Sessió 3

Altres objectes de *R*

A les sessions anteriors hem après a treballar amb vectors, que s'introdueixen fonamentalment amb 3 funcions: `c`, `seq` i `rep` (les constants són un cas particular de vectors) i amb matrius, que s'introdueixen amb la funció `matrix`.

El llenguatge **R** treballa amb diferents tipus d'estructures que es poden assignar a una variable i que s'anomenen objectes. Un objecte pot ser tant una constant, una estructura de dades, una funció o fins i tot un gràfic. Començarem treballant amb els objectes que són estructures de dades més freqüents:

- Objectes atòmics (**tots els seus elements han de ser del mateix tipus**):
  - vectors
  - matrius
  - arrays
  - factors
- Objectes recursius (els seus elements poden ser de tipus diferents i emmagatzemen objectes diferents):
  - llistes
  - data frames

Tots els objectes tenen atributs comuns: `mode`, `class` i cada objecte té atributs específics. Fem un llistat comparatiu dels objectes i d'alguns dels seus atributs a la taula següent.

objecte	funcions	definició	atributs propis
vector	`c(), rep(),seq()`	tira d'elements homogenis(*) (mateix tipus)	length, names
factor	`factor()` `cut(), ordered()`	vector per a variables categòriques	length, levels, labels
matrix	`matrix()` `cbind(),rbind()`	disposició en files i columnes d'elements del mateix tipus elements homogenis(*)	dim, dimnames colnames, rownames ncol, nrow
array	`array()`	matriu generaltizada a 3 o més dimensions elements homogenis(*)	dim, names dimnames
data.frame	`data.frame()`	estructura de dades rectangular files=casos, columnes=variables columnes de diferent tipus	dim, names colnames, rownames ncol, nrow
list	`list()`	elements diversos tipus de longituds diferents	length, names

(*) Si els elements no són homogenis, però hi ha un tipus comú, es converteixen a aquest. El procés de conversió també s'anomena *reciclatge* o *coerció*.

- Per veure i esborrar objectes, s'utilitzen les funcions `ls` i `rm`, respectivament; demaneu ajuda d'aquestes funcions.

```
a<-sqrt(9999) # creem un objecte unidimensional (número o constant)
v<-c(a,a^2) #  creem un vector de longitud 2
ls()       # apareix el llistat d'objectes que hem creat
rm(a)      # esborrem l'objecte "a"
```

  Recordem que si estem treballant amb RStudio, també podem consultar la pestanya "Environment", on apareix la llista d'objectes que hem creat i informació addicional (com ara dimensió si és una matriu o llargada si és un vector).

3.1 Factors

Els `factors` serveixen per emmagatzemar dades categòriques (nominals o ordinals), o dades numèriques que interessa reconvertir en categòriques, per exemple per fer una taula. Són un tipus especial de vectors. Els valors dels factors són els "nivells" (*levels*) que sovint tenen "etiquetes" (*labels*) explicatives; és a dir, noms posats per l'usuari. Per crear factors, s'usen les funcions `factor` i `ordered`.

```
x<-c(1,1,2,3,1,3)
(fx<-factor(x))
(fx<-factor(x,levels=c(1,2,3),labels=c("baix","normal","alt")))
# observeu que sempre mostra les etiquetes
```

Per especificar que es defineix un factor els valors del qual estan en escala ordinal, s'usa la funció `ordered`.

```
x<-c(1,1,2,3,1,3)
fx<-ordered(x,levels=c(1,2,3),labels=c("baix","normal","alt")); fx
```

Crear un factor a partir de dades numèriques (agrupant valors)

Es poden crear factors a partir de variables numèriques utilitzant la comanda `cut`, que agrupa els valors en intervals, que són els valors de la nova variable categòria.

- En primer lloc, creem una mostra aleatòria de 25 notes (funció `sample`), fixant la llavor de partida (`set.seed`) per garantir que obtenim tots la mateixa mostra.

```
set.seed(343) ## iniciem la llavor per generar números aleatoris
(notes<-sample(1:10,25,rep=TRUE))##veiem les notes generades
 [1]  4  2  8  8  6 10  2  4  1  7  5  1  2  7  1
```

  La variable `notes` és, per defecte, un vector numèric. Volem fer-ne una agrupació en intervals: $[0,5), [5,7), [7,9), [9,10]$.

- Seguidament, mitjançant `cut` crearem un nou factor, que anomenem `f1`, amb 4 categories o nivells. Vegem com es fà amb `cut`:

  La funció `cut` té l'argument `breaks` que defineix els intervals. També l'argument `right` que indica si els intervals són tancats per la dreta, és a dir, de la forma $(a, b]$ (`right=TRUE`) o bé si els intervals són de la forma $[a, b)$ (`right=FALSE`). Un altre argument és `include.lowest` per fer que el primer interval sigui de la forma $[a, b]$ (si `right=TRUE`) o bé perquè sigui d'aquesta forma el darrer interval quan `right=FALSE`.

```
(f1<-cut(notes, breaks=c(0,5,7,9,10),right=FALSE))
[1] [0,5) [0,5) [7,9) [7,9) [5,7) NA [0,5) [0,5) [0,5) [7,9) [5,7)
[12] [0,5) [0,5) [7,9) [0,5)
Levels: [0,5) [5,7) [7,9) [9,10)
 # fixem-nos que 4 grups requereixen: min, 3 punts de tall i max
 # right = FALSE fa intervals oberts per la dreta
 # ! atenció: el 10 no està inclòs a cap interval (apareix NA)
 # per incloure'l, fem "include.lowest=TRUE"
(f1<-cut(notes,breaks=c(0,5,7,9,10),right=FALSE,include.lowest=TRUE))
[1] [0,5) [0,5) [7,9) [7,9) [5,7) [9,10] [0,5) [0,5) [0,5) [7,9) [5,7)
[12] [0,5) [0,5) [7,9) [0,5)
Levels: [0,5) [5,7) [7,9) [9,10]
```

- Ara utilitzem l'argument `labels` per afegir etiquetes a les categories: valors *s (suspens), a (aprovat), n (notable) i e (excel·lent)*. També podem crear un factor ordenat amb `ordered_result =TRUE`

```
# posem etiquetes
(f2<-cut(notes,include.lowest = TRUE,right=FALSE, breaks=c(0,5,7,9,10),
  labels=c("s","a","n","e")))
  [1] s s n n a e s s s n a s s n s
  Levels: s a n e
# vector ordinal "ordered_result=TRUE"
(f3<-cut(notes, include.lowest=TRUE, right=FALSE, breaks=c(0,5,7,9,10),
      labels=c("s","a","n","e"),ordered_result=TRUE))
 [1] s s n n a e s s s n a s s n s
Levels: s < a < n < e
```

Aplicació: quan ens interessi tractar les notes com a numèriques, usarem l'objecte `notes` i quan vulguem fer una taula o determinades gràfiques podem usar algun dels factors creats: `f1`, `f2` o `f3`.

Pràctica:
• Genera un vector aleatori (amb llavor 49) de 300 casos d'IMC (Índex de Massa Corporal) amb valors que vagin del 12 al 55 amb un decimal (Pista: l'argument de la funció **seq** que et permet posar decimals és **by=**). Crea un factor amb les etiquetes i els 4 principals intervals que indica la OMS:

Classficació segons IMC	
Insuficiència ponderal	< 18.5
Pes normal	18.5–24.9
Preobesitat	25–29.9
Obesitat	≥ 30
Obesitat classe I	30–34.9
Obesitat classe II	35–39.9
Obesitat classe II	≥ 40

• Es recull l'edat, la despesa i el color preferit dels clients d'una cadena de botigues de roba i s'obtenen els vectors següents: edat <- c(32, 27, 29, 27, 32, 18, 31, 22, 33, 36, 32, 24, 31, 20, 18, 13, 13, 26, 29, 24, 30, 17, 35, 24)
despesa <- c(1, 1, 1, 1, 2, 2, 2, 1, 1, 2, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2)
color <- c('blau', 'verd', 'verd', 'negre', 'blau', 'blau', 'negre', 'negre', 'verd', 'blau', 'vermell', 'vermell', 'negre', 'negre', 'rosa', 'groc', 'blau', 'verd', 'blau', 'rosa', 'rosa', 'verd', 'blau', 'groc')
Crea un factor per cada vector amb les següents premisses i raona si té sentit tractar-los com a factors ordinals:
- Per edat: "teens" (menors de 18 anys); "joves" (entre 18 i 29 anys); "adults" (30 i més anys)
- Per despesa: 1 = menys de 30€; 2 = 30€o més
- Per color: el text que consta en cada cas ja defineix la categoria.

3.2 Llistes

Les llistes són objectes molt generals, heterogenis, de manera que els seus elements poden ser de classe i mode diferent. Molts procediments de **R** donen com a resultat una llista.

- Definim una llista amb la comanda `list` i vegem-ne els atributs

```
a<-5; M<-matrix(1:12,4,3,byrow=TRUE); A<-array(1:24, dim=c(2,4,3))
v<-c("blau","verd"); des<-"1a llista"
     # són els objectes que formaran la nostra  llista
(L<-list(a,M,A,v,des))           # creem la llista
```

- Definim la llista amb noms (per millorar l'output) i vegem els seus atributs

```
(L<-list(escalar=a,matriu=M,array=A,vector=v,descriptor=des))
names(L)
length(L)
mode(L)  class(L)    # en aquest cas dona el mateix
```

- Utilització de les llistes per assignar noms a les dimensions d'un array

```
(A<-array(1:32, dim=c(2,4,4)))  # definim l'array
                    # i ara li posem noms
```

```
dimnames(A)<-list(c("u","dos"),c("a","b","c","d"),
c("m1","m2","m3","m4"))
A ## veiem les tres matrius emmagatzemades juntes com un array
```

Pràctica: Genera els següents objectes i guarda'ls tots en una llista anomenada NOTES. Posa nom a cada un dels objectes quan els tinguis a la llista (decideix tu el nom que has de posar però pensa que ha de ser un nom que descrigui fàcilment les dades que conté cada element de la llista).
n <- 7.45 (nota mitjana de tota la classe)
np <- c(9.45, 9.1, 8.85, 8.8, 8.6) (vector de notes en assistència i lliurament de pràctiques dels 5 millors alumnes)
ne <- matrix(c(10, 10, 9.3, 9, 8.9, 10, 9.3, 9.7, 8, 8.2, 9.9, 8.9, 9, 8.9, 8.5, 9.7, 9.3, 9.5, 9.7, 8.2), nrow=5, byrow=TRUE) (matriu amb les notes dels 4 parcials dels 5 millors alumnes.
Posa nom a les fileres (Alumne1, Alumne2,...), i a les columnes (Parcial1, Parcial2, ...))

3.3 Data frames

Són els objectes més freqüents en les anàlisis estadístiques. L'objecte data frame, com la matriu, té una estructura rectangular, però en realitat els seus elements són les columnes i, per tant, el nombre d'elements (`length`) d'un data frame és el nombre de columnes i el noms (`names`) són els de les columnes. També es diferencia de les matrius que no tot el que apareix en un data frame ha de ser del mateix tipus, només ho han de ser les components d'una la mateixa columna.

- La funció `data.frame`, per crear un data frame, i alhora assignar noms a les seves variables (les seves columnes). Construïm un data frame "de joguina" amb notes de 5 estudiants d'àlgebra i càlcul, i el grup de l'estudiant.

```
df1<-data.frame(alg=c(5,4,8,2,7),cal=c(6,8,8,2,3),
     grup=c("A","B","B","A","B"));
rownames(df1)<-c("Pere", "José M", "Luisa","Lluís","Olga"); df1
df1
       alg cal grup         # aquest és el resultat
Pere     5   6    A
José M   4   8    B
Luisa    8   8    B
Lluís    2   2    A
Olga     7   3    B

# funcions per obtenir informació del data.frame df1
length(df1); dim(df1); names(df1); colnames(df1); rownames(df1);
```

- R té diferents data frames per poder-hi treballar. Podem llegir com a data.frames fitxers de la llibreria base de R o d'altres llibreries. Aquests data.frames es poden carregar amb la funció `data(nom del data.frame)`, en el cas que no estiguin carregats en el paquet base.

```
?iris          ## és de la llibreria base
mode(iris)     ## el que hi ha dins de iris és una llista
class(iris)    ## com a  tipus d'objecte és un data frame
names(iris)
dim(iris)
```

```
nrow(iris)
ncol(iris)
```

- Per tenir informació d'un data frame, tenim, a banda de les funcions dim, `ncol`, `nrow` (nombre de files i columnes), les funcions `head` (capçalera) mostra els 6 primers casos del data.frame i `tail` (cua) els 6 darrers casos. Finalment, la funció `str` ens dona informació de les variables del data.frame (nom i tipus de variable)

```
 ## hi ha data frames a d'altres llibreries:
library(AER)    ## potser cal instal·lar abans el paquet AER
data(Affairs)    # carrega el fitxer a l'espai de treball
af<-data.frame(Affairs)  # reanomena el fitxer
head(af)
tail(af)
str(af)  # obtenim informació de les variables
         # (de l'estructura)
'data.frame':  601 obs. of  9 variables:
$ affairs      : num  0 0 0 0 0 0 0 0 0 0 ...
$ gender       : Factor w/ 2 levels "female","male": 2 1 1 2 2 1 1 2 1 2 ...
$ age          : num  37 27 32 57 22 32 22 57 32 22 ...
$ yearsmarried : num  10 4 15 15 0.75 1.5 0.75 15 15 1.5 ...
$ children     : Factor w/ 2 levels "no","yes": 1 1 2 2 1 1 1 2 2 1 ...
$ religiousness: int  3 4 1 5 2 2 2 2 4 4 ...
$ education    : num  18 14 12 18 17 17 12 14 16 14 ...
$ occupation   : int  7 6 1 6 6 5 1 4 1 4 ...
$ rating       : int  4 4 4 5 3 5 3 4 2 5 ...
```

Pràctica:

• El paquet base de R també té el data frame anomenada `mtcars`. Assigna-la a un objecte anomenat `df.cotxes`. Troba les característiques d'aquest data frame (tipus d'objecte que és, nombre de variables, dimensió, nom de les variables) i mostra el contingut de les 10 primeres fileres i de les 6 darreres.

• Crea un data frame anomenat `df.interes` que contingui els quatre vectors que vas crear a la pràctica de l'apartat 1.5 (`sex, edat, iest, ipol`). Canvia el nom de les variables per `sexe`, `edat`, `interès per l'estadística`, `interès per la política`.

3.4 Conversió del mode de l'objecte

Les funcions `as.factor`, `as.matrix`, etc., canvien el mode, si el canvi és factible. La necessitat dels canvis prové del fet que certes funcions o procediments només són aplicables a objectes de determinada tipologia. Per exemple, per multiplicar matrius no es pot fer si s'han transormat en data frames.

```
v<-c(3,4); v
m<-as.matrix(v); m
as.vector(m)
fa<-as.factor(v); fa
(M<-matrix(1:12,4,3,byrow=TRUE))
(dM<-as.data.frame(M))
```

```
length(M); length(dM)      # ¡! Atenció !! Què passa?
# "%*%" multiplicació matricial
# t(M) matriu transposada
# M i t(M) es poden multiplicar sempre: arguments compatibles
M%*%t(M)
dM%*%t(dM)  # ¡! Atenció !! no es poden multiplicar data frames
as.matrix(dM)%*%as.matrix(t(dM))  # així també, clar
```

Pràctica: Converteix el data.frame `df.interes` de la pràctica anterior en una matriu. Quines característiques (tipus d'objecte, nombre de variables, dimensió, nom de les variables) tenia quan era un data.frame? I quan es transforma en una matriu? Quines diferències hi ha?

3.5 Indexació dels objectes

Els components o elements de vectors, matrius, arrays, llistes i data.frames es poden indexar per posició o per nom, si en tenen. La indexació és important per localitzar i operar, com veurem més endavant.

Indexació de vectors i factors

- Localització d'elements d'un vector per la posició que ocupen. Creació de nous vectors, vegem-ho amb exemples:

```
v<-c(1:4,NA,-2:3)      # definim un vector
v[3];v[5];v[8]         # localitzem elements per posició
v[1:3]; v[c(4,5,9)]    # localitzem "sub-vectors"
w<-v[1:3];w            # nou vector w, sub-vector de v
```

- Localització amb **índex negatiu**

```
v[-1]; v[-(2:5)]       # no els esborra, deixa de mostrar-los
w<-v[-(2:5)];w         # nou vector
v<-v[-(2:5)];v         # ara ho esborra (!)
```

Indexació de matrius i arrays

- Localització (índex positiu i negatiu) d'elements de matrius i arrays i canvis de valors

```
M<-matrix(1:12,4,3,byrow=TRUE)
A<-array(1:24, dim=c(2,4,3))
M[1,3]; M[1,]; M[,1]      # atenció, mostra les columnes com files
M[-1,]; M[,-1]; M[-1,-1] # mostra M excepte fila 1 i/o columna 1
A[,1,]
A[,,1]
A[1,,1]
A[1,1,1]
A[,,-1]    # ens mostra totes les matrius excepte la primera
```

Indexació de llistes

- Localització d'elements de llistes, per la posició o pel nom (*nomllista*`$`*nomobjecte*)

```
L<-list(nombre=a,matriu=M,array=A,vector=v,descriptor=des); L
L[[1]]; L[[2]]        # crida elements de la llista
L$matriu              # crida pel nom el segon element
L[[2]][1,1]; L$matriu[1,1]   # indexa dins la matriu
L[[2]][ ,-1]; L$matriu[,-1]  # elimina dins la matriu
```

Nota: cal distingir entre `L[2]` (2n objecte de la llista `L` com a *subllista* (format llista)) i `L[[2]]` (en format propi de l'objecte, en aquest cas una matriu). Comprovem-ho.

```
class(L[[2]]); class(L[2])
```

Indexació de data frames

Tenim dues formes de referir-nos a una part d'un data frame, pensant-lo com a llista o pensant-lo com a matriu.

- Indexació de *data frames* com a llistes (de les columnes)

```
(df<-data.frame(alg=c(5,4,8),md=c(5,8,8), grup=c("A","B","B")))
df[1] # primera columna, com a data.frame (class?)
df["alg"] # recomanable, si tenen noms les columnes, igual que l'anterior
df[[1]] # primera columna, com a vector
df$alg  # idem
df[["alg"]] # idem
```

- Indexació com a matrius:

```
df[1,] # com a les matrius, és la fila 1 o primer cas de df
df[-2,] # df sense el segon cas, index negatiu !
df[,c(2,3)]  # columnes 2 i 3 de df
```

Indexació mitjançant condicions lògiques

Podem indexar un vector amb un altre vector lògic. Els TRUE corresponen a les components que agafem:

```
w<-c(10,20,30,40,50)  # w té 5 components
seleccio1<-c(2,3,5)
seleccio2<-c(FALSE,TRUE,TRUE,FALSE,TRUE)
# què passa si...
w[seleccio1]
[1] 20 30 50  # components 2, 3 4 de w
w[seleccio2]  # idem
```

Gràcies a això podem utilitzar una condició lògica per seleccionar algunes components:

```
w>25
w[w>25]  # equival a w[c(FALSE,FALSE, TRUE, TRUE,TRUE)]
w2<-c(1:5,NA,NA,2,7,NA)
is.na(w2)   # dona TRUE  a les components NA
!is.na(w2)  # dona FALSE a les components NA
w2[!is.na(w2)] # w2 sense els valors perduts (NA)
```

Canvis en els valors dels objectes mitjançant idexació

```
 v<-c(3,4);
 v[1]<-0; v
 M<-matrix(1:12,nrow=3); M[1,]<-c(0,0,0);M
 A<-array(1:12,dim=c(2,2,3)); A[,,2]<-A[,,1];  A[,,3]<-A[,,1]; A
 # ara totes les matrius de l'array són iguals

 w2[is.na(w2)]<-0; w2     # canviem els NA de w2 per 0

 w3<-c(-5,-3,9,0,4,-8,10)
 w3[w3<0]<- -1;w3[w3>0]<-1;w3  # canviem negatius per -1 i positius per 1

# Exercici 1.3(d): # construir el vector tv
tv<-rep(TRUE,25)
tv[c(15,16,21)]<- FALSE # canviem els que tenen id=14,15,20
```

Afegir una nova columna a un data frame amb indexació

La indexació ens permet afegir una nova columna al data frame `df` anterior. Li posem nom també:

```
> df$calcul<-c(5,3,8)
# ara comprovem que df té una nova columna
> df
  alg md grup calcul
1   5  5    A      5
2   4  8    B      3
3   8  8    B      8
```

Observem el resultat, hem afegit una nova variable al data frame amb el nom `calcul`.

També podem afegir, de la mateixa manera, un cas. Ara mateix `df` és un data frame amb 4 columnes. La tercera és una variable categòrica (un vector de caràcters) i les altres són numèriques. Per això cada fila és una llista (els objectes són de tipus diferent).

```
> df[4,]<-list(7,6,'A',4)
> df
  alg md grup calcul
1   5  5    A      5
2   4  8    B      3
3   8  8    B      8
4   7  6    A      4
```

Pràctica:

• A partir del vector `IMC` creat a la pràctica de l'apartat 3.1. genera els nous vectors que es demanen a continuació:

a) Un vector `imc.1` amb els 35 primers casos

b) Un vector `imc.2` sense els 35 primers casos

c) Un vector `imc.3` amb els casos del 100 al 200 (ambdós inclosos)

d) Un vector `imc.4` amb els casos 2, 19, 75, 127, 225, 299

e) Quin valor hi ha a la posició 37 del vector `imc`?

• De la llista `NOTES` que has creat a la pràctica de l'apartat 3.2., posa la matriu ne en un nou objecte anomenat `notes.parcials`. Recorda que aquesta matriu contenia la nota dels 5 millors alumnes dels 4 exàmens parcials. A partir d'aquesta matriu, i utilitzant la indexació d'objectes respon:

a) Quines són les notes del primer parcial? Posa-les en un vector anomenat Parcial.1.

b) Quines són les notes de l'alumne 3? Posa-les en un vector anomenat Alumne.3.

c) Quina nota va treure al 3er parcial l'alumne 5?

• Treballarem amb el data.frame `df.cotxes` que conté mtcars que es troba al paquet base de R. Fes les següents tasques:

a) Què conté la variable `cyl`?

b) Quants carburadors té el cotxe que està a la 10 posició?

c) Quines característiques té el cotxe que es troba a la 15a posició?

d) Genera un nou data frame anomenat `df.cotxes.0` que no contingui les variables `hp` ni `wt`

e) Afegeix una nova variable al data.frame `df.cotxes`. Anomena-la `identificador` i omple-la amb la sèrie `cotxe.1, cotxe.2,...`

f) Afegeix un nou cotxe amb els mateixos valors que el que es troba a la posició 25, canviant el nom per `Datsun 710.0`. Visualitza els cotxes 25 i 33.