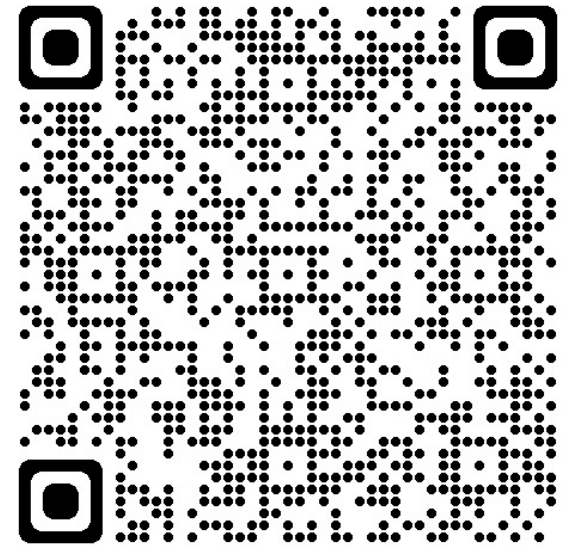


今天上課會.....

- 會講完 dplyr 與 tidyr
- 講完探索式資料分析
- 流程控制與函式基礎

- 請同學填寫課後問卷

(<https://forms.gle/cGyRNPmnRU2CzFuq5>)



R 語言 流程控制與函式

王貿

國立臺灣大學行為與資料科學研究中心助理研究員

國立臺灣大學政治學系博士、兼任講師

maowang01@gmail.com

課程主題重點

- For loops
- Functions



- 假設你有**很多**檔案要**讀取**，你會怎麼做？
- 要**爬取****很多**網頁資料，你會怎麼做？

迴圈 (Loop)

- 使用時機：處理重複模式的動作

```
for(variable in sequence) {  
    statement  
}
```



迴圈 (Loop)

- 具體化：處理重複模式的動作

iterator
variable

sequence

```
for( i in 1:10) {  
  
do something  
}
```

statement

iterator

```
for(x in 1:10){  
    print("Hello")  
}
```

```
for(x in 1:10){  
    print(x)  
}
```

variable

迴圈 (Loop)

- 具體化：處理重複模式的動作

index
variable **sequence**
for(i in 1:10) {

 do something
 statement
}

```
vec <- rep(NA, 10)
for(i in 1:10){
    vec[i] <- 2^(i)
}
```

index **variable**

預分配 (pre-allocation)

- 節省記憶體空間
- 提升迴圈的速度



- `vec <- NULL` (沒有使用預分配 , **不建議**)
- `vec <- rep(NA, 迴圈次數)`
- `lst <- vector(mode = "list", 迴圈次數)`

函式 (function)

- 一般函式：有命名
- 匿名函式

function name default
 argument

```
square <- function(x){  
    x ^ 2  
}
```

body of function

Does **x** exist
in your **global**
environment?