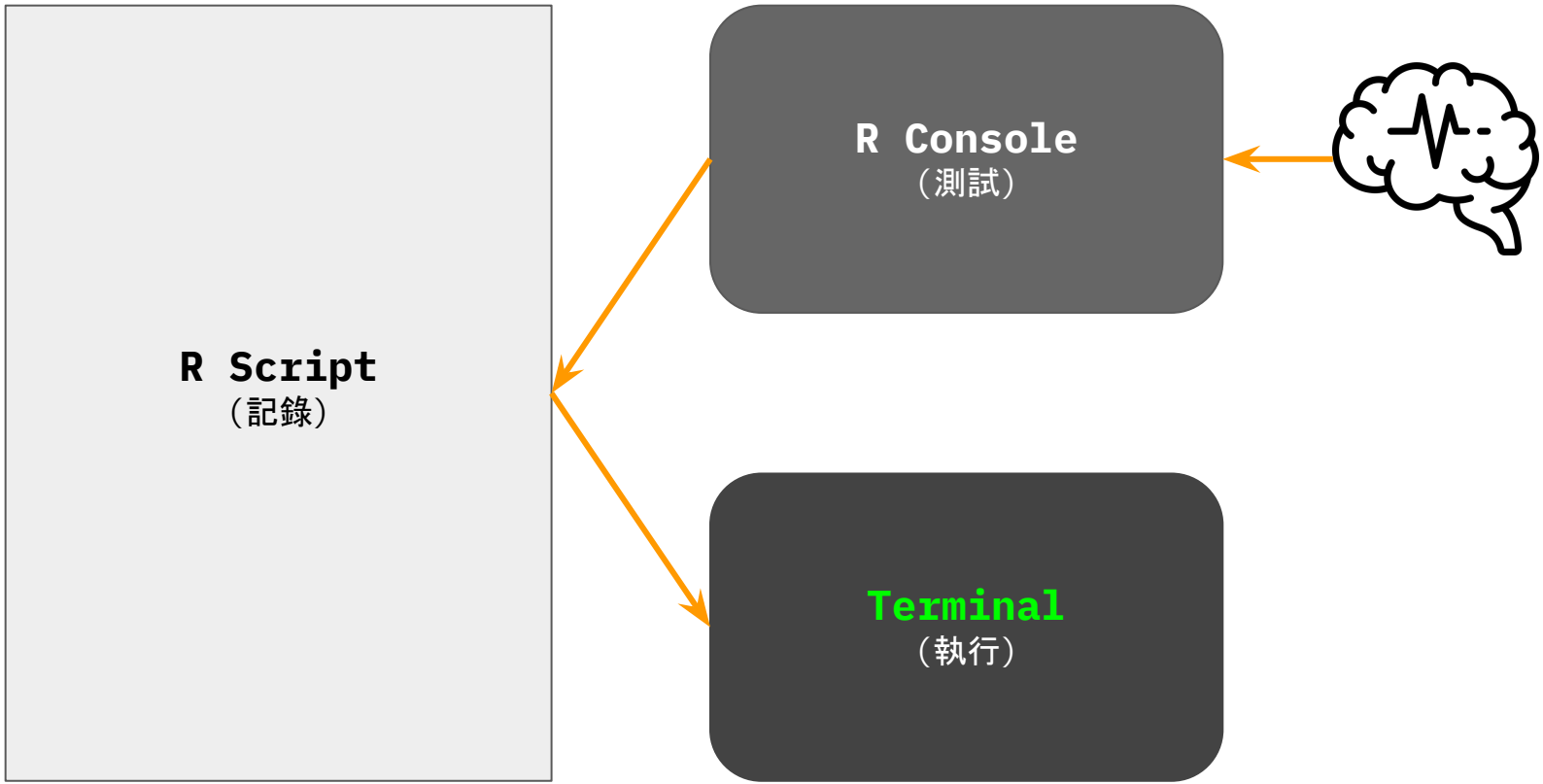
Lab 02

Base R (I)

函數, 向量, 資料類型, 條件式

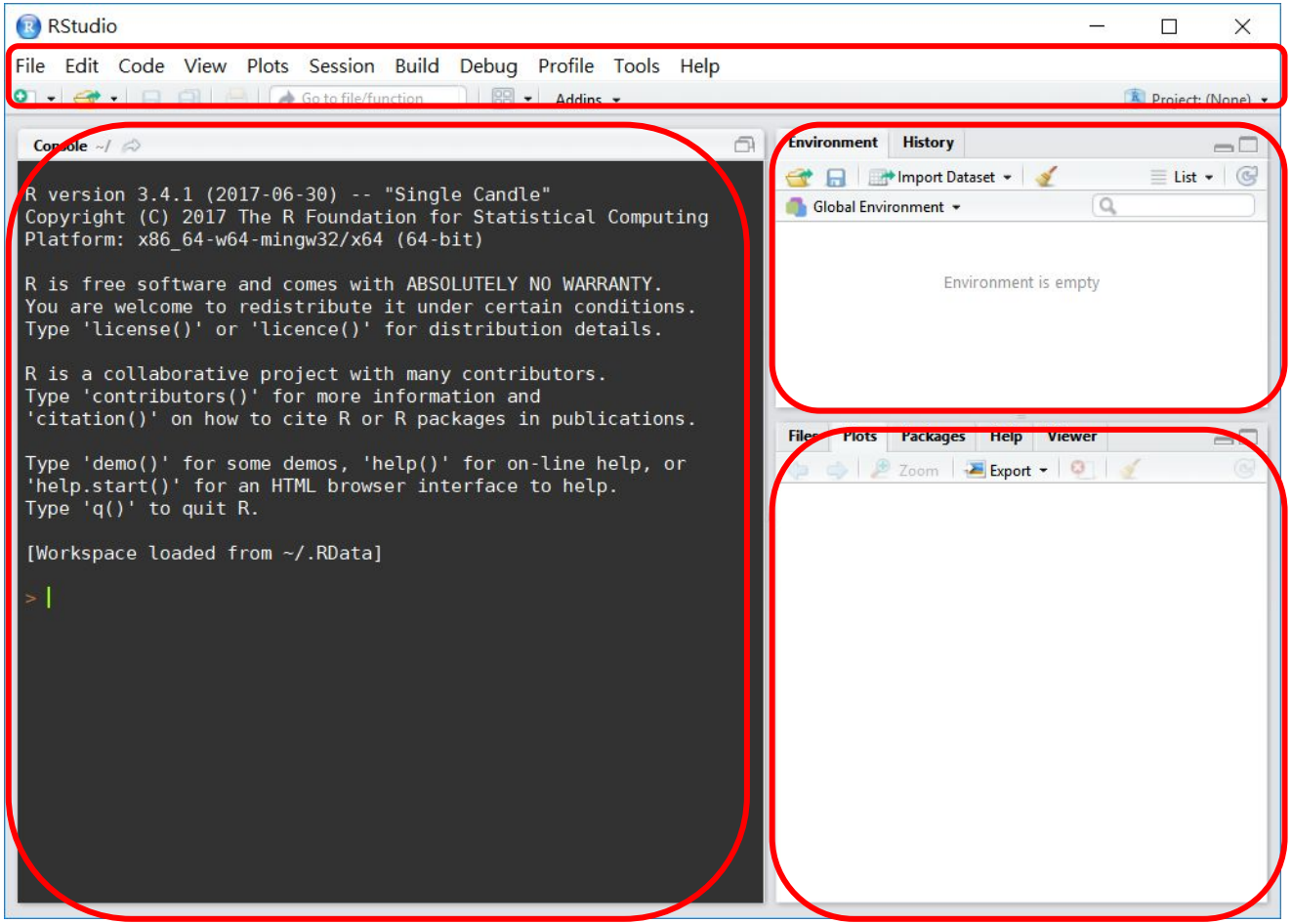
工具：RStudio, R Markdown

撰寫 R 必備的 3 樣工具



RStudio

選單功能



Console
(執行 R)

Source
(R Script)

Environment
History

Files,
Plots,
Packages,
Help,
Viewer

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Console ~/

```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> 
```

Environment History

Import Dataset

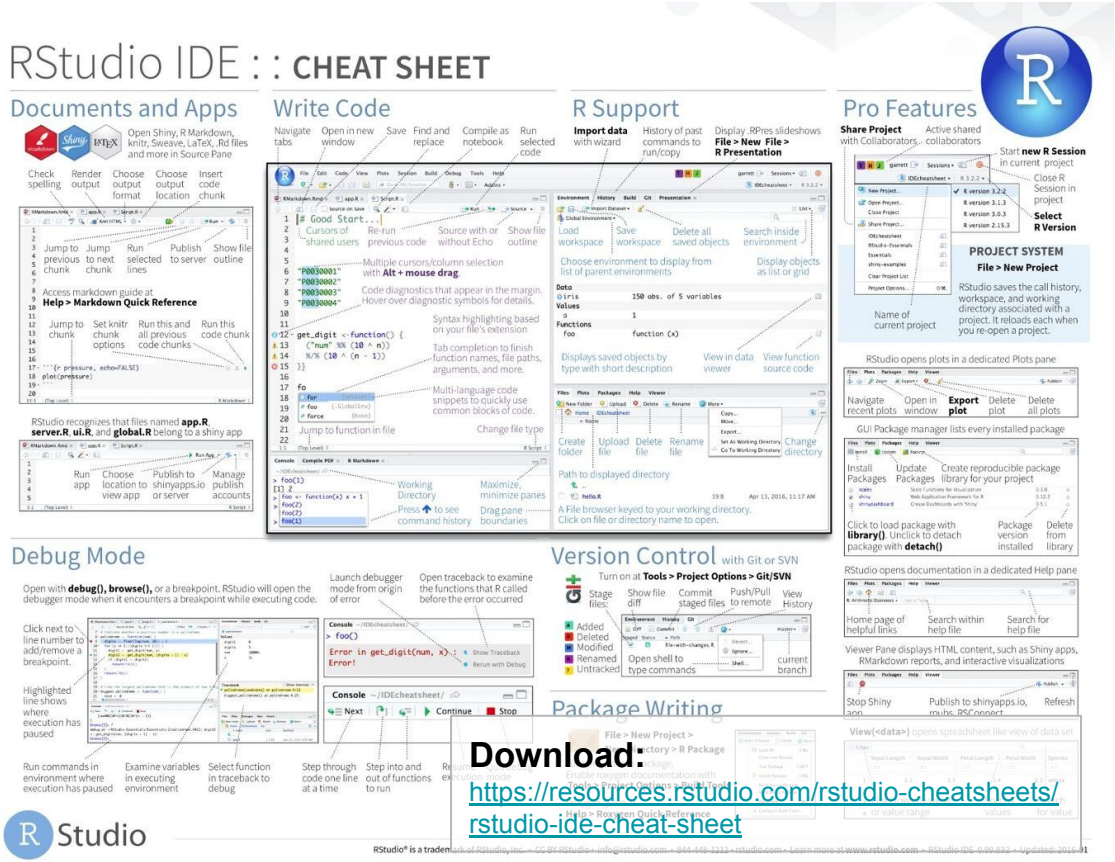
Global Environment

Environment is

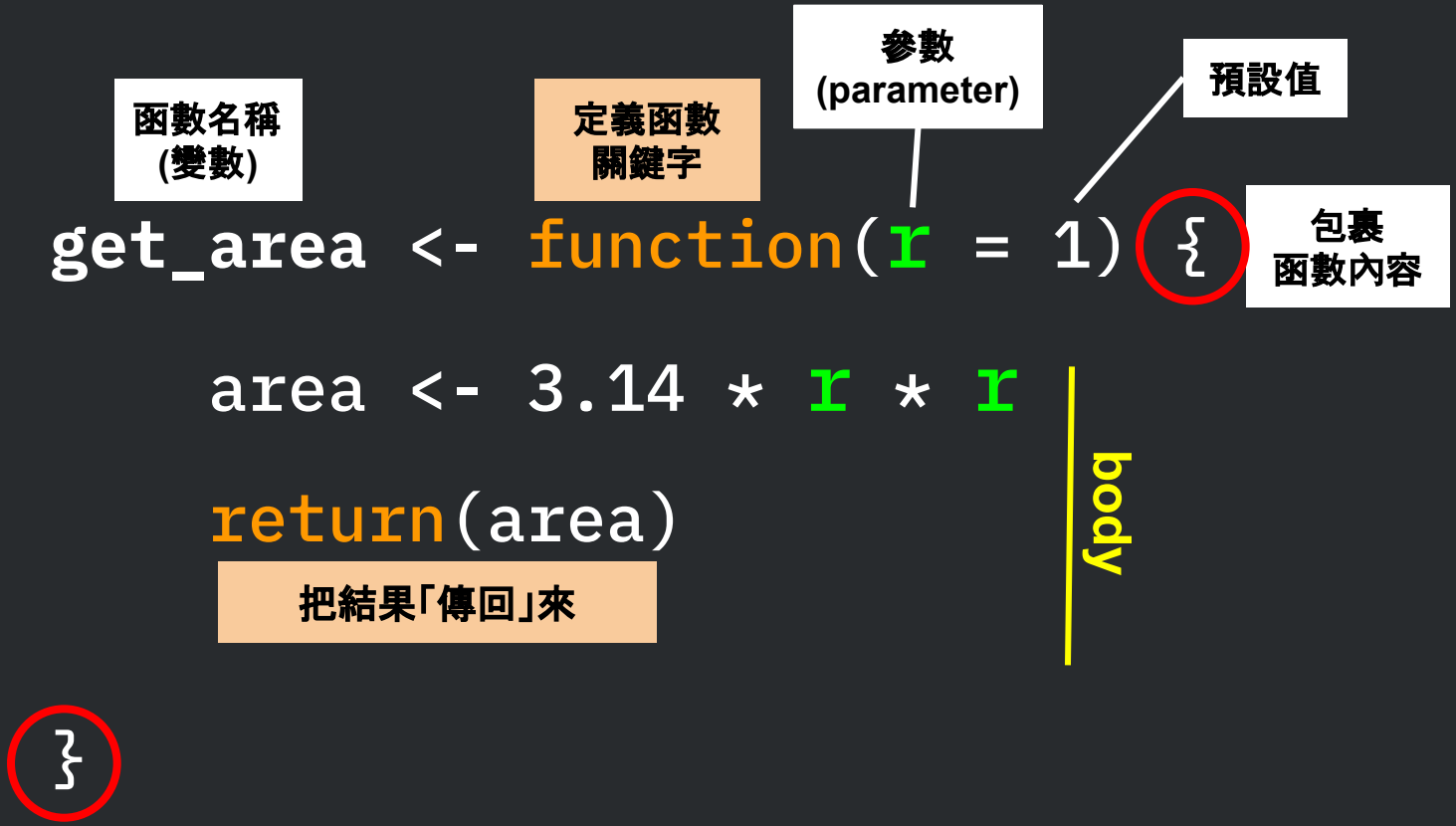
Files Plots Packages Help

Zoom | Export

RStudio Cheat Sheet



自訂函數
(Function)


函數名稱
(變數)
定義函數
關鍵字
參數
(parameter)
預設值
包裹
函數內容
get_area <- function(r = 1) {
area <- 3.14 * r * r
return(area)
把結果「傳回」來
body
}

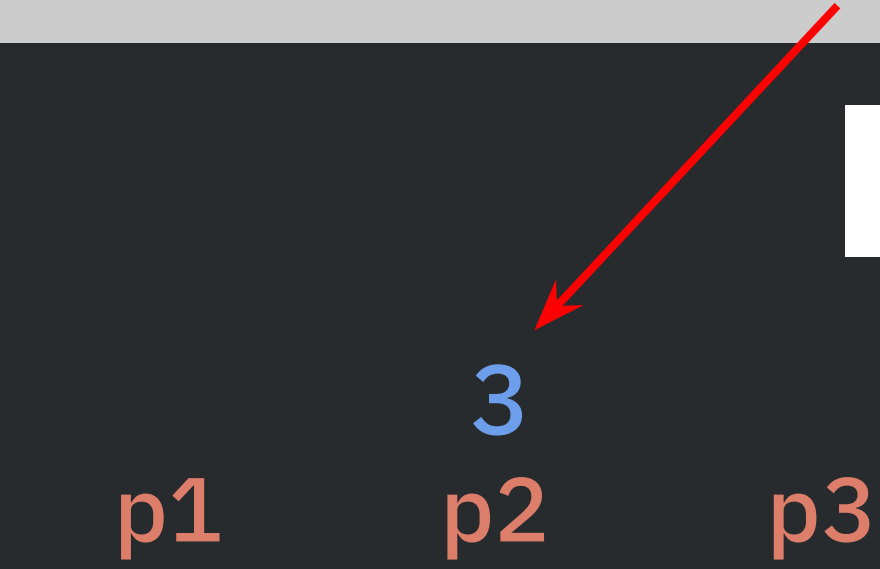
Function Arguments

```
fun <- function(p1, p2, p3) {...}
```

```
fun(1, 2, p2 = 3)
```

1. 先指派**有名字的** (Named) Arguments
2. 再**依序**指派剩下的 Arguments

```
fun(1, 2, p2 = 3)
```



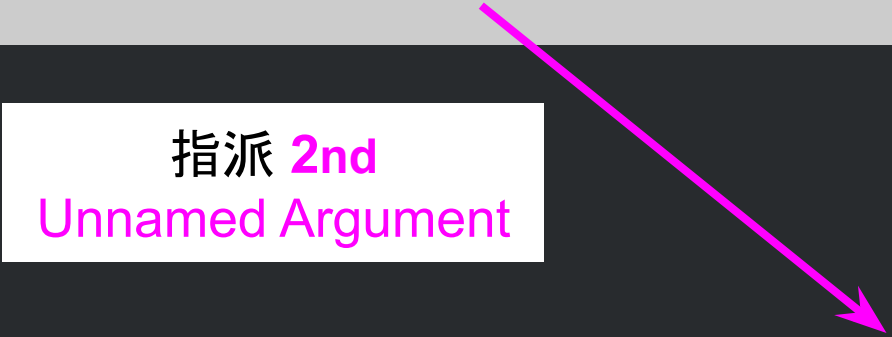
```
fun(1, 2, p2 = 3)
```

指派 **1st**
Unnamed Argument

```
1       3
p1      p2      p3
```

```
fun(1, 2, p2 = 3)
```

指派 **2nd**
Unnamed Argument

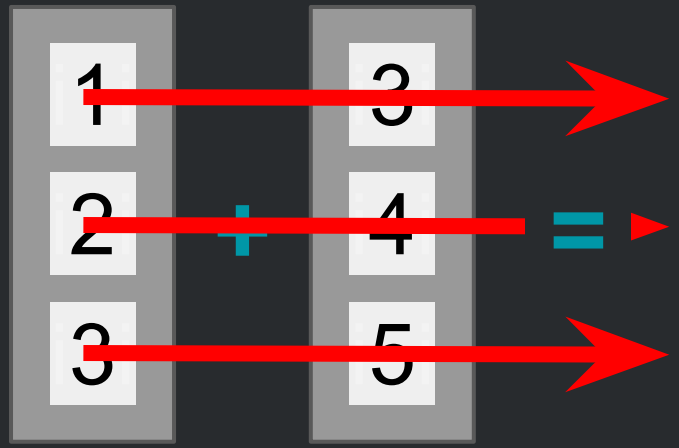


1	3	2
p1	p2	p3

向量
(Vector)

Vector

- Basic units in R
- `length()`
- 4 種 (最常見的) 類型
  - Integer
  - Double
  - Logical
  - Character



Element-wise operation

資料類型
(Data Types)

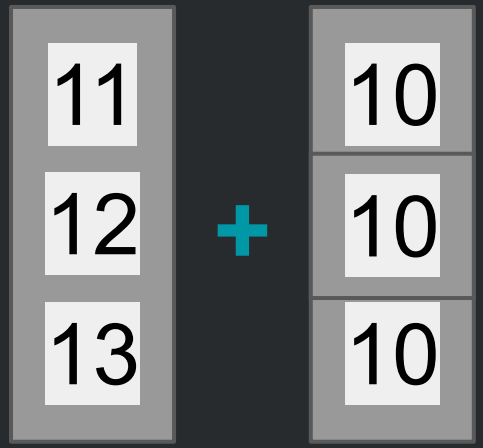
Logical Operators

Operator	語法	意義
`>`	`a > b`	a 大於 b？
`>=`	`a >= b`	a 大於或等於 b？
`<`	`a < b`	a 小於 b？
`<=`	`a <= b`	a 小於或等於 b？
`==`	`a == b`	a 等於 b？
`!=`	`a != b`	a 不等於 b？
`%in%`	`a %in% c(a, b, c)`	a 屬於 c(a, b, c)？

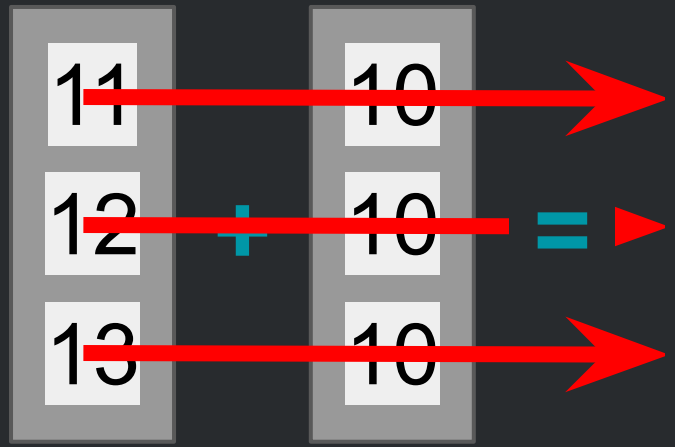
Boolean Operators

Operator	語法	意義
`&`	`cond1 & cond2`	且 (and)
`|`	`cond1 | cond2`	或 (or)
`!`	`!cond1`	非 (~)
`any()`	`any(cond1, cond2, cond3, ...)`	至少一個為真？
`all()`	`all(cond1, cond2, cond3, ...)`	全為真？
`xor()`	`xor(cond1, cond2)`	cond1 與 cond2 其中一個為真？

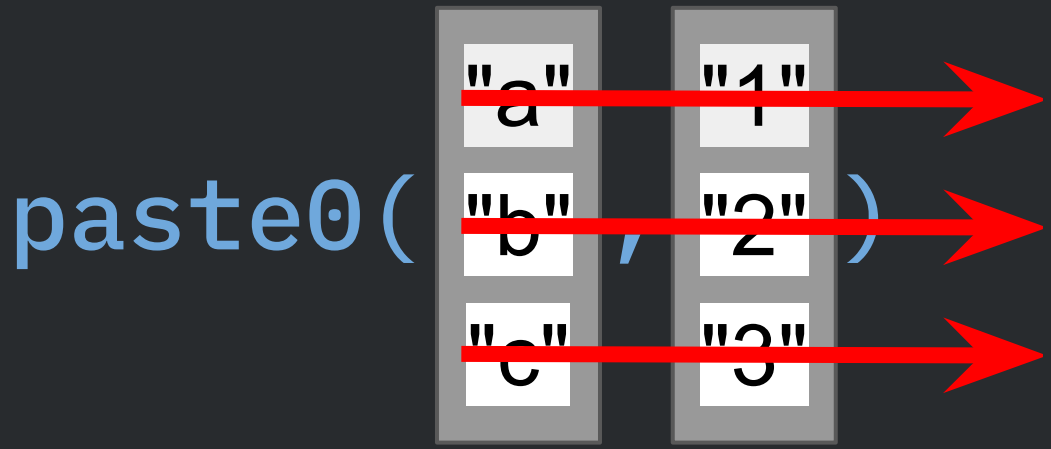
Recycling



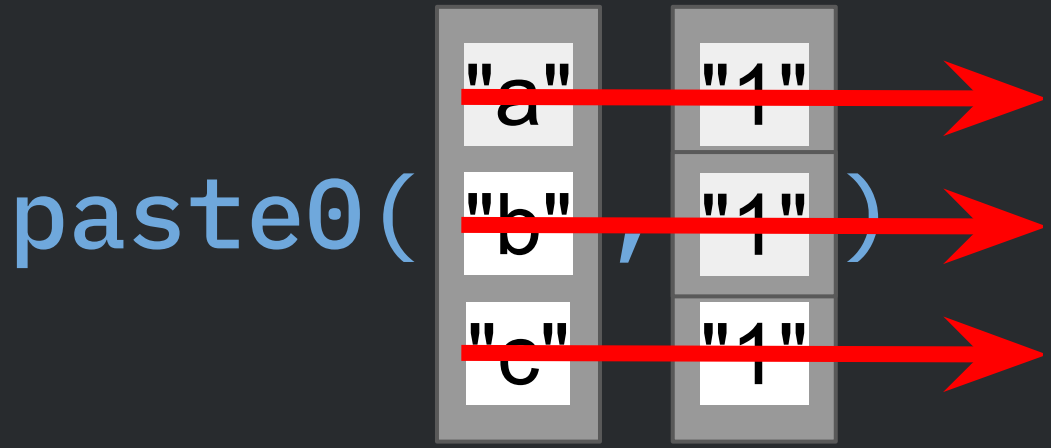
Recycling



Element-wise operation

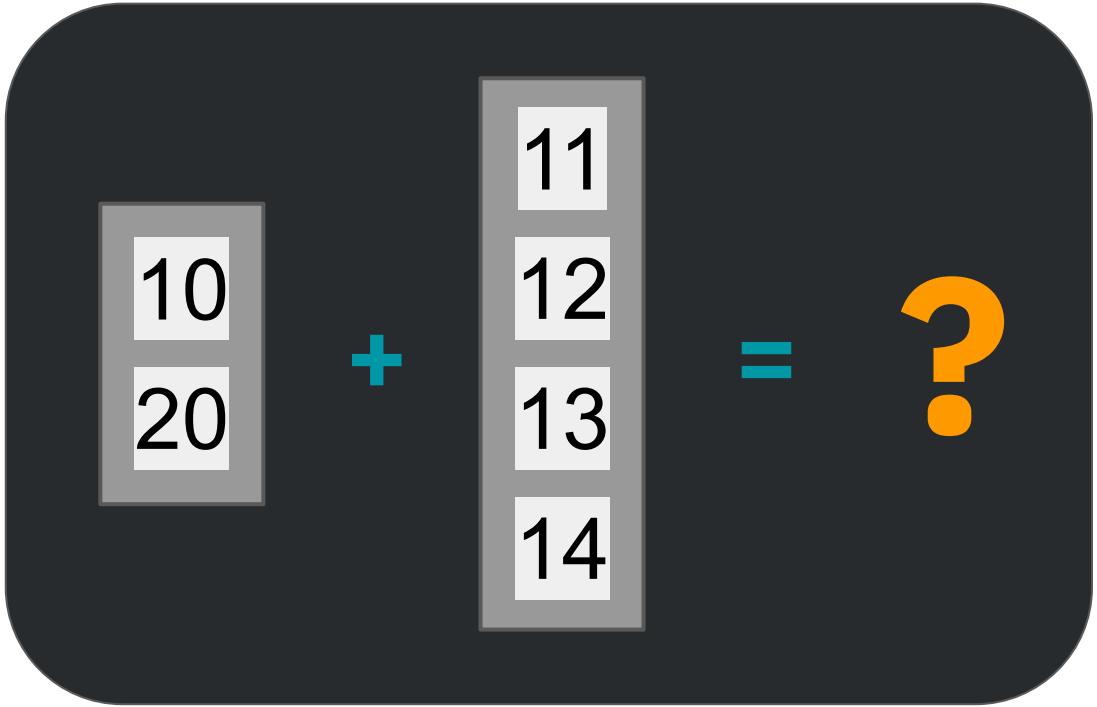


paste0()



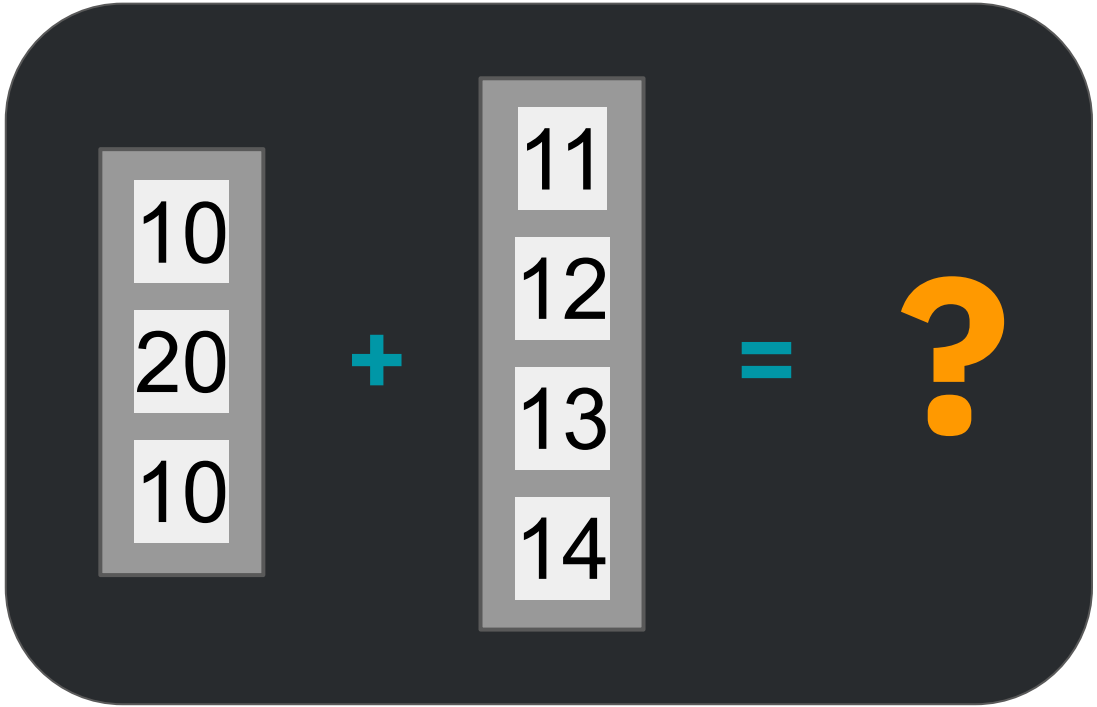
Recycling

Pop Quiz



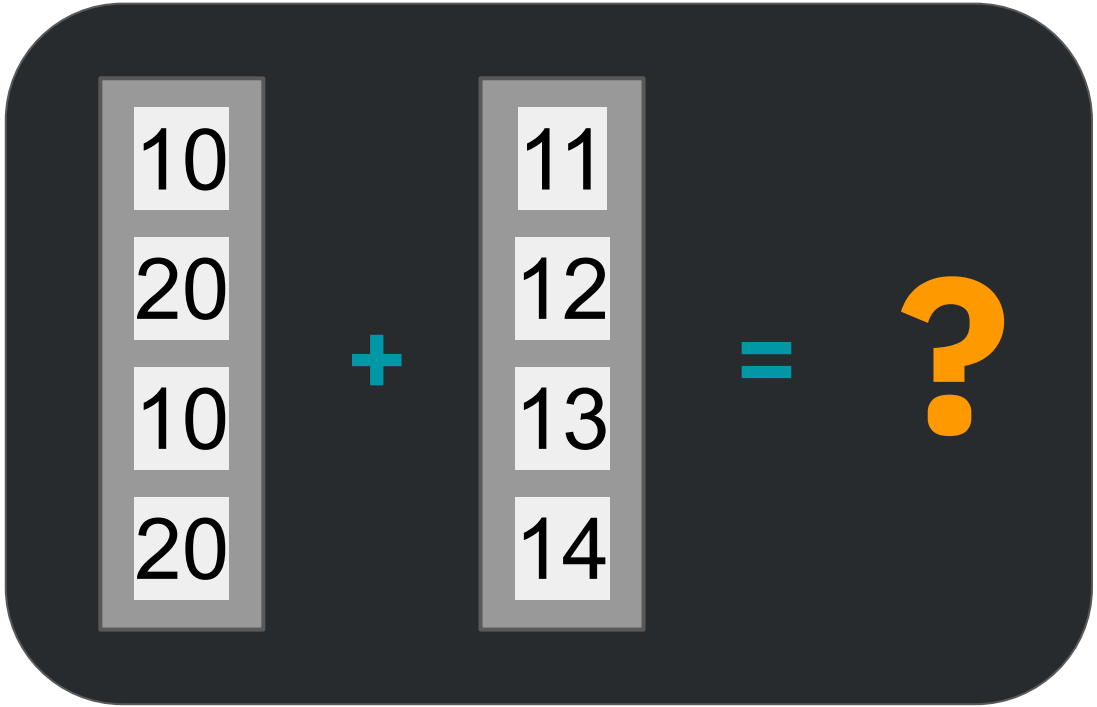
A	B	C	D
21	21	21	21
31	32	22	32
		23	23
		24	34

Pop Quiz



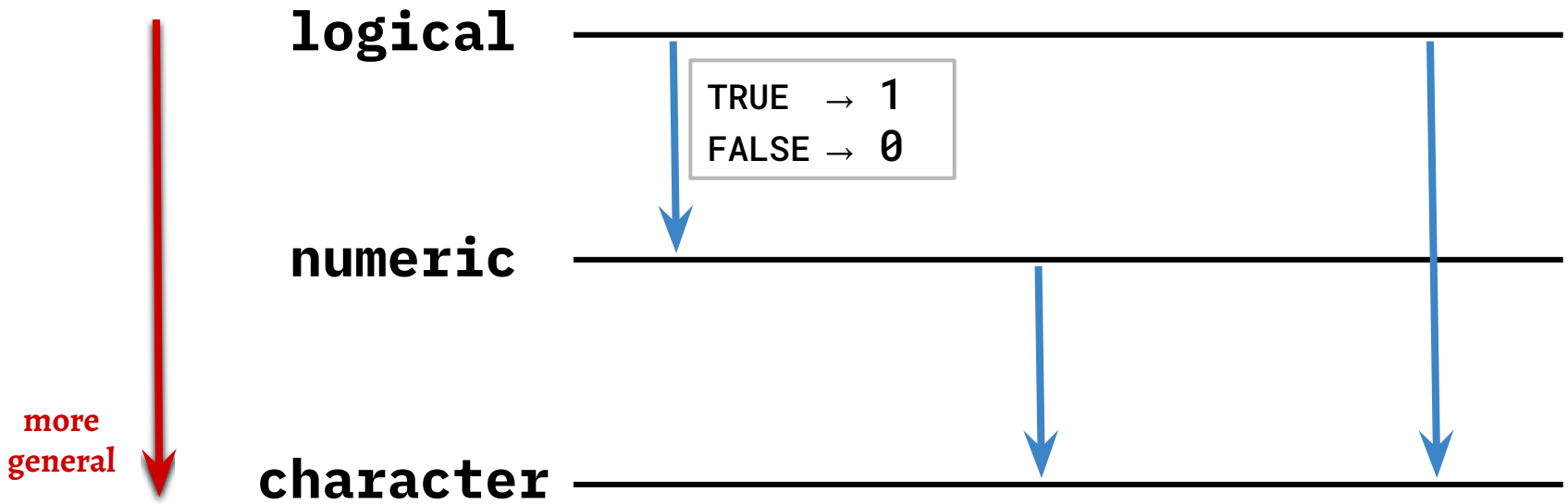
A	B	C	D
21	21	21	21
31	32	22	32
		23	23
		24	34

Pop Quiz



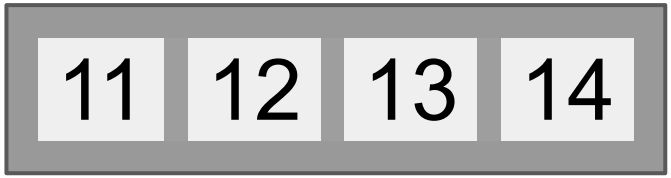
A	B	C	D
21	21	21	21
31	32	22	32
		23	23
		24	34

Rules of Coercion



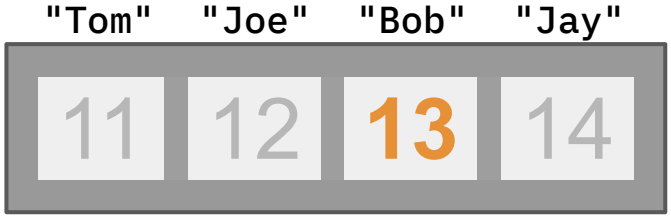
Subsetting a vector

3 種篩選方式

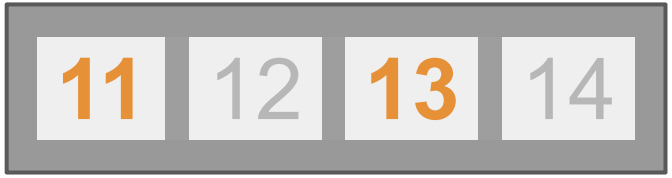


vec

vec[3]



vec["Bob"]



vec[c(T , F , T , F)]

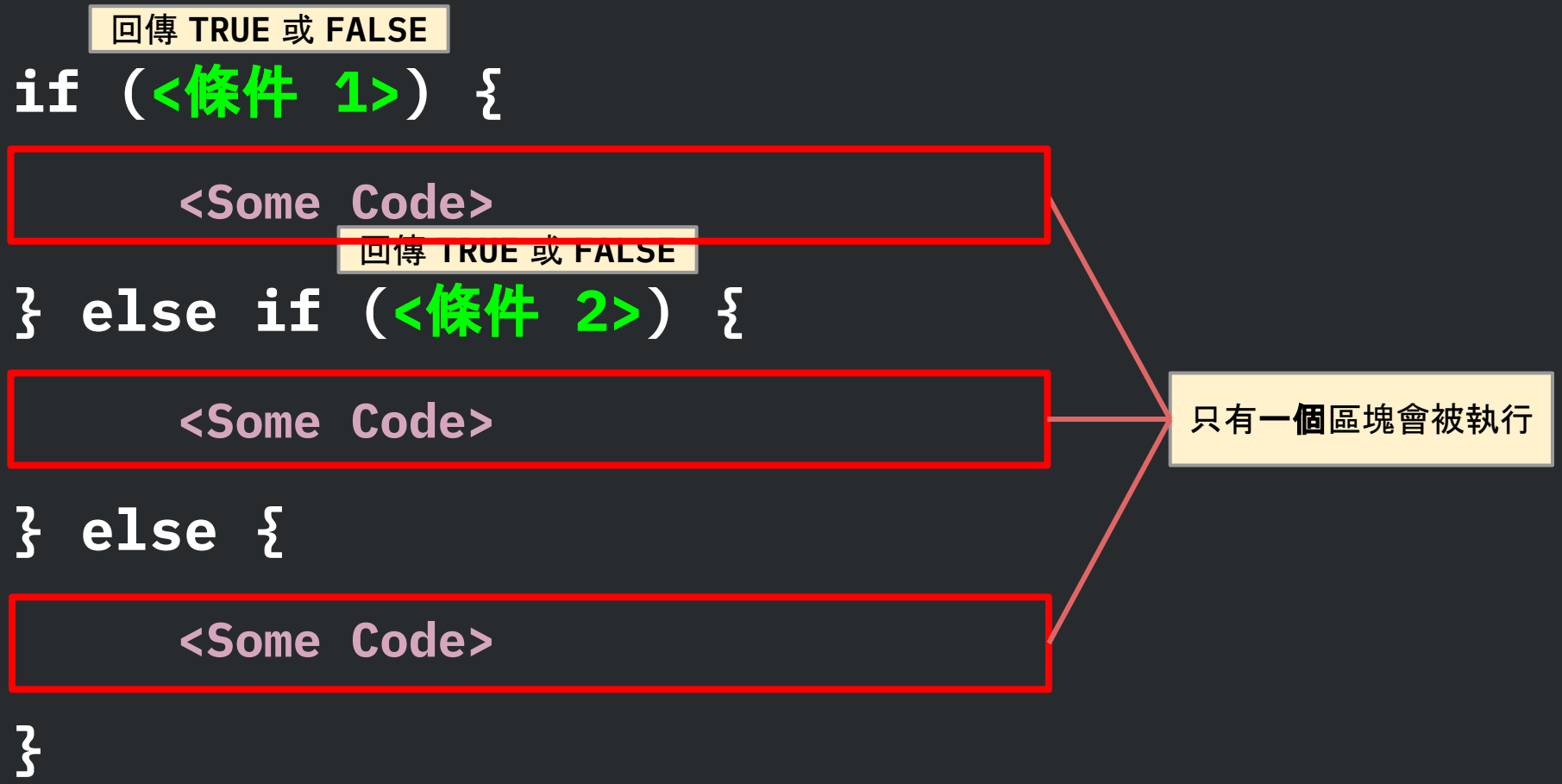
條件式
(if-else)

x 等於 -1

```
x <- -1
if (x > 0) {
      print("x is positive")
} else if (x < 0) {
      print("x is negative")
} else {
      print("x equals zero")
}
print("I'm always printed")
```

x 等於 0

```
x <- 0
if (x > 0) {
      print("x is positive")
} else if (x < 0) {
      print("x is negative")
} else {
      print("x equals zero")
}
print("I'm always printed")
```


回傳 TRUE 或 FALSE
if (<條件 1>) {
<Some Code>
回傳 TRUE 或 FALSE
} else if (<條件 2>) {
<Some Code>
} else {
<Some Code>
}
只有一個區塊會被執行

R Markdown

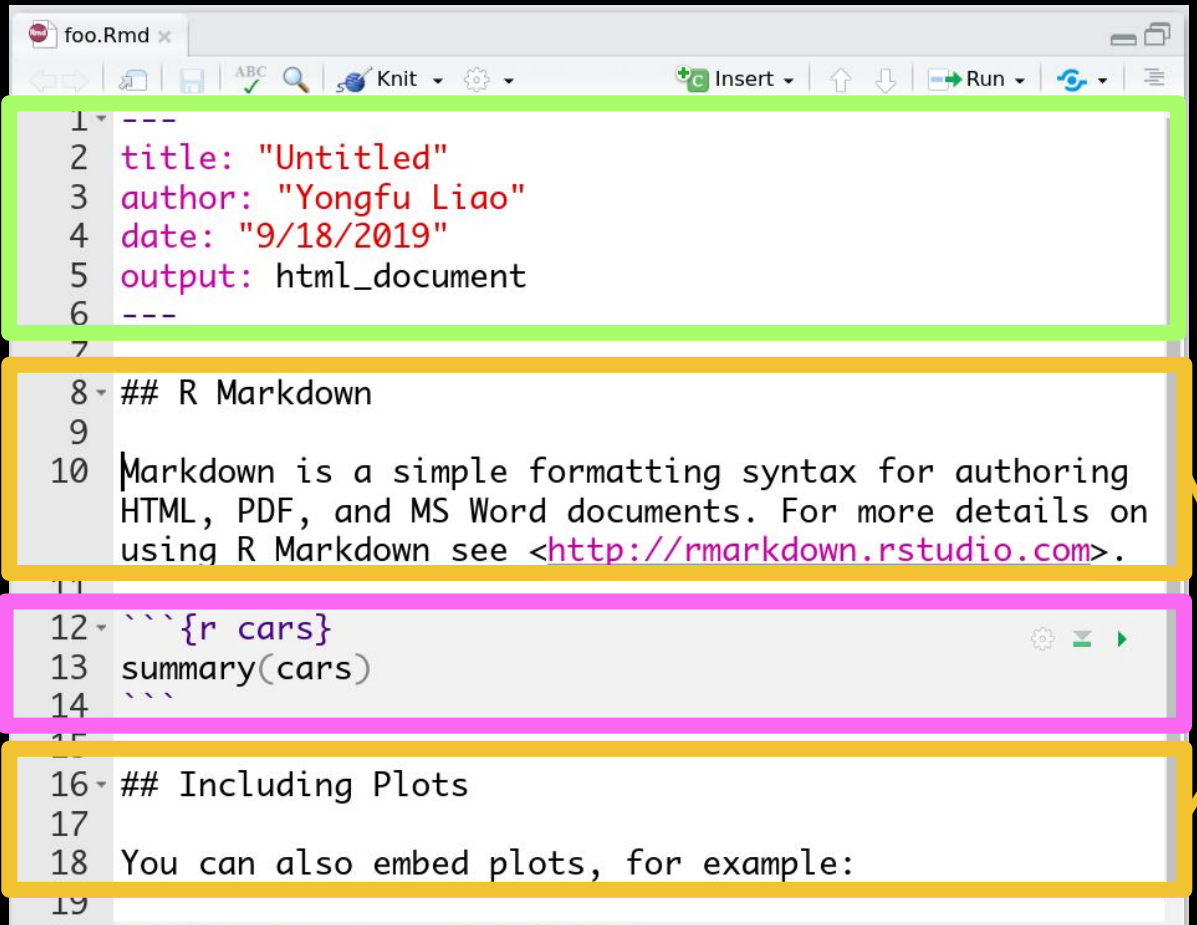

Rmd
knitr
md
pandoc
HTML
PDF

File --> New File --> R Markdown...

R Markdown 的結構

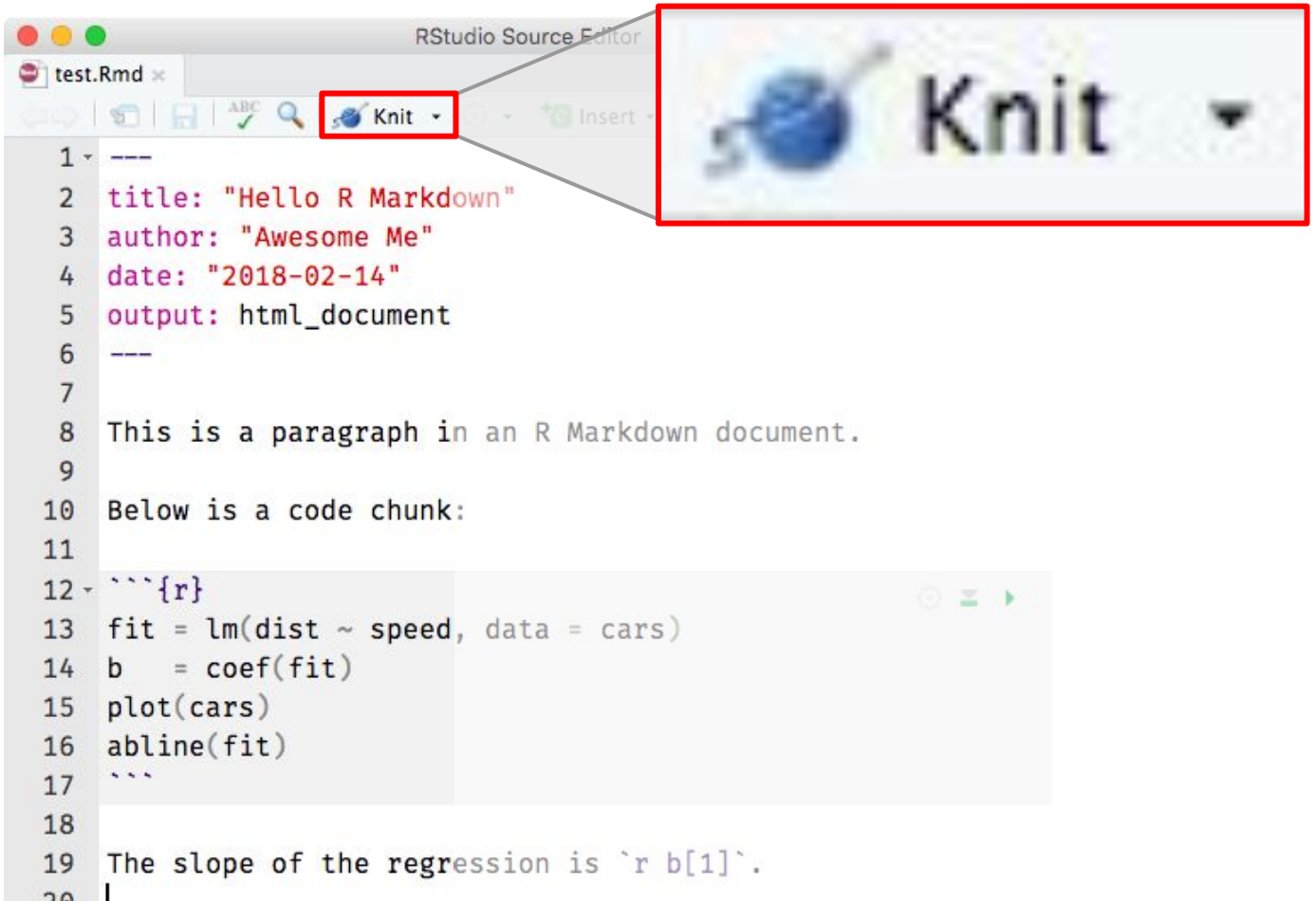
yaml:
控制文件格式

R Code Chunk:
可執行 R 語言的區塊



Markdown
內文

輸出 R Markdown



R Markdown Cheat Sheet