

Capítol 2

Matrius

Una **matriu** m és un objecte **estructurat**, bidimensional, que pot ser vist com una taula. Des del punt de vista d'estructures de dades, una matriu és un vector de vectors. És a dir, un vector on cada component és un vector. Aleshores, al igual que els vectors, tots els components d'una matriu són del mateix tipus base T . Llavors, una matriu és un objecte homogeni, compostat per un nombre finit de files F (entrades horitzontals) i un nombre finit de columnes C (entrades verticals) anomenats components o elements. De manera genèrica, la matriu $m_{F \times C}$ de la que estem parlant és de la forma:

$$\begin{array}{c} \overbrace{\hspace{10em}}^{C \text{ columnes}} \\ \begin{array}{ccccc} m_{1,1} & m_{1,2} & \dots & m_{1,C-1} & m_{1,C} \\ m_{2,1} & m_{2,2} & \dots & m_{2,C-1} & m_{2,C} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{F-1,1} & m_{F-1,2} & \dots & m_{F-1,C-1} & m_{F-1,C} \\ m_{F,1} & m_{F,2} & \dots & m_{F,C-1} & m_{F,C} \end{array} \\ \underbrace{\hspace{10em}}_{F \text{ files}} \end{array}$$

Com podem veure, la matriu m té dimensió $F \times C$ i per tant té $F \times C$ components on $\forall i \in \{1, \dots, F\} \forall j \in \{1, \dots, C\}, m_{i,j} \in T$. És a dir, $m_{i,j}$ es un valor del tipus base T . Una matriu d'enters, de dimensió 4×4 podria ser aquesta de sota:

$$m_{4 \times 4} = \begin{bmatrix} 1 & 2 & 3 & 7 \\ 4 & 4 & 3 & 2 \\ 2 & 1 & 6 & 1 \\ 3 & 1 & 5 & 1 \end{bmatrix}$$

Mentre que una matriu de booleans, de dimensió 3×2 pot ser per exemple:

$$m_{3 \times 2} = \begin{bmatrix} TRUE & TRUE \\ FALSE & TRUE \\ TRUE & FALSE \end{bmatrix}$$

2.1 Creació de matrius al llenguatge R

Al llenguatge R tenim la següent funció per construir matrius.

Funció `matrix()`

Aquest constructor ens permet crear matrius per enumeració dels seus components, i.e. mitjançant un vector.

1. El nombre de files, $matrix(c(e_1, \dots, e_n), nrow = F)$

```
1 > m <- matrix(c(1,2,3,4,5,6,7,8,9,10, 11,12), nrow
  =4)
2 > m
3      [,1] [,2] [,3]
4 [1,]    1    5    9
5 [2,]    2    6   10
6 [3,]    3    7   11
7 [4,]    4    8   12
8 >
```

2. El nombre de columnas, $matrix(c(e_1, \dots, e_n), ncol = C)$

```
1 > m <- matrix(c(1,2,3,4,5,6,7,8,9,10, 11,12), ncol
  =4)
2 > m
3      [,1] [,2] [,3] [,4]
4 [1,]    1    4    7   10
5 [2,]    2    5    8   11
6 [3,]    3    6    9   12
7 >
```

3. Ambdós, $matrix(c(e_1, \dots, e_n), nrow = F, ncol = C)$

```
1 > m <- matrix(c(1,2,3,4,5,6,7,8,9,10, 11,12), nrow
  =4, ncol=3)
2 > m
3      [,1] [,2] [,3]
4 [1,]    1    5    9
5 [2,]    2    6   10
6 [3,]    3    7   11
7 [4,]    4    8   12
8 >
```

4. Cap modalitat $matrix(c(e_1, \dots, e_n))$

```

1 > m <- matrix(c(1,2,3,4,5,6,7,8,9,10, 11,12))
2 > m
3      [,1]
4 [1,]    1
5 [2,]    2
6 [3,]    3
7 [4,]    4
8 [5,]    5
9 [6,]    6
10 [7,]    7
11 [8,]    8
12 [9,]    9
13 [10,]   10
14 [11,]   11
15 [12,]   12
16 >

```

IMPORTANT Els components de una matriu creada amb la funció `matrix()` són emmagatzemats per columnes.

La matriu resultant tindrà dimensió igual al nombre d'elements que s'han fet servir a la crida de la funció `matrix()`, és a dir n . Aleshores, n ha de ser múltiple de les files i de les columnes.

```

1 > m <- matrix(c(1,2,3,4,5,6,7,8,9,10, 11,12), nrow=5)

Mensajes de aviso perdidos
In matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), nrow = 5) :
la longitud de los datos [12] no es un submúltiplo o múltiplo del número
de filas [5] en la matriz

1 > m <- matrix(c(1,2,3,4,5,6,7,8,9,10, 11,12), ncol=7)

Mensajes de aviso perdidos
In matrix(c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12), ncol = 7) :
la longitud de los datos [12] no es un submúltiplo o múltiplo del número
de columnas [7] en la matriz

1 > m <- matrix(c(1,2,3,4,5,6,7,8,9,10, 11,12), nrow=2)
2 > m
3      [,1] [,2] [,3] [,4] [,5] [,6]
4 [1,]    1    3    5    7    9   11
5 [2,]    2    4    6    8   10   12
6 >

```

De tota manera, si volem emmagatzemar els components per files podem fer servir la modalitat `matrix(c(...),nrow=F, byrow=TRUE)`

Com al cas dels vectors, poden crear matrius "buides". En realitat son matrius on els seus components tenen el valor NA.

```

1 > m <- matrix(nrow=4,ncol=3)
2 > m
3      [,1] [,2] [,3]

```

```

4  [1,]    NA    NA    NA
5  [2,]    NA    NA    NA
6  [3,]    NA    NA    NA
7  [4,]    NA    NA    NA
8  > m <- matrix(nrow=4)
9  > m
10         [,1]
11 [1,]      NA
12 [2,]      NA
13 [3,]      NA
14 [4,]      NA
15 > m <- matrix(ncol=3)
16 > m
17         [,1] [,2] [,3]
18 [1,]      NA    NA    NA
19 >
20 > m <- matrix()
21 > m
22         [,1]
23 [1,]      NA
24 >

```

A diferència dels vectors, després d'assignar un valor d'un tipus T a algú dels components de la matriu, la resta dels components mantenen el valor NA.

```

1  > m <- matrix(nrow=4,ncol=3)
2  > m
3         [,1] [,2] [,3]
4 [1,]      NA    NA    NA
5 [2,]      NA    NA    NA
6 [3,]      NA    NA    NA
7 [4,]      NA    NA    NA
8  > m[3,2] <- 100
9  > m
10         [,1] [,2] [,3]
11 [1,]      NA    NA    NA
12 [2,]      NA    NA    NA
13 [3,]      NA  100    NA
14 [4,]      NA    NA    NA
15 > m <- matrix(nrow=4,ncol=3)
16 > m[3,2] <- TRUE
17 > m
18         [,1] [,2] [,3]
19 [1,]      NA    NA    NA
20 [2,]      NA    NA    NA
21 [3,]      NA  TRUE    NA
22 [4,]      NA    NA    NA
23 >

```

2.2 Accés als components d'una matriu

Mitjaçant un **operador d'accés directe**, que té com a paràmetres dos enters que corresponen a un índex per la fila i un índex per a la columna que es desitja accedir, respectivament. És a dir indica la posició que es desitja visitar, es pot accedir a qualsevol component del vector de manera directa. Matemàticament l'especificació d'aquesta funció és la següent:

$$[\] : \text{matriu} \times \text{enter} \times \text{enter} \rightarrow T$$

Llavors, donat una matriu m i un parell d'expressions de tipus enter, exp_f i exp_c , fem servir la crida

$$m[exp_f, exp_c]$$

amb

$$1 \leq \text{avaluacio}(exp_f) \leq F \quad (2)$$

$$1 \leq \text{avaluacio}(exp_c) \leq C \quad (3)$$

per referir-nos al component de la matriu que es troba a la posició $\langle exp_f, exp_c \rangle$.

Aleshores, com al cas dels vectors, tenim accés directe per visitar els components d'una matriu.

```

1 > m
2      [,1] [,2] [,3]
3 [1,]    1    5    9
4 [2,]    2    6   10
5 [3,]    3    7   11
6 [4,]    4    8   12
7 > m[3,2]
8 [1] 7
9 > m[1,3]
10 [1] 9
11 > i <- 2
12 > m[2*i,i]
13 [1] 8
14 > j <- 2
15 > m[i+j,3]
16 [1] 12
17 >

```

Una de les facilitats per manipular matrius que ens ofereix l'R és que podem accedir no només als components de les matrius si no també a submatrius. En particular, podem agafar una fila o una columna.

```

1 > m
2      [,1] [,2]
3 [1,]    1    4
4 [2,]    2    5
5 [3,]    3    6
6 > m[,2]      #accedeix a la columna 2
7 [1] 4 5 6
8 > m[2,]      #accedeix a la fila 2
9 [1] 2 5

```

```

10 > m[1:2,]
11      [,1] [,2]
12 [1,]    1    4
13 [2,]    2    5
14 >

```

És molt important vigilar que es satisfacin les restriccions (2) i (3). En cas que no es respectin aquestes restriccions, rebrem un missatge d'error en intentar accedir a una posició il·legal dins de la matriu considerada.

```

1 > m
2      [,1] [,2] [,3] [,4]
3 [1,]    1    4    7   10
4 [2,]    2    5    8   11
5 [3,]    3    6    9   12
6 > m[3,7]

```

Error en m[3, 7] : subíndice fuera de los límites

Semblant als vectors, els components d'una matriu pertanyen al tipus base T , llavors $m[exp_f, exp_c] \in T$ i que per tant, aquest component es comporta com una variable del tipus T i podem fer servir qualsevol dels seus operadors.

```

1 > b
2      [,1] [,2]
3 [1,] TRUE TRUE
4 [2,] FALSE TRUE
5 [3,] FALSE FALSE
6 [4,] TRUE FALSE
7 > b[1,2] && b[3,1]
8 [1] FALSE
9 > b[1,2] && b[4,1]
10 [1] TRUE
11 > m
12      [,1] [,2]
13 [1,]    1    4
14 [2,]    2    5
15 [3,]    3    6
16 > cat(m[2,2]+m[3,1], "\n")
17 8
18 >

```

A més dels operadors mencionats a dalt, podem conèixer la dimensió d'una matriu i el número de components que té mitjançant els següents operadors:

$$\dim : \text{matriu} \rightarrow \text{enter} \times \text{enter}$$

$$\text{length} : \text{matriu} \rightarrow \text{enter}$$

```

1 > m <- matrix(c(1,2,3,4,5,6), ncol=2)
2 > m
3      [,1] [,2]
4 [1,]    1    4

```

```

5 [2,]      2      5
6 [3,]      3      6
7 > dim(m)
8 [1] 3 2
9 > c <- dim(m)
10 > c
11 [1] 3 2
12 > length(m)
13 [1] 6
14 >

```

2.3 Recorregut

Hi ha molts problemes on s'han de processar tots i cadascun dels components d'una matriu. Com ja hem dit al capítol 1, aquesta classe de problemes se'ls anomena "Recorreguts". El següent esquema és un recorregut per files d'una matriu:

```

1 A
2 for (i in 1:nrow(m)){
3     for (j in 1:ncol(m)){
4         processar(m[i,j])
5     }
6 }
7 B

```

On A correspon a un bloc d'instruccions de preprocessament, per exemple, la inicialització d'algunes variables, i B correspon a un bloc d'instruccions de postprocessament, per exemple, imprimir el resultat.

També es pot fer el recorregut d'una matriu per columnes amb l'esquema de sota:

```

1 A
2 for (j in 1:ncol(m)){
3     for (i in 1:nrow(m)){
4         processar(m[i,j])
5     }
6 }
7 B

```

Ara vegem l'escriptura dels components d'una matriu m :

```

1 escriure_matriu <- function(m){
2     for (i in 1:nrow(m)){
3         for (j in 1:ncol(m)){
4             cat(m[i,j], "\n")
5         }
6     }
7 }

```

I es pot utilitzar d'aquesta forma:

```

1 > m
2      [,1] [,2]
3 [1,]    1    2
4 [2,]    3    4
5 [3,]    5    6
6 > escriure_matriu(m)
7 1
8 2
9 3
10 4
11 5
12 6
13 >

```

Suposem que necessitem una funció per llegir (per files) una matriu. Aquesta funció pot ser con segueix

```

1 llegir_matriu <- function(F,C){
2   m <- matrix(nrow=F,ncol=C)
3   for (i in 1:F){
4     for (j in 1:C){
5       m[i,j] <- scan(n=1,quiet=TRUE)
6     }
7   }
8   return (m)
9 }

```

I la podem fer servir de la següent manera:

```

1 > m <- llegir_matriu(3,2)
2 1: 1
3 1: 2
4 1: 3
5 1: 4
6 1: 5
7 1: 6
8 > m
9      [,1] [,2]
10 [1,]    1    2
11 [2,]    3    4
12 [3,]    5    6
13 >

```

2.4 Cerca

Com al cas dels vectors, tenim una classe de problemes on no cal visitar o processar tots els elements, és a dir, fer un recorregut, perquè ens demanen confirmar un predicat, és a dir, una propietat, $P(m)$, sobre la matriu m . Per resoldre els problemes de cerca, s'han de fer servir les mateixes tècniques que hem explicat en la Secció 1.4. Per exemple:

$P(s)$ = "els components de la matriu $m_{F \times C}$ formen una seqüència estrictament creixent quan es recorren per fila".

Si analitzem aquesta propietat, ens adonarem que en realitat vol dir el següent: $\forall i \forall j : 1 \leq i \leq F : 1 < j \leq C : m[i, (j-1)] \leq m[i, j]$ i $\forall i : 1 < i \leq F : m[(i-1), C] \leq m[i, 1]$. Llavors, això és el que realment hem de comprovar. En aquest cas, la cerca consisteix en trobar un contraexemple d'aquesta propietat. És a dir, si trobem un parell de components consecutius del vector tal que $1 \leq i \leq n-1 : v[i] > v[i+1]$ hem d'aturar la comprovació de la propietat i donar una resposta negativa. En aquest cas, si féssim el recorregut, la resposta seria afirmativa. S'ha de notar que, segons la definició, tota seqüència buida o que tingui un sol element és creixent. Un possible script en R és el següent:

```

1  creixent <- function(m){
2      ant <- m[1,1]
3      for (i in 2:nrow(m)){
4          for (j in 1:ncol(m)){
5              if (m[i,j] <= ant) return (FALSE)
6              else ant <- m[i,j]
7          }
8      }
9      return (TRUE)
10 }
11
12 > m <- matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,byrow=TRUE)
13 > m
14      [,1] [,2] [,3]
15 [1,]    1    2    3
16 [2,]    4    5    6
17 [3,]    7    8    9
18 >
19 > creixent(m)
20 [1] TRUE
21 > m <- matrix(c(1,2,3,3,4,5,5,5,6),nrow=3,byrow=TRUE)
22 > m
23      [,1] [,2] [,3]
24 [1,]    1    2    3
25 [2,]    3    4    5
26 [3,]    5    5    6
27 > creixent(m)
28 [1] FALSE
29 >
30 > m <- matrix(c(1,2,3,3,4,5,5,5,6),nrow=3)
31 > m
32      [,1] [,2] [,3]
33 [1,]    1    3    5
34 [2,]    2    4    5
35 [3,]    3    5    6
36 > creixent(m)
37 [1] FALSE
38 >

```

```

39 > m <- matrix(c(9,2,3,3,4,5,5,5,6),nrow=3,byrow=TRUE)
40 > m
41      [,1] [,2] [,3]
42 [1,]     9     2     3
43 [2,]     3     4     5
44 [3,]     5     5     6
45 > creixent(m)
46 [1] FALSE
47 >
48 > m <- matrix(c(1,2,3,3,4,5,5,5,-1),nrow=3,byrow=TRUE)
49 > m
50      [,1] [,2] [,3]
51 [1,]     1     2     3
52 [2,]     3     4     5
53 [3,]     5     5    -1
54 > creixent(m)
55 [1] FALSE
56 >

```

Deixem com a exercici al lector fer la implementació corresponent a verificar si una matriu m és estrictament creixent quan es recorre (a) per columnes, (b) en ziga-zaga horitzontal (començant des de la posició $m[1,1]$ d'esquerra a dreta) i (c) en ziga-zaga vertical (començant des de la posició $m[1,1]$ d'adalt a baix).