Pipelines For Data Analysis In R, part 2

author: 曾意儒 Yi-Ju Tseng date: 2017/05/06 autosize: true font-family: 'Microsoft JhengHei' navigation: slide

資料分析步驟

- **資料匯入**
- **資料清洗處理**並轉換為Tidy data
- 資料分析
- 資料呈現與視覺化

資料匯入

- 從檔案匯入
- 從網路匯入
- 從Facebook匯入
- 資料匯出

從檔案匯入

type:section

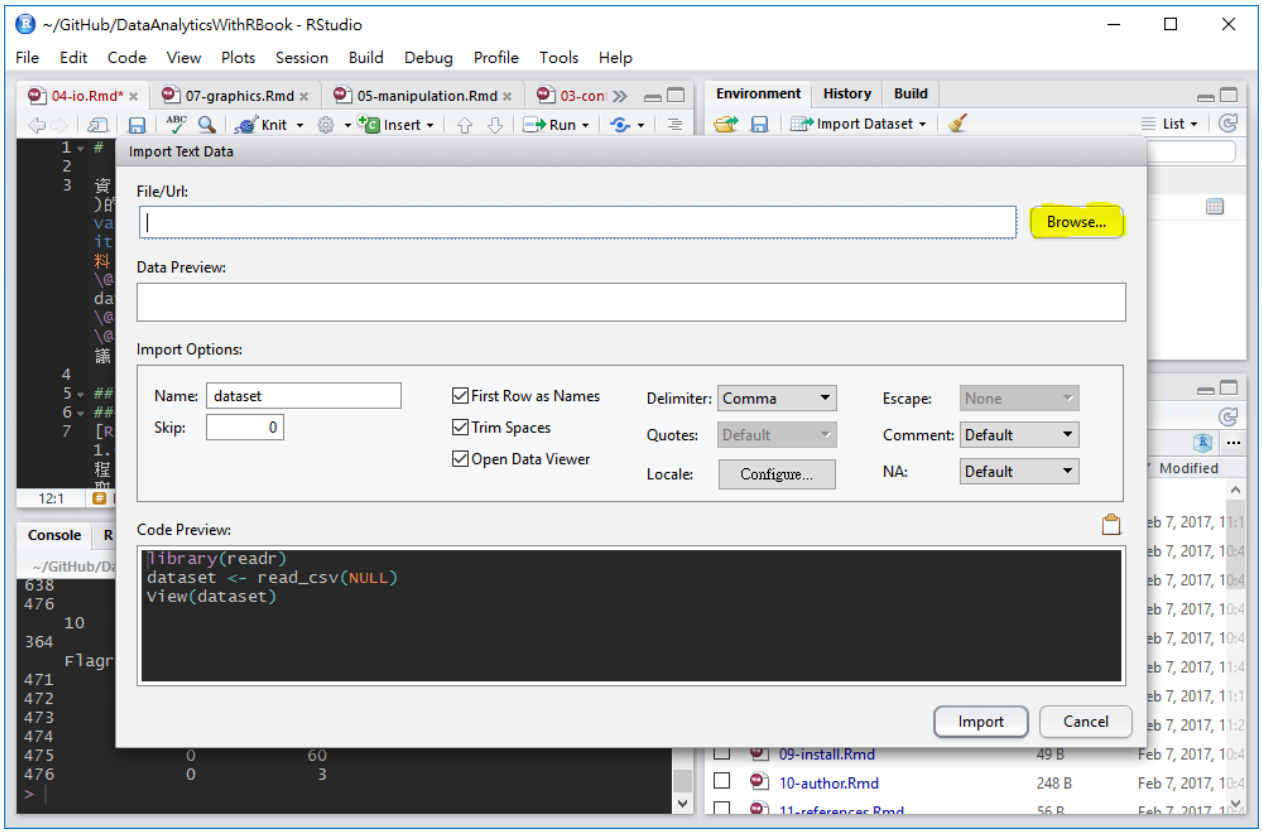
- Import Dataset功能 (RStudio)
- R物件 .rds
- R程式 .R
- 純文字資料 (無分隔)
- 其他格式

Import Dataset功能 (RStudio)

選取RStudio四分割視窗右上角的Environment標籤，選擇Import Dataset

Import Dataset功能 (RStudio)

- 選取 `From CSV`
- 點選 `Browse` 按鈕開啟檔案選取器



Import Dataset功能 (RStudio)

- 利用下方 `Import Options` 的選項微調參數
  - `Delimiter` 分隔符號
  - `First Row as Names` 首列是否為欄位名稱

Import Dataset功能 (RStudio)

type:alert incremental:true

- 操作範例檔案
- 若匯入的檔案為tab**分隔文字檔**? 該如何調整參數？

R物件 .rds

type:sub-section 如果在R程式內處理完資料後，必須儲存一份以供後續分析，使用R物件儲存是最佳的方式

- 檔案小
- 讀取快速
- 推薦使用 `readRDS()` 函數讀取RDS檔案
- A better way of saving and loading objects in R

```
dataset <- readRDS("檔案路徑與名稱")
```

R程式 .R

type:sub-section

- `source` 函數
- 讀R的Obejct or script, **執行**

- **實際操作範例**
  - 有一份example.R檔在工作環境中
  - 一次執行檔案內所有R指令

```
source("example.R")
```

純文字資料 (無分隔)

type:sub-section `readLines`, 逐行讀取文字資料

從網路匯入

type:section

- Open Data
- XML 可延伸標記式語言
- 網頁爬蟲 Webscraping
- API (Application programming interfaces)
- JSON格式檔案

Open Data 開放資料

type:sub-section

- 2011年推動開放政府與開放資料 (維基百科)
- 不受著作權、專利權，以及其他管理機制所限制，任何人都可以自由出版使用
- 常見的儲存方式為:
  - `CSV`
  - `JSON`
  - `XML`

Open Data 開放資料常見平台

- 政府資料開放平台
- Data Taipei
- 開放資料 x 開放桃園
- 內政資料開放平台

XML 可延伸標記式語言

type:sub-section

- Extensible markup language
- 描述**結構化**資料的語言
- 處理XML檔案是網頁Html爬蟲的基礎
- Components
  - Markup 標記 - labels that give the text structure
  - Content 內文 - the actual text of the document
- XML Wiki

XML 可延伸標記式語言

Tags, elements and attributes

- Tags correspond to general labels
  - Start tags `<breakfast_menu>` , `<price>`
  - End tags `</breakfast_menu>` , `</price>`
  - Empty tags `<line-break />`
- Elements are specific examples of tags
  - `<name>Belgian Waffles</name>`
- Attributes are components of the label
  - `<book category="web">`

XML 可延伸標記式語言-讀取

- 臺北市水質監測資訊
- 安裝 `XML` package
- `xmlParse()` 函數將XML檔案匯入

```
library(XML)
waterURL<-"http://data.taipei/opendata/datalist/datasetMeta/download?id=961ca397-4a59-45e8-b312-697f26b059dc&rid=1907
waterQ <- xmlParse(waterURL)
```

xpath?

- XML路徑語言（XML Path Language）
- 基於XML的樹狀結構，提供在資料結構樹中找尋節點的能力
- 維基百科
- 常見語法

XML 可延伸標記式語言-解析

使用 `xpathSApply()` 函數取得指定標籤內的資料

```
#取得所有"code_name"標籤內的資料
xpathSApply(waterQ,"//code_name",xmlValue)[1:10]
```

```
 [1] "雙溪淨水場"                "衛理女中"
 [3] "雙溪國小"                 " "華興加壓站"
 [5] "長興淨水場"                "市政大樓"
 [7] "市議會"                   "捷運忠孝復興站"
 [9] "南港高工"                 "南港加壓站"
```

XML 可延伸標記式語言-解析

使用 `xpathSApply()` 函數取得指定標籤內的資料

```
#取得各監測站的經度longitude
xpathSApply(waterQ,"//longitude",xmlValue)[1:10]
```

```
 [1] "121.56094" "121.54401" "121.55557" "121.53476" "121.54043"
 [6] "121.55661" "121.55360" "121.53551" "121.59892" "121.60829"
```

XML檔案匯入練習

type:alert incremental:true

- 載入桃園捷運車站出入口基本資料
- 嘗試取得各站出入口敘述(LocationDescription)與經緯度(PositionLon,PositionLat)
- 參考剛剛的水站範例

```
library(XML)
waterURL<-"http://data.taipei/opendata/datalist/datasetMeta/download?id=961ca397-4a59-45e8-b312-697f26b059dc&rid=1907
waterQ <- xmlParse(waterURL)
xpathSApply(waterQ,"//longitude",xmlValue)[1:10]
```

API

type:sub-section

- 應用程式介面
- Application Programming Interfaces
- 為了讓第三方的開發者可以額外開發應用程式來強化他們的產品，推出可以與系統溝通的介面
- 有API輔助可將資料擷取過程自動化
  - 以下載Open Data為例，若檔案更新頻繁，使用手動下載相當耗時
- 維基百科

API - Open Data

- 臺北市開放認養動物資料
- 每日更新
- 不可能每日手動下載
- 提供透過API下載的服務
- 透過API下載的資料格式: JSON**格式**

- 臺北市開放認養動物API資訊
  - **資料集**ID: 紀錄資料的基本參數，如包含欄位、更新頻率等
  - **資料**RID: 資料集
  - 擷取範例

JSON格式檔案

type:sub-section

- JSON (Javascript Object Notation)
- 輕量級的資料交換語言
- From application programming interfaces (APIs)
- JavaScript、Java、Node.js應用
- 一些NoSQL資料庫用JSON儲存資料：MongoDB
- Wiki

JSON檔案匯入

- `jsonlite` package (套件使用前必須安裝)
- `fromJSON()` 函數載入JSON資料
- 如果API網址為https，則需使用 `httr` package
  - 使用 `GET()` 函數處理資料擷取網址

```
library(jsonlite)
library(RCurl)
PetData<-fromJSON("http://data.taipei/opendata/datalist/apiAccess?scope=resourceAquire&rid=f4a75ba9-7721-4363-884d-c3
```

JSON檔案匯入

- 轉存為 列表list 的型態
- 五個子元素(offset, limit, count, sort, results)
- results子元素的類別為資料框data.frame

```
str(PetData)
```

```
List of 1
 $ result:List of 5
  ..$ offset : int 0
  ..$ limit  : int 10000
  ..$ count  : int 353
  ..$ sort   : chr ""
  ..$ results:'data.frame':     353 obs. of  20 variables:
  .. ..$ _id            : chr [1:353] "1" "2" "3" "4" ...
  .. ..$ Name           : chr [1:353] "心光" "琥碧" "芽美" "蛋頭" ...
  .. ..$ Sex            : chr [1:353] "雌" "雌" "雌" "雄" ...
  .. ..$ Type           : chr [1:353] "貓" "貓" "貓" "貓" ...
  .. ..$ Build          : chr [1:353] "中" "中" "中" "中" ...
  .. ..$ Age            : chr [1:353] "成年" "成年" "成年" "成年" ...
  .. ..$ Variety        : chr [1:353] "米克斯" "米克斯" "米克斯" "米克斯" ...
  .. ..$ Reason         : chr [1:353] "動物管制" "動物管制" "動物管制" "動物管制" ...
  .. ..$ AcceptNum      : chr [1:353] "106042903" "106042715" "106042714" "106042409" ...
  .. ..$ ChipNum        : chr [1:353] "" "" "" "" ...
  .. ..$ IsSterilization: chr [1:353] "未絕育" "未絕育" "未絕育" "未絕育" ...
  .. ..$ HairType       : chr [1:353] "虎斑白" "虎斑" "黑白" "虎斑白" ...
  .. ..$ Note           : chr [1:353] "右眼混濁\n大家好~我的名字叫心光，我的眼睛不太好，希望還是有好心人願意來帶我回家!\n" '
  .. ..$ Resettlement   : chr [1:353] "臺北市動物之家 收容編號106042903" "臺北市動物之家 收容編號106042715" "臺北市動物之家
  .. ..$ Phone          : chr [1:353] "02-87913062" "02-87913062" "02-87913062" "02-87913062" ...
  .. ..$ Email          : chr [1:353] "tcapoa8@mail.taipei.gov.tw" "tcapoa8@mail.taipei.gov.tw" "tcapoa8@mail.taipei.
  .. ..$ ChildreAnlong  : chr [1:353] "" "" "" "" ...
  .. ..$ AnimalAnlong   : chr [1:353] "" "" "" "" ...
  .. ..$ Bodyweight     : chr [1:353] "" "" "" "" ...
  .. ..$ ImageName      : chr [1:353] "http://163.29.39.183/uploads/images/medium/9144b470-bf9f-4283-87e8-9166eeb0a6c
```

JSON檔案解析

- 使用 $ 符號截取元素與子元素

```
head(PetData$result$results)
```

_id	Name	Sex	Type	Build	Age	Variety	Reason	AcceptNum
1	心光	雌	貓	中	成年	米克斯	動物管制	106042903
2	琥碧	雌	貓	中	成年	米克斯	動物管制	106042715
3	芽美	雌	貓	中	成年	米克斯	動物管制	106042714
4	蛋頭	雄	貓	中	成年	米克斯	動物管制	106042409
5	黑虎	雄	貓	中	老年	米克斯	民眾不擬續養	106042309
6	阿咪	雌	貓	中	成年	米克斯	民眾不擬續養	106042308
7	小花	雌	貓	中	成年	米克斯	民眾不擬續養	106042306
8	Tiger	雌	貓	中	老年	米克斯	民眾不擬續養	106042304

JSON檔案解析

分析各項**開放認養理由**出現次數

```
table(PetData$result$results$Reason)
```

Var1	Freq
	27
動物管制	137
動物救援	112
民眾不擬續養	52
民眾拾獲	25
分析可知開放認養理由以動物管制居多	

JSON檔案匯入練習

type:alert incremental:true

- 練習用資料：「臺北市今日施工資訊」API存取
- 使用檔案匯入**範例**，將資料匯入R中
  - 提示：fromJSON
- 使用str()函數觀察匯入的資料
- 請問今日施工資料有幾筆觀察值？幾個欄位？

網頁爬蟲 Webscraping

type:sub-section

- 不是每個網站都提供API
- 人工複製貼上?!
- 程式化的方式擷取網頁資料: **網頁爬蟲**（Webscraping）（Webscraping Wiki）
- 可能耗費很多網頁流量和資源 －很可能被鎖IP
- 在R的處理辦法
  - 當作XML檔案處理分析
  - 使用 `rvest` package輔助

網頁爬蟲 Webscraping-rvest

載入rvest套件後，經由以下步驟進行網站解析：

- 使用 `read_html("欲擷取的網站網址")` 函數讀取網頁
- 使用 `html_nodes()` 函數擷取所需內容 (條件為CSS或xpath標籤)
- 使用 `html_text()` 函數處理/清洗擷取內容，留下需要的資料
- 使用 `html_attr()` 函數擷取資料參數（如連結url）

網頁爬蟲 Webscraping-rvest

```
library(rvest) ##載入
YahooNewsurl<-"https://tw.news.yahoo.com/"
news_title <- read_html(YahooNewsurl) %>% html_nodes(".tpl-title a") %>% html_text()
news_url <- read_html(YahooNewsurl) %>% html_nodes(".tpl-title a") %>% html_attr("href")
Yahoo_news <- data.frame(title = news_title, url=news_url)
head(Yahoo_news)
```

```
                                title                                                  url
1          曾1妻5妾好風光 男星慘賣豪宅還債 /從1妻5妾的風光到變賣豪宅還債-網友噓雷洪：活該-091741737.html
2           美報告：美棄「一中」台灣更危險                  /美報告-美拋棄-中-台灣處境更危險-081036215.html
```

網頁爬蟲 Webscraping-rvest

- 擷取條件的撰寫會因網頁語法不同而有差異
- 使用Google Chrome**開發工具**輔助觀察擷取資料的條件
- 使用xpath-helper輔助xpath標籤的擷取
- 觀察需要擷取的資料所在HTML片段
  - 新聞清單被包含在 `ul` 標籤下
  - 且css class為 `tpl-title yom-list list-style-none`

```
<ul class="tpl-title yom-list list-style-none" id="yui_3_9_1_1_1486568229946_2408">
<li class="list-story first" id="yui_3_9_1_1_1486568229946_2407">
<div class="txt" id="yui_3_9_1_1_1486568229946_2406">
<a href="/從1妻5妾的風光到變賣豪宅還債-網友噓雷洪：活該-091741737.html" class="title " data-ylk="pkg:96a0ca11-47bc-3100-8
<cite id="yui_3_9_1_1_1486568229946_2405">
<span class="provider" id="yui_3_9_1_1_1486568229946_2404">Yahoo奇摩娛樂新聞</span>
</cite></div></li>
....
```

網頁爬蟲 DCard實作 -1

```
library(rvest) ##(爬蟲結果不代表本人意見)
DCardCGU<-"https://www.dcard.tw/f/cgu?latest=true"
DCardContent<-read_html(DCardCGU)
post_title <- DCardContent %>% html_nodes(".PostEntry_titleUnread_ycJL0") %>% html_text()
post_contentShort<- DCardContent %>% html_nodes(".PostEntry_excerpt_A0Bmh") %>% html_text()
post_author<- DCardContent %>% html_nodes(".PostAuthor_root_3vAJf") %>% html_text()
post_comment<- DCardContent %>% html_nodes(".PostEntry_commentUnread_1cVyd") %>% html_text()
```

網頁爬蟲 DCard實作 -2

```
##(爬蟲結果不代表本人意見)
post_like<- DCardContent %>% html_nodes(".PostLikeCount_likeCount_2uhBH") %>% html_text()
post_url <- DCardContent %>% html_nodes(".PostEntry_entry_2rsgm") %>% html_attr("href")
DCardCGU_posts <- data.frame(title = post_title, author=post_author,
                             content=post_contentShort, commentN=post_comment,
                             likeN=post_like,
                             url=paste0("https://www.dcard.tw",post_url))
```

網頁爬蟲 DCard實作 -3

```
##(爬蟲結果不代表本人意見)
knitr::kable(
    DCardCGU_posts[1:5,c("title","author","commentN")])
```

title	author	commentN
明德寧靜寢室申請	長庚大學	1
尋找球鞋	長庚大學 機械工程學系	1
好漢坡的蛞蝓	長庚大學	2
騎車出去也要插隊	長庚大學	1
長庚看流星雨？	長庚大學	2

爬蟲練習

type:alert

- Ptt PokemonGo 版
- 試著爬出所有**標題**
- 爬出的第三個標題是？

網頁爬蟲 再想想？

incremental:true

- 如何爬評論跟內文呢？
- 其實...DCard是有API的
  - https://www.dcard.tw/_api/forums/cgu/posts
  - https://www.dcard.tw/_api/posts/225917717
  - https://www.dcard.tw/_api/posts/225917717/comments
- 隱私問題 （去年的OkCupid事件）
  - 70,000 OkCupid Users Just Had Their Data Published

進階爬蟲

- CSS Selector 語法介紹 參考資料
  - **.**xxx：select elements with class="xxx"
  - **#**xxx：select elements with id="xxx"
  - [yyy]：select elements with attribute yyy
  - [yyy=zzz]：select elements with attribute yyy="zzz"
- 瀑布式網頁爬蟲
  - 觀察Google Chrome 開發者工具，在Network內找到api呼叫方式
  - 搭配使用RSelenium 模擬瀏覽狀態 DCard實作R Code

其他爬蟲相關參考資源

- 網路爬蟲實作 - 用 r 語言打造自己的爬蟲程式
- rvest GitHub
- R Bloggers 有很多爬蟲範例（英文）
- Ptt爬蟲實作
- 大數學堂 網頁爬蟲課程

從Facebook匯入

type:section

- Graph API in R
- Rfacebook package

Graph API in R

type:sub-section

- Graph API
  - 根據篩選條件，回傳JSON格式的資料
- Graph API Explorer
  - 測試資料撈取方法和結果
- 必須要取得自己的access token (存取權杖)
  - 可在Graph API Explorer視窗右上角的Get Token按鈕取得

- 官方文件

Rfacebook package

type:sub-section

使用 Rfacebook 取得 `tsaiingwen` 粉絲頁的資料

```
library(Rfacebook) #初次使用須先安裝
token<-"your token" #將token複製到此處
getPage("tsaiingwen", token,n = 5)
```

課堂操作

```
4 posts       from_id           from_name
1 46251501064 蔡英文 Tsai Ing-wen
2 46251501064 蔡英文 Tsai Ing-wen
3 46251501064 蔡英文 Tsai Ing-wen
4 46251501064 蔡英文 Tsai Ing-wen
```

Rfacebook package練習

type:alert incremental:true

- 取得Facebook access token
- 使用Rfacebook package取得CGSGA **長庚學生會**粉絲頁面的前五筆資料
- 第一筆資料的likes_count是多少?
- 第二筆資料的shares_count是多少?

Rfacebook package

- 每次擷取資料的比數有上限（大概30筆）
- 如需取得更多資料: 使用迴圈協助
  - since 和 until參數，可設定資料擷取區間。
- 先取得日期向量，供後續迴圈做使用

```
lastDate<-Sys.Date()
DateVector<-seq(as.Date("2017-01-01"),lastDate,by="5 days")
DateVectorStr<-as.character(DateVector)
DateVectorStr
```

```
## "2017-01-01" "2017-01-06" "2017-01-11" "2017-01-16" "2017-01-21" "2017-01-26" "2017-01-31" "2017-02-05"
```

Rfacebook package

利用上述日期向量資料，搭配迴圈，依序設定since 和 until參數

```
totalPage<-NULL
token<-'your token'
for(i in 1:(length(DateVectorStr)-1)){
    tempPage<-getPage("tsaiingwen", token,
                since = DateVectorStr[i],
                until = DateVectorStr[i+1])
    totalPage<-rbind(totalPage,tempPage)
}
nrow(totalPage)
```

```
## [1] 42
```

資料匯出

type:section

- 文字檔 .txt
- CSV檔 .csv
- R物件 .rds

文字檔 .txt write.table()

type:sub-section

```
write.table(iris,file="iris.txt",sep=",",
            row.names = F,col.names = T)
```

- 要匯出的資料
- `file` 檔案名稱
- `append` T/F T:在檔案後加字，F:直接覆蓋檔案 (預設F)
- `quote` 是否需要用雙引號將字串包起 (預設T)
- `sep` 分隔符號 (預設空白)
- `row.names` T/F 是否需要輸出row names
- `col.names` T/F 是否需要輸出column names
- `fileEncoding` 編碼設定

CSV檔 .csv

type:sub-section

與 `write.table()` 類似，使用 `write.csv()` 函數寫入檔案

```
write.csv(iris,file="iris.csv",row.names = F)
```

R物件 .rds

type:sub-section

若是要在R的環境繼續使用，建議匯出成R物件檔案(.rds)

```
saveRDS(iris,"iris.rds")
```

資料清洗與處理

- Tidy Data
- 資料型別轉換處理
- 文字字串處理
- 子集Subset
- 排序
- 資料組合
- 長表與寬表

Tidy Data

type:sub-section

Each column is a variable. Each row is an observation.

- 一個欄位（Column）內只有一個數值，最好要有凡人看得懂的Column Name
- 不同的觀察值應該要在不同行（Row）
- 一張表裡面，有所有分析需要的資料
- 如果一定要多張表，中間一定要有index可以把表串起來
- One file, one table

資料型別轉換處理

type:sub-section 包括**資料型別檢查**與**資料型別轉換**

資料型別:

- 數值 (numeric)
- 字串 (character)
- 布林變數 (logic)
- 日期 (Date)

資料型別檢查 - is.

使用 `is.` 函數檢查資料型別，回傳布林變數，若為**真**，回傳TRUE

- 是否為**數字** `is.numeric(變數名稱)`
- 是否為**文字** `is.character(變數名稱)`
- 是否為**布林變數** `is.logical(變數名稱)`

```
num<-100
is.numeric(num)
```

```
[1] TRUE
```

```
is.character(num)
```

```
[1] FALSE
```

資料型別檢查 - class()

使用 `class(變數名稱)` 函數，直接回傳**資料型別**

```
class(num)
```

```
[1] "numeric"
```

```
class(Sys.Date())
```

```
[1] "Date"
```

資料型別轉換 - as.

使用 as. 函數轉換型別

- 轉換為**數字** as.numeric(變數名稱)
- 轉換為**文字** as.character(變數名稱)
- 轉換為**布林變數** as.logical(變數名稱)

```
cha<-"100"
as.numeric(cha)
```

```
[1] 100
```

資料型別轉換 - as.

若無法順利完成轉換，會回傳空值 NA ，並出現警告訊息

```
as.numeric("abc")
```

```
[1] NA
```

資料型別轉換練習

type:alert incremental:true 回想起DCard(爬蟲結果不代表本人意見)的資料．．．

```
library(rvest) ##載入
DCardCGU<-"https://www.dcard.tw/f/cgu?latest=true"
DCardContent<-read_html(DCardCGU)
post_title <- DCardContent %>% html_nodes(".PostEntry_titleUnread_ycJL0") %>% html_text()
post_comment<- DCardContent %>% html_nodes(".PostEntry_commentUnread_1cVyd") %>% html_text()
post_like<- DCardContent %>% html_nodes(".PostLikeCount_likeCount_2uhBH") %>% html_text()
DCardCGU_posts <- data.frame(title = post_title, commentN=post_comment,
                             likeN=post_like,stringsAsFactors = F)
```

資料型別轉換練習

type:alert 評論數和按讚數都是字串型別 (chr)

```
str(DCardCGU_posts)
```

```
'data.frame':   30 obs. of  3 variables:
 $ title   : chr  "明德寧靜寢室申請" "尋找球鞋\U0001f62d\U0001f62d\U0001f62d\U0001f62d" "好漢坡的蛞蝓" "騎車出去也要插隊"
 $ commentN: chr  "1" "1" "2" "1" ...
 $ likeN   : chr  "1" "1" "4" "6" ...
```

該如何將這兩個欄位轉成數字呢？

文字字串處理

type:sub-section

- 基本處理
- 搜尋字串
- Regular Expression 正規表示式 @ R

基本處理

- 切割 `strsplit()` Split
- 子集 `substr()` sub string
- 大小寫轉換 `toupper()` `tolower()`
- 兩文字連接 `paste()` `paste0()`
- 文字取代 `gsub()` substitute
- 前後空白去除 `str_trim()` 需安裝 `stringr` package trim

基本處理-切割

strsplit (欲切割的字串,用什麼符號切割)

```
strsplit ("Hello World"," ")
```

```
[[1]]
[1] "Hello" "World"
```

基本處理-切割（多字元）|

strsplit (欲切割的字串,切割符號1|切割符號2|...)

```
strsplit ("Hello World"," |o")
```

```
[[1]]
[1] "Hell" ""     "W"    "rld"
```

基本處理-子集（切一小段）

substr(欲做子集的字串,開始位置,結束位置)

```
substr("Hello World", start=2,stop=4)
```

```
[1] "ell"
```

基本處理-大小寫轉換

```
toupper("Hello World")
```

```
[1] "HELLO WORLD"
```

```
tolower("Hello World")
```

```
[1] "hello world"
```

基本處理-兩文字連接

paste(欲連接的字串1, 欲連接的字串2, 欲連接的字串3,.... sep=中間用什麼符號分隔)

```
paste("Hello", "World")
```

```
[1] "Hello World"
```

```
paste("Hello", "World", sep='')
```

```
[1] "HelloWorld"
```

```
paste0("Hello", "World")
```

```
[1] "HelloWorld"
```

基本處理-文字取代

gsub(想要換掉的舊字串,想要換成的新字串,欲作取代的完整字串)

```
gsub("o","0","Hello World")
```

```
[1] "Hell0 W0rld"
```

基本處理-文字取代（多字元）|

gsub(想要換掉的舊字串1|想要換掉的舊字串2|...,想要換成的新字串,欲作取代的完整字串)

```
gsub("o|l","0","Hello World")
```

```
[1] "He000 W0r0d"
```

基本處理-前後空白去除

str_trim要使用前需要安裝與載入stringr套件

```
library(stringr)
str_trim(" Hello World ")
```

```
[1] "Hello World"
```

搜尋字串

- 通常使用在**比對文字向量**
- **有分大小寫**
- 依照回傳值的型態不同，有兩種常用函數
  - 回傳符合條件之向量位置(index) `grep(搜尋條件,要搜尋的向量)`
  - 回傳每個向量是否符合條件(TRUE or FALSE) `grepl(搜尋條件,要搜尋的向量)`

```
##在姓名文字向量中尋找A，回傳包含"A"之元素位置
grep("A",c("Alex","Tom","Amy","Joy","Emma"))
```

```
[1] 1 3
```

搜尋字串 - grepl()

```
##在姓名文字向量中尋找A，回傳各元素是否包含"A"
grepl("A",c("Alex","Tom","Amy","Joy","Emma"))
```

```
[1]  TRUE FALSE  TRUE FALSE FALSE
```

```
##在姓名文字向量中尋找a，回傳各元素是否包含"a"
grepl("a",c("Alex","Tom","Amy","Joy","Emma"))
```

```
[1] FALSE FALSE FALSE FALSE  TRUE
```

搜尋字串 - grep()

```
##在姓名文字向量中尋找A，回傳包含"A"的元素位置
grep("A",c("Alex","Tom","Amy","Joy","Emma"))
```

```
[1] 1 3
```

```
##在姓名文字向量中尋找a，回傳包含"a"的元素位置
grep("a",c("Alex","Tom","Amy","Joy","Emma"))
```

```
[1] 5
```

搜尋字串 - grep()

type:alert 多字元？

子集Subset - 一維資料

type:sub-section

之前有介紹使用 [ ] 取出單一或多個元素的方法

```
letters ##R語言內建資料之一
```

```
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q"
[18] "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

```
letters[1] ##取出letters向量的第一個元素
```

```
[1] "a"
```

子集Subset - 一維資料

也可以用"負號"去掉不要的資料

```
letters[c(1,3,5)] ##取出letters向量的第1,3,5個元素
```

```
[1] "a" "c" "e"
```

```
letters[c(-1,-3,-5)] ##取出letters向量除了第1,3,5個元素之外的所有元素
```

```
 [1] "b" "d" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
[18] "u" "v" "w" "x" "y" "z"
```

子集Subset - 一維資料

若想要快速取得一向量的開頭與結尾元素，可使用 `head()` 和 `tail()` 函數

```
head(letters,5) ##取出letters向量的前五個元素
```

```
[1] "a" "b" "c" "d" "e"
```

```
tail(letters,3) ##取出letters向量的後三個元素
```

```
[1] "x" "y" "z"
```

子集Subset - 二維資料

type:sub-section

- 可針對列(Row)和行(Column)做子集
- 使用 `[ ]` ，但因應二維資料的需求，以 , 分隔列與行的篩選條件
- 資料篩選原則為**前**Row,**後**Column，**前列,後行**
- 若不想篩選列，則在 , 前方保持**空白**即可。
- 篩選方式可輸入**位置**(index)、**欄位名稱**或輸入**布林變數**(TRUE/FALSE)
  - 輸入位置: `dataFrame[row index,column index]`
  - 輸入布林變數: `dataFrame[c(T,F,T),c(T,F,T)]`
  - 輸入欄位名稱: `dataFrame[row name,column name]`

子集Subset - 二維資料 []

```
iris[1,2] ##第一列Row，第二行Column
```

```
[1] 3.5
```

```
iris[1:3,] ##第1~3列Row，所有的行Column
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa

子集Subset - 二維資料 []

```
iris[1:10,c(T,F,T,F,T)] ##第1~10列Row，第1,3,5行Column (TRUE)
```

Sepal.Length	Petal.Length	Species
5.1	1.4	setosa
4.9	1.4	setosa
4.7	1.3	setosa
4.6	1.5	setosa
5.0	1.4	setosa
5.4	1.7	setosa
4.6	1.4	setosa
5.0	1.5	setosa
4.4	1.4	setosa
4.9	1.5	setosa

子集Subset - 二維資料 []

```
iris[,"Species"] ##所有的列Row，名稱為Species的行Column
```

```
  [1] setosa     setosa     setosa     setosa     setosa     setosa
  [7] setosa     setosa     setosa     setosa     setosa     setosa
 [13] setosa     setosa     setosa     setosa     setosa     setosa
 [19] setosa     setosa     setosa     setosa     setosa     setosa
 [25] setosa     setosa     setosa     setosa     setosa     setosa
 [31] setosa     setosa     setosa     setosa     setosa     setosa
 [37] setosa     setosa     setosa     setosa     setosa     setosa
 [43] setosa     setosa     setosa     setosa     setosa     setosa
 [49] setosa     setosa     versicolor versicolor versicolor versicolor
 [55] versicolor versicolor versicolor versicolor versicolor versicolor
 [61] versicolor versicolor versicolor versicolor versicolor versicolor
 [67] versicolor versicolor versicolor versicolor versicolor versicolor
 [73] versicolor versicolor versicolor versicolor versicolor versicolor
 [79] versicolor versicolor versicolor versicolor versicolor versicolor
 [85] versicolor versicolor versicolor versicolor versicolor versicolor
 [91] versicolor versicolor versicolor versicolor versicolor versicolor
 [97] versicolor versicolor versicolor versicolor virginica  virginica
[103] virginica  virginica  virginica  virginica  virginica  virginica
[109] virginica  virginica  virginica  virginica  virginica  virginica
[115] virginica  virginica  virginica  virginica  virginica  virginica
[121] virginica  virginica  virginica  virginica  virginica  virginica
[127] virginica  virginica  virginica  virginica  virginica  virginica
[133] virginica  virginica  virginica  virginica  virginica  virginica
[139] virginica  virginica  virginica  virginica  virginica  virginica
[145] virginica  virginica  virginica  virginica  virginica  virginica
Levels: setosa versicolor virginica
```

子集 [ ] 練習

type:alert incremental:true

- 取出iris資料集"Species"欄位的前十列(Row)
- 取出iris資料集所有單數列(Row)
- 取出iris資料集最後兩個欄位的所有偶數列(Row)

子集Subset - 二維資料 $

也可使用 $ 符號做Column**的篩選**

```
# 等同於iris[,"Species"]
iris$Species ##所有的列Row，名稱為Species的行Column
```

```
  [1] setosa     setosa     setosa     setosa     setosa     setosa
  [7] setosa     setosa     setosa     setosa     setosa     setosa
 [13] setosa     setosa     setosa     setosa     setosa     setosa
 [19] setosa     setosa     setosa     setosa     setosa     setosa
 [25] setosa     setosa     setosa     setosa     setosa     setosa
 [31] setosa     setosa     setosa     setosa     setosa     setosa
 [37] setosa     setosa     setosa     setosa     setosa     setosa
 [43] setosa     setosa     setosa     setosa     setosa     setosa
 [49] setosa     setosa     versicolor versicolor versicolor versicolor
 [55] versicolor versicolor versicolor versicolor versicolor versicolor
 [61] versicolor versicolor versicolor versicolor versicolor versicolor
 [67] versicolor versicolor versicolor versicolor versicolor versicolor
 [73] versicolor versicolor versicolor versicolor versicolor versicolor
 [79] versicolor versicolor versicolor versicolor versicolor versicolor
 [85] versicolor versicolor versicolor versicolor versicolor versicolor
 [91] versicolor versicolor versicolor versicolor versicolor versicolor
 [97] versicolor versicolor versicolor versicolor virginica  virginica
[103] virginica  virginica  virginica  virginica  virginica  virginica
[109] virginica  virginica  virginica  virginica  virginica  virginica
[115] virginica  virginica  virginica  virginica  virginica  virginica
[121] virginica  virginica  virginica  virginica  virginica  virginica
[127] virginica  virginica  virginica  virginica  virginica  virginica
[133] virginica  virginica  virginica  virginica  virginica  virginica
[139] virginica  virginica  virginica  virginica  virginica  virginica
[145] virginica  virginica  virginica  virginica  virginica  virginica
Levels: setosa versicolor virginica
```

子集Subset - 二維資料subset()

Row**的篩選**可使用 `subset()` 函數，使用方法為 `subset(資料表,篩選邏輯)`

```
subset(iris,Species=="virginica") ##Species等於"virginica"的列Row，所有的行Column
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica
104	6.3	2.9	5.6	1.8	virginica
105	6.5	3.0	5.8	2.2	virginica
106	7.6	3.0	6.6	2.1	virginica
107	4.9	2.5	4.5	1.7	virginica
108	7.3	2.9	6.3	1.8	virginica
109	6.7	2.5	5.8	1.8	virginica
110	7.2	3.6	6.1	2.5	virginica
111	6.5	3.2	5.1	2.0	virginica
112	6.4	2.7	5.3	1.9	virginica
113	6.8	3.0	5.5	2.1	virginica
114	5.7	2.5	5.0	2.0	virginica

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
115	5.8	2.8	5.1	2.4	virginica
116	6.4	3.2	5.3	2.3	virginica
117	6.5	3.0	5.5	1.8	virginica
118	7.7	3.8	6.7	2.2	virginica
119	7.7	2.6	6.9	2.3	virginica
120	6.0	2.2	5.0	1.5	virginica
121	6.9	3.2	5.7	2.3	virginica
122	5.6	2.8	4.9	2.0	virginica
123	7.7	2.8	6.7	2.0	virginica
124	6.3	2.7	4.9	1.8	virginica
125	6.7	3.3	5.7	2.1	virginica
126	7.2	3.2	6.0	1.8	virginica
127	6.2	2.8	4.8	1.8	virginica
128	6.1	3.0	4.9	1.8	virginica
129	6.4	2.8	5.6	2.1	virginica
130	7.2	3.0	5.8	1.6	virginica
131	7.4	2.8	6.1	1.9	virginica
132	7.9	3.8	6.4	2.0	virginica
133	6.4	2.8	5.6	2.2	virginica
134	6.3	2.8	5.1	1.5	virginica
135	6.1	2.6	5.6	1.4	virginica
136	7.7	3.0	6.1	2.3	virginica
137	6.3	3.4	5.6	2.4	virginica
138	6.4	3.1	5.5	1.8	virginica
139	6.0	3.0	4.8	1.8	virginica
140	6.9	3.1	5.4	2.1	virginica
141	6.7	3.1	5.6	2.4	virginica
142	6.9	3.1	5.1	2.3	virginica
143	5.8	2.7	5.1	1.9	virginica
144	6.8	3.2	5.9	2.3	virginica
145	6.7	3.3	5.7	2.5	virginica
146	6.7	3.0	5.2	2.3	virginica
147	6.3	2.5	5.0	1.9	virginica
148	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

子集Subset - 二維資料grepl()

Row**的篩選**也可搭配字串搜尋函數 `grepl()`

```
grepl("color",iris$Species)
iris[grepl("color",iris$Species),] ##Species包含"color"的列，所有的行
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
54	5.5	2.3	4.0	1.3	versicolor
55	6.5	2.8	4.6	1.5	versicolor
56	5.7	2.8	4.5	1.3	versicolor

子集Subset - head()

若想要快速取得資料框的前幾列(Row)或後幾列，也可使用 `head()` 和 `tail()` 函數

```
head(iris,5) ##取出iris資料框的前五列
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

子集Subset - tail()

若想要快速取得資料框的前幾列(Row)或後幾列，也可使用 `head()` 和 `tail()` 函數

```
tail(iris,3) ##取出iris資料框的後三列
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
148	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

子集綜合練習

type:alert incremental:true

- 取出內建資料集mtcars中，所有cyl小於等於6的車種資料
  - 方法一 subset()
  - 方法二 [ ]
- 取出內建資料集mtcars中，所有Toyota品牌的車種資料
  - 提示: rownames(), grepl()

    - []

資料組合

type:sub-section

有時需要在資料框新增一列，或新增一行

- Row 列的組合 `rbind()`
- Column 行的組合 `cbind()`

`rbind()` 和 `cbind()` 的參數可以是向量，也可以是資料框

資料組合 - rbind()

使用向量做資料整合範例:

```
rbind(c(1,2,3), #第一列
      c(4,5,6)  #第二列
      )
```

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
```

資料組合 - rbind()

使用資料框與向量做資料整合範例:

```
irisAdd<-rbind(iris, #資料框
      c(1,1,1,1,"versicolor")  #新增一列
      )
```

```
tail(irisAdd,2)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
150	5.9	3	5.1	1.8	virginica
151	1	1	1	1	versicolor

資料組合 - cbind()

使用向量做資料整合範例:

```
cbind(c(1,2,3), #第一行
      c(4,5,6)  #第二行
      )
```

```
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

資料組合 - cbind()

使用資料框與向量做資料整合範例:

```
irisAdd<-cbind(iris, #資料框
      rep("Add",nrow(iris))  #新增一行
      )


tail(irisAdd,1)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	rep("Add", nrow(iris))
150	5.9	3	5.1	1.8	virginica	Add

資料結合 (Join)

除了按照行列順序的組合外，更常有的情形是依照某個欄位的值作為結合依據，如：

- 用學號把以下兩個資料框結合成一個資料框
  - 學號與姓名資料框
  - 學號與宿舍床位資料框
- 用縣市名稱與年度將人口資料與醫療資源資料結合

資料結合 (Join)

原生的R環境可以用 `merge()` 函數將資料框結合，使用方法為 `merge(資料框1,資料框2,by="結合依據欄位")`

```
nameDF<-data.frame(ID=c(1,2,3,4,5),
                   Name=c("Amy","Bob","Chris","David","Emma"))
scoreDF<-data.frame(ID=c(1,2,4),
                   Score=c(60,90,50))
```

資料結合 (Join)

```
nameDF
```

ID	Name
1	Amy
2	Bob
3	Chris
4	David
5	Emma

```
scoreDF
```

ID	Score
1	60
2	90
4	50

資料結合 (Join) 更有效率的做法

`dplyr` 套件提供更有效率的資料結合方法，包括:

- inner_join()：保留有對應到的資料
- left_join()：保留左邊資料框的所有資料
- right_join()：保留右邊資料框的所有資料
- full_join()：保留所有資料
- semi_join()
- anti_join()

資料結合 - inner_join()

只保留兩張表都有的列 使用方法 `inner_join(x, y, by = )`

```
library(dplyr)
inner_join(nameDF,scoreDF,by="ID")
```

```
  ID  Name Score
1  1   Amy    60
2  2   Bob    90
3  4 David    50
```

資料結合 - left_join()

保留左邊的表所有的列 使用方法 `left_join(x, y, by = )`

```
library(dplyr)
left_join(nameDF,scoreDF,by="ID")
```

```
  ID  Name Score
1  1   Amy    60
2  2   Bob    90
3  3 Chris    NA
4  4 David    50
5  5  Emma    NA
```

資料結合 - right_join()

保留右邊的表所有的列 使用方法 `right_join(x, y, by = )`

```
library(dplyr)
right_join(nameDF,scoreDF,by="ID")
```

```
  ID  Name Score
1  1   Amy    60
2  2   Bob    90
3  4 David    50
```

資料結合 - full_join()

保留所有的列 使用方法 `full_join(x, y, by = )`

```
library(dplyr)
full_join(nameDF,scoreDF,by="ID")
```

```
  ID  Name Score
1  1   Amy    60
2  2   Bob    90
3  3 Chris    NA
4  4 David    50
5  5  Emma    NA
```

資料結合 - semi_join()

留下左邊的ID也有出現在右邊的表的列，右表資料不會輸出 使用方法 `semi_join(x, y, by = )`

```
library(dplyr)
semi_join(nameDF,scoreDF,by="ID")
```

```
  ID  Name
1  1   Amy
2  2   Bob
3  4 David
```

資料結合練習

type:alert

- 下載105各村里教育程度資料
- 下載10512各村（里）戶籍人口統計月報表
- 分別讀入兩個csv檔
- 依照區域別與村里名稱，將兩張表格結合，只留下有對應到的資料
- 請問結合後的資料有幾列？

遺漏值處理

type:sub-section

- 遺漏值(Missing Value)常常出現在真實資料內，在數值運算時常會有問題
- 最簡單的方法是將有缺值的資料移除

遺漏值處理 is.na()

如資料為向量，可使用 `is.na()` 來判斷資料是否為空值 `NA` ，若為真 `TRUE` ，則將資料移除。

```
naVec<-c("a","b",NA,"d","e")
is.na(naVec)
```

```
[1] FALSE FALSE  TRUE FALSE FALSE
```

```
naVec[!is.na(naVec)] ##保留所有在is.na()檢查回傳FALSE的元素
```

```
[1] "a" "b" "d" "e"
```

遺漏值處理 complete.cases()

若資料型態為資料框，可使用 `complete.cases` 來選出完整的資料列，如果資料列是完整的，則會回傳真TRUE

```
head(airquality,5)
```

Ozone	Solar.R	Wind	Temp	Month	Day
41	190	7.4	67	5	1
36	118	8.0	72	5	2
12	149	12.6	74	5	3
18	313	11.5	62	5	4
NA	NA	14.3	56	5	5

```
complete.cases(airquality)
```

```
  [1]  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
 [12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
 [23]  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
 [34] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE
 [45] FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE
 [56] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE
 [67]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
 [78]  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
 [89]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE
[100]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
[111]  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE
[122]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[133]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[144]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
```

遺漏值處理 complete.cases()

若資料型態為資料框，可使用 `complete.cases` 來選出完整的資料列，如果資料列(row)是完整的，則會回傳真TRUE

```
head(airquality[complete.cases(airquality),]) ##保留所有在complete.cases()檢查回傳TRUE的元素
```

	Ozone	Solar.R	Wind	Temp	Month	Day
1	41	190	7.4	67	5	1
2	36	118	8.0	72	5	2
3	12	149	12.6	74	5	3
4	18	313	11.5	62	5	4
7	23	299	8.6	65	5	7
8	19	99	13.8	59	5	8