網頁資訊擷取 I

Sung-Chien Lin

2018 年 9 月 16 日

課程簡介

課程內容

- 網頁資料的組織方式(Hypertext Markup Language, HTML)
- 利用 rvest 套件進行網頁資料擷取
- 利用程式結構擷取多個網頁資料
  - if-else 分支結構
  - while 迴圈
- 以聯合報歷史新聞的目錄頁為例

學習目標

- 能夠說明網頁資料的 HTML 原始碼上的內容在網頁中的對應部份
- 能夠利用 rvest 套件擷取網頁上的資料
- 能夠使用 if-else 分支結構和 while 迴圈撰寫程式

網頁資料的基本概念

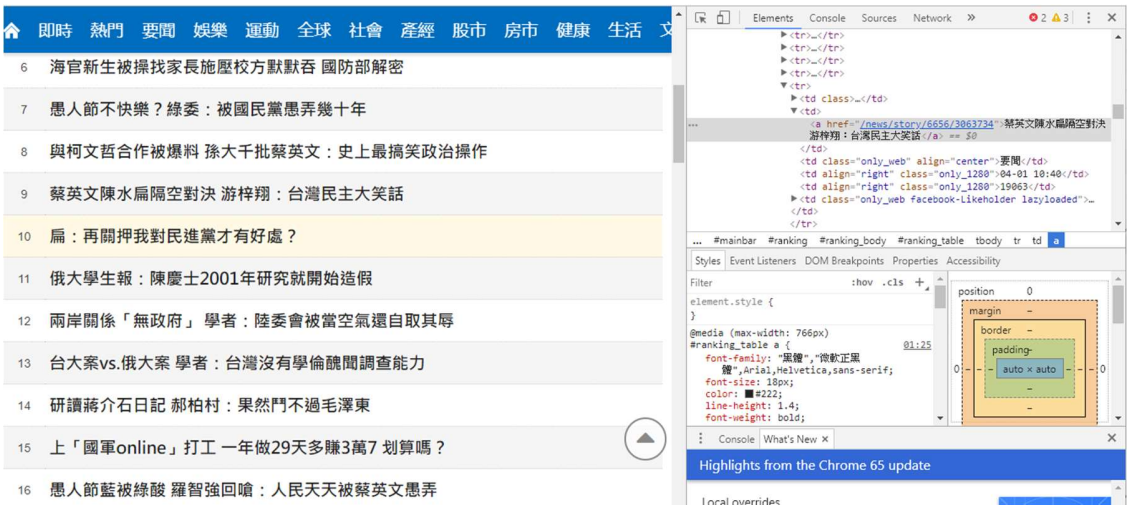
聯合報歷史新聞目錄頁截圖

網頁資料(HTML)

- 網頁由標示語言 HTML(Hypertext markup language)所建構
- 透過多種不同功用的標籤 tag 構成的元件
  - <h2>Yahoo! 奇摩</h2>
  - Google 地圖
  - 繼昨（21）日一天內生成兩個颱風後，中央氣象局表示，今年第七號颱風洛克（Roke，美國命名）也在今日稍早生成。 氣象局指出，位在海南島東南方的熱帶性低氣壓，在剛才已增強為輕度颱風，但預期將會朝廣東移動，對台灣不會有直接影響。對於兩日內連續生成三個颱風，氣象達人彭啟明在臉書指出「超趕進度的！」，連網友都忍不住說「這是馬拉松比賽嗎？」。

聯合報歷史新聞目錄頁的 HTML

- 利用聯合報歷史新聞目錄頁的 HTML 格式，可以取得當天的新聞標題與內文的連結
- 在瀏覽器的「更多工具」下，選擇「開發人員」工具，可以看到這個網頁的 HTML 原始碼



常見的標籤

- h1, h2, ..., h6：標題
- p：段落
- a：連結
- img：圖片
- table, tr, td：表格
- ol, ul, li：列表
- ...

每個元件內包含若干屬性

- Google 地圖
- href：屬性
- "https://www.google.com.tw/maps"：屬性值

常用的屬性

- href：超連結網址
- id：標籤識別。一個網頁內，具有標籤識別的元件其標籤識別都不同。
- class：標籤群組。一個網頁內，多個元件可以具有相同的標籤識別。

網頁資料擷取套件

rvest 套件

- 進行網頁資料蒐集的套件
- 第一次使用套件前，需要安裝
  - 在 Console 上輸入

```
install.packages("rvest")
```

rvest 常用的函數

- read_html()：讀取網頁的 html 碼
- html_nodes()：取得網頁上的元件
- html_text()：取得網頁元件上的文字
- html_attr()：取得網頁元件上的屬性

多個網頁的資料自動擷取

聯合報歷史新聞目錄頁

- 若是新聞資料較多，通常會表示成多個目錄頁

- 可利用程式結構自動取得連續多個目錄頁資料

程式結構

- 靈活地利用程式結構可以使資料處理自動化
- 程式結構
  - 分支：根據情況決定是否執行某些敍述
  - 迴圈：重複執行某些敍述

本次課程程式

```
setwd("rCourse/07")

library(rvest)

continue.flag <- TRUE
news.df = data.frame(title = character(),      #所有的新聞標題與連結資料
                     link = character(),
                     stringsAsFactors = FALSE)

## 準備讀取目錄頁
date <- "2018/09/15" # 設定目錄頁讀取日期
page.url <- paste0("https://udn.com/news/archive/2/6649/", date) # 產生
目錄頁網址

while (continue.flag==TRUE) {
  print(paste("Processing", page.url))
  page.cont <- read_html(page.url) # 讀取目錄頁資料

  # 讀取目錄頁上所有的新聞資料
  title.nodes <- html_nodes(page.cont, css="td a")
  # 將這新聞標題與內文連結一起放在同一個data frame 中
  page.df <- data.frame(title=html_text(title.nodes), #目前目錄頁上的新聞
標題與連結資料
                        link=html_attr(title.nodes, "href"),
                        stringsAsFactors = FALSE)

  # 合併先前與目前頁面的新聞資料
  news.df <- rbind(news.df, page.df)

```

```
  page.nodes <- html_nodes(page.cont, css=".pagelink a") # 取得頁碼標示

  page.text <- html_text(page.nodes) # 取得頁碼文字

  page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則結果為 TRUE，否則為 FALSE

  continue.flag <- any(page.next) #是否有「下一頁」

  if (continue.flag) {
    # 如果有「下一頁」連結，則準備讀取下一個目錄頁
    url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連結
    page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址
  }
}

write.csv(news.df, file=paste0("udn_", gsub("/","_",date),".csv"),
          row.names=FALSE, fileEncoding="UTF-8")
```

預備工作

準備工作目錄與檔案

- 在 rCourse 下，建立工作目錄 05

設定工作目錄

- 首先開啟新的 Script
- 在 Script 上，設定工作目錄

```
setwd("rCourse/07")
```

載入 rvest 套件

- 在 Script 上輸入

```
library(rvest)
```

讀取單一目錄頁

讀取目錄頁的第一頁

- 在 Console 上輸入，設定擷取目錄頁的日期

```
date <- "2018/09/15" #設定擷取目錄頁的日期
```

聯合報當天歷史新聞的第一個目錄頁面的網址

- 在 Console 上輸入，產生目錄頁的網址

```
page.url <- paste0("https://udn.com/news/archive/2/6649/", date)
```

- `page.url` 的值即是上圖網址列上的網址

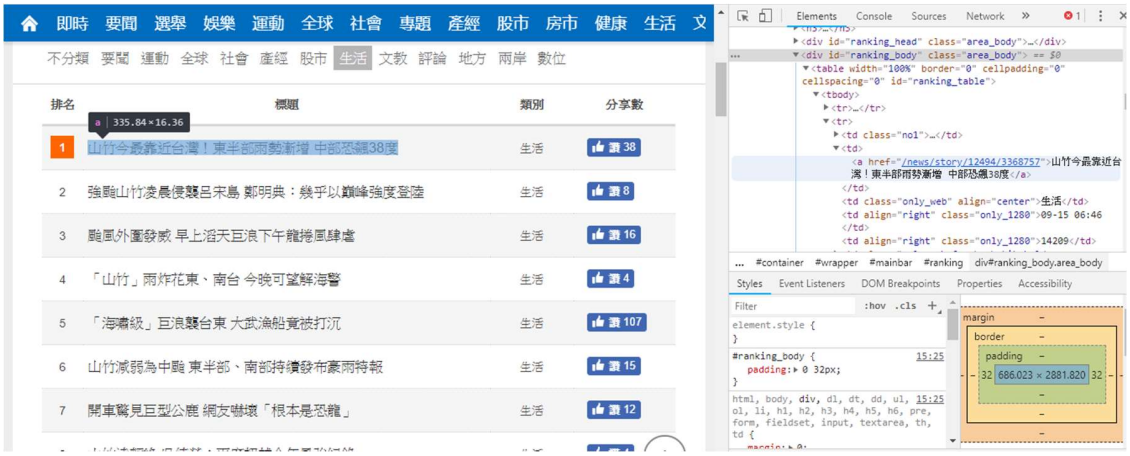
從目錄頁網址取得頁面的 HTML 資料

- read_html(page.url)：從網址 page.url 取得頁面的 HTML 資料
- 在 Console 上輸入

```
page.cont <- read_html(page.url)
```

分析聯合報歷史新聞目錄頁的 HTML

- 從聯合報的目錄頁 HTML 原始碼發現新聞資料的標籤



- 新聞資料的標籤的第一層為 a，但會包含許多無關資料
- 先從第二層 td a 開始嘗試

讀取目錄頁上所有的新聞資料

- 在 Console 上輸入

```
title.nodes <- html_nodes(page.cont, css="td a")
```

- 如果第二層仍包含太多無關資料，再往上一層
- 保險一些的做法是找到有包含 id 的標籤
  - 以此例來說，即是使用 table#ranking_table tbody tr td a

取得標籤內文字資料(即新聞標題)

- 在 Console 上輸入

```
html_text(title.nodes)
```

取得標籤內屬性值資料(即新聞內文連結)

- 在這個例子中，為取得"href"(超連結)屬性的資料
- 在 Console 上輸入

```
html_attr(title.nodes, "href")
```

將這兩種資料一起放在同一個 data frame 中

- 在 Console 上輸入

```
page.df <- data.frame(title=html_text(title.nodes),
                      link=html_attr(title.nodes, "href"),
                      stringsAsFactors = FALSE)
```

Filter

	title	link
1	山竹今最靠近台灣！東半部雨勢漸增 中部恐飆38度	/news/story/12494/3368757
2	強颱山竹凌晨侵襲呂宋島 鄭明典：幾乎以巔峰強度登陸	/news/story/12494/3368761
3	颱風外圍發威 早上滔天巨浪下午龍捲風肆虐	/news/story/12494/3369502
4	「山竹」雨炸花東、南台 今晚可望解海警	/news/story/12494/3368609
5	「海嘯級」巨浪襲台東 大武漁船竟被打沉	/news/story/12494/3368871
6	山竹減弱為中颱 東半部、南部持續發布豪雨特報	/news/story/7266/3369013
7	開車驚見巨型公鹿 網友嚇壞「根本是恐龍」	/news/story/7470/3369309

讀取多個新聞目錄頁

有多個新聞目錄頁，如何讀取？

- 觀察在每個目錄頁中有標示第一頁、目前所在頁面、其他附近頁面、上一頁、下一頁以及最後一頁等目錄頁連結
- 第一個目錄頁

- 最後一個目錄頁

- 策略：在每個頁面讀取下一頁的連結，一直到無法讀取為止

讀取頁面中的頁碼標示

- 在 Console 上執行，注意 page.nodes 和 page.text 的結果

```
## 準備讀取目錄頁
date <- "2018/09/15" # 設定目錄頁讀取日期
page.url <- paste0("https://udn.com/news/archive/2/6649/", date) # 產生目錄頁網址

page.cont <- read_html(page.url) # 讀取目錄頁資料

page.nodes <- html_nodes(page.cont, css=".pagelink a") # 取得頁碼標示

page.text <- html_text(page.nodes) # 取得頁碼文字
```

根據頁碼標示，判斷是否有「下一頁」連結

- grepl("下一頁", page.text)：如果 page.text 的字串中包含「下一頁」，則結果為 TRUE，否則為 FALSE
- any(page.next)：如果 page.next 其中包含任何一個 TRUE，結果為 TRUE；如果全為 FALSE，則為 FALSE
- if (continue.flag)：根據 continue.flag 的值，決定是否執行大括號裡的敘述
- 在 Console 上逐行執行，注意各個敘述的結果

```
page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則結果為 TRUE，否則為 FALSE

continue.flag <- any(page.next) #是否有「下一頁」

if (continue.flag) {
  # 如果有「下一頁」連結，則準備讀取下一個目錄頁
  url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連結
  page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址

  ## 讀取下一個目錄頁
}
```

if 分支結構說明

if 分支結構

- if 是 R 語言的分支結構，如果後面的運算結果為 TRUE，便執行下面的程式
- 以下透過兩個例子，了解 if 的用法(請試著在 Console 上執行看看)

練習

- 例一：當 if 後面的運算結果為 TRUE

```
x <- "原來的字串"
if (4/2==2) {
  x <- "新改的字串"
}
x
```

- 例二：當 if 後面的運算結果為 FALSE

```
x <- "原來的字串"
if (3+5==7) {
  x <- "新改的字串"
}
x
```

練習

- 如果輸入任一整數，便將它列印出來(請試著在 Console 上執行看看)

```
n <- as.integer(readline(prompt = "請輸入任一整數："))
if (is.na(n)==FALSE) {
  print(paste("您輸入的是：", as.character(n)))
}
```

- 改寫上面的程式，加上如果這個整數大於 0 才列印出來

if-else

- 如果有兩種狀況，需要選擇一個進行時，可以使用 if-else 結構
- 如果 if 後面的運算結果為 TRUE，便執行下面的程式；否則便執行 else 後的程式
- 透過下面例子，了解 if-else 的用法(請試著在 Console 上執行看看)
- 例子：
  - menu(choices, graphics, title)會取得使用者選擇第幾個選項

```
if (menu(choices=c("7", "3"), graphics=TRUE, title="請問哪一個數字大於 5?
")==1) {
  print("答對了，您真是天才！")
} else {
  print("答錯了！請再試試。")
}
```

練習

- 試著改寫上面的例子詢問使用者，今年是閏年或平年？

複習

- 什麼情況會進行『取得「下一頁」的連結』和『產生下一個目錄頁網址』等等敘述

```
page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則結果為 TRUE，否則為 FALSE

continue.flag <- any(page.next) #是否有「下一頁」

if (continue.flag) {
  # 如果有「下一頁」連結，則準備讀取下一個目錄頁
  url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連結
  page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址

  ## 讀取下一個目錄頁
}
```

以下的程式為讀取兩個連續目錄頁的片段

- 講解用，請不用輸入！

```
## 準備讀取目錄頁
date <- "2018/04/01" # 設定目錄頁讀取日期
page.url <- paste0("https://udn.com/news/archive/2/6638/", date) # 產生目錄頁網址

page.cont <- read_html(page.url) # 讀取目錄頁資料

page.nodes <- html_nodes(page.cont, css=".pagelink a") # 取得頁碼標示

page.text <- html_text(page.nodes) # 取得頁碼文字

page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則結果為 TRUE，否則為 FALSE

continue.flag <- any(page.next) #是否有「下一頁」

if (continue.flag) {
  # 如果有「下一頁」連結，則準備讀取下一個目錄頁
  url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連結
  page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址

  ## 讀取下一個目錄頁
  page.cont <- read_html(page.url) # 讀取目錄頁資料

  page.nodes <- html_nodes(page.cont, css=".pagelink a") # 取得頁碼標示

  page.text <- html_text(page.nodes) # 取得頁碼文字
```

```
  page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則結果為 TRUE，否則為 FALSE

  continue.flag <- any(page.next) #是否有「下一頁」

  if (continue.flag) {
    # 如果有「下一頁」連結，則準備讀取下一個目錄頁
    url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連結
    page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址

    ## 讀取下一個目錄頁
  }
}
```

- 有許多重複的地方，而且不應該都是以手動的方式取得每一頁的資料

以 while 迴圈的方式改寫

- 請在 Console 上輸入

```
continue.flag <- TRUE
## 準備讀取目錄頁
date <- "2018/04/01" # 設定目錄頁讀取日期
page.url <- paste0("https://udn.com/news/archive/2/6638/", date) # 產生目錄頁網址

while (continue.flag==TRUE) {
  print(paste("Processing", page.url))
  page.cont <- read_html(page.url) # 讀取目錄頁資料

  page.nodes <- html_nodes(page.cont, css=".pagelink a") # 取得頁碼標示

  page.text <- html_text(page.nodes) # 取得頁碼文字

  page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則結果為 TRUE，否則為 FALSE

  continue.flag <- any(page.next) #是否有「下一頁」

  if (continue.flag) {
  # 如果有「下一頁」連結，則準備讀取下一個目錄頁
    url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連結
    page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址
  }
}
```

while 迴圈結構

while 迴圈結構

- while 是 R 語言的條件迴圈結構，如果 while 後面的運算結果為 TRUE，便重複執行下面的程式，一直到 while 後面的運算結果為 FALSE 為止
- 透過下面的例子，了解 while 的用法(請試著在 Console 上執行看看)

```
flag <- TRUE
while (flag) {
  if (menu(choices=c("9", "5"), graphics=TRUE, title="請選擇大於 7 的數值：")==1) {
    flag <- FALSE
    print("謝謝！您的選擇是對的")
  } else {
    print("抱歉！請您再想想看")
  }
}
```

- 以視覺化的方式表示

練習

- 請改寫上面的範例，例如選擇閏年的天數

複習

- 什麼情況會執行 while 迴圈內的敘述，什麼時候會停止

```
continue.flag <- TRUE
## 準備讀取目錄頁
date <- "2018/04/01" # 設定目錄頁讀取日期
page.url <- paste0("https://udn.com/news/archive/2/6638/", date) # 產生
目錄頁網址

while (continue.flag) {
  print(paste("Processing", page.url))
  page.cont <- read_html(page.url) # 讀取目錄頁資料

  page.nodes <- html_nodes(page.cont, css=".pagelink a") # 取得頁碼標示

  page.text <- html_text(page.nodes) # 取得頁碼文字

  page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則
結果為 TRUE，否則為 FALSE

  continue.flag <- any(page.next) #是否有「下一頁」

  if (continue.flag) {
  # 如果有「下一頁」連結，則準備讀取下一個目錄頁
    url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連
結
    page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址
  }
}
```

讀取多個網頁

將讀取新聞標題和連結的程式敘述加入上面的 while 迴圈

- 在 Script 上輸入

```
continue.flag <- TRUE
news.df = data.frame(title = character(),      #所有的新聞標題與連結資料
                     link = character(),
                     stringsAsFactors = FALSE)

## 準備讀取目錄頁
date <- "2018/04/01" # 設定目錄頁讀取日期
page.url <- paste0("https://udn.com/news/archive/2/6638/", date) # 產生
目錄頁網址

while (continue.flag==TRUE) {
  print(paste("Processing", page.url))
  page.cont <- read_html(page.url) # 讀取目錄頁資料

  # 讀取目錄頁上所有的新聞資料
  title.nodes <- html_nodes(page.cont, css="td a")
  # 將這新聞標題與內文連結一起放在同一個data frame 中
  page.df <- data.frame(title=html_text(title.nodes), #目前目錄頁上的新聞
標題與連結資料
                        link=html_attr(title.nodes, "href"),
                        stringsAsFactors = FALSE)

  # 合併先前與目前頁面的新聞資料
  news.df <- rbind(news.df, page.df)
```

```
  page.nodes <- html_nodes(page.cont, css=".pagelink a") # 取得頁碼標示

  page.text <- html_text(page.nodes) # 取得頁碼文字

  page.next <- grepl("下一頁", page.text) # 若頁碼標示文字為「下一頁」，則結果為 TRUE，否則為 FALSE

  continue.flag <- any(page.next) #是否有「下一頁」

  if (continue.flag) {
  # 如果有「下一頁」連結，則準備讀取下一個目錄頁
    url <- html_attr(page.nodes[page.next], "href") # 取得「下一頁」的連結
    page.url <- paste0("https://udn.com", url) # 產生下一個目錄頁網址
  }
}
```

儲存 news.df，便於日後分析

- 儲存為 csv 檔案
- 在 Script 上輸入

```
write.csv(news.df, file=paste0("udn_", gsub("/","_",date),".csv"),
      row.names=FALSE, fileEncoding="UTF-8")
```

- 讀取 csv 檔案
- 在 Console 上輸入，檢視執行結果

```
news.df1 <- read.csv(file=paste0("udn_", gsub("/","_",date),".csv"),
      fileEncoding="UTF-8", stringsAsFactors=FALSE)
View(news.df1)
```

本次課程小結

小結

- 擷取網頁資料，首先針對要擷取的資料，分析對應的網頁 HTML，確認擷取的方式
- 利用 rvest 的 read_html()讀取網頁的 HTML，以 html_nodes()取得內容元件對應的節點資料，分別以 html_text()和 html_attr()取得文字與屬性資料。
- 利用 rvest 的 html_nodes()取得節點資料時，可以特別注意內容元件上的 id 與 class 屬性，使擷取更精確而且容易。

小結

- 如果遇到重複性很高的工作，可以透過程式的方式，節省時間，減少錯誤。
- 本次課程介紹分支結構的 if 和條件迴圈結構的 while，下一次課程將說明列舉迴圈結構的 for 迴圈。

延伸思考

1. 大部分以文章為主的網站，都有類似目錄的結構，本次課程利用 rvest 套件抓取聯合新聞網的歷史新聞目錄。請想想看還有哪些網站也有類似的目錄結構？也請嘗試看看能否運用本次課程的方法抓取該網站的目錄。
2. 擷取目錄的目的是擷取每一篇文章的全文連結，在擷取好目錄後，接下來請思考看看如何擷取文章的全文內容。