出前幾筆 2. 選取欄位
- 找出污染物是細懸浮微粒的偵測站數量？
  - 關鍵點：選擇污染物是細懸浮微粒的偵測站，然後計算它的數量
  - 提示：\_\_\_\_\_
- 計算各種空氣品質狀態的偵測站數量？
  - 關鍵點：對各種空氣品質狀態統計
  - 提示：\_\_\_\_\_

## 雲林縣有哪幾個偵測站

- 找出雲林縣的紀錄
- 選擇偵測站名稱

```
filter(df, County=="雲林縣") %>%  
select(SiteName)
```



## 練習

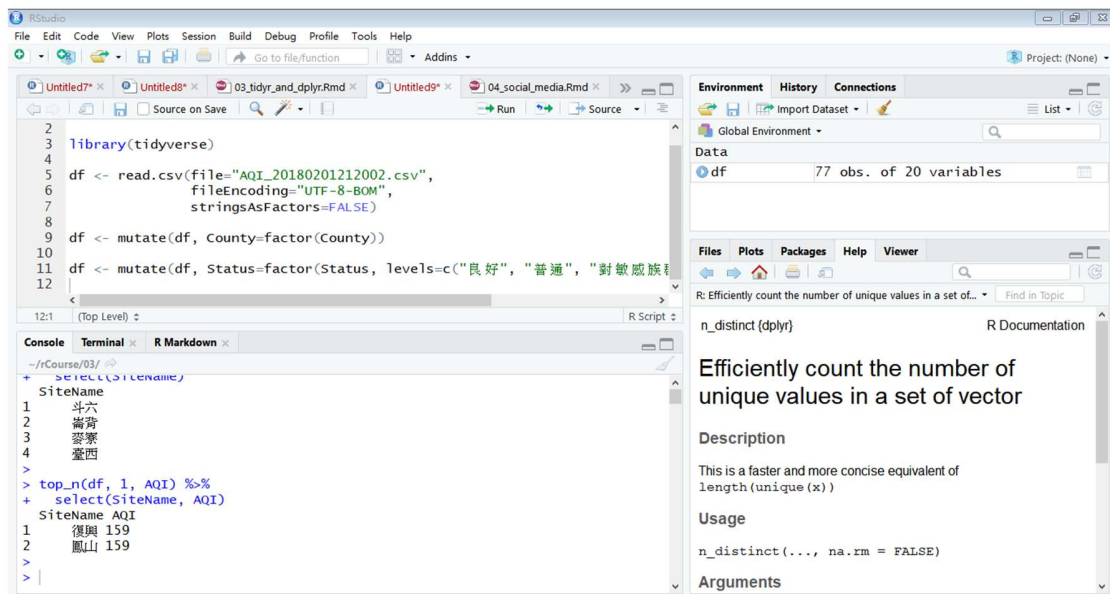
- 新北市各偵測站的空氣品質指數分別是多少  
– 提示：

## 哪個偵測站偵測到的空氣品質最糟糕

- 找到 AQI 最高的偵測站
- 列出偵測站名稱及 AQI 值

```
top_n(df, 1, AQI) %>%
```

```
select(SiteName, AQI)
```



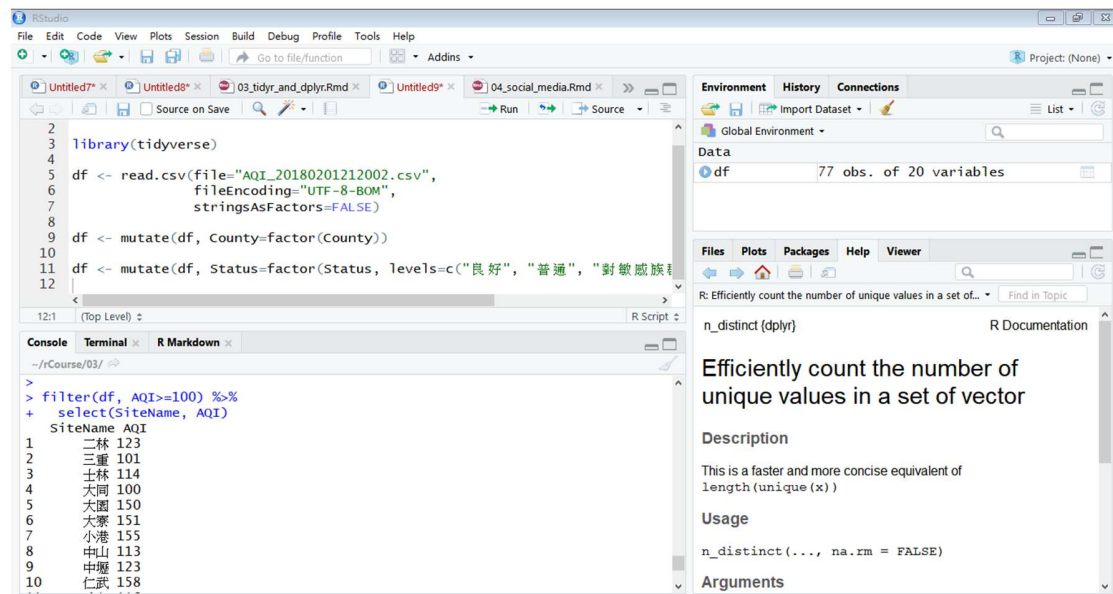
## 練習

- 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？
  - 提示：

## 哪幾個偵測站偵測到的空氣品質超過 100

- 找到 AQI 值超過 100 的紀錄
- 選擇偵測站名稱與 AQI 值

```
filter(df, AQI >= 100) %>%  
select(SiteName, AQI)
```



## 練習

- 空氣品質指標超過 100 的偵測站都在哪幾個縣市？
  - 提示：

## 多少個偵測站的污染物是細懸浮微粒

- 找到污染物是細懸浮微粒的紀錄
- 計算記錄筆數

```
filter(df, Pollutant=="細懸浮微粒") %>%
```

```
summarise(count=n())
```

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains R code for loading the tidyverse library, reading a CSV file, and filtering/summarizing data.
- Environment:** Shows the data frame 'df' with 77 observations and 20 variables.
- Console:** Displays the execution of the code, including the output of the summarise function.
- Documentation:** Shows the documentation for the `n_distinct` function from the dplyr package.

```
library(tidyverse)
df <- read.csv(file="AQI_20180201212002.csv",
               fileEncoding="UTF-8-BOM",
               stringsAsFactors=FALSE)
df <- mutate(df, County=factor(County))
df <- mutate(df, Status=factor(Status, levels=c("良好", "普通", "對敏感族群")))
```

Console Output:

```
> filter(df, Pollutant=="細懸浮微粒") %>%
+ summarise(count=n())
# A tibble: 1 x 1
  count
  <dbl>
1     71
```

Documentation for `n_distinct(dplyr)`:

Efficiently count the number of unique values in a set of vector

Description

This is a faster and more concise equivalent of `length(unique(x))`

Usage

```
n_distinct(..., na.rm = FALSE)
```

Arguments



## 各種 AQI 狀態分別有多少偵測站？

- 根據空氣品質狀態欄位進行分群
- 統計各種狀態上的數量

```
group_by(df, Status) %>%  
summarise(count=n())
```

The screenshot shows the RStudio interface with the following components:

- Script Editor:** Contains the following R code:

```
2 library(tidyverse)
3
4 df <- read.csv(file="AQI_20180201212002.csv",
5               fileEncoding="UTF-8-BOM",
6               stringsAsFactors=FALSE)
7
8 df <- mutate(df, County=factor(County))
9
10 df <- mutate(df, Status=factor(Status, levels=c("良好", "普通", "對敏感族群
11
12
```
- Console:** Shows the output of the code execution:

```
12:1 (Top Level)
> group_by(df, Status) %>%
+ summarise(count=n())
# A tibble: 4 x 2
  Status count
  <ord>   <int>
1 良好     6
2 普通    18
3 對敏感族群不健康 39
4 對所有族群不健康 14
```
- Environment:** Shows the data frame 'df' with 77 observations and 20 variables.
- Viewer:** Displays the documentation for the `n_distinct` function from the `dplyr` package, titled "Efficiently count the number of unique values in a set of vector".

## 本次課程小結

### 小結

- 資料分析時，通常將要分析的資料整理成具有相同欄位的紀錄集合，稱為結構性資料表
- 常見的結構性資料表，如 Excel 的格式
- 在 R 語言中以 **data frame** 或是 **tibble(tidy data format)**的方式表示結構性資料表
- 特別要注意的是結構性資料表中，最好每一個紀錄代表一次的觀察資料

### 小結

- 針對要分析的問題，先思考分析步驟，利用自己的語言將它表示出來
- 將每個分析步驟，表示成一個 **tibble** 方法
- 如果某個步驟無法用一個方法表示，也許這個步驟可以再細分成兩個以上的子步驟
- 然後利用 **pipe(%>%)**將方法串接起來

### 小結

- 選取特定欄位
  - **select()**
- 選取特定紀錄
  - **slice()**按照位置
  - **filter()**按照索引條件
  - **top\_n()**按照索引條件，並只選出前幾筆紀錄
- 排序
  - **arrange()**
- 將記錄分成群組
  - **group\_by()**
- 彙整(對整個資料表或紀錄群組進行統計)
  - **summarise()**

## 小結

- 不需要強記各種方法，經常練習運用，便會自然熟練
- 有需要時，可參考懶人包 <https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf>

## 延伸思考

1. 就你自己的體會，當面對一個問題時，如何將它轉換成一步步的步驟，來解決這個問題？
2. 什麼時候適合使用 **long data format**？什麼時候適合使用 **wide data format**？