

# Índex

<b>1</b>	<b>Repàs: Funcions, Vectors i Esquemes de Recorregut i Cerca</b>	<b>3</b>
1.1	Exercicis . . . . .	3
<b>2</b>	<b>Matrius</b>	<b>7</b>
2.1	Exercicis . . . . .	7
<b>3</b>	<b>Lists</b>	<b>15</b>
3.1	Exercicis . . . . .	15
<b>4</b>	<b>Data Frames</b>	<b>27</b>
4.1	Exercicis . . . . .	27



# Capítol 1

## Repàs: Funcions, Vectors i Esquemes de Recorregut i Cerca

Objectius: Recordar i enfortir els coneixements adquirits a l'assignatura d'Introducció a la Informàtica, que seran bàsics per a poder continuar amb aquesta assignatura.

Primer de tot:

- Creeu una carpeta (o working directory) que es digui **Repas** on emmagatzemareu tots els scripts que codifiquen les diferents funcions d'aquesta part.
- Recordeu d'anar a aquest directori utilitzant `setwd(".../Repas")`
- Aconsellem guardar cada funció dins el directori **Repas** en un fitxer que es digui igual que la funció seguit d'un punt i R. Per exemple si la funció es diu `mi_fu` enmagatzemeu en un script `mi_fu.R`.

### 1.1 Exercicis

1. Fes una funció que donat un vector d'enters, retorni quin és el primer múltiple de 3 que hi ha a la seqüència, o 0 si no hi ha cap.
2. Disseny una funció que donat un vector d'enters retorni un altre vector d'enters amb els elements del primer en ordre invers.
3. Disseny una funció que donats dos vectors d'enters `v1` i `v2` ordenats, retorni un tercer vector que conté tots els elements de `v1` i `v2` també ordenats (si hi ha repetits es repeteixen també en el vector resultat).
4. Fes una funció que donat un vector d'enters i un enter, retorni TRUE si i només si l'enter donat és divisor d'algun dels elements del vector.

#### 4CAPÍTOL 1. REPÀS: FUNCIONS, VECTORS I ESQUEMES DE RECORREGUT I CERCA

5. Fes una funció que donades dues cadenes de caràcters que representen dues paraules o frases diferents, retorni TRUE si aquestes dues paraules són bifronts o falsos palíndroms, és a dir que es llegeixen igual una del dret i l'altra del revés. Per exemple: roma - amor.
6. Fes una funció que donades dues cadenes de caràcters que representen dues paraules, retorni TRUE si aquestes dues paraules són anagrames. Una paraula és anagrama d'una altra si i només si, els caràcters que tenen totes dues paraules són els mateixos, encara que estiguin ordenats de forma diferent. Per exemple: roma - amor - omar - mora, són anagrames totes 4 entre sí, i arbol - labor - borla, també ho són.
7. Tenim la informació dels primers i segons classificats al Campionat Nacional de Lliga de futbol des dels seus inicis (any 1928) en diversos retalls de paper. Però aquesta informació, encara que cronològicament és correcta: sabem qui va ser el primer campió de la primera lliga, i després apareix en algun moment el campió de la 2a lliga abans del campió de la 3a lliga, però estan barrejats amb els subcampions.

Així doncs l'ordre de les lligues es manté. Primer apareix la informació del campió o subcampió d'una lliga abans de les dades de la següent lliga, però estan barrejats els campions i subcampions.

Feu un programa que, donada una seqüència de les dades de les lligues, en format de nom de l'equip i posició (1 o 2), que acabarà en un equip fictici "FI", s'ha de generar la taula dels campions i subcampions de lliga de forma cronològica però aparellant correctament els campions i els subcampions del mateix any. La informació és correcta i hi ha el mateix nombre de campions i de subcampions.

Per exemple, si l'entrada fos:

```
Barça 1
Ath-Bilbao 2
Madrid 2
At-Madrid 2
València 1
Espanyol 1
At-Madrid 1
Barça 2
FI
```

El resultat seria:

#### 1er i 2on CLASSIFICATS DE LES LLIGUES

```
-----
Campions          Subcampions
-----
Barça              Ath-Bilbao
València           Madrid
Espanyol           At-Madrid
At-Madrid          Barça
```

8. Escriu una acció que, donat un vector d'enters, escriu per pantalla, en ordre invers a com arriben, tots els enters que siguin senars havent-los multiplicat abans pel senar mínim dels nombres del vector. Per exemple:

Si el vector és: 3 5 4 8 7 9 10 12 5 2 6 31 22 3 9 0

El resultat a la sortida ha de ser: 27 9 93 15 27 21 15 9 (*ja que el mínim senar és el 3*)

9. *Quicksort*. (Un clàssic). La funció `qs(v)` ha d'ordenar una taula de nombres, per exemple

```
> v<-c(1,5,2,3,4,7,10,9, 6)
> v
[1] 1 5 2 3 4 7 10 9 6
> qs(v)
[1] 1 2 3 4 5 6 7 9 10
```

Us demanem que completeu el codi:

```
qs <- function(vec){
  if(length(vec) > 1){
    pivot <- vec[1]
    low <- qs(vec[vec < pivot])
    mid <- vec[vec == pivot]
    high <- ....)
    return (c(low, ..., ....))
  }
  else { return (vec) }
}
```

Recordeu que cal emmagatzemar la funció dins `Repas` en un script (que recomanem) es digui `qs.R`

6CAPÍTOL 1. REPÀS: FUNCIONS, VECTORS I ESQUEMES DE RECORREGUT I CERCA

## 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 rebí una matriu i la retorni amb tots els seus elements duplicats.
3. *GeneraDiagonal*. Feu una funció que rebí un vector de dimensió  $n$  i retorni una matriu de  $n \times n$  que tingui aquest vector a la diagonal i a tota la resta zeros.
4. *ElementsDiagonal*. Feu una funció que rebí 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 tindrieu:

```
> 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 `quantas_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 tindrieu:

```
> 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("quantas_ca.R")
> n<-quantas_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 `quantas_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 `quantas_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 rebí 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 rebí una matriu i retorni en un vector dos valors, el primer és l'índex 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 rebí una matriu i la retorni amb la seva última columna canviada per la suma de les dues primeres columnes.
- 15. *Sobresurten.* Feu una funció que rebí 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 rebí 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.* Disseny una acció que, donada una matriu A, indiqui per pantalla si aquesta matriu s'ajusta a algun dels casos següents:

- La matriu és *Triangular superior*: Una matriu és triangular superior si:  $A_{i,j} = 0$  per  $j = 1, \dots, n - 1$  i  $i = j + 1, \dots, n$ .
- La matriu és *Diagonal*: Una matriu és diagonal si:  $\forall i, j \ A_{i,j} = 0$  sempre que  $i \neq j$ .
- La matriu és *Triangular inferior*: Una matriu és triangular inferior si:  $A_{i,j} = 0$  per  $i = 1, \dots, n - 1$  i  $j = i + 1, \dots, n$ .

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 rebí 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  $9 \times 9$  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 (òbviamment, sense repetir). De fet, les regles del **sudoku** també demanen que per a cada quadrat  $3 \times 3$  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  $9 \times 9$  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  $3 \times 3$  també hi hagi tots els 9 nombres.

27. *VariacionsCa*. En aquest exercici farem una “variació” de l’exercici *CaVertical*:

- (a) Dissenyeu una funció `quantas_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 `quantas_ca_dia`. La diferència, 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("quantès_ca.R")
> m1<-mostra_general(5, 10, 10, quantès_ca)
> m1
[1] 14 10 10 10 16
> source("quantès_ca_dia.R")
> m2<-mostra_general(5, 10, 10, quantès_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

```

# Capítol 3

## Lists

Objectiu: Introduir l'estructura `list` de R i totes les seves possibilitats com a estructura de dades que permet agrupar dades de diferents tipus.

### 3.1 Exercicis

1. Feu una funció que llegeixi de teclat un text conformat només per paraules i retorni un list que indica per a cada paraula en quines posicions del text està.
2. Feu una funció que rebí el list generat en l'exercici anterior i retorni un altre list amb el nombre de vegades que cada paraula apareix al text.
3. Feu una funció que rebí un vector de paraules i retorni un list amb el nombre de vegades que cada paraula apareix al vector.
4. *Compta paraules.* (Examen recuperació 13/14). Fes una funció que donat un vector de paraules (`text`), retorni un list que compta els cops que han aparegut en el text (en el vector) les paraules "a", "tot" i "que".

Per exemple, si ens entra el vector de paraules següent:

```
"Per" "a" "un" "cop" "que" "tinc" "a" "la" "meva" "mare" "a" "casa"  
"i" "puc" "disfrutar-la" "deixa'm" "que" "li" "dediqui" "tot" "el"  
"temps" "que" "pugui"
```

Hauríem de retornar el list següent:

```
list (a=3, que=3, tot=1)
```

és a dir:

```
$a
[1] 3
```

```
$que
[1] 3
```

```
$tot
[1] 1
```

5. Feu una funció que rebi un vector de paraules i retorni un list que comp-  
ta tots els anagrames que hi ha al vector (per a cada conjunt de lletres  
tindrem quants anagrames d'aquest conjunt de lletres hi ha al vector).

Una paraula és anagrama d'una altra si i només si, els caràcters que tenen  
totes dues paraules són els mateixos, encara que estiguin ordenats de forma  
diferent. Per exemple: roma - amor - omar - mora, són anagrames totes  
4 entre sí, i arbol - labor - borla, també ho són.

6. Feu ara una funció que rebi un vector de paraules i retorni un list amb tots  
els anagrames que hi ha al vector (per a cada conjunt de lletres tindrem  
els anagrames d'aquest conjunt de lletres que hi ha al vector).
7. Feu una funció que donat un list que conté un diccionari on per a cada  
paraula tenim la seva descripció, retorni un list modificat on s'han eliminat  
totes les entrades del diccionari (paraules + descripció) que la paraula té  
menys de 5 lletres (podeu usar la funció `nchar(string)`).
8. *Diccionari nou.* (Examen parcial 15/16). Disposem d'un diccionari que  
se'ns ha quedat antiquat, i volem canviar les seves definicions per les més  
noves. Per a això, comparem el nostre diccionari amb un de nou però  
només volem modificar les paraules que ja teníem, no volem afegir-ne  
més.

Així doncs, es demana que feu una funció que donats dos lists (L1 i L2) on  
L1 és el nostre diccionari i L2 és el nou, es modifiqui el list L1 (el nostre  
diccionari) amb les definicions noves (les de L2) però no afegirem aquelles  
paraules que només estan en L2 i no en L1.

Per exemple:

```
L1 <- list (casa = "lloc on viure",
           lloguer = "diners a pagar per un servei",
           mitjana = "valor promig",
           moneda = "tros de metall amb valor, cada país en té una diferent",
           sabata = "tros de material per a cobrir el peu")
L2 <- list (casa = "vivenda, habitatge, lloc on viure",
           llibre = "conjunt de pàgines per llegir",
```



```
moneda = "tros de metall amb valor",
sabata = "calçat que cobreix el peu",
videojoc = "joc electrònic")
```

el resultat ha de ser:

```
L1 <- list (casa = "vivenda, habitatge, lloc on viure",
           lloguer = "diners a pagar per un servei",
           mitjana = "valor promig",
           moneda = "tros de metall amb valor",
           sabata = "calçat que cobreix el peu")
```

9. Feu una funció que donat un list que conté un diccionari on per a cada paraula tenim la seva descripció, i donat també un vector de paraules, modifiqui el list de manera que si les paraules del vector estaven al list les elimini. Considereu que el diccionari no té paraules repetides.
10. *Sou anual.* (Examen parcial 14/15). Tenim dos lists, un amb informació dels mesos treballats a l'any per un treballador (L1) i l'altre amb el sou per mes que aquests treballadors cobren (L2).

Es demana fer una funció que a partir dels dos lists L1 i L2 retorni un nou list L3 amb el sou guanyat en tot l'any per cada treballador.

Per exemple:

```
L1 <- list (maria=12, joan=10, emma=8)
L2 <- list(joan=800, maria=1000, emma=1200, kim=600)
```

el resultat ha de ser:

```
list (joan=8000, maria=12000, emma=9600, kim=0)
```

és a dir:

```
$joan
[1] 8000
```

```
$maria
[1] 12000
```

```
$emma
[1] 9600
```

```
$kim
[1] 0
```

Fixeu-vos que pot ser que un treballador no estigui en el list L1 i sí en el L2 (com el Kim), i això vol dir que està en nòmina però no ha treballat aquest any (resultat, cobra 0 euros aquest any), però **no us trobareu un treballador que estigui en L1 i no en L2.**

11. *Paguem hipoteca.* (Examen final 15/16). El govern de la Generalitat vol controlar el pagament dels interessos de les hipoteques dels ciutadans de Catalunya, de manera que aquells ciutadans amb més ingressos paguin més interessos que els que en tenen menys. També hi ha un conjunt de ciutadans que no han de pagar res de res (0% d'interessos).

La Generalitat emmagatzema les dades en dos `lists`. En el primer list, L1, per a cada ciutadà (del que tenim el nom com a nom del camp del list) tenim la quantitat de la hipoteca que ha de pagar cada mes. En el segon list, L2, tenim per a cada ciutadà el percentatge d'interessos que li toca pagar de la seva hipoteca. Aquells ciutadans que no han de pagar res dels interessos de la hipoteca, tindran associat un valor 0 en el list L2.

Es demana que implementeu una funció que a partir dels dos lists (L1 i L2), calculi i retorni un nou list, L3, on per a cada nom s'emmagatzema la quantitat total d'interès que ha de pagar. Els ciutadans que no han de pagar res no poden aparèixer a L3.

Per exemple, si els dos lists d'entrada són:

```
L1 <- list (Joan=1000,Pep=1300,Maria=500,Isabel=800,Miquel=1200,
           Nuria=1500,Marc=800,Axel=480,Jana=600)
L2 <- list (Maria=20,Axel=0,Pep=10,Miquel=5,Jana=0,Joan=10,Marc=5,
           Isabel=0,Nuria=10)
```

El list de sortida hauria de ser:

```
L3 <- list (Maria=100,Pep=130,Miquel=60,Joan=100,Marc=40,Nuria=150)
```

Podeu considerar que sempre hi haurà els mateixos noms a L1 i a L2.

12. *Notes definitives.* (Examen recuperació 15/16). Donats dos lists amb les notes dels estudiants de l'assignatura A, `NotesACA` (Notes d'Avaluació Continua) i `NotesEFA` (Notes de l'Examen Final), escriu una funció en R que generi un nou list anomenat `NotesDA` amb les notes definitives dels estudiants d'A, seguint la següent fórmula.

$$NDA = \max(70\% EFA, 60\% ACA + 40\% EFA)$$

Si un estudiant no té nota d'avaluació continua o no es va presentar a l'examen final, no hi apareix al list corresponent.

Per exemple, si els dos lists d'entrada són:

```
NotesACA <- list (Joan=7,Pep=3,Maria=5,Isabel=8,Miquel=4,
                 Nuria=5,Marc=8,Axel=5,Jana=6)
NotesEFA <- list (Maria=2,Axel=0,Pep=10,Miquel=5,Jana=3,
                 Joan=6,Marc=6,Isabel=6,Nuria=9,Didac=7)
```

El list de sortida hauria de ser:

```
NotesDA <- list (Joan=6.6,Pep=7,Maria=3.8,Isabel=7.2,Miquel=4.4,
                 Nuria=6.6,Marc=7.2,Axel=3,Jana=4.8,Didac=4.9)
```

13. Feu una acció que rebi un list construït amb dos camps, un amb un vector amb les notes dels parcials d'una assignatura, i l'altre amb un vector amb les ponderacions de cada un, i escrigui per pantalla la informació de si l'alumne està suspès o aprovat (una cadena de caràcters que val "Suspès" o "Aprovat") i amb quina nota.
14. Feu una funció que rebi un list amb tres camps de tipus vector amb les notes dels parcials de tres assignatures respectivament i retorni un list amb un camp amb les mitjanes de les tres assignatures i un segon camp amb el nombre de notes de cada assignatura que estan per sobre de la mitjana de l'assignatura. *Nota: Podeu usar la funció `mean(v)` que calcula la mitjana d'un vector*
15. Sigui un list `eleccions` amb els resultats de les 5 últimes eleccions al Parlament de Catalunya on el primer camp és un vector amb els anys i els altres camps són vectors amb els resultats obtinguts per cadascun dels 5 primers partits (per aquest ordre, CIU, ERC, PSC, PP i ICV). Sempre tenint com a entrada el list `eleccions`, dissenyeu:
- Una funció que a partir del list `eleccions` retorni quin any el PP va obtenir el seu pitjor resultat i quin resultat va ser.
  - Una funció que a partir del list `eleccions` retorni quin any la diferència d'escons entre CIU i PSC va ser més gran i quina va ser aquesta diferència.
  - Una funció que a partir del list `eleccions` retorni quin partit va obtenir més escons l'any 2003 i quants escons va obtenir.
16. *Guanyats i pèrdues.* (Examen final 13/14). Fes una funció que donats dos lists, `guanyats` i `perdues`, tals que tots dos tenen el format:

clau: nom  
 valor: quantitat de diners que ha guanyat/perdut

(els valors de `guanyats` són tots positius mentre que els valors de `perdues` són tots negatius) modifiqui el list `guanyats` fent la "fusió" de tots dos lists, incloent tots els elements de `guanyats` i els de `perdues` i en el cas d'elements que estan en tots dos fent la suma dels dos valors (`guanyats + perdues`).

Per exemple:

```
guanyats <- list (pepet=3000, maria=2000, joan=1000)
perdues <- list(joan=-500, maria=-1000, kim=-400)
```

el resultat ha de ser:

```
list (pepet=3000, maria=1000, joan=500, kim=-400)
```

és a dir:

```
$pepet
[1] 3000
```

```
$maria
[1] 1000
```

```
$joan
[1] 500
```

```
$kim
[1] -400
```

17. Sigui un list `ed_al` amb dues components de tipus vector que contenen informació sobre l'edat (primera component) i l'alçada (segona component) de 20 persones. Sempre rebent com a entrada el list `ed_al`, dissenyeu les següents funcions:
- Una funció que donat el list `ed_al` retorni quina és l'alçada de la persona més alta.
  - Una funció que donat el list `ed_al` retorni quina edat té la persona més baixa.
  - Una funció que donat el list `ed_al` retorni quantes persones tenen edats compreses entre 30 i 40 anys (inclosos).

18. *Modificant lists.* (Examen final 13/14). Tenim la següent funció de lists ja implementada:

```
ex2 <- function (L1, L2)
{
  for (n in names(L2)) {
    if (!is.null(L1[[n]])) {
      L2[[n]] <- L2[[n]] * L1[[n]] / 100
    }
    else {
      L2[[n]] <- NULL
    }
  }
  return (L2)
}
```

Si els dos lists d'entrada d'aquesta funció són els següents:

```
L1 <- list (Maria=20,Nuria=10,Pep=10,Miquel=5,Jana=15,Joan=15,Marc=10)
L2 <- list (Joan=500,Pep=340,Maria=250,Miquel=700,Nuria=560,Isabel=670,
           Marc=930,Axel=480)
```

quina serà la sortida de la funció?

19. *Fusió*. (Examen parcial 13/14). Fes una funció que donats dos lists, L1 i L2, tals que tots dos tenen el format:

```
clau: nom
valor: nombre de vegades que ha anat al cinema
```

modifiqui el list L1 fent la “fusió” de tots dos lists, incloent tots els elements de L1 i L2 i en el cas d’elements que estan en tots dos fent la suma dels dos valors.

Per exemple:

```
L1 <- list (pepet=3, maria=2, joan=1)
L2 <- list(joan=5, maria=1, kim=4)
```

el resultat ha de ser:

```
list (pepet=3, maria=3, joan=6, kim=4)
```

és a dir:

```
$pepet
[1] 3

$maria
[1] 3

$joan
[1] 6

$kim
[1] 4
```

20. *Aprovats*. (Examen parcial 13/14). Tenim un list on cada component té emmagatzemada informació d’un estudiant de Programació. Per a cada estudiant tenim el seu nom, el niub, l’edat, si té aprovada IntroInf, i la nota final de Programació.

El list està construït de la següent manera:

```
# Creem els estudiants

est1 <- list(nom="Joan Pujol", niub=23543, edat=19,
            introinf=TRUE, nota=6.8)
est2 <- list(nom="Albert Adell", niub=33543, edat=18,
            introinf=FALSE, nota=3.5)
est3 <- list(nom="Pep Borrell", niub=56473, edat=19,
            introinf=TRUE, nota=9.8)
```

```

est4 <- list(nom="Cesc Ferrer", niub=67775, edat=20,
            introinf=TRUE, nota=3.2)
... etc ...

# Creem el list d'estudiants
notesProg <- list(estudiant1=est1, estudiant2=est2, estudiant3=est3,
                 estudiant4=est4, ... etc ...)

```

Es demana fer una funció que a partir del list `notesProg` retorni un vector amb tots els noms d'aquells estudiants que tenen aprovada l'assignatura de Programació.

21. Tenim un list on cada component té emmagatzemada informació d'un client. Per a cada client, el nombre d'unitats que ha comprat d'uns certs productes i el preu per unitat de cada producte.

El list està construït de la següent manera:

```

# Creem els clients

client1<-list(unitats=c(3, 1, 1, 2),
             pxun=c(1.25, 2.55, 2.5, 3))

client2<-list(unitats=c(2, 4, 1),
             pxun=c(2.25, 5.59, 3.05))

client3<-list(unitats=c(1, 1, 1, 1, 1),
             pxun=c(10.25, 3.25, 1.09, 5.99, 0.35))

client4<-list(unitats=c(10, 2, 1, 3),
             pxun=c(0.99, 6.45, 3.99, 1.99))

# Creem la llista de clients
clients<- list(client1=client1, client2=client2, client3=client3,
              client4=client4)

```

Tenint com a entrada en tots els casos el list `clients` es demana:

- (a) Feu una funció que retorni un vector amb el preu total que ha de pagar cada client. Utilitzeu-la per obtenir quin és el client que ha de pagar més diners.
  - (b) Feu una funció que retorni quin és el client que ha comprat més productes i quants productes ha comprat.
22. Tenim un list on la primera component és un vector amb els noms d'uns certs aliments i la segona un vector amb el preu per quilo de cadascun. Tenim un altre list on cada component és una pizza i per a cada pizza,

que també és un list, tenim un vector amb els aliments que es necessiten per a fer-la i un vectos amb la quantitat (en quilos) de cadascun.

L'estructura de dades està construïda de la següent manera:

```
# Aliments i preu per quilo
aliments <- c ("Tomàquet", "Mozzarella", "Pernil dolç", "Gambes",
              "Xampinyons", "Carxofes")
preuxkg <- c (0.99, 0.40, 7.99, 50.5, 3.65, 2.25)

# Llista d'aliments i preu per quilo
preus <- list (aliments=aliments, preu=preuxkg)

# Pizzes
al_mar <- c ("Tomàquet", "Mozzarella")
quant_mar <- c (0.5, 0.2)
mar <- list (pizza = al_mar, quant = quant_mar)

al_per <- c ("Tomàquet", "Mozzarella", "Pernil dolç")
quant_per <- c (0.3, 0.15, 0.15)
per <- list (pizza = al_per, quant = quant_per)

al_est <- c ("Tomàquet", "Mozzarella", "Pernil dolç", "Xampinyons",
            "Gambes", "Carxofes")
quant_est <- c (0.3, 0.2, 0.1, 0.1, 0.05, 0.075)
est <- list(pizza = al_est, quant = quant_est)

al_per_xam <- c ("Tomàquet", "Mozzarella", "Pernil dolç", "Xampinyons")
quant_per_xam <- c (0.35, 0.2, 0.15, 0.15)
per_xam <- list (pizza = al_per_xam, quant = quant_per_xam)

# Llista de pizzas
pizzas <- list (Margarita = mar, Pernil = per, Estacions = est,
               Pernil_Xampinyons = per_xam)
```

- (a) Feu una funció que rebí el list dels aliments i els preus i retorni el nom del producte més car. Quant costarien 300 grams d'aquest producte?
- (b) Feu una funció que rebí el list de les pizzas i el de preus i retorni un list amb els noms de les pizzas i el preu de cada una d'elles tenint en compte que al preu dels ingredients necessaris per fer la pizza se li han de sumar 9 euros de mà d'obra.
- (c) Feu una funció que rebí el list de les pizzas i retorni el mateix list havent-li afegit una nova pizza que portarà tomàquet, mozzarella, xampinyons i gambes amb quantitats 0.3, 0.2, 0.2, 0.15 respectivament.

23. *Més amb lists.* (Examen recuperació 14/15). Tenim la següent funció de lists ja implementada:

```
ex2 <- function (L1, L2, L3)
{
  for (n in names(L1)) {
    if (!is.null(L2[[n]])) {
      L1[[n]] <- L1[[n]] * L2[[n]]
      if (!is.null(L3[[n]])) {
        L1[[n]] <- L1[[n]] + L3[[n]]
      }
    }
    else {
      L1[[n]] <- NULL
    }
  }
  return (L1)
}
```

Si els tres lists d'entrada d'aquesta funció són els següents:

```
L1 <- list (Joan=500, Pep=300, Maria=250, Miquel=700, Nuria=500, Isabel=670,
           Marc=300, Axel=420)
L2 <- list (Axel=2, Maria=2, Nuria=3, Pep=1.5, Jana=2, Joan=10, Marc=3)
L3 <- list (Pep=10, Miquel=200, Joan=50, Nuria=150, Kim=130, Marc=90)
```

quina serà la sortida de la funció?

24. Sigui un list de països (clau) i el nom del continent on són. Per exemple:

```
continents <- list(Catalunya="Europa",Itàlia="Europa",
                  Xile="Amèrica",Austràlia="Oceania",
                  Mèxic="Amèrica",Suècia="Europa",
                  Argentina="Amèrica",Xina="Àsia",
                  Paraguai="Amèrica")
```

i un list de països (clau) i el seu PIB:

```
països_pib <- list(Catalunya=10,Itàlia=5,Xile=7,Austràlia=14)
```

Feu una funció que rebí aquests dos lists, i que torni un altre list que tingui, com a clau, els continents dels països del list de països i pibs, i que tingui, com a dada, almenys, la mitjana dels pibs. Per exemple:

```
[1] "Països i Pibs ====="
$Mèxic
[1] 3

$Xile
[1] 2
```



\$Xina  
[1] 4

\$Itàlia  
[1] 2

\$Paraguai  
[1] 1

\$Austràlia  
[1] 3

\$Catalunya  
[1] 5

[1] "Resultat ====="  
\$Amèrica\$mitjana  
[1] 2

\$Àsia\$mitjana  
[1] 4

\$Europa\$mitjana  
[1] 3.5

\$Oceania\$mitjana  
[1] 3



# Capítol 4

## Data Frames

Objectius: Introduir l'estructura `dataframe` de R i el format i la seva utilització en R.

### 4.1 Exercicis

1. Dissenyeu una funció que rebí un `dataframe` i el nom d'una de les seves variables i retorni la mitjana d'aquesta variable (amb arrodoniment a 4 xifres decimals).
2. Dissenyeu una funció que rebí un `dataframe` i el nom d'una de les seves variables i retorni la mediana d'aquesta variable.
3. Tenim un `dataframe` amb dades dels estudiants de Programació del Grau d'Estadística. El `dataframe` `D` té, per a cada estudiant, les dades següents:

```
D: nom niub nota_prog nota_mitjana
```

Fes una funció que, donat aquest `dataframe` `D` i la nota mitjana històrica de programació a l'escola `MeanProg`, retorni un vector amb els noms dels estudiants que compleixin que la seva nota de programació sigui més gran que la seva pròpia mitjana i també que la mitjana històrica de programació.

4. Dissenyeu una funció que rebí dues variables d'un `dataframe` i en retorni la seva correlació segons la fórmula:

$$\frac{\sum((X_i - \bar{X}) * (Y_i - \bar{Y}))}{\sqrt{\sum(X_i - \bar{X}) * \sum(Y_i - \bar{Y})}}$$

Podeu usar la funció `mean()` de R.

5. Disposem de dades sobre els matrimonis entre dones, entre homes i heterosexuals en cada mes de l'any 2012 (fitxer `matrimonis.txt`). Es demana:

- (a) Dissenyeu una funció que retorni si en algun mes el nombre de matrimonis entre dones va ser més gran que el de matrimonis entre homes.
- (b) Dissenyeu una funció que retorni quin mes la proporció de matrimonis entre parelles del mateix sexe va ser més gran respecte el total de matrimonis.
- (c) Dissenyeu una funció que retorni quantes persones es van casar el mes que es va casar més gent.
6. Tenim un data frame amb els resultats de les eleccions al Parlament de Catalunya des del 1999 al 2012 (5 eleccions) on la primera columna conté els anys en que hi va haver eleccions i les columnes següents són els resultats obtinguts pels 5 primers partits (per aquest ordre, CIU, ERC, PSC, PP i ICV) en cada un dels anys. Sempre rebent com a entrada el dataframe `eleccions`, dissenyeu:
- (a) Una funció que retorni quin any el PP va obtenir el seu pitjor resultat i quin resultat va ser. Podeu retornar les dues dades en un vector.
- (b) Una funció que retorni quin any la diferència d'escons entre CIU i PSC va ser més gran i quina va ser aquesta diferència. Podeu retornar les dues dades en un vector.
- (c) Una funció que retorni quin partit va obtenir més escons l'any 2003 i quants escons van ser. Podeu retornar les dues dades en un vector.
- (d) Una funció que retorni un nou dataframe amb els anys de les eleccions i el partit que va obtenir més escons en cada una.
7. *Segons mitjana (Examen de recuperació 13/14)*. Feu una funció que donats un data frame `dades` i el nom de dues variables (columnes) `var1` i `var2` modifiqui el data frame `dades` afegint una columna `Result` de forma que el valor corresponent en aquesta nova columna es calculi de la següent manera:

Si en la fila corresponent el valor de `var1` és superior a la mitjana d'aquesta mateixa variable (mitjana de la columna `var1`), el resultat serà el valor de `var1` + el valor de `var2`. Si per contra, en la fila corresponent el valor de `var1` no és superior a la mitjana d'aquesta mateixa variable (mitjana de la columna `var1`), el resultat serà el valor de `var1` - el valor de `var2`.

Podeu usar qualsevol de les funcions que ofereix R que necessiteu.

Per exemple, si tenim el data frame

	v1	v2	v3	v4
1	1	2	3	4
2	2	2	3	2
3	2	4	2	3
4	4	2	4	2

i les variables són "v4" i "v1", la mitjana de la columna amb nom "v4" és 2.75 i per tant el data frame resultant ha de ser:

	v1	v2	v3	v4	Result
1	1	2	3	4	5
2	2	2	3	2	0
3	2	4	2	3	5
4	4	2	4	2	-2

8. *Sobre mitjana (Examen final 13/14)*. Feu una funció que donats un data frame `dades` i el nom d'una variable (columna) `var` retoni un altra data frame que contingui totes les files de `dades` tals que el valor corresponent a la variable `var` sigui superior a la mitjana d'aquesta mateixa variable (columna).

Podeu usar qualsevol de les funcions que ofereix R que necessiteu.

Per exemple, si tenim el data frame

	v1	v2	v3	v4
1	1	2	3	4
2	2	2	3	2
3	2	4	2	3
4	4	2	4	2

i la variable és "v4", la mitjana de la columna amb nom "v4" és 2.75 i per tant el data frame resultant ha de ser:

	v1	v2	v3	v4
1	1	2	3	4
3	2	4	2	3

9. El dataframe `trees` conté dades sobre l'alçada, el diàmetre i el volum de 31 cirerers talats. Rebut com a entrada `trees`,

	Girth	Height	Volume
1	8.3	70	10.3
2	8.6	65	10.3
3	8.8	63	10.2
	...		
30	18.0	80	51.0
31	20.6	87	77.0

(a) Feu una funció que retorni quants arbres tenen alçada (Height) més gran que 85 i diàmetre (Girth) més gran que 10.

(b) Afegiu al dataframe les dades de 2 arbres nous amb variables:

```
# Arbre 1: Girth=9.2, Height=80 i Volume=11.2
# Arbre 2: Girth=14, Height=69 i Volume=33.6
```

(c) Feu una funció que retorni quin volum té l'arbre més alt. Utilitzeu-la per saber si és el volum més alt de tots els arbres.

10. *Analfabetisme vs. salari (Examen final única 13/14)*. Fes una funció que donat el data frame `estats`, descrit a continuació, retorni cert si i només si l'estat que té el mínim índex d'analfabetisme (variable `Analfabetisme`) és el mateix que té el màxim salari (variable `Salari`).

Podeu considerar que no hi han valors iguals.

El data frame `estats` té les dades en el format següent:

	Poblacio	Salari	Analfabetisme	Esp_vida	Regio
Alabama	3615	3624	2.1	69.05	South
Alaska	365	6315	1.5	69.31	West
.....					
Wisconsin	4589	4468	0.7	72.48	North_Central
Wyoming	376	4566	0.6	70.29	West

11. El dataframe `Traffic` conté dades d'accidents de trànsit en els anys 1961 i 1962 a Suècia.

L'heu de carregar de la llibreria `MASS` amb el codi següent:

```
library(MASS)
data(Traffic)
```

I les dades són com segueix:

	year	day	limit	y
1	1961	1	no	9
2	1961	2	no	11
3	1961	3	no	9
...				
183	1962	91	yes	15
184	1962	92	yes	9

- (a) Feu una funció que rebí el dataframe `Traffic` i retorni la diferència entre la mitjana dels dies en què hi ha límit de velocitat i la dels dies que no.

- (b) Feu una funció que rebi l'any (1961 o 1962) i retorni el màxim d'accidents d'aquest any i si el dia en que es va donar aquest màxim hi havia límit de velocitat o no.

12. Disposem de dos dataframes que contenen informació sobre els diferents estats dels Estats Units. Cada fila correspon a un estat i cada columna a una variable. Per més informació de les dades podeu consultar la help de R mitjançant

```
?state
```

Després de llegir les dades

```
estats1 <- read.table("estats1.txt", header=T, stringsAsFactors=FALSE)
estats2 <- read.table("estats2.txt", header=T, stringsAsFactors=FALSE)
```

es demana

- (a) Dissenyeu una funció que rebi com a entrada un dels dataframes i retorni quina és la regió de l'estat amb l'àrea més gran.
- (b) Amb la informació obtinguda en l'apartat anterior (i, per tant, utilitzant la funció que heu fet), dissenyeu una funció que rebi els dos dataframes i calculi la mitjana de població per als estats de la regió obtinguda en l'apartat anterior.

13. *Any estrena (Examen final 15/16).*

Tenim un **data frame** on s'emmagatzema informació de pel·lícules de manera que, per a cada títol de pel·lícula, tenim la durada en minuts i l'any en que es va estrenar:

```
D1: titol, durada, any_estrena
```

Implementa una funció que donat un data frame D1 amb les dades de les pel·lícules, retorni l'any en que es van estrenar més pel·lícules. En cas que hi hagi més d'un any que compleixin la condició, retornarà el primer en què es va complir (per exemple, si es compleix en 2008 i 2010, ha de retornar 2008).

Per exemple, si tenim el data frame:

D1:	titol	durada	any_estrena
1	pel11	120	2012
2	pel12	100	2010
3	pel13	90	2010
4	pel14	75	2009
5	pel15	135	2011

la funció hauria de retornar:

```
2010
```

14. El dataset `co2` conté la concentració de CO2 mesurada mensualment des de l'any 1959 fins al 1997. Es vol comprobar si les distribucions anuals es mantenen o no constants al llarg dels anys. És a dir efectuar un contrast d'homogeneïtat:

$H_0$ : Les  $r$  poblacions tenen una distribució en comú

$H_1$ : Les  $r$  poblacions no tenen una distribució en comú

Sent  $r$  els 38 anys.

Cal dissenyar una funció que donat un `data.frame`, calculi l'estadístic  $\chi_{exp}^2$  i el compari amb el valor crític de referència. Ha de retornar TRUE en cas que  $H_0$  sigui rebutjada. Considereu un nivell de significació del 5%.

Podeu carregar les dades executant `data("co2")`. Seguidament, executeu la següent comanda per a transformar les dades de classe `ts` a `matrix`:

```
co2=tapply(co2, list(year = floor(time(co2)), month = month.abb[cycle(co2)]), c)
```

Sent  $r$  i  $c$  el nombre de files i columnes del `data.frame`. I  $n_{ij}$  l'element de la fila  $i$  i la columna  $j$ . L'estadístic  $\chi_{exp}^2$  ve definit:

$$\chi_{exp}^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(n_{ij} - \frac{n_{i\bullet} n_{\bullet j}}{n})^2}{\frac{n_{i\bullet} n_{\bullet j}}{n}} \sim \chi_{(r-1)(c-1)}^2$$

On  $n_{i\bullet} = \sum_{j=1}^c n_{ij}$  i  $n_{\bullet j} = \sum_{i=1}^r n_{ij}$ .

Rebutgeu  $H_0$  si:  $\chi_{exp}^2 > \chi_{(r-1)(c-1); 1-\alpha}^2$ .

Per a calcular el valor crític  $\chi_{(r-1)(c-1); 1-\alpha}^2$  amb R, podeu utilitzar la següent instrucció:

```
qchisq(1-alpha, df=(nrow(co2)-1)*(ncol(co2)-1))
```

15. Per tal d'estudiar l'efecte que causa el tabac sobre l'agregació de les plaquetes en la sang, Levine (1973) va dissenyar un estudi per mesurar el grau al qual les plaquetes es van agregar en individus abans i després de fumar. Les plaquetes estan implicades en la formació de coàguls de sang, i se sap que els fumadors pateixen més sovint de desordres que impliquen la seva formació que no pas les persones no fumadores.

Es van fer dos estudis de l'agregació de plaquetes, amb els mateixos individus, abans i després de fumar tabac i abans i després de fumar enciam. Les dades de totes dues taules les teniu als arxius `tabac.txt` i `enciam.txt` respectivament.

Les dades d'aquests dos arxius tenen el format següent (tots dos iguals):



Persona	Abans	Despres	Diferencia
1	25	24	-1
2	25	30	5
...	...	...	...
10	60	62	2
11	68	70	2

Si denotem per  $X$  i  $Y$  les variables aleatòries que mesuren el percentatge màxim d'agregació de plaquetes a la sang abans i després de fumar un cigarret d'enciam, respectivament columnes **Abans** i **Despres** en l'arxiu, es demana

- Feu una funció que donat un dels data frames anteriors, comprovi si la mitjana de la diferència és la diferència de les mitjanes de les variables  $X$  i  $Y$  i retorni un list amb la següent informació: Un booleà que serà cert o fals responent a la pregunta anterior i la mitjana de la diferència que s'ha calculat. No podeu usar la funció `mean` de R.
- Feu una funció que donat un dels data frames anteriors, calculi i retorni les variàncies de  $X$ , de  $Y$  i de la diferència. Podeu retornar els tres valors en un únic vector.

Recordeu que la fórmula per calcular la variància d'una variable  $X$  és

$$Var(X) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

Es recomana que definiu una funció que us retorni la variància d'una certa variable donada (variància d'un vector de valors).

- Feu una funció que donat un dels data frames anteriors i usant la crida a la funció de l'apartat b), comprovi la següent fórmula i retorni cert o fals depenent de si la fórmula es compleix o no.

$$Var(X - Y) = Var(X) + Var(Y) - 2Cov(X, Y)$$

Recordeu que la fórmula per calcular la covariància entre dues variables  $X$  i  $Y$  és

$$Cov(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

- Feu una funció que donat un dels data frames anteriors i usant les funcions anteriors que us siguin necessàries, retorni el valor de la mesura de discrepància.

$$d = \frac{\bar{D}}{S_D/\sqrt{n}}$$

$$S_D = \sqrt{Var(D)}$$

16. Basant-nos en els mateixos arxius que l'exercici anterior, `tabac` i `enciam`, es demana fer una funció que donats tots dos data frames i usant totes les funcions necessàries de l'exercici anterior, calculi i retorni el valor de la mesura de discrepància entre la diferència d'abans i després de fumar tabac i la diferència d'abans i després de fumar enciam (cal calcular la diferència d'interès  $D = D_{Tabac} - D_{Enciam}$ ).

17. *Què fem?* (Examen final 15/16).

La funció `queFa` `<- function (D1, D2)` està implementada com:

```
queFa <- function(D1,D2){
  DM <- merge (D1, D2)
  a <- vector()
  for (i in 1:nrow(DM)){
    a <- c(a, DM[i,"ingressos"]-DM[i,"despeses"])
  }
  DB <- cbind (DM, S=a)
  DB <- DB[order(DB$nom),]
  return(DB$S)
}
```

El data frame `D1` conté el camp `"nom"` i el camp `"ingressos"` mentre que el data frame `D2` conté el camp `"nom"` i el camp `"despeses"`.

Si els data frames d'entrada de la funció són:

```
D1 <- data.frame (nom=c("Pep","Jana","Robert","Maria","Anna","Albert","Kim"),
  ingressos=c(5500,7000,7200,6400,6700,7500,6000),stringsAsFactors=FALSE)
```

```
D2 <- data.frame (nom=c("Jana","Robert","Maria","Albert","Kim","Pep","Isabel"),
  ingressos=c(5500,6500,6000,8000,5500,5000,4700),stringsAsFactors=FALSE)
```

quina serà la sortida de la funció `queFa (D1, D2)`?

18. *Despeses de la compra* (Examen de recuperació 14/15) Tenim dos dataframes amb informació dels productes que ha comprat cada persona i el preu d'aquests productes. En el primer dataframe `D1` tenim informació de la persona i el que ha comprat (quantitat):

```
D1:  nom producte Kg
```

I en el segon dataframe `D2` tenim la informació del que costa cada producte per Kg:

```
D2:  producte preuKg
```

Es demana implementar una funció que, donats els dos dataframes `D1` i `D2` generi un tercer dataframe `D3` que contingui la despesa de la compra, de manera que per a cada persona tindrem la suma de la despesa de tots els productes que ha comprat. Aquest dataframe resultat ha d'estar ordenat per nom.

D3: nom despesa

Per exemple, si tenim:

D1:	nom	producte	Kg	D2:	producte	preuKg
1	Pep	pomes	5	1	pomes	2.5
2	Kim	peres	3	2	peres	3.0
3	Pep	peres	1.5	3	platans	4.5
4	Maria	pomes	3	4	cireres	6.0
5	Maria	platans	2	5	figues	5.5
6	Albert	cireres	3.5			
7	Pep	figues	5			

El resultat hauria de ser:

D3:	nom	despesa
1	Albert	21
2	Kim	9
3	Maria	16.5
4	Pep	44.5

19. Tenim un dataframe amb dades sobre els desplaçaments d'estudiants universitaris per comarques (fitxer desplaçaments\_uni.txt). Tenint en compte que el total d'universitaris atrets per una comarca és la suma dels interns, dels des de fora i dels que no consta i que el total d'estudiants generats són els interns i els que marxen a fora, es demana
- Feu una funció que retorni quina comarca atreu més estudiants.
  - Feu una funció que retorni quina comarca genera menys estudiants.
  - Dissenyeu una funció que retorni quants desplaçaments interns té la comarca que genera més estudiants.
  - Dissenyeu una funció que retorni quina és la diferència més gran entre universitaris atrets i generats.
  - Feu una funció que retorni quantes comarques hi ha que no atreguin cap estudiant.
20. *Guanyys esportius (Examen final 14/15)*. Tenim dos dataframes amb informació dels tornejos guanyats enguany. En el primer dataframe D1 tenim informació de l'equip que ha guanyat cada torneig:

D1: equip torneig

I en el segon dataframe D2 tenim la informació dels diners que es guanyen amb cada torneig (el campió del torneig):

D2: torneig recompensa

Es demana implementar una funció que, donats els dos dataframes D1 i D2 generi un tercer dataframe D3 que contingui la classificació dels equips pels seus guanys, de manera que per a cada equip tindrem la suma de tots els seus guanys (segons els diferents tornejos que hagi guanyat) i a més els tindrem ordenats de més guanys a menys guanys.

D3: equip guanys

Per exemple, si tenim:

D1:	equip	torneig	D2:	torneig	recompensa
1	Barça	Lliga-1a	1	Lliga-1a	1500
2	Madrid	Lliga-ACB	2	Lliga-ACB	450
3	Girona	Lliga-2a	3	CopaRey	500
4	Girona	Gamper	4	Lliga-2a	150
5	Barça	CopaRey	5	Champions	3000
6	Barça	Champions	6	FinalFour	1000
7	Juventud	FinalFour	7	Gamper	200

El resultat hauria de ser:

D3:	equip	guanys
1	Barça	5000
2	Juventud	1000
3	Madrid	450
4	Girona	350

## 21. Examen de recuperació 15/16

Tenim dos dataframes amb informació dels ingressos i despeses d'un conjunt d'empreses. Els ingressos estan en el primer dataframe (D1) i aquest està ordenat per nom de l'empresa:

D1: NIF nom ingressos

les dades dels ingressos estan en milers d'Euros i poden haver diferents ingressos per a una mateixa empresa.

I en el segon dataframe D2 tenim la informació de les despeses de cada empresa, però en aquest data frame només tenim el NIF de l'empresa no el nom. En el cas de les despeses només hi ha una per empresa. Aquest data frame està ordenat per NIF:

D2: NIF despeses

les dades de les despeses també estan en milers d'Euros.

Es demana implementar una funció que, donats els dos dataframes D1 i D2 generi un tercer dataframe D3 que contingui només el conjunt d'empreses que han hagut de cotitzar a hisenda. Les empreses que han hagut de cotitzar a hisenda són aquelles que han tingut guanys superiors als 500 mil Euros (500 en milers d'Euros). En aquest tercer dataframe (D3), a més de les dades de l'empresa (NIF i nom), han d'aparèixer les columnes **ingressos**, **despeses** i **guanys**, on aquesta última es calcula fent la diferència entre els ingressos totals (que són la suma dels diferents ingressos de l'empresa) i les despeses.

Aquest dataframe resultat ha d'estar ordenat per NIF.

D3: NIF nom ingressos despeses guanys

Per exemple, si tenim:

D1:	nom	NIF	ingressos	D2:	NIF	despeses
1	Abacus	J33244098	50700	1	A33232411	250050
2	Amazon	Q44488877	200000	2	B25880945	700500
3	Amazon	Q44488877	100000	3	F66787231	380000
4	Apple	G28282828	200000	4	G28282828	580500
5	Apple	G28282828	200000	5	H10098477	74700
6	Apple	G28282828	400000	6	J33244098	50500
7	Ikea	F66787231	550000	7	Q44488877	230000
8	Nike	A33232411	250500			
9	Samsung	B25880945	500500			
10	Samsung	B25880945	250200			
11	Watch	H10098477	75050			

El resultat hauria de ser:

D3:	NIF	nom	ingressos	despeses	guanys
1	B25880945	Samsung	750700	700500	50200
2	F66787231	Ikea	550000	380000	170000
3	G28282828	Apple	800000	580500	219500
4	Q44488877	Amazon	300000	230000	70000

22. El dataframe `iris` conté dades sobre les dimensions (longitud i amplada) dels sèpals i pètals, i l'espècie de 150 flors. Rebert com a entrada `iris`:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
	...				
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
	...				
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

- (a) Dissenyeu una funció que rebi el dataframe `iris`, el nom d'una de les seves variables i el nom d'una de les seves espècies i retorni la mitjana de la variable per a l'espècie indicada.
- (b) Dissenyeu una funció que rebi el dataframe `iris` i retorni un altre dataframe amb les mitjanes de cada variable de flor en cada espècie, és a dir es busca un dataframe amb les dades següents: (pots considerar sabut que només hi ha 3 espècies diferents)

	Specie	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	setosa	meanSL-set	meanSW-set	meanPL-set	meanPW-set
2	versicolor	meanSL-ver	meanSW-ver	meanPL-ver	meanPW-ver
3	virginica	meanSL-vir	meanSW-vir	meanPL-vir	menaPW-vir

23. Disposem d'un dataframe amb dades sobre el nombre d'adopcions a Catalunya segons el país d'origen.

- (a) Dissenyeu una funció que rebi el dataframe i retorni de quin país hi va haver més adopcions entre els anys 2005 i 2010.
- (b) Dissenyeu una funció que tingui com a entrada el dataframe, dos anys (entre 2005 i 2012) i un país i retorni la diferència entre les adopcions en aquell país entre els dos anys en valor absolut.
- (c) Dissenyeu una funció que tingui com a entrada el dataframe i un any i retorni quantes adopcions d'Àfrica hi va haver aquell any.
- (d) Feu una funció que rebi com a entrada el dataframe i retorni un altre dataframe on cada fila sigui un any i cada columna un continent, obtenint així quantes adopcions hi va haver en cada any de cada continent.

- (e) Utilitzeu la funció definida en l'apartat anterior per construir-ne una altra que rebí el dataframe original i un any i retorni de quin continent va haver-hi més adopcions aquell any.

24. Disposem de tres dataframes. En el fitxer `estudiants.txt` hi ha informació sobre estudiants i el curs al qual estan matriculats, en `cap_aules.txt` hi trobem informació sobre aules i la seva capacitat i `aules_curs.txt` relaciona els cursos amb les aules on s'imparteixen.

- (a) Feu una funció que retorni els noms dels estudiants matriculats en els cursos que es fan a l'aula A1.
- (b) Tenint en compte el nombre d'estudiants matriculats a cada curs, feu una funció que retorni quins cursos estan programats en aules que no tenen prou capacitat.

Tenim ara un altre dataframe `curs_coord` amb la informació sobre els coordinadors de cada curs i la seva nacionalitat.

- (c) Dissenyeu una funció que retorni quina és la nacionalitat del professor del curs amb menys alumnes matriculats.
- (d) Feu una funció que donada una aula retorni els noms dels professors que fan classe en aquella aula.
- (e) Utilitzant la funció de l'apartat b), feu una funció que retorni els noms dels professors que hauran de buscar aules amb més capacitat pels seus cursos.

25. Disposem de dos dataframes amb informació sobre clients en un supermercat. El primer dataframe, que està en el fitxer `cl_prod.txt` conté informació sobre els productes de cada client. El segon, `caixes`, conté les caixes, el nombre de productes que té assignat cada caixa en aquest moment i el màxim de productes que pot tenir.

- (a) Donat un número de client, feu una funció que retorni a quines caixes podria anar.
- (b) Tenint en compte que les caixes sempre s'han d'omplir per ordre (és a dir, un client no pot anar a la caixa 3 si pot anar a la caixa 2), feu una funció que actualitzi el dataframe `caixes` amb els clients que tenim. La funció ha de retornar el dataframe actualitzat i un vector amb els clients que no han pogut ser col·locats en cap caixa.

26. Es té un `dataframe` amb els resultats de la Lliga d'un cert any. A cada fila hi ha els resultats d'un equip (amb els equips ordenats per ordre alfabètic) i la informació que tenim per a cada equip és: Partits Jugats (PJ), Partits Guanyats (PG), Partits Empatats (PE), Gols a Favor (GF) i Gols en Contra (GC). Fent sempre funcions que rebin com a entrada el `dataframe` amb les dades de la lliga, feu
- (a) Dissenyeu una funció que retorni el nom de l'equip que ha guanyat la Lliga.
  - (b) Dissenyeu una funció que retorni quins equips han baixat a Segona Divisió. Recordeu que els equips que baixen a Segona Divisió són els tres últims de la classificació.
  - (c) Dissenyeu una funció que retorni quin equip té la diferència més petita entre gols marcats i gols rebuts.
  - (d) Dissenyeu una funció que retorni `TRUE` o `FALSE` segons si en la classificació final hi ha hagut equips empatats a punts o no.
  - (e) Dissenyeu una funció que rebi els noms de dos equips i retorni la diferència de punts entre ells.
  - (f) Dissenyeu una funció que rebi els noms de dos equips i retorni quantes posicions els van separar.