

使用 R 軟體－使用 ggplot2 套件呈現熱度圖

陳芬芳 副統計分析師

本期 eNews 為大家介紹如何利用 R 軟體套件 ggplot2 提供的函數—geom_tile 呈現熱度圖，以包含一個連續變數以及二個以上的類別變數的資料進行資料視覺化，示範用地理、時間資訊，繪製出提供豐富訊息的年曆式熱度圖。另外，參考 Sahin (2020) 之研究，將台灣的氣溫與嚴重特殊傳染性肺炎（以下簡稱 COVID-19）病例數之間的相關係數以熱度圖呈現。

此次繪圖範例使用的資料來自政府資料開放平台，分別由交通部中央氣象局（以下簡稱氣象局）、衛生部疾病管制署（以下簡稱疾管署）提供的二個資料集：

1. 一年觀測資料-本局屬地面測站一年觀測資料

 政府資料開放平台
DATA.GOV.TW

網站導覽   EN 客服小幫手 線上客服 會員登入

資料集 資料故事館 互動專區 消息專區 諮詢小組 授權條款 關於平臺

資料集 / 一年觀測資料-本局屬地面測站一年觀測資料

一年觀測資料-本局屬地面測站一年觀測資料

本局屬地面測站一年觀測資料

評分此資料集：
    
平均 4.67 (3 人次投票)

瀏覽次數: 5619 下載次數: 1721 意見數: 2 

主要欄位說明	locationName、elementName(逐時觀測、測站氣壓、溫度、相對溫度、風速、風向、降水量)
*粗體欄位為資料標準欄位	
資料資源下載網址	JSON 檢視資料 本局屬地面測站一年觀測資料 XML 檢視資料 本局屬地面測站一年觀測資料
提供機關	交通部中央氣象局
提供機關聯絡人姓名	胡先生 (02-23491034)
更新頻率	每日
授權方式	政府資料開放授權條款-第1版
計費方式	免費

可以從以下網址取得：

<https://data.gov.tw/dataset/33029>，連結後從『資料資源下載網址』點選該列的 JSON 圖示進行下載。

2. 地區年齡性別統計表-嚴重特殊傳染性肺炎-依個案研判日統計

政府資料開放平臺
DATA.GOV.TW

網站導覽

Twitter

Facebook

EN

客服小幫手

線上客服

會員登入

...

資料集

資料故事館

互動專區

消息專區

諮詢小組

授權條款

關於平臺

資料集 / 地區年齡性別統計表-嚴重特殊傳染性肺炎-依個案研判日統計

地區年齡性別統計表-嚴重特殊傳染性肺炎-依個案研判日統計

2020年起各地區、各年齡層、性別之病例數統計表(疾病名稱：嚴重特殊傳染性肺炎，日期種類：研判日，病例種類：確定病例，感染來源：是否為境外移入)。
本資料集每日依系統固定排程更新一次，呈現截至前一日之統計資訊。
目前嚴重特殊傳染性肺炎境外移入病例多於機場或集中檢疫所採檢確診並即隔離治療，故未標示其縣市資訊。

評分此資料集：
☆☆☆☆☆
平均 3.76 (17 人次投票)

瀏覽次數: 9846

下載次數: 3655

意見數: 17

列印

主要欄位說明	確定病名、個案研判日、縣市、鄉鎮、性別、是否為境外移入、年齡層、確定病例數 <small>*粗體欄位為資料標準欄位</small>
資料資源下載網址	CSV 檢視資料 地區年齡性別統計表-嚴重特殊傳染性肺炎-依個案研判日統計 JSON 檢視資料 地區年齡性別統計表-嚴重特殊傳染性肺炎-依個案研判日統計
提供機關	衛生福利部疾病管制署
提供機關聯絡人姓名	馮小姐 (02-23959825#3114)
更新頻率	每日
授權方式	政府資料開放授權條款-第1版
計費方式	免費

可以從以下網址取得：


<https://data.gov.tw/dataset/120711>，連結後從『資料資源下載網址』點選該列的 CSV 圖示進行下載。

COVID-19 為台灣法定的第五類傳染病，臨床表現主要為發熱，少數病人會呼吸困難，胸部 X 光片呈現雙肺浸潤性病灶。2020 年 1 月 21 日，中央流行疫情指揮中心公布首例確診境外移入 COVID-19 個案。由於氣象局提供的天氣資料為一年觀測資料，所以後續將合併前述二筆資料集，進行整理、擷取出 2020 年 11 月至 2021 年 9 月期間的資料進行繪圖的示範。

從二筆資料集的『主要欄位說明』與『資料資源下載網址』的〔檢視資料〕，可以知道二個檔案中分別包含 locationName（可以對應“縣市”）以及 obsTime（可以對應“個案研判日”），我們利用這些欄位進行資料集的合併。以下透過 jsonlite 套件讀入氣象局資料集檔案 C-B0024-002.json，以 str 確認資料結構、欄位型態與欄列數量後，可以利用 View 進行資料檢視，快速地瞭解資料概況。

```
# install.packages(pkgs = 'jsonlite') # 安裝 jsonlite 套件（如為第一次使用）
library(package = jsonlite)           # 載入讀取 json 格式資料的套件：jsonlite
weather <-
  fromJSON(txt = 'C-B0024-002.json') # 讀入 json 檔案
str(object = weather)                 # 檢視資料結構
View(x = weather)                     # 檢視資料概況
```

Name	Type	Value
weather	list [1]	List of length 1
cwboappendata	list [9]	List of length 9
@xmlns	character [1]	'urn:cwb.gov.tw:cwbcommon:0.1'
identifier	character [1]	'CWB_ANNUAL_DATA_20211004134901'
sender	character [1]	'weather@cwb.gov.tw'
sent	character [1]	'2021-10-04 13:50+08:00'
status	character [1]	'Actual'
msgType	character [1]	'Issue'
dataid	character [1]	'CWB_B0024-002'
scope	character [1]	'Public'
dataset	list [1]	List of length 1
location	list [30 x 3] (S3: data.frame)	A data.frame with 30 rows and 3 columns
locationName	character [30]	'BANQIAO,板橋' 'TAMSUI,淡水' 'ANBU,鞍部' 'TAIPEI,臺北' 'ZHUIZHU,竹子湖' 'KEELUNG,基隆' ...
stationId	character [30]	'466880' '466900' '466910' '466920' '466930' '466940' ...
weatherElement	list [30]	List of length 30
[[1]]	list [2 x 2] (S3: data.frame)	A data.frame with 2 rows and 2 columns
[[2]]	list [2 x 2] (S3: data.frame)	A data.frame with 2 rows and 2 columns
[[3]]	list [2 x 2] (S3: data.frame)	A data.frame with 2 rows and 2 columns
[[4]]	list [2 x 2] (S3: data.frame)	A data.frame with 2 rows and 2 columns
[[5]]	list [2 x 2] (S3: data.frame)	A data.frame with 2 rows and 2 columns
[[6]]	list [2 x 2] (S3: data.frame)	A data.frame with 2 rows and 2 columns

指令 View 顯示的為二列結果：weather 以及下一層的 cwboappendata。我們點一下 cwboappendata 左邊的  就可以看見下一層的結構，依此方式快速的在 dataset 下一層的 location 資料下找到 locationName，即資料中所有測站的名稱。在 weatherElement 之下的則是相對應各測站的天氣觀測資料，逐層展開分別為：

✧ 逐時觀測—測站氣壓、溫度、相對濕度、風速、風向、降水量；

✧ 每日統計—當日最高溫度(°C)、當日最低溫度(°C)、當日平均溫度(°C)。

大致地瞭解天氣資料集結構後，利用 \$ 符號我們就能夠快速取得 30 個測站站名，如下圖所示。

```
# 最新天氣一測站地點(圖) https://www.cwb.gov.tw/V8/C/W/OBS\_Map.html  
weather$cbwopendata$dataset$location$locationName # 各測站名稱
```

```
> weather$cbwopendata$dataset$location$locationName  
[1] "BANQIAO,板橋"      "TAMSUI,淡水"      "ANBU,鞍部"  
[4] "TAIPEI,臺北"       "ZHUZIHU,竹子湖"   "KEELUNG,基隆"  
[7] "PENGJIAYU,彭佳嶼"  "HUALIEN,花蓮"     "XINWU,新屋"  
[10] "SU-AO,蘇澳"        "YILAN,宜蘭"       "KINMEN,金門"  
[13] "TianZhong,田中"    "DONGJIDAO,東吉島" "PENGHU,澎湖"  
[16] "TAINAN,臺南"       "KAOHSIUNG,高雄"   "CHIAYI,嘉義"  
[19] "TAICHUNG,臺中"     "ALISHAN,阿里山"   "DAWU,大武"  
[22] "YUSHAN,玉山"       "HSINCHU,新竹"     "HENGCHUN,恆春"  
[25] "CHENGONG,成功"     "LANYU,蘭嶼"       "SUN MOON LAKE,日月潭"  
[28] "TAITUNG,臺東"      "WUQI,梧棲"        "MATSU,馬祖"
```

依據 COVID-19 全球疫情地圖網站的統計資料，我們得知目前台灣本土病例分布前五名分別為新北市、台北市、桃園市、苗栗縣以及基隆市。由於取得的氣候資料集當中沒有直接對應苗栗縣的測站，所以我們略過苗栗縣，選入累計病例數排名第六名的彰化縣，彙總這五個縣市的每日統計資料，將其命名為 data。

```
# COVID-19 全球疫情地圖網站—台灣本土病例分布  
# https://covid-19.nchc.org.tw/dt\_005-covidTable\_taiwan.php  
# 資料選取：[6] 基隆、[4] 臺北、[1] 新北、[9] 桃園、[13] 彰化  
data <-  
  rbind(weather$cbwopendata$dataset$location$weatherElement[[6]]$time[[2]],  
        weather$cbwopendata$dataset$location$weatherElement[[4]]$time[[2]],  
        weather$cbwopendata$dataset$location$weatherElement[[1]]$time[[2]],  
        weather$cbwopendata$dataset$location$weatherElement[[9]]$time[[2]],  
        weather$cbwopendata$dataset$location$weatherElement[[13]]$time[[2]])
```

接著我們新增一欄縣市資料，設置其形態為因子 (factor)，並制定級別 (levels)，依其地理位置從中部至北部順序設定。另外將包含最高溫度、最低溫度、平均溫度等三項資料的 weatherElement 欄位，一欄分為三欄後刪除原

weatherElement 欄位，同時轉換 obsTime 欄位名稱為日期，將 data 的資料進行適當的調整。

```
# install.packages(pkgs = 'magrittr') # 安裝 magrittr 套件 (如為第一次使用)
library(package = magrittr)           # 載入處理資料流的套件: magrittr
data$縣市 <-
  rep(x = c('基隆市', '台北市', '新北市', '桃園市', '彰化縣'),
      each = 366) %>%
  factor(levels = c('彰化縣', '桃園市', '新北市', '台北市', '基隆市'))

for(i in 1:nrow(data)){
  data$最高溫度[i] <- data[2]$weatherElement[[i]][1,2][[1]]
  data$最低溫度[i] <- data[2]$weatherElement[[i]][2,2][[1]]
  data$平均溫度[i] <- data[2]$weatherElement[[i]][3,2][[1]]
}

data <- data[-2]                      # 刪除 weatherElement 一欄
names(x = data)[1] <- '日期'          # 將 obsTime 更名為日期

# install.packages(pkgs = 'hablar')   # 安裝 hablar 套件 (如為第一次使用時)
library(package = hablar)             # 載入簡單有效率轉換資料型態的套件: hablar
data <-
  convert(.x = data,
          dte(日期),                  # 日期一欄型態轉換為 date (日期)
          num(最高溫度, 最低溫度, 平均溫度)) # 最高溫度、最低溫度、平均溫度此
                                              # 三欄型態轉換為 numeric (浮點數)
str(object = data)                   # 查詢轉換型態後的資料結構
```

```
> str(object = data)
tibble [1,830 x 5] (s3: tbl_df/tbl/data.frame)
 $ 日期      : Date[1:1830], format: "2020-10-03" "2020-10-04" ...
 $ 縣市      : Factor w/ 5 levels "彰化縣","桃園市",...: 5 5 5 5 5 5 5 5 5 ...
 $ 最高溫度: num [1:1830] 31.8 30.5 26.8 25.4 25.7 26.6 26.2 25.7 26 25.8 ...
 $ 最低溫度: num [1:1830] 24.4 26.1 22.4 23.9 23.9 23.4 23.3 23.5 23.8 24.5 ...
 $ 平均溫度: num [1:1830] 27.9 28.2 25 24.6 24.8 25 25.2 24.8 25 25.1 ...
```

由於示範的年曆式熱度圖將需要使用到星期、月份、年度週別、年月、月份週別等欄位做為繪圖以及計算時的資料來源，因此我們先針對日期欄位進行處理，新增前面提到的五個欄位。

```

data$星期 <-
  weekdays(x = data$日期,
           abbreviate = TRUE)          # 指定轉換後的星期以簡稱顯示
data$月份 <-
  months(x = data$日期,
         abbreviate = TRUE)          # 指定轉換後的月份以簡稱顯示

data$年度週別 <-
  as.integer(x = format(x = data$日期,
                        format = '%U')) # 換算以年為單位之累進週別；此結果會
                                         # 出現第 0 週的情況
data$年度週別 <- data$年度週別 + 1     # 全部週別 + 1 排除第 0 週的問題

# install.packages(pkgs = 'zoo')       # 安裝 zoo 套件（如為第一次使用）
library(package = zoo)                # 載入處理時間序列套件：zoo
data$年月 <- as.yearmon(x = data$日期) # 換算年月並同時完成欄位型態設定

data <-
  data[!data$日期<'2020-11-01' & data$日期<'2021-10-01',] # 使用的資料期間

# install.packages(pkgs = 'plyr')     # 安裝 plyr 套件（如為第一次使用）
library(package = plyr)               # 載入運用分割應用於組合資料的套件：plyr
data <-
  ddply(.data = data,
        .variables = .(年月),          # 利用年月欄位分割（組）資料
        .fun = transform,             # 應用 transform 函數創建新變數月份週別
        月份週別 = 1 + 年度週別 - min(年度週別)) # 以年度週別計算出月份週別

data$月份 <-
  factor(x = data$月份,
        levels = c('十一月', '十二月', '一月', '二月', '三月',
                    '四月', '五月', '六月', '七月', '八月', '九月')) # 指定順序

data$星期 <-
  factor(x = data$星期,
        levels = c('週六', '週五', '週四', '週三', '週二', '週一', '週日'))

```

```
data$年度週別 <-  
  factor(x = data$年度週別,  
        levels = c(45:53, 1:40))  
  
data$月份週別 <- factor(x = data$月份週別)
```

前一頁我們以註解（#）方式對日期欄位的處理進行簡單的說明，讀者可以利用 View 或 str 查看初步整理後的資料內容、結構及欄位型態。接下來，我們將預先產生前 3、7、14 日當日最高溫度、最低溫度、平均溫度等欄位，作為下一階段合併五個縣市 COVID-19 病例資料後，繪製相關係數熱度圖所使用的資料來源依據。

```
# 增加四欄最高溫度欄位：前 3、7、10、14 日  
data[paste0('前', c(3,7,14), '日當日最高溫度')] <- NA      # 產生四行值為 NA 的欄位  
data$前 3 日當日最高溫度[4:nrow(data)] <- data$最高溫度[1:(nrow(data)-3)]  
data$前 7 日當日最高溫度[8:nrow(data)] <- data$最高溫度[1:(nrow(data)-7)]  
data$前 14 日當日最高溫度[15:nrow(data)] <- data$最高溫度[1:(nrow(data)-14)]  
  
# 增加最低溫度欄位：前 3、7、10、14 日  
data[paste0('前', c(3,7,14), '日當日最低溫度')] <- NA  
data$前 3 日當日最低溫度[4:nrow(data)] <- data$最低溫度[1:(nrow(data)-3)]  
data$前 7 日當日最低溫度[8:nrow(data)] <- data$最低溫度[1:(nrow(data)-7)]  
data$前 14 日當日最低溫度[15:nrow(data)] <- data$最低溫度[1:(nrow(data)-14)]  
  
# 增加平均溫度欄位：前 3、7、10、14 日  
data[paste0('前', c(3,7,14), '日當日平均溫度')] <- NA  
data$前 3 日當日平均溫度[4:nrow(data)] <- data$平均溫度[1:(nrow(data)-3)]  
data$前 7 日當日平均溫度[8:nrow(data)] <- data$平均溫度[1:(nrow(data)-7)]  
data$前 14 日當日平均溫度[15:nrow(data)] <- data$平均溫度[1:(nrow(data)-14)]
```

利用 paste0 函數連接字串，產生需要的欄位名稱，同時預先設定欄位值為 NA，再以指定欄列資料範圍的方式，將原本的最高、最低、平均溫度資料複製對應至前 3、7、14 天（數值 + 1）位置，即可於後續合併 COVID-19 病例資料後，直接以指定日期的病例數對照其前 3、7、14 日當天的溫度資料，進而計算

出二者之間的相關係數值。接著我們進行資料清理的最後一個步驟，讀入 COVID-19 病例檔案，完成資料型態轉換，合併 COVID-19 病例與天氣資料後，即可開始利用 ggplot2 套件的 geom_tile 函數，進行熱度圖的繪製示範。

```
# COVID-19 病例資料集資料來源 https://data.gov.tw/dataset/120711
covid <-
  read.csv(file = 'Day_Confirmation_Age_County_Gender_19CoV.csv',
           fileEncoding = 'UTF-8')      # 讀取編碼為 UTF-8 的 CSV 檔案
str(object = covid)                    # 查看資料結構

covid <-
  convert(.x = covid[-1],               # covid[-1] 排除不需要的第一行資料
          dte(個案研判日),             # 個案研判日一欄型態轉換為 date (日期)
          fct(縣市))                  # 縣市一欄型態轉換為 factor (因子)

# install.packages(pkgs = 'data.table') # 安裝 data.table 套件 (如為第一次使用)
library(data.table)                   # 載入能夠快速處理資料的套件：data.table
covid.weather <-
  merge(x = data,                      # 合併天氣資料與病例資料
        y = covid,
        by.x = c('日期', '縣市'),     # 利用天氣資料的日期與縣市欄位
        by.y = c('個案研判日', '縣市'), # 利用病例資料的個案研判日與縣市欄位
        all.x = TRUE) %>%             # merge 方法為保留所有的 data 資料
  as.data.table()                     # 轉換為 data.table 型態

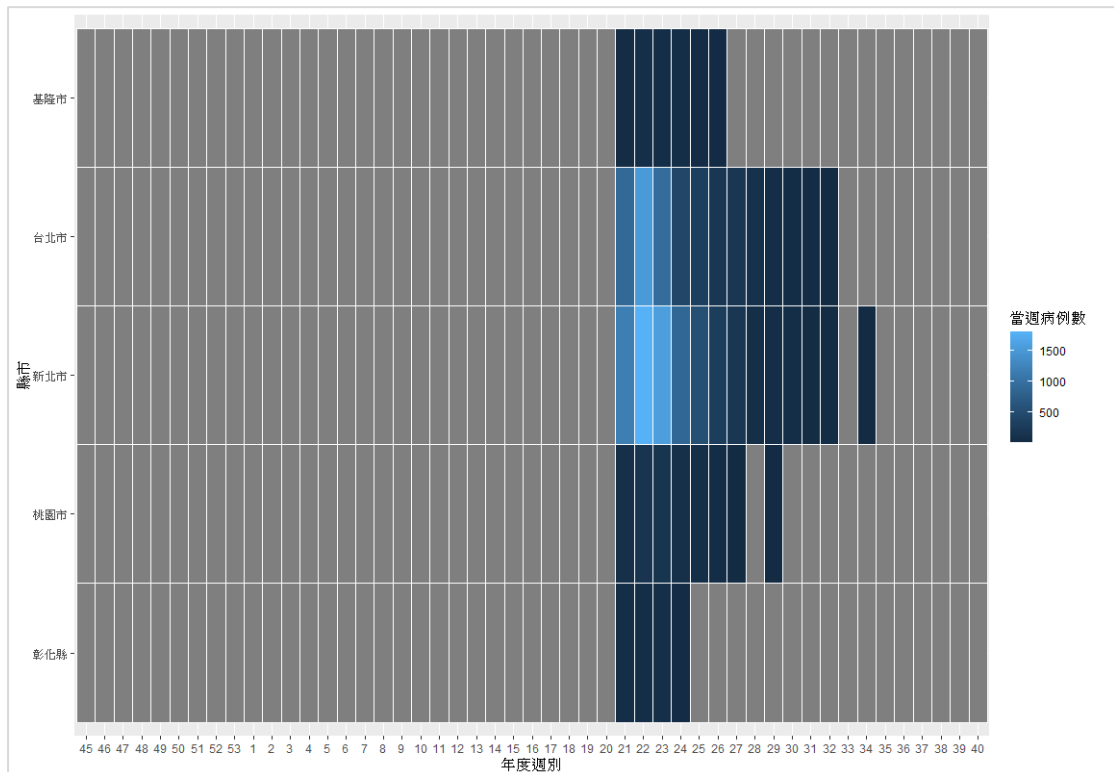
str(object = covid.weather)           # 查看、確認完成前處理後的資料結構
View(x = covid.weather)               # 查看資料內容 (若資料大不建議使用)
```

在這裡提醒讀者，在使用 View 瀏覽資料時，須留意資料的大小，若是資料筆數非常多的話，建議改用 head 或 tail 查看前六或後六筆（預設值），或是自行指定資料的顯示筆數，以避免過於耗時或是消耗過多的記憶體資源。由於我們的資料經過整理後確認筆數不算太多，所以直接使用 View 基本上是沒有問題的。

年曆式熱度圖

我們將先前整理好的資料 covid.weather，設定在以縣市、年度週別為選取條件下，對確定病例數進行加總，並將結果存至命名為當週病例數的欄位中，之後使用管線運算子（%>%）將其傳遞給 ggplot 作為繪圖用的資料來源，指定圖形的 X 軸與 Y 軸，利用 geom_tile 函數畫出以當週病例數作為帶有深淺不同色彩表現病例數多寡的熱度圖。

```
# install.packages(pkgs = 'ggplot2')      # 安裝 ggplot2 套件（如為第一次使用）
library(ggplot2)                          # 載入數據視覺化繪圖套件：ggplot2
covid.weather[,
               .(當週病例數 = sum(確定病例數)), # 加總確定病例新增一欄當週病例數
               by = .(縣市, 年度週別)] %>%   # 設定在縣市、年度週別條件下計算
ggplot() +
  aes(x = 年度週別,                          # 設定 X 軸為年度週別
      y = 縣市,                              # 設定 Y 軸為縣市
      fill = 當週病例數) +                  # 以當週病例數作為熱度圖色彩來源依據
  geom_tile(colour = 'white')               # 設定（格線）顏色為白色繪製熱度圖
```

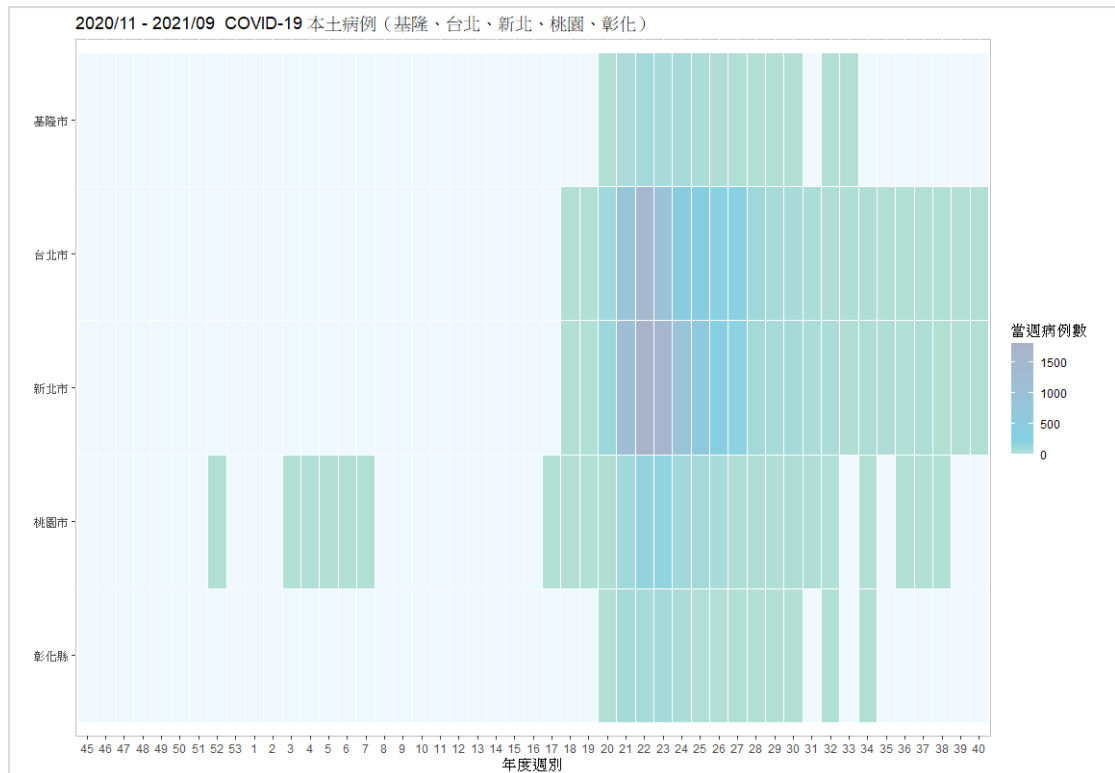


上一頁的熱度圖以不同深淺度的藍色，呈現 2020 年 11 月至 2021 年 9 月基隆、台北、新北、桃園、彰化等五縣市各週累計的 COVID-19 病例數。我們發現除了藍色以外，多數縣市週別被塗上了灰色，這代表的是甚麼意思呢？從台灣的疫情變化可以大膽推測灰色應該是當週無確診病例。進一步利用 View 檢視資料就能印證灰色是代表值為 NA 的情況。

我們可以對確定病例數該欄的 NA 值進行填值為 0 的調整，再重新進行加總、繪製熱度圖。以下即為調整數值、指定熱度圖的顏色，以及移除 Y 軸名稱、增加圖形標題等過程的程式碼示例。在此特別說明，關於 `scale_fill_gradientn` 中的 `values` 與 `colours` 二個參數，分別為設定的切點值及其相對應呈現的顏色，讀者可自行調整，增加或減少設定的區間數，或指定不同的顏色呈現。此處設定的值是以當週病例數為 0、1、210 以及最大值 1802 人進行換算的結果。顏色表示方式與實際視覺可以參考 R CHARTS 網站此一網頁：<https://r-charts.com/colors/>。

```
covid.weather[is.na(確定病例數), 確定病例數:=0] # 將確定病例數為 NA 的調整為 0
covid.weather[,
  .(當週病例數 = sum(確定病例數)),
  by = .(縣市, 年度週別)] %>%

ggplot() +
  aes(x = 年度週別,
    y = 縣市,
    fill = 當週病例數) +
  geom_tile(colour = 'white') +
  scale_fill_gradientn(colours = c('#F0F8FF', '#91D1C2B2', '#4DBBD5B2',
    '#8491B4B2'),
    values = c(0, .0005, .1165, 1)) + # 數值對應 0,1,30*7,1802
  labs(y = "",
    title = '2020/11 - 2021/09 COVID-19 本土病例（基隆、台北、新北、桃園、彰化）') +
  theme(panel.background = element_rect(fill = 'white', # 繪圖框內填充為白色
    colour = 'grey75')) # 繪圖框線設定為灰色
```



比較第 9 頁的 COVID-19 病例热度圖與上圖，將會明顯發現二張圖形的 2020 年的第 52 週至 2021 年的第 7 週出現差異。這個差異主要是因為我們調整了確定病例數數值 NA 的欄位，同時設定當週病例數大於 0 時（value = 0.0005），热度圖即以青色(#91D1C2B2)顯示此一資訊。至於 panel.background 中的 fill 與 colour 則是調整繪圖區預設的框線為灰色、底色為白色，以消除繪圖區最上方與最下方的淺灰區塊（此部分請對照二張圖形進行觀察），增強視覺上的簡潔感。

完成前面二張簡單的病例热度圖繪製之後，接下來我們正式介紹如何加入日期資訊繪製出年曆式热度圖。此種热度圖因為可以同時提供多個欄位的訊息，在視覺上的效果十分的出色。透過 data.table 套件提供的 by 參數，以縣市、月份、星期與月份週別這四個欄位條件篩選，加總確定病例數得出各縣市每日的當日病例數，搭配 facet_grid 函數（此函數可依據資料分組與指定的類別變數，以矩陣形式呈現圖形），重新調整、指定圖形呈現的資料欄位位置，繪製出如同下一頁般的年曆式热度圖。

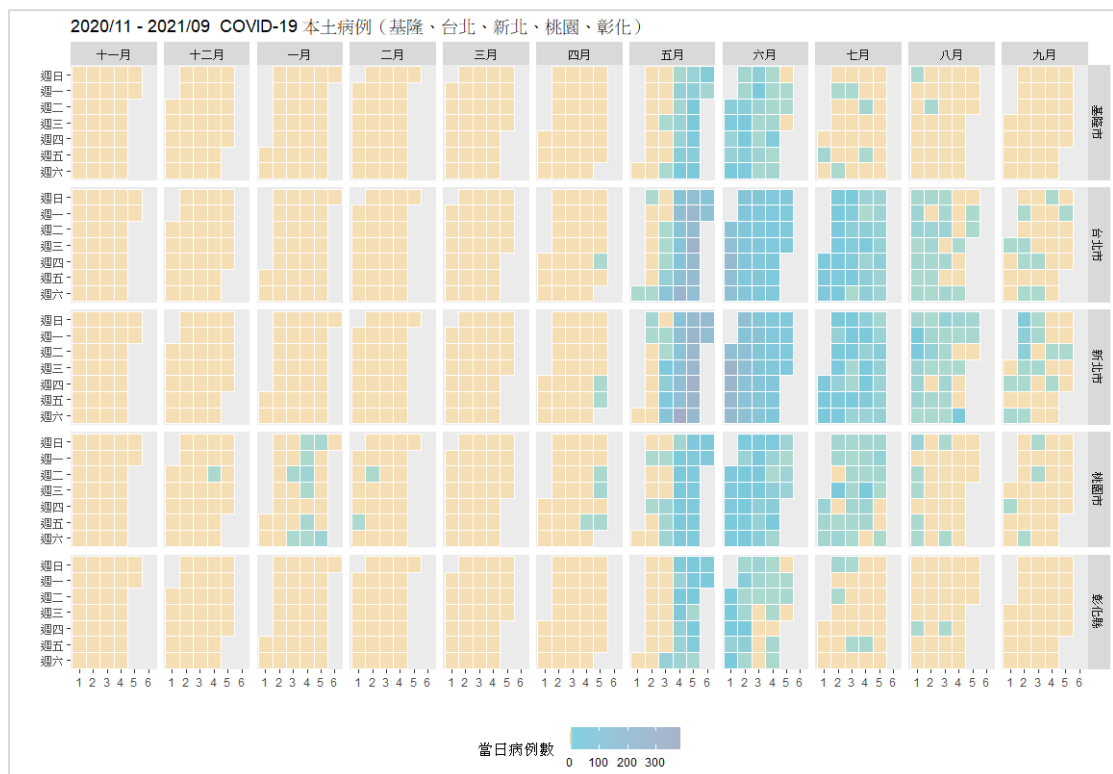
```

covid.weather[,
  .(當日病例數 = sum(確定病例數)),
  by = .(縣市, 月份, 星期, 月份週別)] %>% # 調整計算的篩選條件

ggplot() +
aes(x = 月份週別,
  y = 星期,
  fill = 當日病例數) +
geom_tile(colour = 'white') +
facet_grid(vars(factor(x = 縣市, # 加入縣市欄位(圖右方)
  levels = rev(levels(縣市)))), # 指定縣市呈現的順序
  vars(月份)) + # 加入月份欄位(圖左方)
scale_fill_gradientn(colours = c('#F5DEB3', '#91D1C2B2', '#4DBBD5B2',
'#8491B4B2'),
  values = c(0, .0025, .0259, 1)) + # 數值對應 0,1,10,385

labs(x = "",
  y = "",
  title = '2020/11 - 2021/09 COVID-19 本土病例 (基隆、台北、新北、桃園、彰化)') +
theme(panel.grid = element_blank(), # 取消矩陣圖中空白處的格線
  legend.position = 'bottom') # 圖例位置設定於圖下方

```



從上一頁的年曆式熱度圖我們可以直接觀察到在 2021 年 5 月份的第三週，即 5 月 10 號開始，新北市與台北市開始有許多 COVID-19 確定病例數，且在接下來的第四週與第五週出現更大量的確定病例。年曆式熱度圖在這兒就發揮了優勢，展現多個欄位的資訊，比較同一時間點不同地方的病例數以及發展趨勢。

在進入第二部分的最高、最低、平均溫度與確定病例數的相關係數熱度圖繪製之前，我們先簡單的替換前一段程式碼中 fill 填入的欄位為平均溫度，觀察一下五個縣市在 2020 年至 2021 年這 11 個月當中的氣溫變化。從平均溫度的年曆式熱度圖當中，可以很容易的發現北部三個縣市：基隆、台北、新北市在六月的第三週平均溫度明顯高於桃園及彰化縣，而且也是將近一年的期間中，平均溫度最高的一個月份。對於最高、最低溫度在熱度圖上呈現的樣貌，有興趣的讀者也可以將 fill 替換為最高溫度或者最低溫度這二個欄位進行觀察。

```
covid.weather %>%
  ggplot() +
  aes(x = 月份週別,
      y = 星期,
      fill = 平均溫度) + # 欄位設定為平均溫度
  geom_tile(colour = 'white') +
  facet_grid(vars(factor(x = 縣市,
                        levels = rev(levels(縣市)))),
            vars(月份)) +
  scale_fill_gradientn(colours = c('#FFFFFF', '#4DBBD5B2', '#00A087B2',
                                    '#DC0000B2'),
                      values = c(0, .2980, .6902, 1)) + # 7.4, 15, 25, 32.9
  labs(x = "",
      y = "",
      title = '2020/11 - 2021/09 基隆、台北、新北、桃園、彰化五縣市平均溫度')
+
  theme(panel.grid = element_blank(),
        legend.position = 'bottom')
```



相關係數熱度圖

接下來我們將介紹如何利用 `geom_tile` 函數繪製相關係數熱度圖。正式繪製圖形前，需要針對資料做一些處理，並事先計算前 3、7、14 日當日最高、最低、平均溫度以及當日病例數這些欄位數值之間的相關係數。我們先利用設定條件為縣市與日期之下加總出得到的當日病例數，以及需要使用到的欄位，另存為 `cw1` 這個變數。這邊提醒讀者，由於原始的病例資料並不是以不同縣市、單日、單筆的條件為記錄方式，是故在加總之後會出現重複的資料列，所以利用 `unique` 函數取得沒有重複列的資料使用。

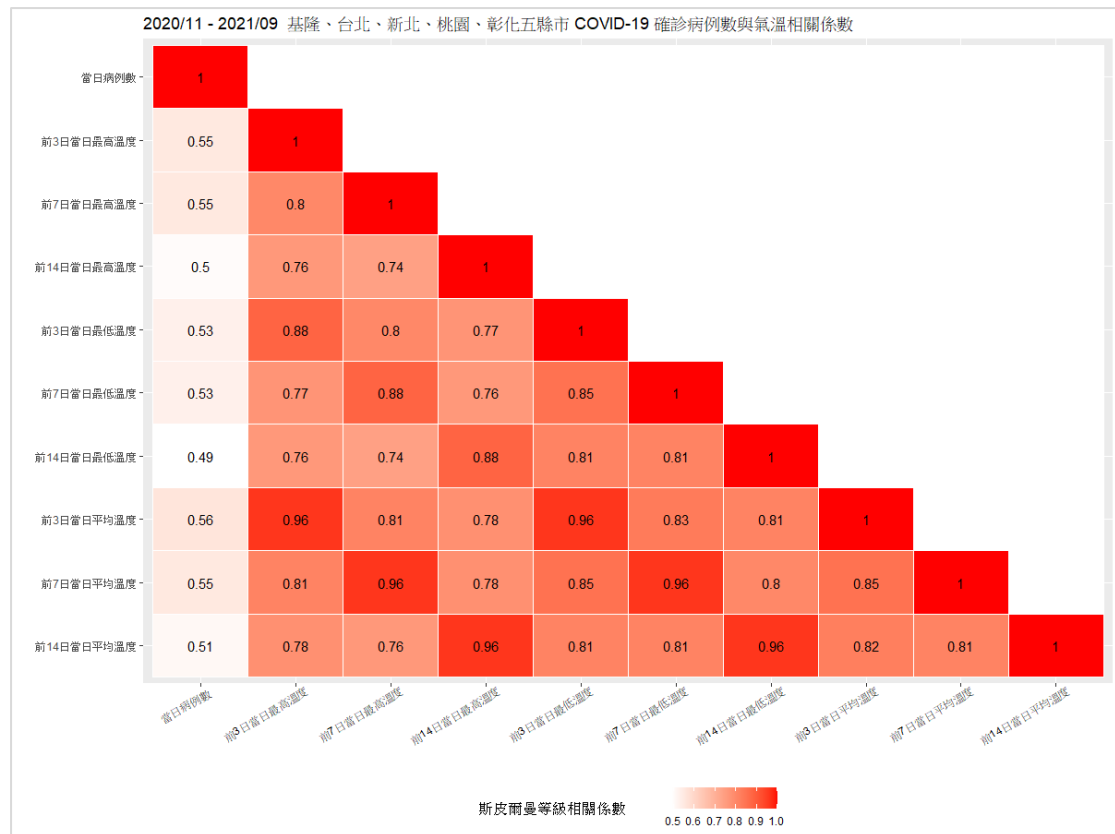
利用 `cor` 函數計算斯皮爾曼等級相關係數，以 `round` 函數取得小數點後二位的結果，並將其轉換為 `data.table` 的格式方便後續使用。如果大家利用 View 查看此結果，就會發現計算完的相關係數呈現上三角與下三角結果對稱的情況，也就是說其實我們僅需要取得上三角或下三角其中一個部份的結果進行繪圖即可。利

用 `upper.tri` 函數，我們將不需要的上三角區塊的值轉換為 NA，以 `stack` 函數將資料由寬資料轉換為長資料格式，並加入繪圖需要用到的列名稱，最後再將繪圖時我們希望看到的資料名稱排列順序，事先以 `factor` 函數的 `levels` 參數設定好，就完成相關係數熱度圖繪製前的資料處理了。

```
cwl <-  
  covid.weather[,  
    .(當日病例數 = sum(確定病例數),  
      前3日當日最高溫度, 前7日當日最高溫度,  
      前14日當日最高溫度,  
      前3日當日最低溫度, 前7日當日最低溫度,  
      前14日當日最低溫度,  
      前3日當日平均溫度, 前7日當日平均溫度,  
      前14日當日平均溫度),  
    by = .(縣市, 日期)] %>%  
  unique() # 僅取不重複的數列資料  
  
cwl.cor <-  
  round(x = cor(x = cwl[, -1:-2], # 計算數值型態欄位的相關係數  
    use = 'pairwise.complete.obs', # 採用成對完整無 NA 的觀察值計算  
    method = 'spearman'), # 方法為不受離群值影響的斯皮爾曼  
    digits = 2) %>% # 等級相關係數方法取二位小數結果  
  as.data.table()  
View(x = cwl.cor)  
  
cwl.cor[upper.tri(x = cwl.cor)] <- NA # 將上三角的相關係數值替換為 NA  
  
cw2 <-  
  data.frame(stack(x = cwl.cor), # 將資料型態由寬資料轉換為長資料  
    y = rep(x = colnames(x = cwl.cor), # 列與欄名相同，利用 colnames 產生  
      times = 10)) # 總共十個不同變數，所以進行 10 次  
  
View(x = cw2)  
  
cw2$y <-  
  factor(x = cw2$y,  
    levels = rev(colnames(x = cwl.cor)))
```


在以下的這一段程式中，我們將 fill 的值改填入 values，也就是資料轉形後相關係數的欄位名稱，ind 及 y 則為實際上內容相同的資料標籤。另外我們以 scale_fill_gradient 指定低相關係數與高相關係數的顏色，分別用白色與紅色之間的深淺度來展現，NA 值（即被替換的上三角部分）以白色呈現。同時 geom_text 指定熱度圖上顯示相關係數值文字的色彩與大小，並將 X 軸的資料標籤參數 angle 設定為 30 度（逆時針為正），水平位置的 hjust 參數設定為 0.9 進行微調，即向上微偏向 X 軸，並指定以黑色呈現軸標籤的文字。

```
cw2 %>%
  ggplot() +
  aes(x = ind,
      y = y,
      fill = values) + # 熱度圖填入相關係數值
  geom_tile(colour = 'white') +
  scale_fill_gradient(low = 'white', high = 'red', na.value = 'white') +
  geom_text(aes(label = values), color = 'black', size = 4) + # 設定文字顯示格式
  labs(x = "",
      y = "",
      title = '2020/11 - 2021/09 基隆、台北、新北、桃園、彰化五縣市 COVID-19
確診病例數與氣溫相關係數',
      fill = '斯皮爾曼等級相關係數') +
  theme(axis.text.x = element_text(angle = 30, # X 軸標籤向左旋轉 30 度
                                     hjust = .9, # 同時水平向上 0.1 單位
                                     color = 'black'),
        legend.position = 'bottom')
```



從上圖我們可以發現溫度的相關係數，以平均溫度的數值，不管是前 3 日、前 7 日或前 14 日均高於最高溫度與最低溫度的部分。但是不同的變數之間是否具有顯著相關，則需要透過統計檢定的結果來確認。在顯著水準設定為 0.05 時，從檢定的結果可以發現所有兩兩變數之間的 P 值均小於 0.05，呈現顯著相關。從熱度圖中，我們也可以發現所有變數之間是呈現正相關，也就是說，這五個縣市的當日病例數與前 3 日、前 7 日、前 14 日的當日最高、最低、平均溫度，都具有顯著的正相關。疫情的變化受到許多因素的影響，我們在此不做深入的討論，僅借用 Sahin (2020) 研究中的概念，新增氣溫欄位演繹相關係數熱度圖的繪製。

```
# install.packages(pkgs = 'rstatix')      # 安裝 rstatix 套件 (如為第一次使用)
library(package = rstatix)               # 載入提供友好資料流的基礎統計檢定方法套件：rstatix
cor_mat(data = cwl[, -1:-2],              # 計算數值欄位的兩兩相關係數
          method = 'spearman') %>%
  cor_get_pval() %>%                      # 計算相關係數的 P 值
  View()
```

	rowname	當日 病例 數	前3日 當日 最高 溫度	前7日 當日 最高 溫度	前14日 當日 最高 溫度	前3日 當日 最低 溫度	前7日 當日 最低 溫度	前14日 當日 最低 溫度	前3日 當日 平均 溫度	前7日 當日 平均 溫度	前14日 當日 平均 溫度
1	當日病例數	0.00e+00	1.49e-133	9.87e-131	1.910000e-104	1.710000e-122	5.09e-120	6.87e-102	6.6e-136	3.19e-132	1.02e-110
2	前3日當日最高溫度	1.49e-133	0.00e+00	0.00e+00	1.120000e-317	0.000000e+00	0.00e+00	2.04e-313	0.0e+00	0.00e+00	0.00e+00
3	前7日當日最高溫度	9.87e-131	0.00e+00	0.00e+00	4.510000e-289	0.000000e+00	0.00e+00	1.43e-288	0.0e+00	0.00e+00	2.69e-307
4	前14日當日最高溫度	1.91e-104	1.12e-317	4.51e-289	0.000000e+00	1.570002e-318	9.31e-317	0.00e+00	0.0e+00	0.00e+00	0.00e+00
5	前3日當日最低溫度	1.71e-122	0.00e+00	0.00e+00	1.570002e-318	0.000000e+00	0.00e+00	0.00e+00	0.0e+00	0.00e+00	0.00e+00
6	前7日當日最低溫度	5.09e-120	0.00e+00	0.00e+00	9.310000e-317	0.000000e+00	0.00e+00	0.00e+00	0.0e+00	0.00e+00	0.00e+00
7	前14日當日最低溫度	6.87e-102	2.04e-313	1.43e-288	0.000000e+00	0.000000e+00	0.00e+00	0.00e+00	0.0e+00	0.00e+00	0.00e+00
8	前3日當日平均溫度	6.60e-136	0.00e+00	0.00e+00	0.000000e+00	0.000000e+00	0.00e+00	0.00e+00	0.0e+00	0.00e+00	0.00e+00
9	前7日當日平均溫度	3.19e-132	0.00e+00	0.00e+00	0.000000e+00	0.000000e+00	0.00e+00	0.00e+00	0.0e+00	0.00e+00	0.00e+00
10	前14日當日平均溫度	1.02e-110	0.00e+00	2.69e-307	0.000000e+00	0.000000e+00	0.00e+00	0.00e+00	0.0e+00	0.00e+00	0.00e+00

本期 eNews 以 ggplot2 套件的 geom_tile 函數說明如何畫出具有多樣資訊的年曆式熱度圖，以及統計分析上經常用到的相關係數其圖形的表示。R 軟體中有許多其他的套件均可直接、快速的套繪變數之間相關係數圖表，我們在這裡的示範主要提供讀者們不同的繪圖方式。

在處理資料的過程中，選擇不同的資料集型態，演示了使用 jsonlite 套件以及 read.csv 讀取資料集，也利用 data.frame、以及 data.table 的特性，協助我們進行資料欄、列的整理以及計算、轉換。同時用原有的日期欄位，轉換出繪圖需要用到年、月、週、星期等欄位，並一併介紹方便好用的 hablar 套件，利用 convert 函數一次處理多個欄位的型態轉換。最後，利用整理好的資料進行圖形繪製，透過 ggplot2 疊層調整、增加圖形資料的功能，將我們想要呈現的結果繪製出來。

希望這一期 eNews 的內容，能夠帶給讀者在使用 ggplot2 套件更多的助益與進一步的認識。在下一期的 eNews 當中，我們將為大家介紹如何利用視覺化方法，展現存活分析的結果，敬請大家期待唷！

參考資料

1. Sahin, M. (2020). Impact of weather on COVID-19 pandemic in Turkey. *Science of The Total Environment*, 728, 138810.
2. R CHARTS (2021). *Colors in R*. <https://r-charts.com/colors/>
3. 國家發展委員會 (2021). 交通部中央氣象局. 一年觀測資料-本局屬地面測站一年觀測資料 [資料集]. <https://data.gov.tw/dataset/33029>
4. 國家發展委員會 (2021). 衛生福利部疾病管制署. 地區年齡性別統計表-嚴重特殊傳染性肺炎-依個案研判日統計 [資料集]. <https://data.gov.tw/dataset/120711>
5. 交通部中央氣象局 (2021). 最新天氣. https://www.cwb.gov.tw/V8/C/W/OBS_Map.html
6. 國家高速網路與計算中心 (2021). 台灣疫情報告. https://covid-19.nchc.org.tw/dt_005-covidTable_taiwan.php