成為初級資料分析師 I R 程式設計與資料科學應用

其他資料結構

郭耀仁

R's base data structures can be organized by their dimensionality and whether they're homogeneous or heterogeneous.

Hadley Wickham

大綱

- 其他資料結構一覽
- `list`
- `factor`
- `data.frame`
- `matrix` 與 `array`

其他資料結構一覽

除了向量以外的資料結構

- 必修
    - `list`
    - `factor`
    - `data.frame`
- 選修
    - `matrix`與`array`

其他資料結構的特性

- 皆是 ITERABLE 可迭代的
- list 用來儲存 KEY-VALUE 組
- factor 用來記錄文字向量的等級
- data.frame 用來儲存表格資料
- matrix 用來處理矩陣運算
- array 用來處理高維矩陣

list

`list` 在 R 中的定位

- 用來儲存不同類型的向量
- 未命名的 `list`（unnamed list）與 Python 的 `list` 呼應
- 命名的 `list`（named list）用來儲存 `KEY-VALUE` 組合的資料，與 Python 的 `dict` 呼應
- 支援 `$` 作為索引

使用 list() 函數創建 list

In [ ]:
```
endgame <- list(
  "Avengers: Endgame",
  2019,
  8.7,
  c("Action", "Adventure", "Sci-Fi")
)
class(endgame)
```

檢視外觀

In [ ]:
```
endgame
```

使用 [[INDEX]] 索引 list

In [ ]:
```
endgame[[1]]
endgame[[2]]
endgame[[3]]
endgame[[4]]
```

list 是可迭代的

In [ ]:
```
for (i in endgame) {
    print(i)
}
```

命名的 `list`（named list）

In [ ]:

```
endgame <- list(
  movieTitle = "Avengers: Endgame",
  releaseYear = 2019,
  rating = 8.7,
  genre = c("Action", "Adventure", "Sci-Fi")
)
class(endgame)
endgame
```

使用 [["KEY"]] 來索引 list

In [ ]:
```
endgame[["movieTitle"]]
endgame[["releaseYear"]]
endgame[["rating"]]
endgame[["genre"]]
```

亦可以使用 $KEY 來索引 list

In [ ]:
```
endgame$movieTitle
endgame$releaseYear
endgame$rating
endgame$genre
```

獲取 list 中的 KEYS

In [ ]:
```
names(endgame)
```

`list` 中的每個向量都維持本來的 class

In [ ]:
```
for (i in endgame) {
  print(class(i))
}
```

加入 list 中 KEY-VALUE

In [ ]:
```
endgame[["movieTime"]] <- 181
endgame
```

更新 list 中 VALUE

In [ ]:
```
endgame[["movieTime"]] <- "3h 1min"
endgame
```

删除 list 中 KEY-VALUE

In [ ]:
```
endgame[["movieTime"]] <- NULL
endgame
```

隨堂練習：將 5 個球員的姓氏（last name）擷取出來並轉換成大寫

In [1]:
```
fav_players <- c("Steve Nash", "Paul Pierce", "Dirk Nowitzki", "Kevin Garnett", "Hakeem Olajuwon")
# ?strsplit
# ?toupper
```

In [3]:
```
ans
```

'NASH' 'PIERCE' 'NOWITZKI' 'GARNETT' 'OLAJUWON'

factor

`factor` 在 R 中的定位

- 特殊的文字向量
- 獨一的文字值會以 **Levels** 紀錄
- 每個獨一的文字值會以一個整數編碼，支援有序文字
- 預設的文字變數類型

使用 factor() 函數創建

In [ ]:
```
avengers <- c("The Avengers", "Avengers: Age of Ultron", "Avengers: Infinity War",
"Avengers: Endgame")
class(avengers)
avengers <- factor(avengers)
class(avengers)
```

獨一的文字值以 Levels 編碼

In [ ]:
```
rgbs <- factor(c("red", "green", "blue", "blue", "green", "green"))
rgbs
```

factor 支援有序文字

In [ ]:
```
temperatures <- factor(c("freezing", "cold", "cool", "warm", "hot"), ordered = TRU
E)
temperatures
temperatures[1] > temperatures[3]
```

調整 factor 的順序

In [ ]:
```r
temperatures <- factor(c("freezing", "cold", "cool", "warm", "hot"),
                       ordered = TRUE,
                       levels = c("freezing", "cold", "cool", "warm", "hot"))
temperatures
```

每個獨一的文字值會以一個整數編碼

In [ ]:
```
temperatures <- c("freezing", "cold", "cool", "warm", "hot")
as.numeric(temperatures) # Error
temperatures <- factor(c("freezing", "cold", "cool", "warm", "hot"))
as.numeric(temperatures)
```

factor 有時難以掌控

In [ ]:
```
avengers <- factor(c("The Avengers", "Avengers: Age of Ultron", "Avengers: Infinit
y War"))
avengers <- c(avengers, "Avengers: Endgame")
avengers
```

R 為何使用 `factor` 作為預設的文字變數類型

- `factor` 具有整數的編碼
- 不需要做額外的 One-hot encoding
- 讓資料成為 `modeling-ready` 的狀態

使用建議

- 在資料處理階段使用 `character`
- 在資料預測階段使用 `factor`

data.frame

`data.frame` 在 R 中的定位

- 處理表格資料的首選
- 多數資料科學家處理的資料是表格形式
- 具有兩個維度，`m x n`（列 x 欄）
- 列常被稱為觀測值、欄常被稱為變數
- 每個欄都是一個向量，具有個別的 class
- 支援 `$` 作為取出單一欄的索引方式

使用 data.frame() 函數創建資料框

In [ ]:

```
avengers <- c("The Avengers", "Avengers: Age of Ultron", "Avengers: Infinity War",
"Avengers: Endgame")
ratings <- c(8.1, 7.3, 8.5, 8.7)
release_year <- c(2012, 2015, 2018, 2019)
is_good <- ratings > 8
avengers_df <- data.frame(title = avengers, rating = ratings, release_year, is_goo
d)
```

In [ ]:
```
avengers_df
```

文字向量預設以 factor 型態儲存

In [ ]:
```
#?str
str(avengers_df)
```

加入參數 stringsAsFactors = FALSE 可以調整為文字向量

In [ ]:
```
avengers_df <- data.frame(title = avengers, rating = ratings, release_year, is_good, stringsAsFactors = FALSE)
str(avengers_df)
```

常見用來觀察 data.frame 的函數

- View()：顯示漂亮的資料框外觀
- head(n)：顯示前 n 列
- tail(n)：顯示後 n 列
- summary()：顯示描述性統計
- str()：顯示結構
- dim()：顯示維度
- nrow()：顯示列數
- ncol()：顯示欄數

從資料框中選出欄位成為一個向量

In [ ]:
```
avengers_df[["title"]]
avengers_df[, "title"]
avengers_df[, 1]
```

或者使用 $

In [ ]:
```
avengers_df$title
```

篩選觀測值：指定列數

In [ ]:
```
avengers_df[c(1, 3, 4), ]
```

篩選觀測值：使用邏輯值向量

In [ ]:
```
avengers_df[c(FALSE, FALSE, TRUE, TRUE), ]
```

利用邏輯運算符產生邏輯值向量

In [ ]:
```
avengers_df$release_year >= 2018
avengers_df[avengers_df$release_year >= 2018, ] # putting logical vector as row in
dex
```

隨堂練習：將三巨頭 Michael Jordan, Scottie Pippen 還有 Dennis Rodman 選出來

In [4]:
```
csv_url <- "https://storage.googleapis.com/ds_data_import/chicago_bulls_1995_1996.csv"
chicago_bulls <- read.csv(csv_url)
```

In [6]:
```
ans
```

	No.	Player	Pos	Ht	Wt	Birth.Date	College
7	23	Michael Jordan	SG	6-6	195	February 17, 1963	University of North Carolina
11	33	Scottie Pippen	SF	6-8	210	September 25, 1965	University of Central Arkansas
12	91	Dennis Rodman	PF	6-7	210	May 13, 1961	Southeastern Oklahoma State University

matrix 與 array

使用 matrix() 函數創建矩陣

In [ ]:
```
my_mat <- matrix(1:4, nrow = 2)
class(my_mat)
```

簡單的矩陣運算符

- `*` 用來處理元素級別的乘法
- `t` 用來轉置
- `%*%` 用來處理矩陣相乘

$$AB_{i,j} = \sum A_{i,k} B_{k,j}$$

In [ ]:
```
my_mat <- matrix(1:4)
my_mat * my_mat
t(my_mat) %*% my_mat
```

隨堂練習：創建一個九九乘法矩陣

In [8]:
```
ans
```

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

使用 array() 函數創建 array

In [ ]:

```
my_arr <- array(1:24, dim = c(4, 3, 2))
my_arr
class(my_arr)
my_arr
```