

# 原住民族資料分析線上訓練工作坊：R的基礎 與應用

第一週

---

**Kacing 廖彥傑**

英國艾賽克斯大學博士候選人

# 課程設計

## 我們課程主要包括：

- 作業環境、基礎R語言、資料結構與套件導入、R的社群生態
- 向量(vector)、序列(list)、矩陣(matrix)、資料框架(dataframe)
- 資料類型結構、Tidyverse 相關模組、函數式編程(functional programming)、循環
- 資料視覺化EDA (explore data analysis)、基礎統計分析與回歸、討論個人研究計畫
- 用R檢視地圖資料、製作互動式地圖、文本分析與應用、基礎機器學習

# 目的

## 為什麼我們要學習資料科學？

The New York Times

### ***Training the Next Generation of Indigenous Data Scientists***

A new workshop explores the right of Indigenous people to govern the collection, ownership and use of their biological and cultural data.

- 人工智能對政府部門運作、變革與影響
- 如果當政府部門開始用人工智能進行政策制定與評估時，會需要資料來建立模型。
- 原住民資料與代表性的問題？哪一天我們的政策會不會也被機器決定？

# 目的



# 目的

希望完成課程，大家都能夠：

- 基礎的資料分析知識
- 獨立用R完成分析
- 應用這些知識在自己的研究或工作計畫上
- 持續進修

開使吧！

# 基礎 (Some Basics)

## 列印

```
print("Hello World")
```

```
#> [1] "Hello World"
```

```
print(pi)
```

```
#> [1] 3.141593
```

```
print(sqrt(2))
```

```
#> [1] 1.414214
```

# 基礎 (Some Basics)

```
print(matrix(c(1, 2, 3, 4), 2, 2))
```

```
#>      [,1] [,2]  
#> [1,]    1    3  
#> [2,]    2    4
```

```
print(list("a", "b", "c"))
```

```
#> [[1]]  
#> [1] "a"  
#>  
#> [[2]]  
#> [1] "b"  
#>  
#> [[3]]  
#> [1] "c"
```

```
print("The zero occurs at", 2 * pi, "radians.")
```

```
num <- readline(prompt="有多少人: ")  
cat("\n", "原住民族資料分析線上訓練工作坊：R的基礎與應用", "\n", "總共有", num, "\n")
```



# 基礎 (Some Basics)

## 建立變數

```
x <- 3
```

```
x <- 3  
y <- 4  
z <- sqrt(x + y)
```

```
print(z)
```

```
#> [1] 2.645751
```

```
x <- c("Lbak", "Uking", "是", "Truku")
```

```
print(x)
```

```
#> [1] "Lbak" "Uking" "是" "Truku"
```

# 基礎 (Some Basics)

## 列出變數

```
x <- 10  
y <- 50  
z <- c("Kacing", "David", "Liao")
```

```
ls()
```

```
#> [1] "black"      "blue"       "blue_green" "brown"  
#> [5] "green"     "grey_dark"  "grey_light" "grey_mid"  
#> [9] "magenta_green" "magenta_red" "magenta_yellow" "orange"  
#> [13] "purple"    "red"        "red_green"  "red_pink"  
#> [17] "turquoise" "x"          "y"          "z"
```

# 基礎 (Some Basics)

```
ls.str()
```

```
#> black : chr "#000000"  
#> blue : chr "#3b3b9a"  
#> blue_green : chr "#4d599b"  
#> brown : chr "#9b684d"  
#> green : chr "#8bb174"  
#> grey_dark : chr "grey20"  
#> grey_light : chr "grey70"  
#> grey_mid : chr "grey50"  
#> magenta_green : chr "#4d9b68"  
#> magenta_red : chr "#9b4d80"  
#> magenta_yellow : chr "#9b8f4d"  
#> orange : chr "#FFA500"  
#> purple : chr "#6A5ACD"  
#> red : chr "#fb6107"  
#> red_green : chr "#9b4d59"  
#> red_pink : chr "#e64173"  
#> turquoise : chr "#20B2AA"  
#> x : num 10  
#> y : num 50  
#> z : chr [1:3] "Kacing" "David" "Liao"
```

# 基礎 (Some Basics)

## 刪除變數

```
david <- "David is hot "  
rm(x)
```

```
rm(david, y, z)
```

```
ls()
```

```
#> [1] "black"      "blue"       "blue_green" "brown"  
#> [5] "green"      "grey_dark"  "grey_light" "grey_mid"  
#> [9] "magenta_green" "magenta_red" "magenta_yellow" "orange"  
#> [13] "purple"     "red"        "red_green"  "red_pink"  
#> [17] "turquoise"
```

```
rm(list = ls())  
ls()
```

```
#> character(0)
```

# 基礎 (Some Basics)

## 製造簡單向量 (Vector)

```
c(1, 1, 2, 3, 5, 8, 13, 21)
```

```
#> [1] 1 1 2 3 5 8 13 21
```

```
c("好想", "吃", "momo", "paradise")
```

```
#> [1] "好想"      "吃"        "momo"      "paradise"
```

```
c(TRUE, TRUE, FALSE, TRUE)
```

```
#> [1] TRUE TRUE FALSE TRUE
```

# 基礎 (Some Basics)

## 製造簡單序列 (Sequences)

1:100

```
#>  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
#> [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
#> [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
#> [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
#> [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
#> [91] 91 92 93 94 95 96 97 98 99 100
```

100:1

```
#>  [1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83
#> [19] 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65
#> [37] 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47
#> [55] 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29
#> [73] 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11
#> [91] 10  9  8  7  6  5  4  3  2  1
```

# 基礎 (Some Basics)

```
seq(from=1,to=100)
```

```
#>  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
#> [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
#> [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
#> [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
#> [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
#> [91] 91 92 93 94 95 96 97 98 99 100
```

```
seq(from=1,to=100,by=2)
```

```
#>  [1]  1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
#> [26] 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
```

# 基礎 (Some Basics)

```
seq(from = 0, to = 100, length.out = 5)
```

```
#> [1] 0 25 50 75 100
```

```
rep(1, times = 5)
```

```
#> [1] 1 1 1 1 1
```



# 比較(Comparison)

## 語法

	定義
==	等於
!=	不等於
<	小於
>	大於
<=	小於等於
>=	大於等於

# 比較(Comparison)

## 條件比較

```
a <- 10  
b <- 11  
a == b
```

```
#> [1] FALSE
```

```
a != b
```

```
#> [1] TRUE
```

```
a > b
```

```
#> [1] FALSE
```

```
a < b
```

```
#> [1] TRUE
```

```
a >= b
```

```
#> [1] FALSE
```

# 比較(Comparison)

```
v <- c(3, "david", 4)
w <- c("david", "david", 1)
```

```
v == w
```

```
#> [1] FALSE TRUE FALSE
```

```
v != w
```

```
#> [1] TRUE FALSE TRUE
```

```
v < w
```

```
#> [1] TRUE FALSE FALSE
```

```
v <= w
```

```
#> [1] TRUE TRUE FALSE
```

# 比較(Comparison)

```
v > w
```

```
#> [1] FALSE FALSE TRUE
```

```
v >= w
```

```
#> [1] FALSE TRUE TRUE
```

```
v <- c(3, 3, 4)  
v == 4
```

```
#> [1] FALSE FALSE TRUE
```

```
v != 4
```

```
#> [1] TRUE TRUE FALSE
```

# 比較(Comparison)

```
a <- 1:10
```

```
a > 5
```

```
#> [1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
```

```
a < 5
```

```
#> [1] TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
a == 5
```

```
#> [1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE
```

# 比較(Comparison)

```
all(a > 5)
```

```
#> [1] FALSE
```

```
any(a > 5)
```

```
#> [1] TRUE
```

```
b <- a > 5
```

```
any(b)
```

```
#> [1] TRUE
```

```
all(b)
```

```
#> [1] FALSE
```

# 比較(Comparison)

```
c <- c(a, NA)
c > 5
```

```
#> [1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE NA
```

```
all(c > 5)
```

```
#> [1] FALSE
```

```
any(c > 5)
```

```
#> [1] TRUE
```

```
all(c < 20)
```

```
#> [1] NA
```

```
any(c > 20)
```

```
#> [1] NA
```

# 比較(Comparison)

```
is.na(a)
```

```
#> [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
is.na(c)
```

```
#> [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
```

```
any(is.na(c))
```

```
#> [1] TRUE
```

```
all(is.na(c))
```

```
#> [1] FALSE
```



# 比較(Comparison)

## 提取資料 (Selecting Vector Elements)

```
truku <- 1:10  
truku
```

```
#> [1] 1 2 3 4 5 6 7 8 9 10
```

```
truku[1]
```

```
#> [1] 1
```

```
truku[2]
```

```
#> [1] 2
```

# 比較(Comparison)

```
truku[1:3]
```

```
#> [1] 1 2 3
```

```
truku[c(2, 5, 10)]
```

```
#> [1] 2 5 10
```

# 比較(Comparison)

```
truku[-1] # Ignore first element
```

```
#> [1] 2 3 4 5 6 7 8 9 10
```

```
truku[1:3] # As before
```

```
#> [1] 1 2 3
```

```
truku[-(1:3)] # Invert sign of index to exclude instead of select #> [1] 2 3 5 8 13 21 34
```

```
#> [1] 4 5 6 7 8 9 10
```

```
num <- truku < 5 # This vector is TRUE wherever fib is less than 10
```

```
truku[num]
```

```
#> [1] 1 2 3 4
```

# 比較(Comparison)

```
v <- c(3,6,1,9,11,16,0,3,1,45,2,8,9,6,-4)
v[ v > median(v)]
```

```
#> [1]  9 11 16 45  8  9
```

```
v[ (v < quantile(v, 0.05)) | (v > quantile(v, 0.95)) ]
```

```
#> [1] 45 -4
```

```
v[ abs(v - mean(v)) > sd(v)]
```

```
#> [1] 45 -4
```

```
v<-c(1,2,3,NA,5)
v[!is.na(v) & !is.null(v)]
```

```
#> [1] 1 2 3 5
```

# 比較(Comparison)

```
years <- c(1986, 1964, 1976, 1994)
names(years) <- c("Kennedy", "Johnson", "Carter", "Clinton")
years
```

```
#> Kennedy Johnson Carter Clinton
#>    1986    1964    1976    1994
```

```
years["Carter"]
```

```
#> Carter
#>    1976
```

```
years[1]
```

```
#> Kennedy
#>    1986
```

```
years[c("Carter", "Clinton")]
```

```
#> Carter Clinton
#>    1976    1994
```

# 算數 (Arithmetic)

## 基本加減乘除

```
v <- c(11, 12, 13, 14, 15)
w <- c(1,2,3,4,5)
```

```
v + w
```

```
#> [1] 12 14 16 18 20
```

```
v * w
```

```
#> [1] 11 24 39 56 75
```

```
v / w
```

```
#> [1] 11.000000 6.000000 4.333333 3.500000 3.000000
```

# 算數 (Arithmetic)

```
w
```

```
#> [1] 1 2 3 4 5
```

```
w + 2
```

```
#> [1] 3 4 5 6 7
```

```
w - 2
```

```
#> [1] -1 0 1 2 3
```

```
w * 2
```

```
#> [1] 2 4 6 8 10
```

```
w / 2
```

```
#> [1] 0.5 1.0 1.5 2.0 2.5
```

# 預告

Table 2-1. Operator precedence

Operator	Meaning	See also
[ [[	Indexing	<a href="#">Recipe 2.9</a>
:: :::	Access variables in a namespace (environment)	
\$ @	Component extraction, slot extraction	
^	Exponentiation (right to left)	
- +	Unary minus and plus	
:	Sequence creation	<a href="#">Recipe 2.7</a> , <a href="#">Recipe 7.13</a>
%any% (including %>%)	Special operators	Discussion (this recipe)
* /	Multiplication, division	Discussion (this recipe)
+ -	Addition, subtraction	
== != < > <= >=	Comparison	<a href="#">Recipe 2.8</a>
!	Logical negation	
& &&	Logical “and,” short-circuit “and”	
	Logical “or,” short-circuit “or”	
~	Formula	<a href="#">Recipe 11.1</a>
-> ->>	Rightward assignment	<a href="#">Recipe 2.2</a>
=	Assignment (right to left)	<a href="#">Recipe 2.2</a>
<- <<-	Assignment (right to left)	<a href="#">Recipe 2.2</a>
?	Help	<a href="#">Recipe 1.8</a>

參考書目：R Cookbook Proven Recipes for Data Analysis, Statistics, and Graphics by J. D. Long, Paul Teetor