

數字預設是實數型物件(numeric) 類別:字元型(character) 實數型(numeric) 整數型(integer) 複數型(complex) 邏輯型(logical: TRUE/FALSE)

✧ 想要整數物件的話，需要在數字最後加 L

Ex. c(class(1),class(1L)) [1] "numeric" "integer"

✧ 特殊數位 Inf, 表示無窮大，-Inf 表示無窮小

✧ 特殊的值 NaN 是未定義值，可以看成非數字，也可認為是缺失值

✧ 冒號用來創建等差數列 Ex. x<-1:20; x #整數向量 1~20

✧ c()函數用來創建向量

x<-c(T,F); x #TRUE 可以縮寫為 T，數字 1 代表 TURE [1] TRUE FALSE x<-c(1+0i,2+4i); x [1] 1+0i 2+4i

✧ 利用 vector()創建初始向量，或類型對應的函數如 integer() x<-1:3; names(x)<-c("a","b","c"); x a b c
x<-vector("numeric",length=10); x 等於 integer(10)#會針對資料物件類型給預設值 1 2 3

如果創建的資料物件類型不一致，會創建一個最低級公共類型的向量，即強制向低級轉換

Ex. x<-c(T,2);x [1] 1 2 x<-c("a",T);x [1] "a" "TRUE" gl(2,4) [1] 1 1 1 1 2 2 2 2 Levels: 1 2

字元型 < 數字型 < 邏輯型 gl(2, 2, 20)[1] 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 Levels: 1 2 c d

✧ 以 as. 開頭的函數轉換 EX. x<-0:6; class(x) as. 想轉換的型態(x) m<-matrix(1:4,2,2) a 1 3

轉換一定要符合邏輯，不成功會返回 NA 值，比如字元轉換數字 dimnames(m)<-list(c("a","b"),c("c","d")); m b 2 4

(命名列表)mylist<-list(name1=object1, name2=object2,...

✧ (矩陣)args(matrix)<-要設矩陣時一定要先寫

function (data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)

m<-matrix(nrow=2,ncol=3,dimnames=list(c("r1","r2"),c("c1","c2","c3")));m

c1 c2 c3 attributes(m)和 dim(m) #查看維度

r1 NA (1) NA(3) NA (5) #默認按 Column 排列(1-1 -> 2-1 -> 2-1 2-3)

r2 NA (2) NA(4) NA (6) m<-1:6; dim(m)<-c(2,3);m #通過 dim()對向量創建矩陣

通過 cbind() & rbind()綁定(binding)Column 或 Row

x<-1:3; y<-10:12

cbind(x,y)		rbind(x,y)			, , c1			, , c2			foo bar			c(nrow(x),ncol(x))
x	y	[,1]	[,2]	[,3]	B1	B2	B3	B1	B2	B3	1	1	TRUE	[1] 4 2
[1,] 1 10	x	1	2	3	A1	1	3	5	A1	7	9	11	2 2 TRUE	
[2,] 2 11	y	10	11	12	A2	2	4	6	A2	8	10	12	3 3 FALSE	
[3,] 3 12											4	4	FALSE	

陣列(array)和矩陣的區別在於 array 維度大於 2

dimn1<-c("A1","A2"); dimn2<-c("B1","B2","B3");dimn3<-c("c1","c2")

x<-array(1:12,dim=c(2,3,2),dimnames = list(dimn1,dimn2,dimn3));x

✧ (數據框)可以通過讀取表格函數 read.table()或 read.csv()讀取數據框亦可通過調用 data.matrix()將數據框轉化為矩陣

x<-data.frame(foo=1:4,bar=c(T,T,F,F));x gl(2, 3, labels = c("女","男"),ordered=T)[1] 女 女 女 男 男 男 Levels: 女 < 男

(因子)一般由函數 factor()創建，有序變數需要指定參數 ordered=TRUE，等級從小到大

每個整數有一個標籤(level)，可以用 unclass(x)查看 x<-factor(c("yes","yes","no","yes"),ordered=T);x

table(x) #看每個等級的頻數#levels 定義因子順序，預設是按字母順序(a>b)

(gl 創建因子) args(gl) (判斷 NAN 值)使用函數 is.na(), is.nan()NaN 屬於 NA，但是 NA 不屬於 NaN，NaN 僅用於數學運算

names()給向量、列表、數據框命名 dimnames() 給矩陣命名

文字編輯器：mydata<-data.frame(a=1:2,b=c("a","b")) fix(mydata) #fix()函數打開文字編輯器編輯

代碼輸入：直接將資料登錄成字符串格式，用 read.table 讀取，由 text 指定此字符串

(讀取數據) read.table(),read.csv()讀取表格資料

[]從向量中提取多個元素 [[]]從列表或數據框中提取一個元素，無法提取多個 \$可以從有命名的列表或數據框中提取元素，和[][]一樣

x>"a" ##字母可以通過字典排序法來排序 x<-matrix(1:12,2,6); x#通过(i,j)索引

\$後面的名字不能是計算得到的#索引不到結果就會為 NULL x<-list(a=1:4,b=0.6,c="hello") x[c(1,3)]== x[c("a","c")]

x<-list(a=1:4,b=0.6,c="hello") name<- "a" x[name]== x\$a x<-list(a=list(10,12,14),b=c(3.14,2.81)) x[[c(1,3)]] [1] 14

x[[1]][[3]] ##連著取兩次子集 [1] 14x[[c(2,1)]] ##只能是數位索引 [1] 3.14

名字索引部分匹配(縮寫僅能唯一匹配)，只能用在雙引號和\$符號中 x<-list(aardvark=1:5) x\$a x[["a"]] ##預設情況下不會部分匹配

(生成新元素) x<-1:3;x[4]<-100;x [1] 1 2 3 100 x[7]<-0.01;x [1] 1e+00 2e+00 3e+00 1e+02 NA NA 1e-02

x<-data.frame(a=T);x x[2]<- "lalala";x x["c"]<-1+2i;x x\$d<-5L;x a V2 c d

x<-c(1,2,NA,4,NA,5) bad<-is.na(x);bad x[!bad] (去除N) 1 TRUE lalala 1+2i 5

is.na() & complete.cases()得到邏輯向量，透過取子集剔除缺失值 na.omit()直接剔除(物件一般是數據框)

x<-c(1,2,NA,4,NA,5) y<-c("a","b",NA,"d","e",NA) good<-complete.cases(x,y);good x[good] y[good]

airquality[4:6,] good<-complete.cases(airquality) airquality[good,][4:6,]== na.omit(airquality)[4:6,]

attach()可簡化欄位的提取 當R遇到一個變數後，先在全域環境中查找，再到此搜索路徑中查找

summary(airquality\$Ozone)== attach(airquality) summary(Ozone) detach(airquality)

另一種方式是使用函數 with(數據框名，語句) with(airquality, summary(Ozone))

x<-matrix(1:4,2,2);y<-matrix(rep(10,4),2,2);x*y x%*%y #矩陣運算

函數	功能	示例
mean	均值	mean(1:10)
sd	标准差	sd(1:3)
scale	正态化	scale(1:3)
var	方差	var(1:3)
median	中位数	median(1:10)
sum	求和	sum(1:10)
rowSums	每行和	rowSums(matrix(1:10,2,5))
unique	去重	unique(c(1,2,1,3,2))

函数	功能	示例
abs	绝对值	abs(-1.2)
sqrt	平方根	sqrt(1:10)
ceiling	取整+1	ceiling(pi)
floor	取整	floor(pi)
round	小数位数	round(pi,digits=2)
sin	三角函数	sin(pi/6)
log,exp	对数指数	exp(1);log(exp(2))
类似的还有cos,log2,log10		

rmnorm:模擬出 n 個正態隨機變量 rmnorm(n, mean=0, sd=1)

sample(1:10, 4, replace=T) sample(letters, 5, replace=F) sample(1:10) sample(1:10 , replace=T) (一般隨機亂數)

設置隨機種子(seed)是非常重要的，它可以做到重複生成同樣一組隨機數 set.seed(1);rmnorm(4) rmnorm(4)

創建重複數 rep(x, times, each) rep(1:2,4) [1] 1 2 1 2 1 2 1 2 rep(1:2, each=4) [1] 1 1 1 1 2 2 2 2

創建等差數列 seq(2,7,0.5)[1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 x<-c("a","b","c");seq_along(x) #相當於

seq(1:length(x))[1] 1 2 3 sequence(3:5) #相當於 c(1:3,1:4,1:5)[1] 1 2 3 1 2 3 4 1 2 3 4 5

(字串連接)paste(c("ai", "ya", "ya"),c("1", "2"), sep="\$", collapse="-")

[1] "ai\$1-ya\$2-ya\$1

paste0("ai", "ya", "ya")

[1] "aiyaya"

x <- c(as="asfef", "yuiop[, "stuff.blah.yech")

strsplit(x, "e")(先除掉 e 後分開) strsplit("abcd", NULL)(分開 abcd)

x<-c("bizarre", "love", "triangle", "we're", "meant[") substr(x, 2, 5)

substr(x, 1, 4)<- " 好玩 ";x

x<-list(foo=1:4, bar=0.6)(取子集列表)

x[1] x[[1]] x\$bar x[["bar"]] x["bar"]

\$foo [1] 1 2 3 4 [1] 0.6 [1] 0.6 \$bar

[1] 1 2 3 4 [1] 0.6

read.table(), read.csv()讀取表格資料 header：是否有表頭(read.table 默認為 F，read.csv 默認為 T) sep：分割字元(read.table 默認為空格，read.csv 默認為逗號)