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, composat 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:

$$\overbrace{\begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,C-1} & m_{1,C} \\ m_{2,1} & m_{2,2} & \dots & m_{2,C-1} & m_{2,C} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ m_{F-1,1} & m_{F-1,2} & \dots & m_{F-1,C-1} & m_{F-1,C} \\ m_{F,1} & m_{F,2} & \dots & m_{F,C-1} & m_{F,C} \end{bmatrix}}^{C\ columnes}$$
$$F\ files$$

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, $m_{i,j}$ és un valor del tipus base T. Una matriu d'enters, de dimensió 4×4 podria ser aquesta:

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

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, $matrix(c(e_1, ..., e_n), nrow = F)$

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

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

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

3. Ambdos, $matrix(c(e_1, ..., e_n), nrow = F, ncol = C)$

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

4. Cap modalitat $matrix(c(e_1, ..., e_n))$. En aquest cas els elements es colůloquen en una matriu d'una única columna.

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

IMPORTANT Els components d'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.

```
> 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
```

```
> 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
```

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

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, es poden crear matrius "buides". En realitat són matrius on els seus components tenen el valor `NA`.

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

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

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

```
> m <- matrix(nrow=4,ncol=3)
> m
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
[3,]   NA   NA   NA
[4,]   NA   NA   NA
> m[3,2] <- 100
> m
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
[3,]   NA  100   NA
[4,]   NA   NA   NA
> m <- matrix(nrow=4,ncol=3)
> m[3,2] <- TRUE
> m
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
[3,]   NA TRUE   NA
[4,]   NA   NA   NA
>
```

2.2 Accés als components d'una matriu

S'accedeix als components d'una matriu mitjaçant un **operador d'accés directe**, que té com a paràmetres dos enters que corresponen a un índex per a la fila i un índex per a la columna que es desitja accedir, respectivament. És a dir, indicant 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:

$$[\ \ ] : matriu \times enter \times enter \to T$$

Llavors, donada 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

$$1 \leq avaluacio(exp_f) \leq F \qquad (2)$$

$$1 \leq avaluacio(exp_c) \leq C \qquad (3)$$

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.

```
> m
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
> m[3,2]
[1] 7
> m[1,3]
[1] 9
> i <- 2
> m[2*i,i]
[1] 8
> j <- 2
> m[i+j,3]
[1] 12
>
```

Una de les facilitas 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 sencera.

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

```
> m[1:2,]
     [,1] [,2]
[1,]    1    4
[2,]    2    5
>
```

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 ilůlegal dins de la matriu considerada.

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

```
> b
      [,1]  [,2]
[1,]  TRUE  TRUE
[2,] FALSE  TRUE
[3,] FALSE FALSE
[4,]  TRUE FALSE
> b[1,2] && b[3,1]
[1] FALSE
> b[1,2] && b[4,1]
[1] TRUE
> m
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> cat(m[2,2]+m[3,1],"\n")
8
>
```

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:

$$dim : matriu \to enter \times enter$$

$$length : matriu \to enter$$

$$nrow : matriu \times enter \to enter$$

$$ncol : matriu \times enter \to enter$$

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

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 problemas se'ls anomena "Recorreguts". Per recórrer tots els elements d'una matriu hem d'accedir a tots ells. Això ho podem fer per files o per columnes. El següent esquema és un recorregut per files d'una matriu:

```
A
for (i in 1:nrow(m)){
   for (j in 1:ncol(m)){
      processar(m[i,j])
   }
}
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:

```
A
for (j in 1:ncol(m)){
   for (i in 1:nrow(m)){
      processar(m[i,j])
   }
}
B
```

Com a exemple podem veure l'escriptura dels components d'una matriu m:

```
escriure_matriu <- function(m){
   for (i in 1:nrow(m)){
      for (j in 1:ncol(m)){
         cat(m[i,j],"\n")
      }
   }
}
```

I aquesta funció es pot utilitzar d'aquesta forma:

```
> m
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> escriure_matriu(m)
1
2
3
4
5
6
>
```

També podem donar format de matriu a aquesta sortida fent la funció com segueix:

```
escriure_matriu <- function(m){
   for (i in 1:nrow(m)){
      for (j in 1:ncol(m)){
         cat(m[i,j]," ")
      }
                cat(m[i,j],"\n")
   }
}
```

I això donarà la sortida següent:

```
> m
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> escriure_matriu(m)
1 2
3 4
5 6
>
```

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

```
llegir_matriu <- function(F,C){
   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 seqüencia 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 recorregut per files de la matriu tal que el primer sigui més gran que el segon hem d'aturar la comprovació de la propietat i donar una resposta negativa. 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 seguent:

```
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(3,5,6,7,9,11),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] TRUE
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).