Pipelines For Data Analysis In R, part 1

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**

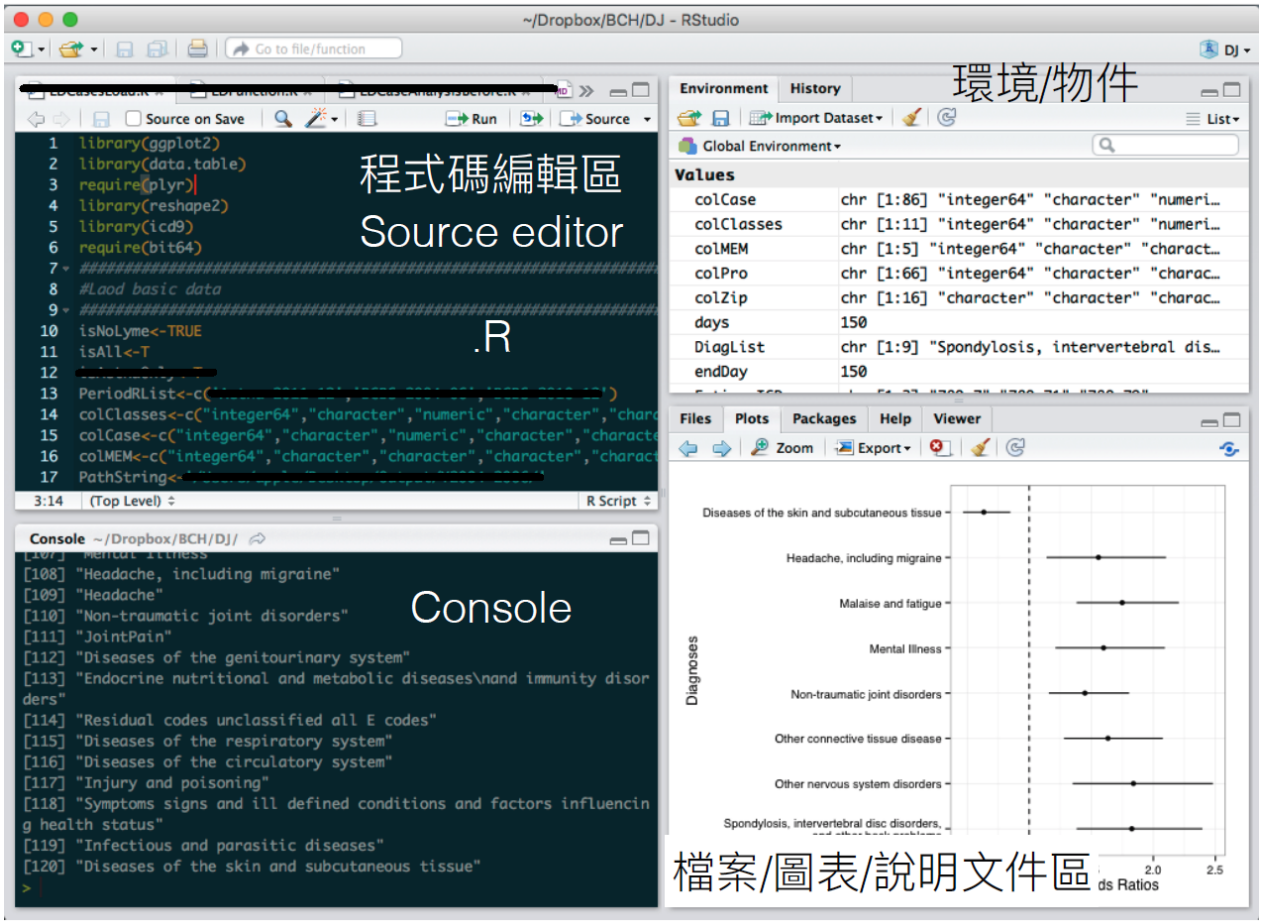
在分析之前

- RStudio怎麼用？
- R基本語法
- R資料結構

RStudio Interface

left: 30% 共有四個區塊，分別為：

- 程式碼編輯區 Source editor
- 執行視窗 Console
- 環境/物件
- 檔案/圖表/說明文件



RStudio 使用步驟

- New Project (如果原本沒有的話)
- New R Script (如果原本沒有的話)
- 在左上方**程式碼編輯區** Source editor撰寫程式碼
- 將需要執行的程式碼反白，點選Run，執行程式碼
- 將游標移至需要執行的程式碼，點選Run 也可執行該行程式碼
- 程式碼會在左下方Console視窗執行，顯示結果
- 如果有畫圖，會出現在右下方視窗
- 可在右上方視窗檢查所有變數

Console視窗

- 可直譯的語言
- 可在執行視窗(Console)直接打程式碼
- `>`：輸入指令
- `+`：表示前面的程式碼還沒打完
  - 鍵入完整的程式碼
  - Esc跳出

R = Base + Other Packages

- 安裝套件Package的方法如下：

```
install.packages("套件名稱")
```

- 套件名稱需要加上雙引號

```
install.packages("ggplot2")
```

- 載入**已安裝**的套件：

```
library(ggplot2)
```

- **不用**在套件名稱前後加雙引號

Functions - Basic

- 內建Functions函數
- 安裝Packages套件後各套件也會提供多種函數
- 使用方式: `函數名稱(參數1,參數2,....)`
- `?函數名稱` 查詢所需參數與說明 以計算平均數為例，可使用 `mean()` 函數:

```
mean(c(1,2,3,4,5,6)) ##計算1~6的平均數
```

```
[1] 3.5
```

Functions - Arguments 順序

- 函數的參數設定有順序性
- 不想照順序-->指定參數名稱

如序列產生函數 `seq()` ，參數順序為 `from, to, by` ，代表序列起點、序列終點，以及相隔單位。

```
seq(from=1,to=9,by=2)#1~9，每隔2產生一數字
```

```
[1] 1 3 5 7 9
```

Functions - Arguments 順序

```
seq(from=1,to=9,by=2)#1~9，每隔2產生一數字
```

```
[1] 1 3 5 7 9
```

```
seq(1,9,2)#按照順序輸入參數，可省去參數名稱
```

```
[1] 1 3 5 7 9
```

```
seq(by=2,to=9,from=1)#不照順序輸入，需要參數名稱
```

```
[1] 1 3 5 7 9
```

Variable [<-]

- 使用 `<-` 設定變數
- **變數名稱** `<-` **變數內容(值)**

- **變數名稱**可依箭頭方向放置於左側 <- 或右側 ->
- 但為方便閱讀，**變數名稱**多放置於左側

```
a<-1
a
```

```
[1] 1
```

```
2->b
b
```

```
[1] 2
```

Variable 命名規則

- 不可使用保留字
  - break, else, FALSE, for, function, if, Inf, NA, NaN, next, repeat, return, TRUE, while等
- 開頭只能是英文字，或 .
- 大小寫敏感

資料型態

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

數值 (numeric)

數值包括

- 整數（沒有小數點）
- 浮點數（有小數點）

```
num1<-100
num2<-1000.001
```

字串 (character)

- 用雙引號 " 框起：字串格式
- 數字前後加上雙引號：字串格式
  - 無法進行數值的加減乘除

```
char1<-"abcTest"
char2<-"100"
char3<-"200"
#char2+char3
#會輸出Error message: non-numeric argument to binary operator
```

布林變數 (logic)

- 用於邏輯判斷
- 大寫TRUE或T代表**真**

- 大寫FALSE或F代表假。

```
boolT<-TRUE
boolT1<-T
boolF<-FALSE
boolF1<-F
```

日期 (Date)

- 表示日期
- `Sys.Date()` 可得系統日期

```
dateBook<-Sys.Date()
dateBook
```

```
[1] "2017-05-04"
```

日期 (Date) - lubridate

- `lubridate` package：日期與字串的相關轉換操作
- `ymd()` 函數：將 `年/月/日` 格式的文字轉換為日期物件
  - y表年year
  - m表月month
  - d表日day

```
library(lubridate)
ymd('2012/3/3')
```

```
[1] "2012-03-03"
```

日期 (Date) - lubridate

- `mdy()` 函數：將 `月/日/年` 格式的文字轉換為日期物件
  - y表年year
  - m表月month
  - d表日day
- 以此類推

```
mdy('3/3/2012')
```

```
[1] "2012-03-03"
```

- 其他使用方法：The Yhat Blog

基本運算子: 數學運算

數學運算與其他程式語言相同

- 加 `+`
- 減 `-`

```
num1<-1
num2<-100
```

```
num1+num2
```

```
[1] 101
```

```
num1-num2
```

```
[1] -99
```

- 乘 *
- 除 /
- 餘數 %%
- 次方 ^

```
num1*num2
```

```
[1] 100
```

```
num1/num2
```

```
[1] 0.01
```

基本運算子: 邏輯運算

常用之邏輯判斷也可在R中直接使用

- 大於 >
- 小於 <

```
num1<-1
num2<-100
num1>num2
```

```
[1] FALSE
```

```
num1<num2
```

```
[1] TRUE
```

- 等於 == ，**雙等號**
- 大於等於 >=
- 小於等於 <=

```
num1==num2
```

```
[1] FALSE
```

```
1==1
```

```
[1] TRUE
```

基本運算子: 邏輯運算

文字字串也可比較大小

```
char1<-"abcTest"
char2<-"defTest"
char1>char2
```

```
[1] FALSE
```

基本運算子: 邏輯判斷

在R中使用**單符號**即可表示且 & 和或 |

- 且 &

```
TRUE & TRUE
```

```
[1] TRUE
```

```
TRUE & FALSE
```

```
[1] FALSE
```

在R中使用**單符號**即可表示且 & 和或 |

- 或 |

```
TRUE | TRUE
```

```
[1] TRUE
```

```
TRUE | FALSE
```

```
[1] TRUE
```

基本運算子: 反向布林變數 !

```
!TRUE
```

```
[1] FALSE
```

```
!FALSE
```

```
[1] TRUE
```

解讀錯誤訊息

- Message：有可能的錯誤通知，程式會繼續執行
- Warning：有錯誤，但是不會影響太多，程式會繼續執行
- Error：有錯，而且無法繼續執行程式
- Condition：可能會發生的情況

```
log(-1)
```

```
[1] NaN
```

```
mena(NA)
```

```
Error in eval(expr, envir, enclos): 沒有這個函數 "mena"
```

解讀錯誤訊息 範例

```
# Error: could not find function "fetch_NBAPlayerStatistics"
# 找不到"fetch_NBAPlayerStatistics" function
```

可能原因：沒安裝或沒讀入SportsAnalytics package

```
# Error in library(knitr): there is no package called 'knitr'
# 找不到"knitr" package
```

可能原因：沒安裝knitr package

Help

- R語言與套件均有完整的文件與範例可以參考
- 輸入 `?函數名稱` 或 `?套件名稱`

```
?ggplot2
?ymd
```

- Google
- Stack Overflow也有許多問答

R 常見的資料結構

- 向量 Vector
- 因子 Factor
- 列表 List
- 矩陣 Matrix
- 資料框 data.frame
- 屬性查詢函數

向量 Vector

type:sub-section

- 一維資料
- 所有元素之**資料型態必須相同**
- `c()` 函數 定義向量

```
vec<-c('a','b','c','d','e')
```

- a~e: **元素**(element)
- 順序固定
  - `a`：第1個元素
  - `b`：第2個元素

向量 Vector 取值

若要將 `vec` 向量的第4個元素取出，可使用**向量名稱[元素位置]**:

```
vec[4] ## 第4個元素
```

```
[1] "d"
```

也可同時取出多個元素

```
vec[c(2,3)] ## 第2與第3個元素
```

```
[1] "b" "c"
```

向量 Vector 元素設定

```
vec[3]
```

```
[1] "c"
```

```
vec[3]<-'z' ##第三個元素值設定為“z”
vec[3]
```

```
[1] "z"
```

產生向量函數

若要產生連續向量，如1~20，可使用：來串連首字與最後一字

```
1:20 ## c(1,2,...,19,20)
```

```
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

向量取值練習

type:alert incremental:true

- 新增一向量a，包含數字1到10
- 新增一向量b，包含數字1到20中的所有偶數
- 取出a向量的第4個元素
- 取出b向量的第5,6,7個元素

向量運算

向量也可直接做加減乘除運算，如

```
numvec<-1:10 ## c(1,2,3,4,5,6,7,8,9,10)
numvec+3 ## 所有元素+3
```

```
 [1]  4  5  6  7  8  9 10 11 12 13
```

```
numvec*2 ## 所有元素*2
```

```
 [1]  2  4  6  8 10 12 14 16 18 20
```

向量運算

向量和向量也可做運算，如

```
numvec1<-1:3 ## c(1,2,3)
numvec2<-4:6 ## c(4,5,6)
numvec1+numvec2
```

```
[1] 5 7 9
```

```
numvec1*numvec2
```

```
[1]  4 10 18
```

因子 factor

type:sub-section

- 由向量轉換而成
- 表示**類別**數據
- 使用方法為 `factor(資料向量,levels=類別次序)`

```
factor(c("大學生","碩士班學生","博士班學生"),
       levels = c("大學生","碩士班學生","博士班學生"))
```

```
[1] 大學生     碩士班學生 博士班學生
Levels: 大學生 碩士班學生 博士班學生
```

因子變量一但決定其類別的種類與數目時，通常不會再作更動，也就是任何新增的元素都要是大學生、碩士班學生與博士班學生其中一種。

列表 list

type:sub-section

- **向量**和**因子**都只能儲存一種元素
- 保有彈性: **列表**list
- 元素可分屬不同資料類別，
- **數值**、**文字**、**向量**、**因子**
- `list()` 函數

列表 list

```
listSample<-list(Students=c("Tom","Kobe","Emma"),
                 Year=2017,
                 Score=c(60,50,80,40),
                 School="CGU")
listSample
```

```
$Students
[1] "Tom"  "Kobe" "Emma"

$Year
[1] 2017

$Score
[1] 60 50 80 40

$School
[1] "CGU"
```

列表資料擷取

列表可用 `$` 符號做資料擷取

```
listSample$Students ##取得listSample列表中的Students變量
```

```
[1] "Tom"  "Kobe" "Emma"
```

列表資料擷取

- 若要取得**值**，要使用雙中括號 `[[]]`

```
listSample[[1]] ##第一個變量的值
```

```
[1] "Tom"  "Kobe" "Emma"
```

- 只使用單中括號 `[]` ，回傳的資料型態會是**列表**list

```
listSample[1] ##第一個變量（列表型態）
```

```
$Students
[1] "Tom"  "Kobe" "Emma"
```

列表list取值練習

type:alert incremental:true

- 依投影片，新增listSample列表
- 取出分數資料 (值)
  - 方法一
  - 方法二
- 取出分數資料 （列表）

列表資料編輯設定

列表資料和向量資料一樣，可重新編輯設定

```
listSample[[1]]
```

```
[1] "Tom"  "Kobe" "Emma"
```

```
listSample[[1]]<-c("小明","大雄","胖虎","小新","大白") ##將Students變量重新設定
listSample[[1]]
```

```
[1] "小明" "大雄" "胖虎" "小新" "大白"
```

列表資料編輯設定

列表資料也能用 $ 符號與 <- 變數設定符號新增

```
listSample$Gender<-c("M","F","M","F","M") ##新增Gender變量，並設定向量值
```

資料框 data.frame

type:sub-section

- 二維資料格式 (Excel試算表)
- 由欄位（Column）和列（Row）組成
- 使用 data.frame() 來創建新的資料框

```
StuDF <- data.frame(StuID=c(1,2,3), ##欄位名稱=欄位值
                    name=c("小明","大雄","胖虎"),
                    score=c(80,60,90))
StuDF
```

```
  StuID name score
1     1 小明    80
2     2 大雄    60
3     3 胖虎    90
```

資料框 data.frame

- 每列：觀察值 / 每欄：變數
- 欄位需有名稱
  - StuID, name, score
  - 若沒有設定，自動指派V1~Vn
- 欄位的資料型態相同
- 每一列有列名
  - 依序指派1~n作為列名

資料框 data.frame

- 檢查欄位名稱 `colnames()`
- 檢查欄位列名， `rownames()`

```
colnames(StuDF) #檢查欄位名稱
```

```
[1] "StuID" "name"  "score"
```

```
rownames(StuDF) #檢查列名
```

```
[1] "1" "2" "3"
```

```
nrow(StuDF) #幾列
```

```
[1] 3
```

```
ncol(StuDF) #幾欄
```

```
[1] 3
```

資料框 data.frame

如需檢查個欄位之資料型別，可使用 `str()` 函數

解釋iris資料框???

```
str(iris)
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

資料框資料擷取

資料框可用 `$` 符號做**欄位**資料擷取

```
iris$Species ##取得iris資料框中的Species欄位
```

```
  [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
```

資料框資料編輯

資料框可用 $ 符號做**欄位**資料擷取後，當成向量，並使用**[]**做資料編輯。

```
iris$Species[2]<-"versicolor"
head(iris$Species)
```

```
[1] setosa     versicolor setosa     setosa     setosa     setosa
Levels: setosa versicolor virginica
```

資料框資料編輯練習

type:alert incremental:true

- iris$Time<-1 會發生什麼事情?