

# 成為初級資料分析師 I R 程式設計與資料科學應用

流程控制: *while* 迴圈

郭耀仁

*When you've given the same in-person advice 3 times, write a blog post.*

*David Robinson*

# 大綱

- 邏輯值的應用場景
- `while` 迴圈

## 邏輯值的應用場景

## 邏輯值會出現在

- 條件判斷
- **while** 迴圈
- 資料篩選

**迴圈是用來解決需要反覆執行、大量手動複製貼上程式碼的任務**

將介於 1 至 100 的偶數印出

2

4

# ...

100

**while 迴圈**



## while 迴圈的 Code Block

- 保留字 `while`
- 起始值
- 終止值：被評估為 `logical` 的 `EXPR`
- 更新

```
1 i <- 1 # initial value
2 while (EXPR) {
3   # do something iteratively when EXPR is evaluated as TRUE
4   i <- i + 1 # update
5 }
```

```
In [ ]: i = 2
        while (i <= 100) {
            print(i)
            i <- i + 2
        }
```

## 常見的迴圈任務

- `print()`
- 加總 (Summation)
- 計數 (Counter)
- 合併 (Combine)：在後面章節討論

計算 1 到 100 之間的偶數總和：[The Story of Gauss](https://www.nctm.org/Publications/Teaching-Children-Mathematics/Blog/The-Story-of-Gauss/)  
(<https://www.nctm.org/Publications/Teaching-Children-Mathematics/Blog/The-Story-of-Gauss/>).

```
In [ ]: i = 2
        even_summation = 0

        while (i <= 100) {
            even_summation <- even_summation + i
            i <- i + 2
        }
        even_summation
```

計算 x 到 y 之間的偶數個數（包含 x 或 y 如果它們為偶數）

```
In [ ]: x = 55
        y = 66
        even_counter = 0

        while (x <= 66) {
            if (x %% 2 == 0) {
                even_counter <- even_counter + 1
            }
            x <- x + 1
        }
        even_counter
```

## 隨堂練習：判斷質數

在大於 1 的正整數中，除了 1 和該數自身外，無法被其他正整數整除的數字

```
In [1]: x <- 89
```

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

```
'89 是質數'
```

```
In [4]: x <- 56
```

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

'56 不是質數'



## 可搭配使用的保留字

- `break`
- `next`

**更快地判斷質數**

```
In [ ]: x <- 56
i <- 1
divisor_counter <- 0
while (i <= x**0.5) {
  print(sprintf("第 %s 次檢查因數", i))
  if (x %% i == 0) {
    divisor_counter <- divisor_counter + 1
  }
  i <- i + 1
}
if (x == 1) {
  ans <- sprintf("%s 不是質數", x)
} else if (divisor_counter == 1) {
  ans <- sprintf("%s 是質數", x)
} else {
  ans <- sprintf("%s 不是質數", x)
}
ans
```

In [ ]:

```
x <- 56
i <- 1
divisor_counter <- 0
while (i <= x**0.5) {
  print(sprintf("第 %s 次檢查因數", i))
  if (x %% i == 0) {
    divisor_counter <- divisor_counter + 1
  }
  i <- i + 1
  if (divisor_counter > 1) {
    break
  }
}
if (x == 1) {
  ans <- sprintf("%s 不是質數", x)
} else if (divisor_counter == 1) {
  ans <- sprintf("%s 是質數", x)
} else {
  ans <- sprintf("%s 不是質數", x)
}
ans
```

## 樓層的忌諱

```
In [ ]: i <- 1
        n_floors <- 10

        while (i <= n_floors) {
          if (i == 4) {
            i <- i + 1
            next
          }
          print(sprintf("%s 樓", i))
          i <- i + 1
        }
```

隨堂練習：判斷介於  $x$  與  $y$  之間的質數個數（包含  $x$  與  $y$  如果他們也是質數）

```
In [7]: x <- 1  
        y <- 5
```

```
In [9]: msg
```

'介於 1 與 5 之間的質數有 3 個'

```
In [10]: x <- 5  
        y <- 19
```

```
In [12]: msg
```

'介於 5 與 19 之間的質數有 6 個'