

Chapter 3

COLLECTION OF DATA

集結變數

將變數做有效的整理

資料型態	Example
一維	向量 (vector)
	因素向量 (factor vector)
二維	矩陣 (Matrix)
	資料框 (data.frame)
三維	陣列 (Array)
	清單 (list)

向量 (Vector)

c() 函數可將變數集結在一個向量中，並可用 [] 存取某一個索引值

Note: R 語言的索引值從 1 開始

```
1 GSW <- c("Curry", "Thompson", "Durant", "Green", "McGee")
2
3 print(GSW)
4 print(GSW[1])
5 |
```

```
> GSW <- c("Curry", "Thompson", "Durant", "Green", "McGee")
>
> print(GSW)
[1] "Curry"      "Thompson"    "Durant"      "Green"       "McGee"
> print(GSW[1])
[1] "Curry"
> |
```

向量 (Vector)

`c()` 函數可將變數集結在一個向量中，並可用 `[]` 存取某一個索引值

Note: 可以使用負的索引值來刪除資料

```
1 GSW <- c("Curry", "Thompson", "Durant", "Green", "McGee")
2 print(GSW)
3
4 GSW <- GSW[c(-2, -5)]
5 print(GSW)
```

```
Console Terminal x
~/
> GSW <- c("Curry", "Thompson", "Durant", "Green", "McGee")
> print(GSW)
[1] "Curry"    "Thompson" "Durant"    "Green"    "McGee"
>
> GSW <- GSW[c(-2, -5)]
> print(GSW)
[1] "Curry"    "Durant"    "Green"
>
```

向量 (Vector)

`c()` 函數可將變數集結在一個向量中，並可用 `[]` 存取某一個索引值

Note: 將整數，數值，文字都放入向量中，資料形態會變成文字

```
1 GSW <- c("Curry", 30, "Durant", 35L)
2 print(GSW)
3
4 print(class(GSW))
```

```
Console Terminal x
~/
> GSW <- c("Curry", 30, "Durant", 35L)
> print(GSW)
[1] "Curry" "30"      "Durant" "35"
>
> print(class(GSW))
[1] "character"
>
```

向量 (Vector)

可使用邏輯 & (and), | (or) 或判斷運算子 ==, != 來取出向量的資料

```
1 GSW <- c("Curry", 30, "Thompson", 11, "Durant", 35L)
2 print(GSW)
3
4 SC <- GSW == "Curry"
5 print(SC)
6 print(GSW[SC])
7
8 splash <- GSW == "Curry" | GSW == "Thompson"
9 print(GSW[splash])
```

```
Console Terminal x
~/
> GSW <- c("Curry", 30, "Thompson", 11, "Durant", 35L)
> print(GSW)
[1] "Curry"      "30"          "Thompson"    "11"          "Durant"
[6] "35"
>
> SC <- GSW == "Curry"
> print(SC)
[1] TRUE FALSE FALSE FALSE FALSE FALSE
> print(GSW[SC])
[1] "Curry"
>
> splash <- GSW == "Curry" | GSW == "Thompson"
> print(GSW[splash])
[1] "Curry"      "Thompson"
> |
```

向量 (Vector)

rep() 函數可以產生重複變數的向量

Note: 可以使用 times 參數來指定要重複幾次

```
1  
2 a <- rep(7, times = 11)  
3 print(a)  
4  
5 b <- rep("7-11", times = 50)  
6 print(b)
```

```
> a <- rep(7, times = 11)  
> print(a)  
[1] 7 7 7 7 7 7 7 7 7 7 7  
>  
> b <- rep("7-11", times = 50)  
> print(b)  
[1] "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11"  
[9] "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11"  
[17] "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11"  
[25] "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11"  
[33] "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11"  
[41] "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11" "7-11"  
[49] "7-11" "7-11"  
>
```

向量 (Vector)

seq() 函數可以產生等差級數

Note: from 起點 / to 終點 / by 間距

```
1 a <- seq(from = 3, to = 76, by = 5)
2 print(a)
3
4 b <- seq(from = 1, to = 100, by = 3)
5 print(b)
6
7 c <- 1:30
8 print(c)
```

```
> a <- seq(from = 3, to = 76, by = 5)
> print(a)
[1] 3 8 13 18 23 28 33 38 43 48 53 58 63 68 73
>
> b <- seq(from = 1, to = 100, by = 3)
> print(b)
[1] 1 4 7 10 13 16 19 22 25 28 31 34 37 40 43
[16] 46 49 52 55 58 61 64 67 70 73 76 79 82 85 88
[31] 91 94 97 100
>
> c <- 1:30
> print(c)
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
[21] 21 22 23 24 25 26 27 28 29 30
>
```


因素向量 (factor Vector)

`factor()` 為帶有階層 `levels` 資訊的向量, (適合幫文字排順序)

Note: 若沒指定 `levels`, 則會依據字母順序排序

```
1 player <- c("Curry","Jordan","Kobe","Iverson","LBJ")
2
3 rank <- factor(player, ordered=TRUE,
4   levels = c("LBJ","Kobe","Jordan","Iverson","Curry") )
5
6 print(rank)
7
8 rank2 <- factor(player, ordered=TRUE)
9 print(rank2)
```

```
Console Terminal x
~/
> player <- c("Curry","Jordan","Kobe","Iverson","LBJ")
>
> rank <- factor(player, ordered=TRUE,
+   levels = c("LBJ","Kobe","Jordan","Iverson","Curry") )
>
> print(rank)
[1] Curry   Jordan  Kobe    Iverson LBJ
Levels: LBJ < Kobe < Jordan < Iverson < Curry
>
> rank2 <- factor(player, ordered=TRUE)
> print(rank2)
[1] Curry   Jordan  Kobe    Iverson LBJ
Levels: Curry < Iverson < Jordan < Kobe < LBJ
> |
```

隨堂練習 1

12 星座分別為

白羊，金牛，雙子，巨蟹，獅子，處女，天秤，天蠍，射手，魔羯，水瓶，雙魚

1. 建立一個 12 星座的向量

```
consts <- c("白羊", "金牛", "雙子",  
            "巨蟹", "獅子", "處女",  
            "天秤", "天蠍", "射手",  
            "魔羯", "水瓶", "雙魚")
```

2. 刪掉雙魚座以及天蠍座並輸出剩餘的星座

```
> print(consts)
[1] "白羊" "金牛" "雙子" "巨蟹" "獅子" "處女" "天秤" "射手"
[9] "魔羯" "水瓶"
>
```

隨堂練習 2

牛排的熟度分別是

Blue Rare < Rare < Medium Rare < Medium < Medium Well < Well Done

(微煎 < 一分 < 三分 < 五分 < 七分 < 全熟

1. 建立一個牛排熟度的因素向量

2. 輸出如下所示

```
[1] BR          Rare          Med_R          Medium          Med          Well Done
Levels: BR < Med < Med_R < Medium < Rare < Well Done
> |
```

矩陣 (Matrix)

矩陣可儲存列 (水平) 和欄 (垂直) 的一種資料結構

Ex. 使用 `matrix()` 產生一個矩陣，設定列數 = 3，並將數字 1~9 依序填入

```
1 marion <- matrix(1:9, nrow=3)
2 print(marion)
3 |
```

Console	Terminal x
~/	
> marion <- matrix(1:9, nrow=3)	
> print(marion)	
	[,1] [,2] [,3]
[1,]	1 4 7
[2,]	2 5 8
[3,]	3 6 9
>	

矩陣 (Matrix)

矩陣可使用 `[]` 搭配索引值選出變數值

Ex. 取出第 1 列第 3 行的值，取出第 2 列的值以及取出第 1 行的值

```
1 marion <- matrix(1:9, nrow=3)
2 print(marion)
3
4 a <- marion[1,3]
5 print(a)
6
7 b <- marion[2,]
8 print(b)
9
10 c <- marion[,3]
11 print(c)
12
13
```

```
Console Terminal x
~/
> marion <- matrix(1:9, nrow=3)
> print(marion)
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> a = marion[1,3]
> print(a)
[1] 7
>
> b = marion[2,]
> print(b)
[1] 2 5 8
>
> c = marion[,3]
> print(c)
[1] 7 8 9
>
```

矩陣 (Matrix)

矩陣可使用搭配判斷運算子篩選資料

Ex. 取出 3,4,5,6,7 五個數字

```
1 marion <- matrix(1:9, nrow=3)
2 print(marion)
3
4 get <- marion >2 & marion <8
5 new_marion = marion[get]
6
7 print(new_marion)
8
```

```
Console Terminal x
~/
> marion <- matrix(1:9, nrow=3)
> print(marion)
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
>
> get <- marion >2 & marion <8
> new_marion = marion[get]
>
> print(new_marion)
[1] 3 4 5 6 7
> |
```

矩陣 (Matrix)

矩陣可只會有一種資料型態

Ex. 把布林, 整數跟數值都放進去矩陣, 會發生什麼事?

```
1 marion <- matrix(c(1,2,3,4L,5L,TRUE), nrow=2)
2 print(marion)
3 print(class(marion[,3]))
4
5 marion2 <- matrix(c(1L,2L,3L,4L,5L,TRUE), nrow=2)
6 print(marion2)
7 print(class(marion2[,3]))
8
```

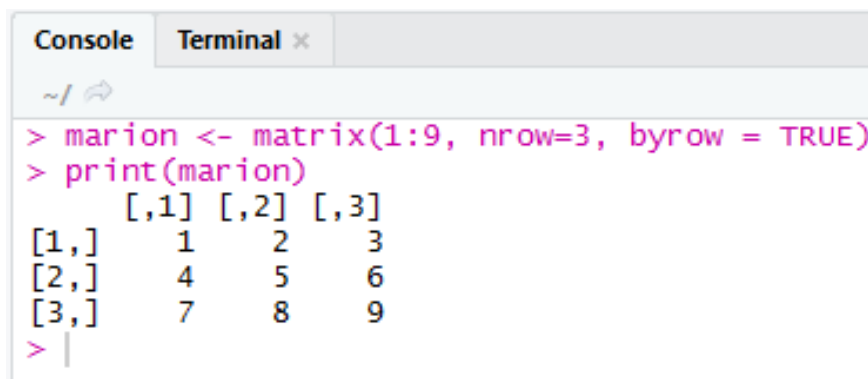
```
Console Terminal x
~/
> marion <- matrix(c(1,2,3,4L,5L,TRUE), nrow=2)
> print(marion)
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    1
> print(class(marion[,3]))
[1] "numeric"
>
> marion2 <- matrix(c(1L,2L,3L,4L,5L,TRUE), nrow=2)
> print(marion2)
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    1
> print(class(marion2[,3]))
[1] "integer"
>
```

矩陣 (Matrix)

矩陣放資料的方式跟其他程式語言不一樣怎麼辦

Ex. 使用 `byrow = TRUE` 來變更排列方式

```
1 marion <- matrix(1:9, nrow=3, byrow = TRUE)
2 print(marion)
3 |
```



The screenshot shows a R console window with two tabs: 'Console' and 'Terminal x'. The 'Console' tab is active. The prompt is '~ / ↩'. The user has entered the following commands:

```
> marion <- matrix(1:9, nrow=3, byrow = TRUE)
> print(marion)
```

The output of the `print(marion)` command is displayed as follows:

	[,1]	[,2]	[,3]
[1,]	1	2	3
[2,]	4	5	6
[3,]	7	8	9

The prompt is now `> |`.

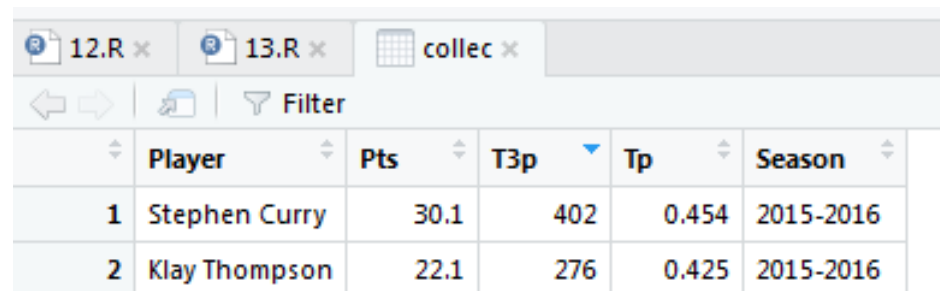
資料框 (data.frame)

資料框在 R 語言是非常重要的資料結構，可讓不同欄位有不同的資料型態

之後學到讀取外部資料 (ex, excel) 時，都會用 data.frame 來儲存資料

```
1 Player <- c("Stephen Curry", "Klay Thompson")
2 Pts <- c(30.1, 22.1)
3 T3p <- c(402L, 276)
4 Tp <- c(0.454, 0.425)
5 Season <- c("2015-2016", "2015-2016")
6
7 collec <- data.frame(Player, Pts, T3p, Tp, Season)
8 view(collec)
```

```
Console Terminal x
~/
> Player <- c("Stephen Curry", "Klay Thompson")
> Pts <- c(30.1, 22.1)
> T3p <- c(402L, 276)
> Tp <- c(0.454, 0.425)
> Season <- c("2015-2016", "2015-2016")
>
> collec <- data.frame(Player, Pts, T3p, Tp, Season)
> view(collec)
> |
```



	Player	Pts	T3p	Tp	Season
1	Stephen Curry	30.1	402	0.454	2015-2016
2	Klay Thompson	22.1	276	0.425	2015-2016

資料框 (data.frame)

但資料框有一個問題，預設將文字都儲存成 factor vector，怎麼處理？

```
9 |  
10 name = collec[,1]  
11 print(name)  
12 print(class(name))  
13  
14
```

```
> name = collec[,1]  
> print(name)  
[1] Stephen Curry Klay Thompson  
Levels: Klay Thompson Stephen Curry  
> print(class(name))  
[1] "factor"  
> |
```

資料框 (data.frame)

Solution 1: 設定 stringsAsFactors = FALSE

```
1 Player <- c("Stephen Curry", "Klay Thompson")
2 Pts <- c(30.1, 22.1)
3 T3p <- c(402L, 276)
4 Tp <- c(0.454, 0.425)
5 Season <- c("2015-2016", "2015-2016")
6
7 collec <- data.frame(Player, Pts, T3p, Tp, Season, stringsAsFactors = FALSE)
8 View(collec)
9
10 name = collec[,1]
11 print(name)
12 print(class(name))
13
14
```

Console	Terminal x
~/	
> Player <- c("Stephen Curry", "Klay Thompson")	
> Pts <- c(30.1, 22.1)	
> T3p <- c(402L, 276)	
> Tp <- c(0.454, 0.425)	
> Season <- c("2015-2016", "2015-2016")	
>	
> collec <- data.frame(Player, Pts, T3p, Tp, Season, stringsAsFactors = FALSE)	
> View(collec)	
>	
> name = collec[,1]	
> print(name)	
[1] "Stephen Curry" "Klay Thompson"	
> print(class(name))	
[1] "character"	
>	

資料框 (data.frame)

Solution 2: 利用 `as.character()` 進行轉換

```
1 Player <- c("Stephen Curry", "Klay Thompson")
2 Pts <- c(30.1, 22.1)
3 T3p <- c(40.2, 27.6)
4 Tp <- c(0.454, 0.425)
5 Season <- c("2015-2016", "2015-2016")
6
7 collec <- data.frame(Player, Pts, T3p, Tp, Season)
8 View(collec)
9
10 collec[,1] <- as.character(collec[,1])
11
12 a <- collec[,1]
13 print(a)
14 print(class(a))
15
```

```
> collec[,1] <- as.character(collec[,1])
>
> a <- collec[,1]
> print(a)
[1] "Stephen Curry" "Klay Thompson"
> print(class(a))
[1] "character"
>
```

資料框 (data.frame)

資料框支援利用變數名稱來存取資料

Note: 可以使用 \$ 變數名稱 或 [, “變數名稱”]

```
1 Player <- c("Stephen Curry", "Klay Thompson")
2 Pts <- c(30.1, 22.1)
3 T3p <- c(40.2, 27.6)
4 Tp <- c(0.454, 0.425)
5 Season <- c("2015-2016", "2015-2016")
6
7 collec <- data.frame(Player, Pts, T3p, Tp, Season,
8                       stringsAsFactors = FALSE)
9
10 name1 = collec$Player
11 name2 = collec[, "T3p"]
12 print(name1)
13 print(name2)
14 |
15
```

```
> name1 = collec$Player
> name2 = collec[, "T3p"]
> print(name1)
[1] "Stephen Curry" "Klay Thompson"
> print(name2)
[1] 40.2 27.6
> |
```

資料框 (data.frame)

資料框也可使用搭配判斷運算子篩選資料

Ex. 找出 >400 顆 3 分球的是誰

```
1 Player <- c("Stephen Curry", "Klay Thompson")
2 Pts    <- c(30.1, 22.1)
3 T3p    <- c(402L, 276)
4 Tp     <- c(0.454, 0.425)
5 Season <- c("2015-2016", "2015-2016")
6
7 collec <- data.frame(Player, Pts, T3p, Tp, Season,
8                       stringsAsFactors = FALSE)
9
10 filter <- collec$T3p >= 400
11 a = collec[filter,]
12 print(a$Player)
13
14
```

```
Console Terminal x
~/
> Player <- c("Stephen Curry", "Klay Thompson")
> Pts    <- c(30.1, 22.1)
> T3p    <- c(402L, 276)
> Tp     <- c(0.454, 0.425)
> Season <- c("2015-2016", "2015-2016")
>
> collec <- data.frame(Player, Pts, T3p, Tp, Season,
+                       stringsAsFactors = FALSE)
>
> filter <- collec$T3p >= 400
> a = collec[filter,]
> print(a$Player)
[1] "Stephen Curry"
>
```

資料框 (data.frame)

最後用 `str()` 函數 來觀察所有變數的資料型態

```
1 Player <- c("Stephen Curry", "Klay Thompson")
2 Pts    <- c(30.1, 22.1)
3 T3p    <- c(402L, 276)
4 Tp     <- c(0.454, 0.425)
5 Season <- c("2015-2016", "2015-2016")
6
7 collec <- data.frame(Player, Pts, T3p, Tp, Season,
8                       stringsAsFactors = FALSE)
9
10 str(collec)
11
12
```

```
> str(collec)
'data.frame':  2 obs. of  5 variables:
 $ Player: chr  "Stephen Curry" "Klay Thompson"
 $ Pts   : num  30.1 22.1
 $ T3p   : num  402 276
 $ Tp    : num  0.454 0.425
 $ Season: chr  "2015-2016" "2015-2016"
> |
```

隨堂練習 3

1. 如先前的例子產生一個 3*3 的矩陣

```
> print(icy)
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

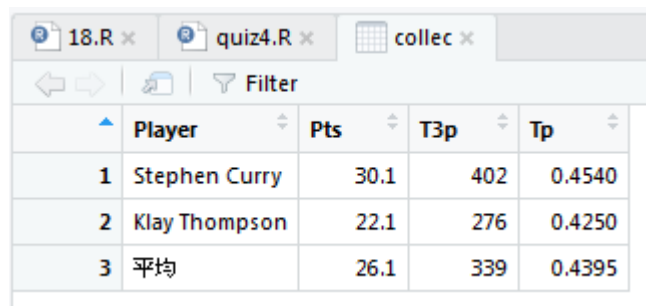
2. 輸出矩陣中的奇數

```
> print(icy)
[1] 2 4 6 8
> |
```


隨堂練習 4

接續先前資料框的例子，請算出 2015-2016 年 Curry 和 Thompson

1. 得分的平均
2. 三分球投進的平均
3. 三分球命中率的平均
4. 將資料框更新成下圖所示



The screenshot shows an RStudio interface with three tabs: '18.R', 'quiz4.R', and 'collec'. The 'collec' tab is active, displaying a data frame with the following columns: Player, Pts, T3p, and Tp. The data is as follows:

	Player	Pts	T3p	Tp
1	Stephen Curry	30.1	402	0.4540
2	Klay Thompson	22.1	276	0.4250
3	平均	26.1	339	0.4395

陣列 (Array)

陣列是矩陣的加強版，也就是我們可以建立二維以上的多維資料

Ex. 建立一個三維陣列（也就是三個二維陣列）

```
1 arrays <- array(1:18, dim= c(2,3,3))
2 print(arrays)
```

```
> arrays <- array(1:18, dim= c(2,3,3))
> print(arrays)
, , 1

      [,1] [,2] [,3]
[1,]     1     3     5
[2,]     2     4     6

, , 2

      [,1] [,2] [,3]
[1,]     7     9    11
[2,]     8    10    12

, , 3

      [,1] [,2] [,3]
[1,]    13    15    17
[2,]    14    16    18

>
```

陣列 (Array)

陣列是矩陣的加強版，可利用陣列建立二維以上的多維資料

Ex. 存取不同的資料（第三個維度代表第幾個陣列）

```
1 arrays <- array(1:18, dim= c(2,3,3))
2 print(arrays)
3
4 print(arrays[1,2,3])
5
6 print(arrays[1, ,1])
7
8 print(arrays[ ,2,2])
```

```
> print(arrays[1,2,3])
[1] 15
>
> print(arrays[1, ,1])
[1] 1 3 5
>
> print(arrays[ ,2,2])
[1] 9 10
```

Console Terminal x

```
> print(arrays)
, , 1
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

, , 2
      [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12

, , 3
      [,1] [,2] [,3]
[1,]   13   15   17
[2,]   14   16   18
```

清單 (List)

清單是一個無底洞，可以把所以物件都放進去

```
1 nickname <- "Splash Brothers"
2 Player <- c("Stephen Curry", "Klay Thompson")
3 Pts <- c(30.1, 22.1)
4 T3p <- c(402L, 276)
5 Tp <- c(0.454, 0.425)
6
7 collec <- data.frame(Player, Pts, T3p, Tp,
8                       stringsAsFactors = FALSE)
9
10 big_list <- list(nickname, collec)
11 print(big_list)
12
```

```
Console Terminal x
~/
> nickname <- "Splash Brothers"
> Player <- c("Stephen Curry", "Klay Thompson")
> Pts <- c(30.1, 22.1)
> T3p <- c(402L, 276)
> Tp <- c(0.454, 0.425)
>
> collec <- data.frame(Player, Pts, T3p, Tp,
+                       stringsAsFactors = FALSE)
>
> big_list <- list(nickname, collec)
> print(big_list)

[[1]]
[1] "Splash Brothers"

[[2]]
      Player  Pts T3p  Tp
1 Stephen Curry 30.1 402 0.454
2 Klay Thompson 22.1 276 0.425
```

清單 (List)

存取清單元素，利用 `[[]]` 二層中括號

```
1 nickname <- "Splash Brothers"
2 Player <- c("Stephen Curry", "Klay Thompson")
3 Pts <- c(30.1, 22.1)
4 T3p <- c(402, 276)
5 Tp <- c(0.454, 0.425)
6
7 collec <- data.frame(Player, Pts, T3p, Tp,
8                       stringsAsFactors = FALSE)
9
10 big_list <- list(nickname, collec)
11 print(big_list)
12
13 big_list[[2]]
14 big_list[[2]][1]
15 big_list[[2]][1,]
16 big_list[[2]][,2]
```

```
> print(big_list)
[[1]]
[1] "Splash Brothers"

[[2]]
      Player  Pts T3p   Tp
1 Stephen Curry 30.1 402 0.454
2 Klay Thompson 22.1 276 0.425

>
> big_list[[2]]
      Player  Pts T3p   Tp
1 Stephen Curry 30.1 402 0.454
2 Klay Thompson 22.1 276 0.425
> big_list[[2]][1]
      Player
1 Stephen Curry
2 Klay Thompson
> big_list[[2]][1,]
      Player  Pts T3p   Tp
1 Stephen Curry 30.1 402 0.454
> big_list[[2]][,2]
[1] 30.1 22.1
```

隨堂練習 5

1. 將 1-250 放入 10 個 $5*5$ 的陣列
2. 找出 125 這個數值在那一個索引

Any Questions !?