

Capítol 3

Lists

Un **list** l és un objecte **estructurat** compostat per objectes anomenats components o elements. A diferència d'un vector, els components d'un list poden ser de tipus diferents. Aleshores, els lists són objectes heterogenis. De manera genèrica, un list l és de la forma:

t_1	t_2	t_3	t_4	t_5	t_6	\dots	t_N
-------	-------	-------	-------	-------	-------	---------	-------

on $\forall i \in \{1, \dots, N\}$, $t_i \in T_i$. És a dir, t_i és un valor del tipus T_i i $\forall i \neq j$, $i, j \in \{1, \dots, N\}$ no necessàriament $T_i = T_j$.

Per exemple, considerem el següent list:

<i>Enric</i>	<i>Canals</i>	21	<i>H</i>	1.77	<i>FALSE</i>	66543211	93 - 4444446
--------------	---------------	----	----------	------	--------------	----------	--------------

En aquest exemple el list té components de diferents tipus: cadena de caràcters, enter, real i booleà.

Els components poden rebre un nom que facilita l'accés als components i permet un nivell d'abstracció més alt.

nom_1	nom_2	nom_3	nom_4	nom_5	nom_6	\dots	nom_N
t_1	t_2	t_3	t_4	t_5	t_6	\dots	t_N

En aquest list, $\forall i \in \{1, \dots, N\}$, $t_i \in T_i$ i nom_i és l'identificador del i -èssim component. Com veurem a l'apartat 3.1, els noms d'un list poden ser dinàmicament assignats. A sota podem veure el list de l'exemple anterior amb noms per als components:

<i>Nom</i>	<i>Cognom</i>	<i>Edat</i>	<i>Sexe</i>	<i>Alçada</i>	<i>Treballa</i>	<i>Mobil</i>	<i>Telefon</i>
<i>Enric</i>	<i>Canals</i>	21	<i>H</i>	1.77	<i>FALSE</i>	66543211	93 - 4444446

En aquest exemple el list té components anomenats “Nom” i “Cognom” de tipus cadena de caràcters, “Edat” de tipus enter, “Sexe” de tipus caràcter (‘H’/‘D’), “Alçada” de tipus real, “Treballa” de tipus booleà (TRUE/FALSE) i, “Mobil” i “Telefon” que podrien ser de tipus cadena de caràcters també.

3.1 Creació de lists

Per a construir un list tenim la següent funció constructora:

```
list()
```

Aquest constructor ens permet crear lists buits, lists per enumeració dels seus components i lists amb *tags* o noms per als seus components, com segueix:

1. list buit

```
1 > l <- list()
2 > l
3 list()
4 >
```

2. list per enumeració

```
1 > l <- list("Enric", "Canals", 21, "H", 1.77, FALSE
, 66543211, "93 - 44444446")
2 > l
3 [[1]]
4 [1] "Enric"
5
6 [[2]]
7 [1] "Canals"
8
9 [[3]]
10 [1] 21
11
12 [[4]]
13 [1] "H"
14
15 [[5]]
16 [1] 1.77
17
18 [[6]]
19 [1] FALSE
20
21 [[7]]
22 [1] 66543211
23
24 [[8]]
25 [1] "93 - 44444446"
26 >
```

3. list amb noms per als components

```

1 > l <- list(Nom="Enric", Cognom="Canals",Edat=21,
             Sexe="H",Alcada=1.77,Treballa=FALSE,Mobil
             =66543211,Telefon="93 - 4444446")
2 > l
3 $Nom
4 [1] "Enric"
5
6 $Cognom
7 [1] "Canals"
8
9 $Edat
10 [1] 21
11
12 $Sexe
13 [1] "H"
14
15 $Alcada
16 [1] 1.77
17
18 $Treballa
19 [1] FALSE
20
21 $Mobil
22 [1] 66543211
23
24 $Telefon
25 [1] "93 - 4444446"

```

Adicionalment, podem consultar els noms dels components d'un list mitjançant la funció `names(l)` que donat un list `l` ens retorna un vector amb els noms dels seus components:

```

1 > names(l)
2 [1] "Nom"      "Cognom"    "Edat"     "Sexe"
3 "Alcada"   "Treballa" "Mobil"    "Telefon"

```

De vegades és necessari donar-li noms als components d'un list creat prèviament per enumeració. Això es pot fer assignant a `names(l)` el vector amb els noms que es vol:

```

1 > l<-list("Maria Perez","FIB",2011)
2 > l
3 [[1]]
4 [1] "Maria Perez"
5
6 [[2]]
7 [1] "FIB"
8
9 [[3]]
10 [1] 2011

```

```

11
12 #Fins aquí els components del list l no tenen
    noms. Ara li afegim els noms:
13
14 > names(l) <- c("Nom", "Centre", "Ingres")
15 > l
16 $Nom
17 [1] "Maria Perez"
18
19 $Centre
20 [1] "FIB"
21
22 $Ingres
23 [1] 2011

```

3.2 Accés als components d'un list

Existeixen diferents maneres d'accedir directament a cada component d'un list. En efecte, mitjançant **operadors d'accés directe**, que tenen com a paràmetre o bé el nom d'un component o bé un índex que indica la posició del component que es desitja visitar, es pot accedir a qualsevol component del list de manera directa. Les especificacions d'aquests operadors són les següents:

1. $\$: list \times id \rightarrow T_{id}$

Essent l un list, id el nom d'un component de l i T_{id} el tipus del component amb nom id . La crida a aquest operador té la següent sintaxi

$$l\$id$$

```

1 > l<-list(Nom="Maria Perez", Centre="FIB", Ingres
    =2011)
2 > l$Nom
3 [1] "Maria Perez"
4 > l$Centre
5 [1] "FIB"
6 > l$Ingres
7 [1] 2011
8 >

```

2. $[[]]: list \times enter \rightarrow T$

En aquest operador d'accés, els lists són tractats de manera semblant als vectors i fem servir la crida

$$l[[exp]]$$

per referir-nos al component del list l que es troba a la posició que resulti d'avaluar l'expressió entera exp . Cal notar, però, una diferència important

amb els vectors, i és que amb el list, l'operador és un doble gafet ([[]]) i cal parar atenció a aquesta diferència.

De la mateixa manera que passava amb els vectors, els accessos als components d'un list han de ser necessàriament posicions permeses en el list, i per tant, cal tenir la possibilitat de conèixer la dimensió d'un list. Per fer-ho disposem d'una funció especificada com segueix:

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

Així doncs, si un list l té N components, la crida

$$\text{length}(l)$$

ens tornarà N .

```

1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres
      =2011)
2 > l
3 $Nom
4 [1] "Maria Perez"
5
6 $Centre
7 [1] "FIB"
8
9 $Ingres
10 [1] 2011
11
12 > length(l)
13 [1] 3
14 >
15 > l1 <-list()
16 > length(l1)
17 [1] 0

```

Ara podem demanar que

$$1 \leq \text{avaluacio}(\text{exp}) \leq \text{length}(l) \quad (2)$$

```

1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres
      =2011)
2 > l[[1]]
3 [1] "Maria Perez"
4 > l[[2]]
5 [1] "FIB"
6 > l[[3]]
7 [1] 2011
8 > l[[1+1]]
9 [1] "FIB"
10 > l[[2*1]]
11 [1] "FIB"

```

Com en el cas dels vectors, és molt important vigilar que es satisfaci la restricció (2). En cas que no la respectem tindrem un problema en intentar accedir a una posició il·legal dins del list. En aquests casos, l'R ens respon con segueix

```
1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres
      =2011)
2 > l[[4]]
3 Error en l[[4]] : subindice fuera de los limites
4 >
```

S'ha de tenir en compte que existeix una bijecció del conjunt de noms d'un list l i la posició en que aquest nom apareix al list. Per exemple, al list l obtingut fent la instrucció

```
1 l<-list(Nom="Maria Perez",Centre="FIB",Ingres
      =2011) :
```

tindríem la bijecció següent:

	<i>posicio</i>	
<i>Nom</i>	→	1
<i>Centre</i>	→	2
<i>Ingres</i>	→	3

3. Entre les dues opcions anteriors per accedir als components d'un list l , també existeix una altra manera d'indexar-lo amb la crida següent:

```
l[[quote(id)]]
```

on *quote(id)* és el nom corresponent al *id* dins de doble cometes:

```
1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres
      =2011)
2 > l[["Centre"]]
3 [1] "FIB"
4 >
```

Com a conseqüència, tenim tres maneres d'accedir al mateix component:

```
1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres
      =2011)
2 >
3 > l[["Centre"]]
4 [1] "FIB"
5 >
6 > l$Centre
7 [1] "FIB"
8 >
9 > l[[2]]
10 [1] "FIB"
11 >
```

3.3 Afegir i eliminar elements d'un list

Els lists poden créixer i decreïxer dinàmicament. És a dir, podem afegir nous components a un list i eliminar qualsevol dels components que existeixin.

Per a afegir un nou element directament li assignem un valor al component i això ja crea el component:

```

1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres=2011)
2 > l
3 $Nom
4 [1] "Maria Perez"
5
6 $Centre
7 [1] "FIB"
8
9 $Ingres
10 [1] 2011
11
12 # Afegim l'element "Universitat":
13
14 > l$Universitat <- "UPC"
15 > l
16 $Nom
17 [1] "Maria Perez"
18
19 $Centre
20 [1] "FIB"
21
22 $Ingres
23 [1] 2011
24
25 $Universitat
26 [1] "UPC"
27 >

```

Per a eliminar un element el que farem és assignar-li el valor NULL al component que volem eliminar:

```

1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres=2011,
2           Universitat="UPC")
3 > l
4 $Nom
5 [1] "Maria Perez"
6
7 $Centre
8 [1] "FIB"
9
10 $Ingres
11 [1] 2011
12
13 $Universitat

```

```

13 [1] "UPC"
14
15 # Eliminem l'element "Centre":
16
17 > l$Centre<-NULL
18 > l
19 $Nom
20 [1] "Maria Perez"
21
22 $Ingres
23 [1] 2011
24
25 $Universitat
26 [1] "UPC"
27 >

```

També es pot preguntar si un element existeix en un list o no mitjançant la funció *is.null*:

is.null : *component_del_list* → *bool*

```

1 > is.null(l$Centre)
2 [1] TRUE
3
4 > is.null(l[["Ingres"]])
5 [1] FALSE
6 >

```

3.4 Accés a sublists

Per obtenir un sublist a partir d'un list, es fa servir la següent forma d'indexada:

$[] : list \times interval \rightarrow list$

on *interval* és un interval de la forma $[exp]$ o $[exp1 : exp2]$. La crida es fa com segueix:

$l[exp]$

```

1 > l<-list(Nom="Maria Perez",Centre="FIB",Ingres =2011,
           Universitat="UPC")
2 > l[1]
3 $Nom
4 [1] "Maria Perez"
5 >
6 > l[[1]]
7 [1] "Maria Perez"
8 >

```

S'ha de notar la diferència de significat quan es fa servir gafet simple i quan es fa servir doble gafet. En el primer cas torna un list mentre que en el segon cas torna l'èssim component.

També podem fer un sublist amb més d'un element del list original usant un subrang per a l'índex:

$$l[\text{exp1}:\text{exp2}]$$

```

1 > l[1:3]
2 $Nom
3 [1] "Maria Perez"
4
5 $Centre
6 [1] "FIB"
7
8 $Ingres
9 [1] 2011
10
11 > l[3:4]
12 $Ingres
13 [1] 2011
14
15 $Universitat
16 [1] "UPC"
17 >
18 > l[3:1]
19 $Ingres
20 [1] 2011
21
22 $Centre
23 [1] "FIB"
24
25 $Nom
26 [1] "Maria Perez"
27 >
28 > l[(2+1):(3+1)]
29 $Ingres
30 [1] 2011
31
32 $Universitat
33 [1] "UPC"
34 >

```

3.5 Recorreguts i cerques

Hi ha molts problemes on s'han de recórrer els components d'un list i també problemes on és necessari fer servir l'esquema de cerca. Aquestes dues famílies de problemes les hem caracteritzat al Capítol 1, quan parlàvem de vectors, a les seccions 1.3 i 1.4. Com a exemple, aquí podem veure una funció que escriu un list:

```
1 escriure_list <- function(l){
2   n <- names(l)
3   for (i in 1:length(l)){
4     cat(n[i],": ",l[[i]],"\n")
5   }
6 }
```

que fa un recorregut del list i que es pot utilitzar d'aquesta forma:

```
1 > l <- list(Nom="Joan",Cognom="Martinez",Edat=23,Grau=
   "Informatica",Universitat="UPC")
2 > escriure_list(l)
3 Nom : Joan
4 Cognom : Martinez
5 Edat : 23
6 Grau : Informatica
7 Universitat : UPC
8 >
```