Capítol 2

Matrius

Objectius: Aprendre a treballar amb l'estructura de dades `matrix`. Per això es dissenyaran un conjunt de funcions que permetin resoldre problemes bàsics sobre aquesta estructura de dades. S'utilitzaran algunes d'aquestes funcions per introduir (o recordar) algunes construccions bàsiques de l'estadística. Finalment es consideraran alguns casos de mostreig.

Primer de tot:

- Creeu una carpeta (o working directory) que es digui `Matrius` i emmagatzameu dins tots els fitxers que codifiquen les diferents funcions d'aquesta part.

- Recordeu d'anar a aquest directori utilitzant `setwd("..../Matrius")`

2.1 Exercicis

1. *QuantsCops*. Feu una funció que donat un enter i una matriu retorni el nombre de vegades que apareix l'enter en la matriu.

2. *Duplicats*. Feu una funció que rebi una matriu i la retorni amb tots els seus elements duplicats.

3. *GeneraDiagonal*. Feu una funció que rebi un vector de dimensió n i retorni una matriu de n x n que tingui aquest vector a la diagonal i a tota la resta zeros.

4. *ElementsDiagonal*. Feu una funció que rebi una matriu quadrada i retorni un vector amb els elements de la diagonal de la matriu.

5. *CaVertical.* Aquest exercici es compon de vàries funcions a dissenyar:

   (a) Dissenyeu una funció `mat_lletres` que donats dos nombres, corresponents al nombre de columnes i de files de la matriu, ompli aleatòriament de lletres {`a`, `b`, `c`} una matriu i la retorni. Aconsellem que l'script on escriviu la funció es digui igual que la funció.

   Com a exemple d'execució tindríeu:

```
>  setwd ("..../Matrius")
>  source ("mat_lletres.R")
>  M <- mat_lletres(5,10)
> M
      [,1] [,2] [,3] [,4] [,5]
 [1,] "b"  "c"  "c"  "c"  "a"
 [2,] "b"  "a"  "b"  "b"  "a"
 [3,] "c"  "c"  "b"  "a"  "b"
 [4,] "b"  "c"  "c"  "b"  "b"
 [5,] "b"  "a"  "b"  "a"  "b"
 [6,] "b"  "b"  "c"  "b"  "c"
 [7,] "a"  "a"  "a"  "a"  "b"
 [8,] "a"  "c"  "c"  "b"  "c"
 [9,] "b"  "a"  "c"  "a"  "c"
[10,] "b"  "a"  "c"  "b"  "a"
```

   (b) Dissenyeu una funció `quantes_ca` que donada una matriu de lletres calcula quantes vegades apareix (en vertical) la combinació "ca". La matriu haurà estat generada aplicant `mat_lletres`.

   Com a exemple d'execució tindríeu:

```
> M
      [,1] [,2] [,3] [,4] [,5]
 [1,] "b"  "c"  "c"  "c"  "a"
 [2,] "b"  "a"  "b"  "b"  "a"
 [3,] "c"  "c"  "b"  "a"  "b"
 [4,] "b"  "c"  "c"  "b"  "b"
 [5,] "b"  "a"  "b"  "a"  "b"
 [6,] "b"  "b"  "c"  "b"  "c"
 [7,] "a"  "a"  "a"  "a"  "b"
 [8,] "a"  "c"  "c"  "b"  "c"
 [9,] "b"  "a"  "c"  "a"  "c"
[10,] "b"  "a"  "c"  "b"  "a"
> source("quantes_ca.R")
>  n<-quantes_ca(M)
> n
[1] 5
```

(c) Donat que executant varies vegades `mat_lletres` s'obtenen resultats diferents (la generació de la matriu és aleatòria) com es pot veure en l'exemple següent:

```
> M1<-mat_lletres(5,10)
> M2<-mat_lletres(5,10)
> M1
      [,1] [,2] [,3] [,4] [,5]
 [1,] "b"  "b"  "a"  "c"  "c"
 [2,] "b"  "b"  "a"  "c"  "a"
 [3,] "a"  "c"  "c"  "c"  "a"
 [4,] "a"  "c"  "c"  "b"  "b"
 [5,] "a"  "c"  "b"  "a"  "b"
 [6,] "c"  "c"  "b"  "a"  "a"
 [7,] "a"  "b"  "c"  "b"  "a"
 [8,] "c"  "b"  "b"  "a"  "a"
 [9,] "a"  "c"  "a"  "b"  "a"
[10,] "a"  "c"  "b"  "a"  "a"
> M2
      [,1] [,2] [,3] [,4] [,5]
 [1,] "b"  "c"  "a"  "c"  "c"
 [2,] "c"  "a"  "a"  "c"  "b"
 [3,] "a"  "b"  "b"  "b"  "c"
 [4,] "c"  "c"  "b"  "a"  "a"
 [5,] "b"  "b"  "a"  "a"  "c"
 [6,] "c"  "c"  "b"  "c"  "c"
 [7,] "a"  "a"  "a"  "b"  "b"
 [8,] "c"  "b"  "a"  "c"  "b"
 [9,] "a"  "a"  "c"  "a"  "b"
[10,] "a"  "c"  "a"  "b"  "a"
>
```

Dissenyeu una funció `mostra_ca(s,c,f)` que a partir del nombre de columnes `c` i de files `f` i d'un nombre `s` que indica el nombre de resultats que es volen, retorni un vector de `s` components on cada component `i` s'obté amb el resultat d'aplicar la funció `quantes_ca` a una nova matriu generada amb `mat_lletres(c,f)`. És a dir, per a calcular cada component del vector resultant s'ha de generar una nova matriu amb `mat_lletres(c,f)` i executar `quantes_ca` sobre aquesta matriu.

6. *Combinacio31H*. Feu una funció que rebi una matriu amb 1, 2 i 3 i retorni quantes vegades hi apareix (en horitzontal) la combinació 31.

7. *Notes*. Disposem d'una matriu amb les notes de tots els estudiants en les assignatures del curs (matriu estudiants/assignatures amb contingut numèric que és la nota).

   Es demanen tres funcions que donada la matriu de notes de tots els estudiants calculin respectivament:

a) La mitjana de notes per a cada estudiant (retorna un vector amb totes les mitjanes).

b) La mitjana de notes per a cada assignatura (retorna un vector amb totes les mitjanes).

c) La mitjana de notes de tots els estudiants, és a dir la mitjana dels resultats de a) (retorna un valor).

Nota: No es pot usar la funció `mean` *de R*

8. *Matriu dispersa.* (Examen parcial 13/14). Direm que una matriu és **dispersa** quan la quantitat de zeros que té la matriu supera el 70% dels nombres totals de la matriu (el 70% es calcula multiplicant el nombre total d'elements de la matriu per 0.7).

   Donada aquesta descripció, fes una funció que donada una matriu retorni TRUE si aquesta matriu és dispersa i FALSE en cas contrari.

9. *Combinacio23.* Feu una funció que rebi una matriu amb 1, 2 i 3 i retorni quantes vegades hi apareix la combinació 23 (en qualsevol direcció: horitzontal, vertical o diagonal).

10. *MaximaMitjana.* Feu una funció que rebi una matriu i retorni en un vector dos valors, el primer és l'index de la columna que té la mitjana més gran, i el segon és el valor d'aquesta mitjana.

11. *Transposada de Matrius.* Feu una funció que rep una matriu i en torna la seva transposada. Òbviament, no podeu fer servir la funció `t(M)`, però la podeu fer servir per verificar el vostre resultat.

12. *ColumnaOrdenada.* Dissenyeu una funció que donada una matriu A retorni `TRUE` si aquesta matriu és *columna-ordenada*. Una matriu és *columna-ordenada* si per a cada columna els seus elements estan ordenats de més petit a més gran, recorrent la columna per a cada fila. És a dir: $\forall i, j,\ A_{i,j} \leq A_{i+1,j}$.

13. *Matriu Simètrica.* Feu una funció que rep una matriu A quadrada i torna `TRUE` si i només si és una matriu simètrica. Una matriu és simètrica si i només si $\forall i, j\ A_{i,j} = A_{j,i}$.

14. *ModificaColumna.* Feu una funció que rebi una matriu i la retorni amb la seva última columna cambiada per la suma de les dues primeres columnes.

15. *Sobresurten.* Feu una funció que rebi una matriu i retorni quants dels seus elements són més grans que la mitjana de tots els elements de la matriu.

16. *Quadrat semi-màgic.* (Examen final 13/14) Fes una funció que donada una matriu quadrada de nombres enters més grans que 0, retorni cert si i només si cada una de les files de la matriu sumen 27 (és a dir, la suma dels elements per a cada fila és igual a 27).

    Nota: Aquest exercici té un punt extra si s'assoleix la màxima eficiència

17. *MultMatrius.* Feu una funció que rebi dues matrius i en torni la seva multiplicació. Suposeu que les dimensions de les matrius són tals que es poden multiplicar. Podeu fer servir la instrucció de R A%*%B per comprovar que la vostra funció dóna el resultat correcte.

18. *TipusMatriu.* Dissenya una acció que, donada una matriu A, indiqui per pantalla si aquesta matriu s'ajusta a algun dels casos següents:

    a) La matriu és *Triangular superior*: Una matriu és triangular superior si: $A_{i,j} = 0$ per $j = 1, \ldots, n-1$ i $i = j+1, \ldots, n$.

    b) La matriu és *Diagonal*: Una matriu és diagonal si: $\forall i, j\ A_{i,j} = 0$ sempre que $i \neq j$.

    c) La matriu és *Triangular inferior*: Una matriu és triangular inferior si: $A_{i,j} = 0$ per $i = 1, \ldots, n-1$ i $j = i+1, \ldots, n$.

19. *Diagonal dominant.* (Examen parcial 14/15). Direm que una matriu és **diagonal dominant** quan per a cada fila el valor de la diagonal és més gran o igual que tots els altres elements de la mateixa fila.

    Donada aquesta descripció, fes una funció que donada una matriu retorni TRUE si aquesta matriu és diagonal dominant i FALSE en cas contrari.

20. *Diagonal ben ponderada.* (Examen final 14/15). Donada una matriu M quadrada de dimensió $n \times n$ amb $n \geq 3$ i un element (i, j) a l'interior (que no està a la vora) direm que $M[i, j]$ és *ben ponderat* si

    $M[i, j] \geq M[i-1, j] + M[i+1, j] + M[i, j-1] + M[i, j+1]$.

    Direm que M és *ben ponderada* si existeixen n punts interiors ben ponderats. Feu una funció que, donada la matriu i el nombre n, retorni TRUE si la matriu està ben ponderada i FALSE en cas contrari.

21. *Què cerquem?.* (Examen final 14/15 - única). Tenim la següent funció de matrius ja implementada:

```
ex4 <- function(mat) {
    a = nrow(mat)
    b = ncol(mat)
    t <- c(rep(FALSE,b))
    for (j in 1:b) {
        i <- 1
        while (i <= a && !t[j]) {
```

```
            if (mat[i,j] < 0) {
                t[j] <- TRUE
            }
            i <- i+1
        }
    }
    return (t)
}
```

Si la matriu d'entrada d'aquesta funció és la següent:

```
10  2  0  3 -5 22  3  4
 4 20  6 -7  0  5  2  2
-9 23  7 33 77 13 15  5
10 13 45  0 85  0 77 64
12 15  9 -4 32 34 90 -2
```

quina serà la sortida de la funció?

22. *Rotació de Matrius.* Feu una funció que rep una matriu (no necessàriament quadrada) i torna la `rotació` de la matriu. La rotació de la matriu és una altra matriu amb els mateixos valors però amb la diferència que se li ha aplicat una rotació en el sentit invers de les agulles del rellotge. Per exemple, si tenim la matriu:

```
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

la seva rotació (a l'esquerra) serà:

```
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    1    3    5
```

23. *DiaDia.* Feu una funció que donats dos vectors de la mateixa mida retorni una matriu quadrada que tingui aquests dos vectors com a diagonals superiors i inferiors i a la resta de posicions, 1.

24. *Matrius Superposades.* Feu una funció que rebi 2 matrius (no necessàriament de la mateixa mida) i que en torna la superposició.

    La superposició de dues matrius es defineix com la matriu que obtenim posant una matriu a sobre de l'altra (coincidint en la coordenada $(1, 1)$) tal que en la posició (i, j) hi ha el màxim dels elements que hi ha en la mateixa posició en totes dues matrius. En cas que una matriu no tingui cap element en aquesta posició, assumirem que hi ha un zero.

    Per exemple, donades les matrius:

```
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    1    3    5

     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

la superposició és:

```
     [,1] [,2] [,3]
[1,]    2    4    6
[2,]    3    4    5
[3,]    5    6    0
```

25. *Sudoku.*

    Feu una funció que rep una matriu 9×9 amb nombres naturals del 1 al 9 i torna cert si i només si és un `sudoku`.

    Direm que una matriu és un sudoku si i només si per a cada fila (i per a cada columna) hi ha tots els nombres del 1 al 9 (òbviament, sense repetir). De fet, les regles del `sudoku` també demanen que per a cada quadrant 3×3 també hi ha tots els 9 nombres, però aquesta regla ara no la tindrem en compte per a aquest exercici.

26. *Sudoku (II).*

    Feu una funció que rep una matriu 9×9 amb nombres naturals del 1 al 9 i torna cert si i només si és un `sudoku`.

    En aquest exercici, només verificarem la condició que demana que, per a cada submatriu 3×3 també hi hagi tots els 9 nombres.

27. *VariacionsCa.* En aquest execici farem una "variació" de l'exercici *CaVertical*:

    (a) Dissenyeu una funció `quantes_ca_dia` que donada una matriu com la de l'exercici *CaVertical* calcula quantes vegades apareix la combinació"ca" en diagonal.

    (b) Ara dissenyeu una funció per generar mostres a partir de `quantes_ca_dia`. La diferencia, però, és que volem passar la funció com a paràmetre. Cal que completeu (substituint els punts suspensius pel que calgui) el codi de `mostra_general` (en què `F` és una funció qualsevol que rep una matriu i retorna un enter):

```
mostra_general<-function(s,c,f, F){
   v<-1:s
   for(i in 1:s){
      M<-mat_lletres(c,f)
      v[i]<- .....
   }
   return(v)
}
```

Mireu que tot funcioni com cal, executant per exemple:

```
> source("quantes_ca.R")
> m1<-mostra_general(5, 10, 10, quantes_ca)
> m1
[1] 14 10 10 10 16
> source("quantes_ca_dia.R")
> m2<-mostra_general(5, 10, 10, quantes_ca_dia)
> m2
[1]  9  9 11  8  6
```

28. *PermsRec.*

Dissenyeu una funció `perm` que donat un nombre n emmagatzemi les permutacions dels n primers nombres en una matriu. La funció `perm(n)` ha d'utilitzar (de manera intel·ligent) `perm(n-1)`. En el cas de `perm(3)` obteniu:

```
> source("perm.R")
> P<-perm(3)
> P
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    3    2
[3,]    2    1    3
[4,]    2    3    1
[5,]    3    1    2
[6,]    3    2    1
```