

# (5) R程式設計風格及範例講解

吳漢銘

淡江大學 數學系  
資料科學與數理統計組

<http://www.hmwu.idv.tw>





# 程式風格

2/21

```
> source("Calculator.R")
> demo(2, 3, 1)
> my.sum(2, 3)
```

```
1 #####
2 # Name: Calculator.R                               #
3 #   for demonstration                             #
4 # Author: Han-Ming Wu                             #
5 # Date: 2008/03/19                               #
6 #####
7
8 demo <- function(a, b, item){
9
10     c <- switch(item,
11                 my.sum(a, b),
12                 my.diff(a, b),
13                 my.prod(a, b),
14                 my.divide(a, b))
15     list(answer=c)
16 }
17
18 # sum
19 my.sum <- function(a, b){
20     c <- a+b
21     c
22 }
23
24 # diff
25 my.diff <- function(a, b){
26     c <- abs(a-b)
27     c
28 }
29
30 #prod
31 my.prod <- function(a, b){
32     c <- a*b
33     c
34 }
35
36 #divide
37 my.divide <- function(a, b){
38     c <- a/b
39     c
40 }
41
```

## Google's R Style Guide

<http://google-styleguide.googlecode.com/svn/trunk/Rguide.xml>

Google's R Style Guide

R is a high-level programming language used primarily for statistical computing and graphics. The goal of the R Programming Style Guide is to make our R code easier to read, share, and verify. The rules below were designed in collaboration with the entire R user community at Google.

### Summary: R Style Rules

1. **File Names:** end in .R
2. **Identifiers:** variable.name (or variableName), FunctionName, kConstantName
3. **Line Length:** maximum 80 characters
4. **Indentation:** two spaces, no tabs
5. **Spacing**
6. **Curly Braces:** first on same line, last on own line
7. **else:** Surround else with braces
8. **Assignment:** use <-, not =
9. **Semicolons:** don't use them
10. **General Layout and Ordering**
11. **Commenting Guidelines:** all comments begin with # followed by a space; inline comments need two spaces before the #
12. **Function Definitions and Calls**
13. **Function Documentation**
14. **Example Function**
15. **TODO Style:** TODO(username)

### Summary: R Language Rules

1. **attach:** avoid using it
2. **Functions:** errors should be raised using stop()



# 程式風格範例

3/21

Paul E. Johnson, R Style. An Rchaeological Commentary

<http://cran.r-project.org/web/packages/rockchalk/vignettes/Rstyle.pdf>

R Coding Conventions (RCC)

<http://www.aroma-project.org/developers/RCC>

```
1 nor=matrix(rnorm(Nsim*p),nrow=p)
2 risk=matrix(0,ncol=150,nrow=10)
3 a=seq(1,2*(p-2),le=10)
4 the=sqrt(seq(0,4*p,le=150)/p)
5 for (j in 1:150){
6   nornor=apply((nor+rep(the[j],p))^2,2,sum)
7   for (i in 1:10){
8     for (t in 1:Nsim)
9       risk[i,j]=risk[i,j]+sum((rep(the[j],p)-
10 max(1-a[i]/nornor[t],0)*(nor[,t]+rep(the[j],p)
11 }}
12 risk=risk/Nsim
```

```
1 nor <- matrix(rnorm(Nsim*p), nrow=p)
2 risk <- matrix(0, ncol=150, nrow=10)
3 a <- seq(1, 2*(p-2), le=10)
4 the <- sqrt(seq(0, 4*p, le=150)/p)
5 for(j in 1:150){
6   nornor <- apply((nor + rep(the[j],p))^2, 2, sum)
7   for(i in 1:10){
8     for(t in 1:Nsim){
9       s <- max(1-a[i]/nornor[t], 0)
10      t <- nor[,t] + rep(the[j], p)
11      u <- sum((rep(the[j], p) - s*t)^2)
12      risk[i,j] <- risk[i,j] + u
13    }
14  }
15 }
16 risk <- risk/Nsim
```



# 請不要寫這種「程式」！

4/21

```
25 d11 <- ((x1-x2)^2+(y1-y2)^2)^(1/2)
26 d11
27
28 d12 <- ((x1-x3)^2+(y1-y3)^2)^(1/2)
29 d12
30
31 d13 <- ((x1-x4)^2+(y1-y4)^2)^(1/2)
32 d13
33
34 d14 <- ((x1-x5)^2+(y1-y5)^2)^(1/2)
35 d14
36
37 d21 <- ((x2-x3)^2+(y2-y3)^2)^(1/2)
38 d21
39
40 d22 <- ((x2-x4)^2+(y2-y4)^2)^(1/2)
41 d22
42
43 d23 <- ((x2-x5)^2+(y2-y5)^2)^(1/2)
44 d23
45
46 d31 <- ((x3-x4)^2+(y3-y4)^2)^(1/2)
47 d31
48
49 d32 <- ((x3-x5)^2+(y3-y5)^2)^(1/2)
50 d32
51
52 d41 <- ((x4-x5)^2+(y4-y5)^2)^(1/2)
53 d41
```

```
1 > if(
2 sqrt((m8[1,1]-m8[2,1])^2+(m8[1,2]-m8[2,2])^2)+
3 sqrt((m8[2,1]-m8[3,1])^2+(m8[2,2]-m8[3,2])^2)>
4 sqrt((m8[3,1]-m8[1,1])^2+(m8[3,2]-m8[1,2])^2)
5 &
6 sqrt((m8[1,1]-m8[2,1])^2+(m8[1,2]-m8[2,2])^2)+
7 sqrt((m8[3,1]-m8[1,1])^2+(m8[3,2]-m8[1,2])^2)>
8 sqrt((m8[2,1]-m8[3,1])^2+(m8[2,2]-m8[3,2])^2)
9 &
10 sqrt((m8[2,1]-m8[3,1])^2+(m8[2,2]-m8[3,2])^2)+
11 sqrt((m8[3,1]-m8[1,1])^2+(m8[3,2]-m8[1,2])^2)>
12 sqrt((m8[1,1]-m8[2,1])^2+(m8[1,2]-m8[2,2])^2)
13 ){
```

```
89 (x == my.data[1,1]) |(x == my.data[1,2]) |(x == my.data[1,3]) |(x ==
my.data[1,4]) |(x == my.data[1,5]) |(x == my.data[1,6])
90 第一筆 <-
c(my.data[1,1],my.data[1,2],my.data[1,3],my.data[1,4],my.data[1,5],my.data[1,6])
91
92 (x == my.data[2,1]) |(x == my.data[2,2]) |(x == my.data[2,3]) |(x ==
my.data[2,4]) |(x == my.data[2,5]) |(x == my.data[2,6])
93 第二筆 <-
c(my.data[2,1],my.data[2,2],my.data[2,3],my.data[2,4],my.data[2,5],my.data[2,6])
94
95
96 (x == my.data[3,1]) |(x == my.data[3,2]) |(x == my.data[3,3]) |(x ==
my.data[3,4]) |(x == my.data[3,5]) |(x == my.data[3,6])
97 第三筆 <-
c(my.data[3,1],my.data[3,2],my.data[3,3],my.data[3,4],my.data[3,5],my.data[3,6])
98
99
100
101 (x == my.data[4,1]) |(x == my.data[4,2]) |(x == my.data[4,3]) |(x ==
my.data[4,4]) |(x == my.data[4,5]) |(x == my.data[4,6])
102 第四筆 <-
c(my.data[4,1],my.data[4,2],my.data[4,3],my.data[4,4],my.data[4,5],my.data[4,6])
```



# 勉勵的話

5/21

全文摘錄自: <http://www.csie.stu.edu.tw/資料下載/課程資料/計算與邏輯思考/計算與邏輯思考ch2.doc>

學資訊相關科系的同學或多或少都要修習一些程式設計的課程，對許多人來說，學習程式設計是一件令人苦惱的事，枯燥的指令與失敗的挫折似乎比趣味與成就感的機會要多的多，而且事實上這些人之中，將來以設計程式為主要工作的比率也並不高。

那麼是否一定要花如此多的時間與精力來學呢？

事實上，程式除了是指揮電腦工作的工具之外，學習程式至少還有三個好處：

1. 學習程式是了解計算機運作原理的最佳途徑。
2. 培養邏輯思考的能力。
3. 構思一個有條理的程序來處理所面對的問題。

正如同很多事物的學習過程，學習程式設計者往往一開始面對的是很多不知其所以然的指令語法，然是一個一個的範例，從模仿中學習，但除非學習動機或興趣特別強，很可能在產生興趣之前就已經被枯燥與挫折打敗。

如果能多增加一些背景知識，其實應該可以對學習者有些助益。學習程式設計所應具備的能力其實不是很多，但是在中學以前的課程中，並未被特別提出來，對於資訊工程的學生，往往在修了離散數學這門課時，才恍然大悟原來是怎麼回事。我們的目的在提出學習程式所需的背景知識與基本技巧，以便於增加在學習任何一種程式語言之前，對計算機原理與程式設計的方法能夠有所了解，而不致感到枯燥乏味。

大家應該要知道，指令語法不過是程式設計的枝微末節，在懂了原理之後，使用任何一種程式語言不過是套用它所規定的語法而已，

針對初學者來說，我們並不著重於程式的效率，而是著重於培養如何找出一個正確的計算機程序來處理問題，同時，我們也不著重於某個特定程式語言的語法，而只著重於觀念的建立。





<http://www.thenewslens.com/post/49248/>  
<http://www.thenewslens.com/post/28632/>

6/21

連5歲小孩都會：英國將寫程式列為中小學必修，台灣卻連大學生都不想學

孫德明 (Jim) 2014/06/19 發表於：國際・教育・科技

上個月我參加史丹福大學 (Stanford) 校友會及學術文教基金會舉辦的「雲端科技與人文生活」論壇，聆聽到一場母校工學院院長Dr. James Plummer的演講，提到近年來在美國大學裡，選讀電腦 / 計算機課程及學位的學生大幅增加，史丹福大學的工學院也根據現代職場及技術領域最需要的「T型人才」的培養，進行了許多的課程及活動改造。我認為這個資訊教育的趨勢十分值得我們關注，而對應的人才培訓機制也很值得我們參考。

我上網找了一些資料，果然看到具體的數據，顯示這幾年選擇資訊科學科系的學生比例增加很多。

四大名校主修資訊系的學生逐年增加

Year	Stanford	MIT
2000	400	600
2001	450	650
2002	500	700
2003	550	750
2004	600	800
2005	650	850
2006	700	900
2007	750	950
2008	800	1000
2009	850	1050
2010	900	1100

歐巴馬呼籲美國年輕人學寫程式，台灣的教育鼓勵我們學什麼？

MrLow-Tech 2014/03/12 發表於：國際・教育・科技



# 範例 1

7/21

## 問題想法:

配對正確 (輸出)



左括號個數 = 右括號個數



計數字串「左，右」括號個數



從字串中找出「左，右」括號



字串由螢幕輸入

輸入包含左右小括號之字串(最長為40字元),  
請判斷是否左右小括號配對正確。

(例1) 輸入:  $((1+2)-3)*(4/5)$

輸出: 括號配對正確。

(例3) 輸入:  $((1+2+3)$

輸出: 括號配對不正確。

(例3) 輸入:  $((1+2)*(3+4)*(5+6))/(7+8)$

輸出: 括號配對正確。



# 程式想法

8/21

從輸入字串中找出「左，右」括號

scan

從字串中，並找出所要的pattern

**Pattern Matching and Replacement**  
`grep, sub, gsub, regexpr, gregexpr`

查 **help, google,...**

**Substrings of a Character Vector**  
`substr, substring`





# 一般程式設計流程

1. 輸入格式應有一特別規定。(例如一個點座標)
  - `x1 <- 3, y1 <- 2; demo <- function(x1, y1){....}`
  - `x <- c(3, 2); demo <- function(x){....}`
2. 輸入/輸出測試OK。(先給定Input, 測試運算過程)
3. 加入判別。(輸入型態, 參數範圍, 長度大小, 真或偽)
4. 加入提示。(互動程式, 給提示, 減少輸入錯誤)
5. 加入註解。(便於日後維護)
6. 改變數名。(有義意的名稱)
7. 群組, 結構化。(重複的動作有哪一些?)
8. 寫作風格。(格式, 標頭, 空格, 對齊)
9. 好、巧、妙。(三境界)



# 範例1: 實作<sup>(1)</sup>

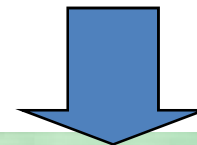
10/21

## 程式

```
cat("第一題")  
string <- "((1+2)*(3+4)*(5+6))/(7+8)"  
gregexpr("[()]", string)[[1]]  
length(gregexpr("[()]", string)[[1]])
```

## 執行

```
> cat("第一題")  
第一題> string <- "((1+2)*(3+4)*(5+6))/(7+8)"  
> gregexpr("[()]", string)[[1]]  
[1] 1 2 8 14 21  
attr(,"match.length")  
[1] 1 1 1 1 1  
> length(gregexpr("[()]", string)[[1]])  
[1] 5
```





# 範例1: 實作<sup>(2)</sup>

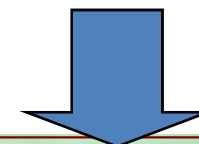
11/21

## 程式

```
cat("第一題\n")
string <- "((1+2)*(3+4)*(5+6))/(7+8)"
left.num <- length(gregexpr("[(", string)[[1]])
cat("left.num: ", left.num, "\n")
right.num <- length(gregexpr(")]", string)[[1]])
cat("right.num: ", right.num, "\n")
if(left.num== right.num){
  cat("OK")
} else{
  cat("Not OK")
}
```

## 執行

```
> cat("第一題\n")
第一題
> string <- "((1+2)*(3+4)*(5+6))/(7+8)"
> left.num <- length(gregexpr("[(", string)[[1]])
> cat("left.num: ", left.num, "\n")
left.num: 5
> right.num <- length(gregexpr(")]", string)[[1]])
> cat("left.num: ", right.num, "\n")
left.num: 5
> if(left.num== right.num){
+   cat("OK")
+ } else{
+   cat("Not OK")
+ }
OK>
```





# 範例1: 實作<sup>(3)</sup>

12/21

程式

```
ex1 <- function(){  
  cat("第一題\n")  
  
  #string <- "((1+2)*(3+4)*(5+6))/(7+8)"  
  cat("輸入包含左右小括號之字串(最長為40字元)，請判斷是否左右小括號配對正確")  
  string <- scan(what="character", nmax=1, quiet=TRUE)  
  
  left.num <- length(gregexpr("[(", string)[[1]])  
  #cat("left.num: ", left.num, "\n")  
  right.num <- length(gregexpr(")", string)[[1]])  
  #cat("right.num: ", right.num, "\n")  
  
  if(left.num== right.num){  
    cat("OK")  
  }else{  
    cat("Not OK")  
  }  
}
```

執行

```
ex1()
```

```
> ex1()  
第一題  
輸入包含左右小括號之字串(最長為40字元)，請判斷是否左右小括號配對正確1: ((1+2)*(3+4)*(5+6))/(7+8)  
OK>
```





# 範例1: 實作<sup>(4)</sup>

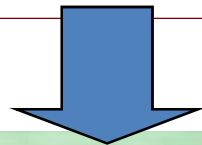
13/21

```
ex1 <- function(){
  cat("第一題\n")
  cat("#####\n")
  cat("# 輸入包含左右小括號之字串(最長為40字元),      #\n")
  cat("# 請判斷是否左右小括號配對正確                  #\n")
  cat("# 例如輸入: {\\tt ((1+2)-3)*(4/5)}              #\n")
  cat("#####\n")

  ##輸入
  string <- scan(what="character", nmax=1, quiet=TRUE)

  ##找出 "(" 和 ")", 並計數
  left.num <- length(gregexpr("(", string)[[1]])
  right.num <- length(gregexpr(")", string)[[1]])

  ##判斷是否相等
  if(left.num== right.num){
    ##是的話, 輸出OK
    cat("OK")
  }
  else{
    ##不是的話, 輸出NOT OK
    cat("Not OK")
  }
}
```







# 繼續練習這一題(Y/N)

14/21

```
Y.or.N <- "y"

while(Y.or.N=="y"){
  ex1()
  cat("繼續練習這一題(Y/N): ")
  Y.or.N <- scan(what="character", nmax=1, quiet=TRUE)
  if(Y.or.N!="Y" || Y.or.N!="N"){
    cat("輸入錯誤，再輸入一次 ")
  }
}
```



# 範例1: 完成

15/21

經過 n次修改及測試，存檔成ex1.R

```
1 #####
2 # Name: 範例主程式                                     #
3 # Author: Han-Ming Wu                                   #
4 # Date: 2008/11/05                                     #
5 #####
6
7 #####
8 # ex1                                                    #
9 #####
10 ex1 <- function(){
11   cat("#####\n")
12   cat("# 第一題                                         #\n")
13   cat("# 輸入包含左右小括號之字串(最長為40字元)       #\n")
14   cat("# 請判斷是否左右小括號配對正確                 #\n")
15   cat("# 例如輸入: ((1+2)-3)*(4/5)                     #\n")
16   cat("#####\n")
17
18   repeat{
19     ##輸入
20     cat("請輸入包含左右小括號之字串(最長為40字元): ")
21     string <- scan(what="character", nmax=1, quiet=TRUE)
22     if(nchar(string) < 40){
23
24       ##找出"(" ")", 並計數
25       left.num <- length(gregexpr("[{]", string)[[1]])
26       right.num <- length(gregexpr("[})", string)[[1]])
27       cat("左小括號個數為: ", left.num, "\n")
28       cat("右小括號個數為: ", right.num, "\n")
29
30       ##判斷是否相等
31       if(left.num== right.num){
32         ##是的話，輸出"配對正確"
33         cat("括號配對正確!\n")
34       }
35       else{
36         ##不是的話，輸出"配對不正確"
37         cat("括號配對不正確!\n")
38       }
39       break
40     }
41   }
42   else{
43     cat("輸入錯誤!\n")
44   }
45 }
46 }
```

```
48 #####
49 # 是否繼續                                             #
50 #####
51
52 ask <- function(){
53   cat("繼續練習這一題(y/n): ")
54   Y.or.N <- scan(what="character", nmax=1, quiet=TRUE)
55   return(Y.or.N)
56 }
57
58 #####
59 # 檢查Input                                           #
60 #####
61 input.check <- function(answer){
62
63   if(answer=="N" || answer=="n"){
64     cat("謝謝練習，再會!\n")
65   }
66   else if(!(answer=="Y" || answer=="y")){
67     cat("輸入錯誤!\n")
68     Y.or.N <- ask()
69     Y.or.N <- input.check(Y.or.N)
70   }
71   else{
72     Y.or.N <- answer
73   }
74   return(Y.or.N)
75 }
76
77 #####
78 # Example                                             #
79 #####
80 Y.or.N <- "y"
81 while(Y.or.N=="y" || Y.or.N=="Y"){
82   ex1()
83   Y.or.N <- ask()
84   Y.or.N <- input.check(Y.or.N)
85 }
86
87 }
```



# 範例1: 執行

16/21

```
> source("ex1.R")
#####
# 第一題                                     #
# 輸入包含左右小括號之字串(最長為40字元)   #
# 請判斷是否左右小括號配對正確             #
# 例如輸入: ((1+2)-3)*(4/5)                 #
#####
請輸入包含左右小括號之字串(最長為40字元): 1: ((1+2)*(3+4)*(5+6))/(7+8)
左小括號個數為: 5
右小括號個數為: 5
括號配對正確!
繼續練習這一題(y/n): 1: y
#####
# 第一題                                     #
# 輸入包含左右小括號之字串(最長為40字元)   #
# 請判斷是否左右小括號配對正確             #
# 例如輸入: ((1+2)-3)*(4/5)                 #
#####
請輸入包含左右小括號之字串(最長為40字元): 1: (3+5)/(5*7*(9-3))
左小括號個數為: 3
右小括號個數為: 4
括號配對不正確!
繼續練習這一題(y/n): 1: n
謝謝練習, 再會!
```



## 範例 2

17/21

某國發行了1,5,10,50,100不同面額的鈔票，若有人要從銀行領出 $N$ 元，銀行行員要如何發給鈔票，使用的張數會最少？

(例) 輸入: 478

輸出: 1元3張，5元1張，10元2張，50元1張，100元4張，共478元。

**問題想法:**

張數最少



面額最大的開始換



剩下的錢，再換次大面額的



重複，直到換完。



## 範例 3

18/21

平面上兩點 $(x_1, y_1)$ ,  $(x_2, y_2)$ 之距離式為:  $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。  
給定 $n$ 個點( $n \leq 10$ ), 找出構成最小周長的三角形的三個點。

(例) 輸入:  $(1, 1)$   $(0, 0)$   $(4, 3)$   $(2, 0)$   $(7, 8)$

輸出: 三點為 $(1, 1)$   $(0, 0)$   $(2, 0)$ , 其周長為4.828428。

問題想法: 三角形周長



三個點的兩兩距離



三個點是否成一個三角形?



10點任取三個點有多少可能?



跑完所有可能



取最小的周長的那一組





# 範例 3: 可能用到之副程式

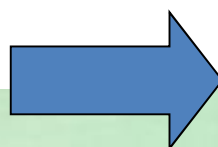
19/21

```
1 #  
2 # 兩點之間距離  
3 #  
4 my.dis <- function(a, b){  
5   dis <- sqrt((a[1]-b[1])^2 + (a[2]-b[2])^2)  
6   return(dis)  
7 }  
8  
9 #  
10 # 判斷三點可否形成一個三角形  
11 #  
12 is.triangle <- function(a, b, c){  
13  
14   is.tri <- FALSE  
15   dis.ab <- my.dis(a, b)  
16   dis.bc <- my.dis(b, c)  
17   dis.ac <- my.dis(a, c)  
18   count.1 <- ifelse(dis.ab + dis.bc > dis.ac, 0, 1)  
19   count.2 <- ifelse(dis.ab + dis.ac > dis.bc, 0, 1)  
20   count.3 <- ifelse(dis.bc + dis.ac > dis.ab, 0, 1)  
21   if(count.1 + count.2 + count.3 == 0){  
22     is.tri <- TRUE  
23   }  
24   return(is.tri)  
25 }  
26
```

```
27 #  
28 # 三角形之周長  
29 #  
30 my.perimeter <- function(a, b, c){  
31  
32   if(is.triangle(a, b, c){  
33     length <- my.dis(a,b)+ my.dis(a,c)+ my.dis(b,c)  
34   }  
35   else{  
36     length <- Inf  
37   }  
38   return(length)  
39 }  
40
```

```
for(i in 1: n){  
  for(j in (i+1): n){  
    for(k in (j+1): n){  
      ....  
    }  
  }  
}
```

找出規則  
(重復、順序、整批)



寫成function  
用apply, lapply, tapply,...等等



# 範例 4: 找出規律

20/21

輸入任何一個正整數  $n(n \leq 10)$ , 輸出 $n$ 階層的 Pascal 三角形。

(例) 輸入: 5

輸出:

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

問題想法:

觀察數列規則



細部修正(空格)

```
100  else if(x==6){
101      cat("      1\n")
102      cat("     1 1\n")
103      cat("    1 2 1\n")
104      cat("   1 3 3 1\n")
105      cat("  1 4 6 4 1\n")
106      cat(" 1 5 10 10 5 1\n")
107  }
```


choose( $n$ , 0: $n$ )




# R Documentation

21/21

```
312 #####
313 #
314 #
315 #####
316 #' @title Box Plots For Interval Data
317 #' @description Produce box-and-whisker plot of the interval data
318 #' @param idata an IntervalData object
319 #' @details ...
320 #' @author Han-Ming Wu
321 #' @seealso boxplot.sbs.i, boxplotdou.i
322 #' @return The percentiles of the interval data
323 #' @examples
324 #' #' data(face)
325 #' idata.x <- face$x
326 #' y.C <- face$y
327 #' title <- "face data"
328 #' boxplot.i(idata.x)
329
330 boxplot.i <- function(idata, ...){
331
332
333   p <- dim(idata)[2]
334   n
335   st
```

Exploratory Symbolic Data Analysis 



Documentation for package 'exploreSDA' version 0.0.0.9000

- [DESCRIPTION file](#).
- [Code demos](#). Use [demo\(\)](#) to run them.

Help Pages

<a href="#">boxplot.i</a>	Box Plots For Interval Data
<a href="#">boxplot.sbs.i</a>	The Side-by-Side Box Plots For Interval Data
<a href="#">cbind.h</a>	Combine HistData Objects by Rows or Columns
<a href="#">cbind.i</a>	Combine IntervalData Objects by Rows or Columns
<a href="#">get.subset</a>	The Subset of the Histogram Data
<a href="#">hM.size</a>	The Dimensions of the Histogram Data
<a href="#">plot.index.i</a>	The Index Plot For Interval Data

boxplot.i {exploreSDA} R Documentation

Box Plots For Interval Data

Description

Produce box-and-whisker plot of the interval data

Usage

```
## S3 method for class 'i'
boxplot(idata, ...)
```

Arguments

idata an IntervalData object

Details

...

Value

The percentiles of the interval data

Author(s)

Han-Ming Wu

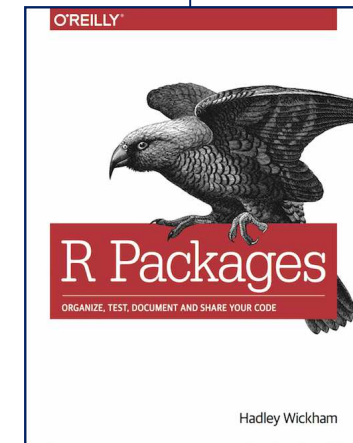
See Also

[boxplot.sbs.i](#), [boxplotdou.i](#)

Examples

```
boxplot.i(idata.x)
```

[Package *exploreSDA* version 0.0.0.9000 [Index](#)]



<http://r-pkgs.had.co.nz/>

## Create an R Package in RStudio

<https://www.youtube.com/watch?v=9PyQlbAEujY>