

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.

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

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 rebí 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 rebí 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:
- (a) Una funció que a partir del list `eleccions` retorni quin any el PP va obtenir el seu pitjor resultat i quin resultat va ser.
 - (b) 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.
 - (c) Una funció que a partir del list `eleccions` retorni quin partit va obtenir més escons l'any 2003 i quants escons va obtenir.
16. *Guany i pèrdues.* (Examen final 13/14). Fes una funció que donats dos lists, `guany` i `perdues`, tals que tots dos tenen el format:

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

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

Per exemple:

```
guany <- 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 vector 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 pizzes
pizzes <- 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 pizzes i el de preus i retorni un list amb els noms de les pizzes 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 pizzes 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