

# Chapter 5

---

## FUNCTION

# 函數 (Function)

---

還記得第一章的 `class()`，第二章的 `c()`，第三章的 `str()` ？

這些都是內建且好用的函數

而本章節要來介紹一些簡單的的數學，文字，描述統計函數

以及自訂函數

# 數值函數

函數	Meaning
abs()	絕對值
sqrt()	開根號
ceiling( )	無條件進入
floor()	無條件捨去

```
1 a <- abs(-3)
2 print(a)
3
4 b <- sqrt(8)
5 print(b)
6
7 c <- ceiling(c(1.23, 2.34, 3.45, 1.0))
8 print(c)
9
10 d <- floor(c(1.23, 2.34, 3.45, 1.0))
11 print(d)
```

```
Console Terminal x
~/
> a <- abs(-3)
> print(a)
[1] 3
>
> b <- sqrt(8)
> print(b)
[1] 2.828427
>
> c <- ceiling(c(1.23, 2.34, 3.45, 1.0))
> print(c)
[1] 2 3 4 1
>
> d <- floor(c(1.23, 2.34, 3.45, 1.0))
> print(d)
[1] 1 2 3 1
```

# 數值函數

函數	Meaning
round()	四捨五入
exp(x)	$e^x$
log10()	取出以 10 為底的對數

```
1 a <- round(9.87)
2 print(a)
3
4 b <- round(9.87, digit=1)
5 print(b)
6
7 c <- exp(0)
8 print(c)
9
10 d <- log10(100)
11 print(g)
```

```
Console Terminal x
~/
> a <- round(9.87)
> print(a)
[1] 10
>
> b <- round(9.87, digit=1)
> print(b)
[1] 9.9
>
> c <- exp(0)
> print(c)
[1] 1
>
> d <- log10(100)
> print(g)
[1] 2
> |
```

# 文字函數

函數	Meaning
toupper()	轉大寫
tolower()	轉小寫
substr()	擷取文字
floor()	無條件捨去

```
1 a <- toupper("Linkin Park")
2 print(a)
3
4 b <- tolower("STEPhen cuRRy")
5 print(b)
6
7 #start是起點, end是終點
8 c <- "I love Golden State Warriors"
9 d <- substr(c, start = 1, stop = 13)
10 print(d)
```

```
> a <- toupper("Linkin Park")
> print(a)
[1] "LINKIN PARK"
>
> b <- tolower("STEPhen cuRRy")
> print(b)
[1] "stephen curry"
>
> #start是起點, end是終點
> c <- "I love Golden State Warriors"
> d <- substr(c, start = 1, stop = 13)
> print(d)
[1] "I love Golden"
>
```

# 文字函數

函數	Meaning
grep()	搜尋文字向量
sub()	替換文字向量樣式

```
1 a <- c("pen", "apple", "pineAPPLE" )
2 b <- grep(pattern="apple", a)
3 c <- grep(pattern="iphone", a)
4 print(b)
5 print(c)
6
7 d <- grep(pattern="APPLE", a,
8           ignore.case=TRUE)
9 print(d)
10
11 e <- sub(pattern="pineAPPLE", replacement ="ipohne", a)
12 print(e)
```

```
Console Terminal x
~/
> a <- c("pen", "apple", "pineAPPLE" )
> b <- grep(pattern="apple", a)
> c <- grep(pattern="iphone", a)
> print(b)
[1] 2
> print(c)
integer(0)
>
> d <- grep(pattern="APPLE", a,
+           ignore.case=TRUE)
> print(d)
[1] 2 3
>
> e <- sub(pattern="pineAPPLE", replacement ="ipohne", a)
> print(e)
[1] "pen"      "apple"    "ipohne"
>
```

# 文字函數

函數	Meaning
strsplit()	將文字切割成向量
paste()	將文字連接起來

```
1 test <- "I love Golden State Warriors"
2 cut<-strsplit(test, split=" ")
3 print(cut)
4
5 print(class(cut))
6
7 a <- "Curry"
8 b <- "Durant"
9 c <- "Thompson"
10 d <- "Green"
11 e <- "McGee"
12
13 f <- paste(a, b, c, d, e)
14 print(f)
```

```
> test <- "I love Golden State Warriors"
> cut<-strsplit(test, split=" ")
> print(cut)
[[1]]
[1] "I"          "love"       "Golden"    "State"     "Warriors"

>
> print(class(cut))
[1] "list"

>
> a <- "Curry"
> b <- "Durant"
> c <- "Thompson"
> d <- "Green"
> e <- "McGee"
>
> f <- paste(a, b, c, d, e)
> print(f)
[1] "Curry Durant Thompson Green McGee"
```

# 描述統計函數

函數	Meaning
mean()	算平均
sd()	算標準差
median()	找中位數
range()	找最大值和最小值

```
1 vc <- 1:9
2
3 a <- mean(vc)
4 print(a)
5
6 b <- sd(vc)
7 print(b)
8
9 c <- median(vc)
10 print(c)
11
12 d <- range(vc)
13 print(d)
```

Console

Terminal x

~/

```
> vc <- 1:9
>
> a <- mean(vc)
> print(a)
[1] 5
>
> b <- sd(vc)
> print(b)
[1] 2.738613
>
> c <- median(vc)
> print(c)
[1] 5
>
> d <- range(vc)
> print(d)
[1] 1 9
```



# 描述統計函數

函數	Meaning
sum()	算總和
max()	找到最大值
min()	找最小值

```
1 vc <- 1:9
2
3 a <- sum(vc)
4 print(a)
5
6 b <- max(vc)
7 print(b)
8
9 c <- min(vc)
10 print(c)
```

```
Console Terminal x
~/
> vc <- 1:9
>
> a <- sum(vc)
> print(a)
[1] 45
>
> b <- max(vc)
> print(b)
[1] 9
>
> c <- min(vc)
> print(c)
[1] 1
>
```

# 隨堂練習 1

---

假設有一個直徑為 150 高為 40 的圓柱體

Q. 算出圓柱體底面積和體積

```
> print(area)
[1] 17671.46
>
> vol <- area * 40
> print(vol)
[1] 706858.3
> |
```

# 隨堂練習 2

---

1. 輸出今天的日期 ex. 2018 2/26

```
> print(a)  
[1] "Today: 2018-02-26"
```

2. 輸出下個星期的今天 ex. 2018 3/5

```
> print(b)  
[1] "Next week: 2018-03-05"
```

3. 將 Q2 的西元改成民國 ex. 107 3/05

```
> print(c)  
[1] "Next week: 107-03-05"  
> |
```

# 自訂函數

---

除了內建函數之外

我們可以自行撰寫一些函數來滿足特定需求

# 自訂函數

## 自定函數

```
函數名稱 <- function( 參數  
1,2,3...)  
{  
  函數內容...  
  return( 資料 ) #option  
}
```

```
1 hello <- function()  
2 {  
3   print("hello R")  
4 }  
5  
6 hello()
```

```
> hello <- function()  
+ {  
+   print("hello R")  
+ }  
>  
> hello()  
[1] "hello R"  
> |
```

# 自訂函數

## 自定函數 + 參數及回傳

```
函數名稱 <- function( 參數  
1,2,3...)  
{  
  函數內容...  
  return( 資料 ) #option  
}
```

```
1 sums <- function(a,b)  
2 {  
3   c <- a+b  
4   return(c)  
5 }  
6  
7 a <- 3  
8 b <- 4  
9 d <- sums(a,b)  
10  
11 print(d)|
```

```
Console Terminal x  
~/  
> sums <- function(a,b)  
+ {  
+   c <- a+b  
+   return(c)  
+ }  
>  
> a <- 3  
> b <- 4  
> d <- sums(a,b)  
>  
> print(d)  
[1] 7  
> |
```

# 自訂函數

## 自定函數 + if else

```
函數名稱 <- function( 參數  
1,2,3...)  
{  
  函數內容...  
  return( 資料 ) #option  
}
```

```
1 cal <- function(e,opt)  
2 {  
3   if(opt==1){  
4     return (e^2)  
5   } else{  
6     return (e^3)  
7   }  
8 }  
9 edge = 5  
10 a <- cal(5,1)  
11 print(a)  
12  
13 b <- cal(5,2)  
14 print(b)  
15
```

Console	Terminal x
~/	
> cal <- function(e,opt) + { +   if(opt==1){ +     return (e^2) +   } else{ +     return (e^3) +   } + }	
> edge = 5 > a <- cal(5,1) > print(a) [1] 25 > > b <- cal(5,2) > print(b) [1] 125 >	

## 隨堂練習 3

---

請自定一個函數輸入一個  $n$

Q. 算出  $n!$

```
> ans <- fact(5)
> print(ans)
[1] 120
> |
```



# 隨堂練習 4

---

第四章我們寫過請尋找離某數字最近的二個質數

Ex `a<-100`, output : 97, 101

`a<-93`, output : 89, 97

請將找質數的程式碼重新整理成自訂函數 + 主程式

ex. `prime(a)`, output : 97, 101

Any Questions !?