

# 網頁資訊擷取 II

Sung-Chien Lin

2018 年 9 月 16 日

## 課程簡介

### 課程內容

- 以聯合報歷史新聞的內文頁為例，利用 **rvest** 套件進行網頁資料擷取
- 將內文頁的資料擷取寫成函數，在程式中呼叫函數
- R 的錯誤處理
- 配合「網頁資訊擷取 I」的目錄頁擷取，完成歷史新聞資料擷取
- 列舉迴圈- **for**

### 學習目標

- 能夠在程式中撰寫函數
- 能夠撰寫有關 **for** 迴圈的程式
- 完成聯合報歷史新聞的網頁擷取

## 擷取新聞內文

- 前次課程已經利用 rvest 套件取得聯合報歷史新聞目錄頁上的新聞標題與連結
- 本次課程將進一步根據新聞連結，利用 rvest 套件取得新聞的內文

### 看到一張手繪孝親「服藥地圖」 家訪社工眼眶都紅了

f 分享 留言 列印 存新聞 A- A+

2018-04-01 09:43 聯合報 記者范榮達／即時報導 讚 6,150 分享

「早飯前血糖藥、一定要吃飯」、「早飯後跟藥盒的藥丸一起吃」...還有「牛奶、豆漿不能喝」，42歲鄭姓婦人為獨居的69歲父親貼心設計一張圖文並茂「服藥地圖」，叮嚀提醒按時服藥，滿溢孝親之情，家訪的社工人員看到不禁紅了眼眶。

苗栗縣政府社會處今年2月底接獲社會救助通報，社工家訪發現，鄭姓老翁的太太離家多年，他到工地當板模工及打零工，獨自扶養2女1男，但兒子目前不知去向，次女因重度身心障礙安置在機構，長女嫁到台中市。鄭姓老翁過去沒有申請任何社會福利幫忙，咬牙苦撐，直到去年底，因病倒下。

來回票價由台幣\*

經濟艙	15,111 元起
商務艙	82,222 元起

立即訂票 ▶

\*票價不含稅及機場附加費。  
優惠及條款及限制請參閱。

Emirates  
阿聯酋航空

WEEKEND SALE

抽繩綁帶連身長褲

NT\$288 <sup>590</sup>

GENQUO

訂閱電子報

## 函數

- 可以將擷取新聞內文的程式片段包裝成一個函數
- 只要提供某一個新聞連結做為函數的參數，便可以取得對應的新聞內文

## 函數

- 進一步來說，可以將擷取某一天的歷史新聞包裝成一個函數
- 提供需要擷取新聞的日期做為參數
- 先根據提供的日期，取得當天新聞目錄中所有標題與連結
- 再依據新聞連結，取得新聞內文

## 錯誤處理

- 許多程式在一般情況下運作都很正常
- 但有時候會出現一些事先無法預料的情形，例如新聞突然被刪除或運算超過程式能處理的範圍，會產生意外的錯誤
- 所以對於容易發生意外錯誤的部分程式，需要進行錯誤處理

## 程式結構

- 前次課程說明了 if 分支結構以及 while 條件迴圈，此次課程將說明 for 列舉迴圈
- 比較 for 迴圈與 while 迴圈：
  - while 迴圈以某一個運算式的結果為 TRUE 或 FALSE(條件)，決定迴圈是否繼續執行，通常在執行前不一定能夠事先知道程式執行的次數
  - for 迴圈則與某一個 vector 的元素有關，執行次數為這些元素的數量，也就是 vector 的長度

## 本次課程程式

```
setwd("rCourse/06")

library(tidyverse)
library(rvest)

newsExtractor <- function(addr) {
  print(paste0("https://udn.com", addr))      #目前讀取新聞頁面
  tryCatch({
    #監看網頁取得時是否會產生錯誤
    story <- paste0("https://udn.com", addr) %>%      #新聞頁面網址
      read_html() %>%      #讀取 HTML 資料
      html_nodes(css="div#story_body_content p") %>% #取得新聞內文節點
      html_text() %>%      #抽出新聞內文
      paste(collapse="\n")      #彙整成一段
  }, error = function(e) {
    print(e)      #一旦錯誤發生
    return(NA)      #列印錯誤原因
  })
}

dayStory <- function(date) {
  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
  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
  }
}

news.df <- news.df %>%
  rowwise() %>%
  mutate(text=newsExtractor(link))

return(news.df)
}

for (i in 1:15) {
  date <- sprintf("2018/09/%02d", i)
  day.news <- dayStory(date)

  write.csv(day.news, 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 06

### 設定工作目錄

- 首先開啟新的 Script
- 在 Script 上，設定工作目錄

```
setwd("rCourse/06")
```

### 載入此次課程所需套件

- 在 Script 上輸入

```
library(tidyverse)  
library(rvest)
```

## 讀取新聞內文網頁

### 設定聯合報歷史新聞頁面的網址

- 在 Console 上執行

```
news.addr <- "news/story/12494/3368757"  
news.url <- paste0("https://udn.com/", news.addr)
```

### 從新聞頁面網址讀取 HTML 資料

- 在 Console 上執行

```
news.cont <- news.url %>%  
  read_html()
```

### 取得新聞內文節點

- 先找到 div(具有一個 id 屬性為 story\_body\_content)
- 然後找到 div 的子元件 p
- css="div#story\_body\_content p"
- 在 Console 上執行

```
p_nodes <- news.cont %>%  
  html_nodes(css="div#story_body_content p")
```

## 抽出新聞內文

- 在 Console 上執行

```
p_story <- p_nodes %>%  
  html_text()
```

## 彙整成一段文字

- 在 Console 上執行

```
story <- p_story %>%  
  paste(collapse="\n")
```

## 將上面的程式寫成一段

- 在 Console 上執行

```
news.addr <- "/news/story/12494/3368757"

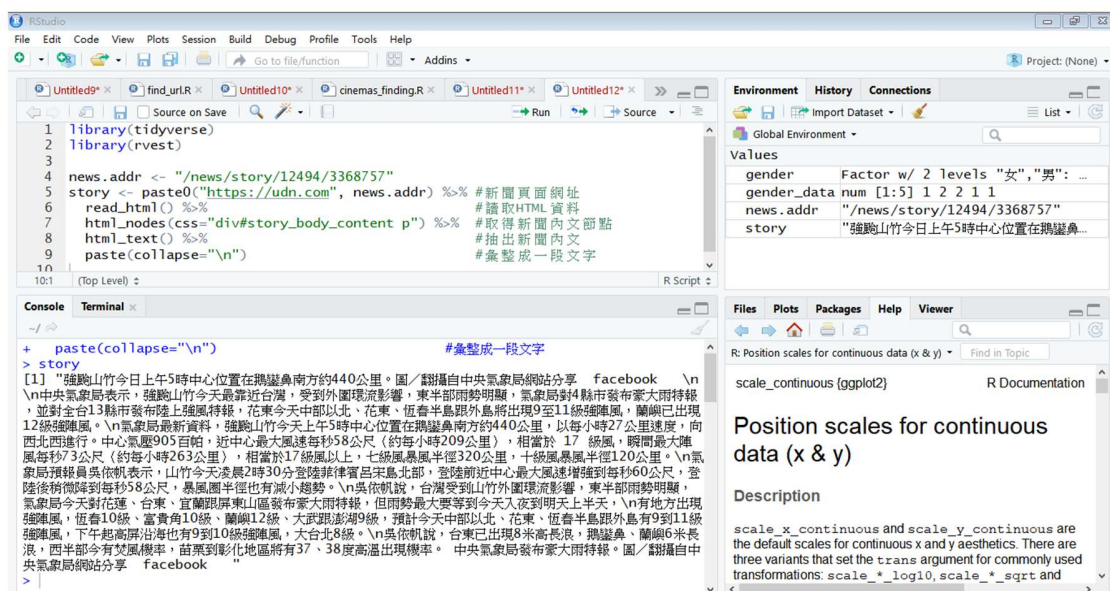
story <- paste0("https://udn.com", news.addr) %>% #新聞頁面網址

read_html() %>% #讀取 HTML 資料

html_nodes(css="div#story_body_content p") %>% #取得新聞內文節點

html_text() %>% #抽出新聞內文

paste(collapse="\n") #彙整成一段文字
```



## 抽取第二則新聞

- 在 Console 上執行

```
news.addr <- "/news/story/12494/3368761"

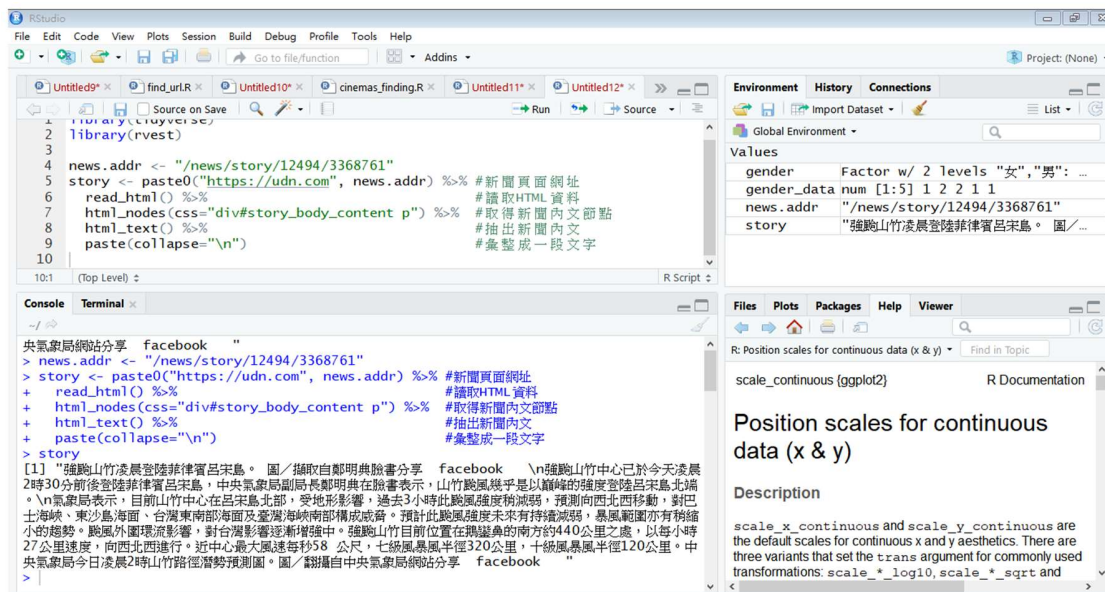
story <- paste0("https://udn.com", news.addr) %>% #新聞頁面網址

read_html() %>% #讀取 HTML 資料

html_nodes(css="div#story_body_content p") %>% #取得新聞內文節點

html_text() %>% #抽出新聞內文

paste(collapse="\n") #彙整成一段
```



## 抽取多則新聞

- 兩段程式中有許多重複的地方
- 可以將重複部分運用函數(Functions)編寫

# 函數

## 使用函數的優點

- 提高程式師的效能
- 避免重複編寫
- 可運用在其他程式
- 減少出錯的可能性
- 提高程式的可讀性

## 函數的寫法

- 根據上面的例子改寫
- 在 Script 上輸入

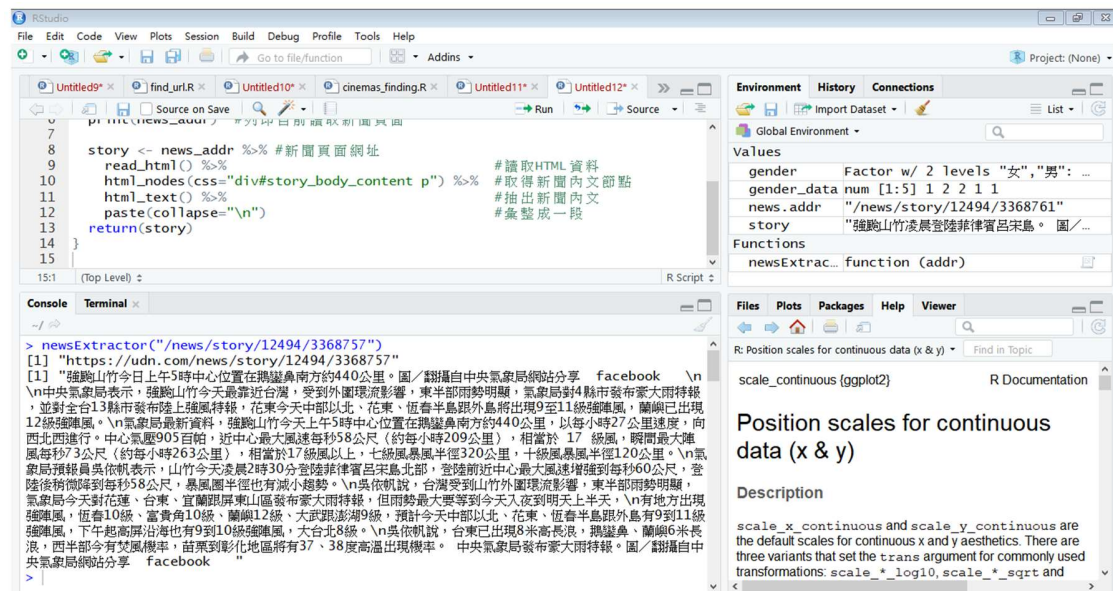
```
newsExtractor <- function(addr) {  
  news_addr <- paste0("https://udn.com", addr)  
  print(news_addr) #列印目前讀取新聞頁面  
  
  story <- news_addr %>% #新聞頁面網址  
    read_html() %>% #讀取 HTML 資料  
    html_nodes(css="div#story_body_content p") %>% #取得新聞內文節點  
    html_text() %>% #抽出新聞內文  
    paste(collapse="\n") #彙整成一段  
  return(story)  
}
```

- newsExtractor：函數名稱
- addr：函數的參數

## 使用函數的方法

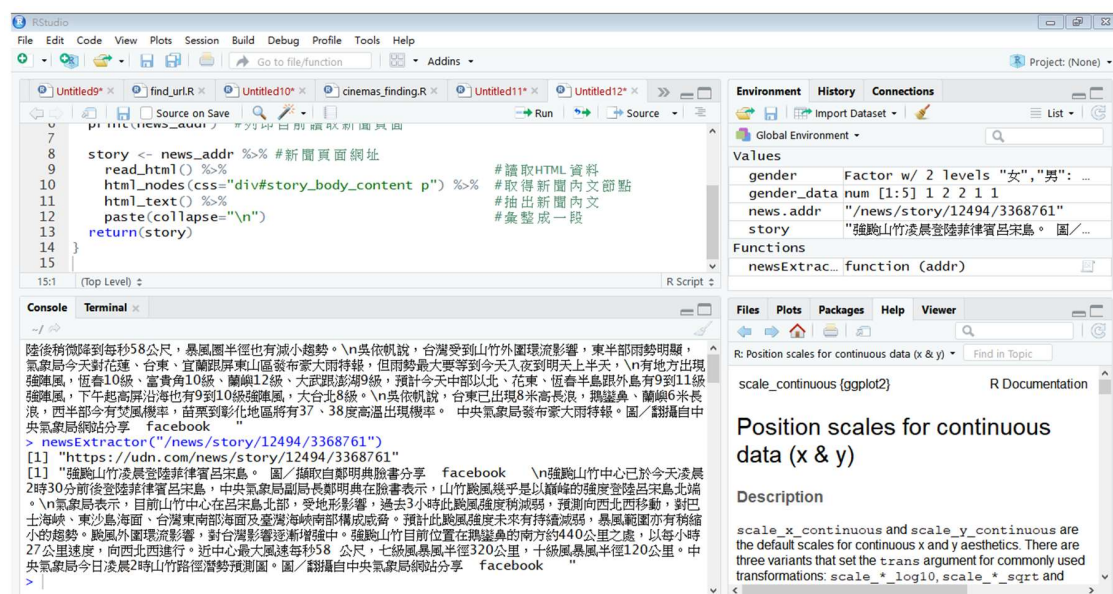
- 在 Console 上輸入

```
newsExtractor("/news/story/12494/3368757")
```



- 在 Console 上輸入

```
newsExtractor("/news/story/12494/3368761")
```



## 從目錄頁網址取得所有新聞頁面的內文

### 讀取所有的目錄頁資料

- 複製上次課程的程式，在 Script 上貼上，執行產生新聞標題與連結的 data frame news.df

```
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 <- "2018/09/01" # 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
}
}

```

## 從目錄頁的連結資料取得新聞內文

- rowwise 函數每次將 tibble(data frame)的一個 row，傳到下一個函數
- 以 news.df 的欄位資料 link 做為 newsExtractor()的參數，取得新聞內文
- 在 Script 上輸入

```
news.df <- news.df %>%  
  rowwise() %>%  
  mutate(text=newsExtractor(link))
```

The screenshot shows the RStudio interface. The top pane displays a dataframe with columns 'title', 'link', and 'text'. The 'link' column contains URLs from udn.com. The bottom pane shows the console output, which lists the URLs being processed. The right pane shows the Environment tab with a list of objects: news.df (71 obs. of 3 variables), page.cont (List of 2), page.df (11 obs. of 2 variables), page.nodes (List of 4), and title.nodes (List of 11). The bottom right pane shows the R Documentation for 'Position scales for continuous data (x & y)'.

- 由於原先的 storyExtractor 函數並沒有處理網頁取得時發生錯誤的問題
- 所以可能會發生錯誤

## 錯誤處理

### 網頁資料取得錯誤

- 由於某些緣故，有些新聞內文的頁面可能已經被移除，或者發生其他網路錯誤
- 如果不處理這些錯誤，也就是略過這些網頁，無法繼續擷取以下的新聞內文
- 所以在 `newsExtractor()` 函數中可加入網頁資料取得錯誤的處理
- 監看新聞內文網頁資料的取得，如果發生錯誤，便將錯誤列印出來，便傳回 NA

### 加入錯誤處理

- `tryCatch(expr, error)`：監看某段程式 `expr`，一但發生錯誤，進行 `error` 的處理
- 改寫 Script 上原先的 `newsExtractor()` 函數

```
newsExtractor <- function(addr) {  
  print(paste0("https://udn.com", addr))      #目前讀取新聞頁面  
  tryCatch({  
    #監看網頁取得時是否會產生錯誤  
    story <- paste0("https://udn.com", addr) %>%      #新聞頁面網址  
      read_html() %>%      #讀取 HTML 資料  
      html_nodes(css="div#story_body_content p") %>% #取得新聞內文節點  
      html_text() %>%      #抽出新聞內文  
      paste(collapse="\n")      #彙整成一段  
  
    return(story)  
  }, error = function(e) {      #一旦錯誤發生  
    print(e)      #列印錯誤原因  
    return(NA)      #傳回"無資料"  
  })  
}
```



## 再次嘗試從目錄頁取得新聞頁面內容

- 在 Console 上輸入

```
news.df <- news.df %>%  
  rowwise() %>%  
  mutate(text=newsExtractor(link))
```

The screenshot shows the RStudio environment with a data frame named `news.df` containing 71 rows and 3 columns: `title`, `link`, and `text`. The console shows the output of the `newsExtractor` function, which returns a list of URLs for each row. The Environment pane shows the data frame structure: `news.df` (71 obs. of 3 variables), `page.cont` (List of 2), `page.df` (11 obs. of 2 variables), `page.nodes` (List of 4), and `title.nodes` (List of 11). The Values pane shows the structure of the `news.df` object, including `continue.f` (FALSE), `date` ("2018/09/01"), `gender` (Factor w/ 2 levels "女", "男"), and `gender.data` (num [1:51] 1 2 2 1 1).

title	link	text
1 麥當勞之亂 10:30之後大部分餐廳暫停營業	/news/story/7270/3342674	板橋麥當勞10點30分真的打烊了，門口擺滿了準備補給的...
2 老司機拿出超厚自製地圖找路 網友感嘆「看了好想哭」	/news/story/7266/3343358	民衆設計程車時，司機直接拿出一本厚厚的地圖找路...
3 麥當勞之亂！員工：累得跟狗一樣 睡覺做惡夢	/news/story/7270/3342739	麥當勞今天上午10時30分起，大部分餐廳暫停營業。1...
4 聞花生醬測失智症？出現這3種警訊更應該當心	/news/story/7266/3343148	近期網路傳出「聞不到花生醬，就是得失智症」訊息...
5 麥當勞之亂？排隊、堵車、塞車「員工皆難日」	/news/story/7270/3342651	昨麥當勞推出大麥克買一送一，民衆排隊到門外。記...
6 小心受騙！搭大麥克熱潮 假優惠「資、姿」不分	/news/story/7266/3342622	詐騙駭客頁面把球后戴資穎寫成戴「姿」穎。圖／擷...
7 吳德榮：燕子增強為今年最強颱風 下周二起轉日	/news/story/7266/3342639	強烈颱風燕子明日至下周一掃近日本南方海域。圖／...

```
[1] "https://udn.com/news/story/7270/3342937"  
[1] "https://udn.com/news/story/7266/3343168"  
[1] "https://udn.com/news/story/7266/3341573"  
[1] "https://udn.com/news/story/7266/3343558"  
[1] "https://udn.com/news/story/7270/3342929"  
[1] "https://udn.com/news/story/7266/3343385"  
[1] "https://udn.com/news/story/7266/3341842"  
[1] "https://udn.com/news/story/7266/3343683"  
[1] "https://udn.com/news/story/7266/3341841"  
[1] "https://udn.com/news/story/7736/3343277"  
> View(news.df)  
>
```

## 擷取一天的新聞內容

- 在 Script 上，將取得每日新聞資料的程式改寫成函數
  - 將日期作為函數的變數

```
dayStory <- function(date) {  
  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  
  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
}
}

news.df <- news.df %>%
  rowwise() %>%
  mutate(text=newsExtractor(link))

return(news.df)
}

```

## 嘗試取得 9 月 1 日的生活新聞資料

- 在 Console 上輸入

```
day.news <- dayStory("2018/09/01")
```

The screenshot shows the RStudio environment with the following components:

- Environment:** Lists objects created by the script: `day.news` (71 obs. of 3 variables), `news.df` (71 obs. of 3 variables), `page.cont` (List of 2), `page.df` (11 obs. of 2 variables), `page.nodes` (List of 4), and `title.nodes` (List of 11).
- Console:** Displays the output of the `dayStory` function, showing a list of 71 URLs from udn.com/news/story/7270/3342937 to udn.com/news/story/7736/3343277.
- Viewer:** Shows the R Documentation for `scale_continuous (ggplot2)`, titled "Position scales for continuous data (x & y)".

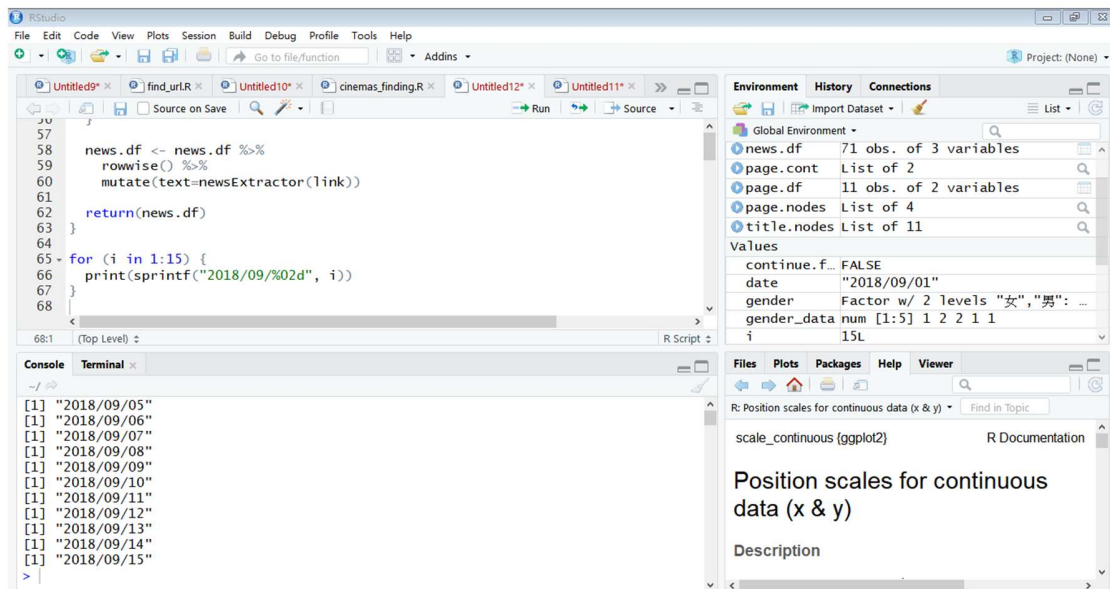


## 取得一個月的新聞

### 產生 2018 年 9 月 1~15 日每天的日期

- 在 Console 上輸入

```
for (i in 1:15) {  
  print(sprintf("2018/09/%02d", i))  
}
```



## for 迴圈

- for 迴圈是包含一個列舉的表示，也就是 `x in someVector`
  - `someVector` 代表一個 `vector`
  - `x` 是一個變數
- 第一次執行 for 迴圈內的敘述時，將 `x` 的值設為 `someVector` 的第一個元素，第二次則將 `x` 的值設為第二個元素，以此類推，最後一次執行時，`x` 的值為 `someVector` 的最後一個元素。
- for 迴圈執行的次數是 `someVector` 的長度，也就是 `length(someVector)`

## 練習

- 從 1 到 N 的總和
- 在 Console 上輸入

```
charFin <- readline("請輸入一個正整數：")
intFin <- as.integer(charFin)
if (!is.na(intFin) & intFin>0) {
  intSum <- 0
  for (i in 1:intFin) {
    intSum <- intSum + i
  }
  print(paste("從 1 到", charFin, "的總和為：", as.character(intSum)))
} else {
  print(paste(charFin, "不是正整數"))
}
```

## 練習

- 修改上面的程式，計算 1 到某一個數之間所有奇數的總和
- `for (i in which((1:intFin)%%2==1))`

## 抓取聯合報 9 月 1~15 日的要聞新聞資料

- 在 Script 上輸入
- 因為執行時間相當長，請回家後再試試

```
for (i in 1:15) {  
  date <- sprintf("2018/09/%02d", i)  
  day.news <- dayStory(date)  
  
  write.csv(day.news, file=paste0("udn_", gsub("/", "_", date), ".csv"),  
            row.names=FALSE, fileEncoding="UTF-8")  
}
```

## 本次課程小結

### 小結

- 這兩次課程運用了擷取文字資料的一般做法：先透過目錄頁取得文章標題與連結，再以連結取得文章內文。
- 除了聯合報之外，蘋果日報、中國時報和自由時報等新聞媒體以及 PTT BBS 等也都是相同做法，只需要注意各媒體目錄頁與內文頁的 HTML 寫法。

### 小結

- 這兩次課程使用了 if、while、for 等程式結構自動取得文字資料。
- if 是分支結構，根據設定的條件(if 後的運算式)，決定某些程式敘述是否被執行的選擇機制。
- while 與 for 則是迴圈結構，提供某些程式敘述可以執行多次的機制。
- for 是列舉迴圈。在程式執行前，便已經利用設定的項目(for 後的向量)，預先決定執行次數。
- while 是條件迴圈，當設定的條件(while 後的運算式)成立，便會重複執行程式。

### 小結

- 若有一段程式經常需要使用，可以將其包裝為函數，節省開發時間。
- 使用函數時，可以改變輸入的參數。
- 事實上，R 語言本身便提供許多基礎函數，而且大量的套件也都以函數的形式提供程式開發者運用。

## 延伸思考

1. 在從聯合報網站擷取歷史新聞網頁之後，請想想如何運用擷取的網頁內容進行研究。
2. 這兩次課程介紹網頁擷取的方法，或許你也會想嘗試看看擷取其他新聞網站上的新聞資料。