Capítol 4

Data Frames

Un **data frame** *df* és un objecte **estructurat**, de dues dimensions, on les columnes (també anomenades *variables*) són dades d'un mateix tipus i les files (o *individus*) són un conjunt de dades de diferent tipus. Un data frame es pot veure com un list de vectors i també com una matriu no heterogènia. Vist com una matriu, un data frame és un objecte composat per un nombre finit de files F (entrades horitzontals) i un nombre finit de columnes C (entrades verticals) anomenats components o elements. De manera genèrica, podem considerar que un data frame *df* és de la forma:

$$
\begin{array}{c}
 \\
F\ files
\end{array}
\overbrace{\begin{bmatrix}
df_{1,1} & df_{1,2} & \dots & df_{1,C-1} & df_{1,C} \\
df_{2,1} & df_{2,2} & \dots & df_{2,C-1} & df_{2,C} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
df_{F-1,1} & df_{F-1,2} & \dots & df_{F-1,C-1} & df_{F-1,C} \\
df_{F,1} & df_{F,2} & \dots & df_{F,C-1} & df_{F,C}
\end{bmatrix}}^{C\ columnes}
$$

Com podem veure, en aquest cas el data frame *df* té F files i C columnes i per tant té $F \times C$ components on es compleix

$$\forall i \in \{1, \dots, F\}, \forall j \in \{1, \dots, C\}, df_{i,j} \in T_j$$

Dit d'una altra manera, amb el data frame *df* hom pot associar la dimensió $F \times C$. En un data frame, cada fila i és un objecte de tipus *list* amb dades no necessàriament homogènies mentre que cada columna j és un objecte de tipus vector i, per tant, homogeni. És a dir, $df_{i,j}$ es un valor del tipus base T_j. Per fixar la idea, si considerem una enquesta, un data frame és el tipus d'objecte de l'R que permet representar les dades recollides mitjançant l'enquesta. Vist d'aquesta manera, cada columna correspon als valors d'una variable estadística (i possiblement el seu nom és a la capçalera) i cada fila correspon a una resposta donada per una persona enquestada.

Considerem per exemple el data frame següent:

$$df = \begin{bmatrix} Edat & Sexe & Alçada & Hobby \\ 18 & H & 1.81 & Fútbol \\ 20 & M & 1.67 & Lectura \\ 15 & M & 1.60 & Dança \\ 19 & H & 1.76 & Bowling \\ 23 & M & 1.70 & Música \\ 18 & H & 1.65 & Lectura \end{bmatrix}$$

Interpretant aquest exemple com les respostes a una enquesta, les variables estadístiques serien `Edat, Sexe, Alçada` i `Hobby` on els seus valors serien els que es troben a la columna que encapçala cadascuna. Addicionalment, cada fila seria una possible resposta a l'enquesta. Com hem dit abans, cada fila pot ser vista com un *list*, com ara ⟨`23, M, 1.70, Música`⟩ i cada columna com a un vector, per exemple el vector de seqüència de caràcters ⟨`"Fútbol", L-Lectura", "Dança", "Bowling", "Música", L·Lectura"`⟩.

4.1 Creació de data frames al llenguatge R

El llenguatge R ofereix diverses formes de crear data frames. Ací en veurem dos: la creació directa i la creació a partir de fitxers.

4.1.1 Creació directa

La creació directa d'un data frame utilitza el constructor

`data.frame()`

Aquest constructor rep com a paràmetres les columnes del data frame a crear. Com en el cas dels *list*, hom pot assignar un nom a cadascuna de les columnes (variables) del data frame. Per exemple podem construir un data frame fent:

```
data.frame(nomcol_1 = valcol_1, ..., nomcol_k = valcol_k, stringsAsFactors=FALSE)
```

Un exemple concret de construcció d'un data frames és

```
> df <- data.frame(Nom=c("Maria","Kim","Pep"),
                   Edat=c(18,20,23),
                   stringsAsFactors=FALSE)
> df
    Nom Edat
1 Maria   18
2   Kim   20
3   Pep   23
>
```

Però a diferència del *list*, en aquest cas també poden usar-se els noms nom de les variables d'R per a donar nom als camps del data frame. Així doncs podem fer el següent:

```
> Edat <- c(18,20,15,19,23,18)
> Sexe <- c("H","M","M","H","M","H")
> Alcada <- c(1.81,1.67,1.60,1.76,1.70,1.65)
```

```
> Hobby <- c("Futbol","Lectura","Danca","Bowling","
    Musica","Lectura")
> df <- data.frame(Edat,Sexe,Alcada,Hobby,
    stringsAsFactors=FALSE)
> df
  Edat Sexe Alcada   Hobby
1   18    H   1.81  Futbol
2   20    M   1.67 Lectura
3   15    M   1.60   Danca
4   19    H   1.76 Bowling
5   23    M   1.70  Musica
6   18    H   1.65 Lectura
>
```

Recordem que el paràmetre `stringsAsFactors` defineix la manera en la que R interpretarà les cadenes de caràcters. Si no es defineix de manera explícita el paràmetre `stringsAsFactors=FALSE`, el valor que pren per defecte és `TRUE` i això vol dir que les cadenes de caràcters seran tractades d'una forma diferent a la que estem acostumats. L'explicació en detall d'aquest paràmetre està fora de l'abast d'aquest document i per tant, sempre que calgui crear data frames amb columnes on hi hagi cadenes de caràcters s'inclourà l'assignació `stringsAsFactors=FALSE`.

També es poden crear data frames buits. Per exemple, si a l'intèrpret R introduïm l'assignació

```
> df1 <- data.frame()
```

i tot seguit demanem pel valor de la variable `df1`, s'obté

```
> df1
data frame with 0 columns and 0 rows
>
```

4.1.2 Creació de data frames a partir de fitxers

Generalment els data frames són objectes grans i interessarà construir-los a partir de les dades d'un fitxer CSV, format en el qual les dades estan usualment separades per un espai. En aquest cas, R ofereix un operador de lectura de fitxers `read.table` que s'utilitza com segueix:

```
read.table(quote(nomfitxer), header=TRUE)
```

El paràmetre `header = TRUE` defineix que la primera línia conté les capçaleres del data frame, és a dir, els noms de les variables. Si no s'explicita pren el valor per defecte `FALSE`. Per exemple, el codi R

```
> df <- read.table("mydata.txt", header=TRUE)
```

crea el data frame `df` llegint les dades del fitxer de nom `"mydata.txt"` amb capçaleres.

4.2 Operadors del tipus *list* i data frames

Ja hem explicat que un data frame es pot veure com un *list*, per tant se li podran aplicar tots els operadors estudiats al capítol anterior.

4.2.1 Accés a una columna

En tot el que segueix, la variable `df` denotarà un data frame. suposem el data frame de la Secció 4.1.1, la primera columna corresponent al nom `Edat` es pot seleccionar usant el valor enter de líndex

```
1 > df[[1]
2 [1] 18 20 15 19 23 18
```

El nom de de la columna amb l'operador $

```
1 > df$Edat
2 [1] 18 20 15 19 23 18
```

O bé emprant el nom com a índex

```
1 > df[["Hobby"]]
2 [1] "Futbol"  "Lectura" "Danca"   "Bowling" "Musica"
     "Lectura"
```

4.2.2 Obtenir la longitud

Podem demanar la longitud del data frame amb la funció `length()` i ens retornarà el nombre de columnes (variables) del data frame:

```
1 > length(df)
2 [1] 4
3 >
```

4.2.3 Noms

Podem accedir als noms de les variables:

names(): Retorna un vector amb els noms de les variables o columnes.

colnames(): Retorna un vector amb els noms de les variables. És equivalent a `names()`.

o també als noms de les files:

rownames(): Retorna un vector amb els noms de les files.

En tots els casos, igual que passa en les estructures *list*, també es poden assignar a aquestes funcions el vector de noms que hom vulgui definir. Exemples d'utilització d'aquestes operacions aplicades al data frame definit al llistat 4.1.1 són:

```
1 > names(df)
2 [1] "Edat"   "Sexe"   "Alcada" "Hobby"
3 > colnames(df)
4 [1] "Edat"   "Sexe"   "Alcada" "Hobby"
5 > rownames(df)
6 [1] "1" "2" "3" "4" "5" "6"
7 >
```

Considerem el data frame del llistat de la Secció 4.1.1. Si ara hom defineix el conjunt de noms de files amb

```
> rownames(df) <- c("Joan","Anna","Maria","Albert","
    Jana","Josep")
>
```

el data frame `df` resultant serà

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
> length(df)
[1] 4
>
```

4.3 Operadors del tipus matriu i data frames

Com ha estat explicat més amunt, un data frame també es pot veure com una matriu, per tant també es podem aplicar els operadors definits per a les matriu i accedir als elements del data frame de forma matricial, és a dir, variables amb dos índexs. Recordem-ne algunes d'aquestes operacions.

nrow(): Retorna el nombre de files.

ncol(): Retorna el nombre de columnbes.

dim(): Retorna el nombre de files i columnes, és a dir, les dimensions de la matriu.

Considerem ara el data frame `df` del llistat de la Secció 4.2.3. Aleshores tindrem

```
> nrow(df)
[1] 6
> ncol(df)
[1] 4
> dim(df)
[1] 6 4
>
```

Important En particular, quan s'usen operadors d'indexat, cal assegurar-se'n que el valor resultant de l'avaluació de l'expressió de l'índex caigui dins d'un rang de valors [1..max] on max és el valor màxim de files o columnes del data frame. En general cal tenir molta cura amb el fet que les funcions s'apliquin sobre data frames que estiguin correctament definits. Un exemple concret de situació en la qual una funció s'aplica de manera errònia és el següent

```
> df2
data frame with 0 columns and 0 rows
> nrow(df2)
[1] 0
> ncol(df2)
[1] 0
> df[1,2]
NULL
```

4.3.1 Accés a una columna sencera

Recordem que cada columna d'un data frame és un vector, per tant, un objecte amb informació homogènia. Considerem un altre cop el data frame `df` del llistat de la Secció 4.2.3. Aleshores tindrem

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
> df[,2]
[1] "H" "M" "M" "H" "M" "H"
> df[,3]
[1] 1.81 1.67 1.60 1.76 1.70 1.65
>
```

4.3.2 Accés a una fila sencera

Recodem que cada fila és un list per tant, un objecte heterogeni. Fixeu-vos que en fer la petició d'una única fila en realitat ens retorna un data frame amb només aquesta fila.

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
> df[2,]
       Edat Sexe Alcada   Hobby
Anna     20    M   1.67 Lectura
> df[4,]
       Edat Sexe Alcada   Hobby
Albert   19    H   1.76 Bowling
```

4.3.3 Accés a una component d'un data frame

De manera anàloga al que passava en el càlcul matricial, tot sovint interessa tractar cada element d'un data frame de manera individual. Considerant el data frame `df` del llistat de la Secció 4.2.3 tindrem

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
> df[1,3]
[1] 1.81
> df[3,4]
[1] "Danca"
> df[2,5]
NULL
> df[7,3]
NULL
>
```

Noteu que, quan es tracta d'accedir a elements del data frame que no estan definits, l'operador retorna la constant `NULL`, és a dir, un valor invàlid. Recordeu el que ha estat dit més amunt respecte d'haver d'usar amb cura els índexs.

També es pot accedir a un element determinat usant com a índex el nom de la columna

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
> df[1,"Alcada"]
[1] 1.81
> df[3,"Hobby"]
[1] "Danca"
>
```

4.4 Subdata frames

De la mateixa manera que de les matrius es poden extreure submatrius, dels data frames es poden extreure subdata frames. Les diverses extracccions es poden fer per rang d'índex, amb filtres o selecció de files

4.4.1 Seleccció per rang d'índex

La seleccció de subdata frames per rang d'índex segueix la mateixa sintaxi que en el cas de l'extracció de submatrius

- L'absència del segon índex, `[i, ]` designa la fila `i`-èssima.
- L'absència del primer índex, `[, j]` designa la columna `j`-èssima.
- La presència d'un rang, `[min:max]` indica el conjunt de files o columnes tals que la primera és la corresponent a l'índex `min` i la darrera és la corresponent a l'índex `max`.

Vegem tot seguit alguns exemples.

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
>
> df[2:4,]
       Edat Sexe Alcada   Hobby
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
>
> df[,1:3]
       Edat Sexe Alcada
Joan     18    H   1.81
Anna     20    M   1.67
Maria    15    M   1.60
Albert   19    H   1.76
Jana     23    M   1.70
Josep    18    H   1.65
>
> df[2:4,1:3]
       Edat Sexe Alcada
Anna     20    M   1.67
Maria    15    M   1.60
Albert    19    H   1.76
>
> df[2:4,3]
[1] 1.67 1.60 1.76
```

Noteu que el resultat de la darrera línea és un vector.

Com hem vist, l'accés a les dades d'una única columna del data frame retorna les dades de la columna en un vector. Si el que volem és que el resultat retornat sigui una columna del data frame expressada com un altre data frame cal afegir el paràmetre `drop = FALSE` com segueix:

```
> df[2:4,3, drop=FALSE]
       Alcada
Anna     1.67
Maria    1.60
Albert   1.76
```

4.4.2 Filtres

Moltes vegades el que es vol és obtenir un subdata frame segons alguna condició o propietat que es pot expressar com una expressió lògica repecte d'una columna. Si el data frame és `df`, la sentència seria:

```
df[condició(df$colnom, valor), rangcol]
```

per exemple

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
> df[df$Alcada < 1.70,]
       Edat Sexe Alcada   Hobby
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Josep    18    H   1.65 Lectura
> df[df$Alcada > 1.70, 2:4]
      Sexe Alcada   Hobby
Joan     H   1.81  Futbol
Albert   H   1.76 Bowling
>
```

4.4.3 Selecció de files

Si hom vol seleccionar un subdata frame que contingui totes les columnes del data frame original però només inclogui les files que compleixin una determinada condició, el llenguatge R ofereix la funció `subset`. La sintaxi és

```
subset(df, cond(colnoms, valors))
```

Exemples d'utilització són

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
>
```

```
> subset(df, Sexe == "H")
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Albert   19    H   1.76 Bowling
Josep    18    H   1.65 Lectura
> subset(df, Sexe == "H" & Alcada >= 1.65)
      Edat Sexe Alcada   Hobby
Joan    18    H   1.81  Futbol
Albert  19    H   1.76 Bowling
Josep   18    H   1.65 Lectura
>
```

4.5 Extensió de data frames

Els data frame es poden extendre afegint tant files com columnes usant els operadors específics del llenguatge R.

4.5.1 Afegir una fila

Afegir una fila a un data frame significa afegir un individu, és a dir un valor concret per a cadascuna de les variables. La funció corresponent és

```
rbind()
```

Per a poder afegir un individu (una fila) al data frame, com que les dades a afegir són heterogènies, possiblement de diferents tipus, cal afegir un altre *data frame* o un *list*. Aquest nou element cal que tingui necessàriament el mateix nombre de columnes que el data frame que es vol augmentar. Un exemple que il.lustra aquesta funció és:

```
> df
      Edat Sexe Alcada   Hobby
Joan    18    H   1.81  Futbol
Anna    20    M   1.67 Lectura
Maria   15    M   1.60   Danca
Albert  19    H   1.76 Bowling
Jana    23    M   1.70  Musica
Josep   18    H   1.65 Lectura
> rbind(df, data.frame(Edat=22,Sexe="M",Alcada=1.70,
    Hobby="Tenis",
                       stringsAsFactors=FALSE))
      Edat Sexe Alcada   Hobby
Joan    18    H   1.81  Futbol
Anna    20    M   1.67 Lectura
Maria   15    M   1.60   Danca
Albert  19    H   1.76 Bowling
Jana    23    M   1.70  Musica
Josep   18    H   1.65 Lectura
7       22    M   1.70   Tenis
>
```

Fixeu-vos que, afegint data frame, cal tenir cura de definir l'opció `stringsAsFactors=FALSE` si algun dels camps del data frame és una cadena de caràcters.

Afegir una nova fila a un data frame expressada com un *list* s'aconsegueix amb la sintaxi

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
> NovaFila <- list(Edat=22, Sexe="M", Alcada=1.70,
    Hobby="Tenis")
> rbind(df, NovaFila)
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
7        22    M   1.70   Tenis
>
```

També es poden afegir diverses files, és a dir, es pot afegir un data frame a un altre data frame.

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
>
> df1 <- data.frame(Edat=c(17,21,15),Sexe=c("H","M","H
    "),
                          Alcada=c(1.70,1.71,1.65),
                             Hobby=c("Bowling","Danca",
                             "Musica"),
                          stringsAsFactors=FALSE)
>
> df1
  Edat Sexe Alcada   Hobby
1   17    H   1.70 Bowling
2   21    M   1.71   Danca
3   15    H   1.65  Musica
>
```

```
> rbind(df, df1)
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
7        17    H   1.70 Bowling
8        21    M   1.71   Danca
9        15    H   1.65  Musica
>
```

4.5.2 Afegir una nova columna

Afegir una columna a un data frame significa afegir una variable, és a dir, afegir un nou valor a tots i cadascun dels individus del data frame. La funció és

```
cbind()
```

Per a poder afegir una variable o columna al data frame, les dades poden estar emmagatzemades directament en un vector, perquè són homogènies. L'única restricció que cal tenir en compte és que el nombre de components del vector ha de ser igual al nombre de files del data frame.

```
> Ciutat <- c("Paris","Barcelona","Barcelona","Roma","
    Caracas","Barcelona","Paris")
> df <- rbind(df, list(22,"M",1.70,"Tenis"))
> df <- cbind(df, Ciutat)
>
> df
       Edat Sexe Alcada   Hobby    Ciutat
Joan     18    H   1.81  Futbol     Paris
Anna     20    M   1.67 Lectura Barcelona
Maria    15    M   1.60   Danca Barcelona
Albert   19    H   1.76 Bowling      Roma
Jana     23    M   1.70  Musica   Caracas
Josep    18    H   1.65 Lectura Barcelona
7        22    M   1.70   Tenis     Paris
>
```

També es pot afegir una columna al data frame tal i com es feia en el cas del *list*, és a dir, aplicant directament l'operador "$":

```
> df
       Edat Sexe Alcada   Hobby
Joan     18    H   1.81  Futbol
Anna     20    M   1.67 Lectura
Maria    15    M   1.60   Danca
Albert   19    H   1.76 Bowling
Jana     23    M   1.70  Musica
Josep    18    H   1.65 Lectura
```

```
> df$CodiPostal <- c
   (08027,08003,08014,08034,08034,08006)
> df
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H   1.81  Futbol       8027
Anna     20    M   1.67 Lectura       8003
Maria    15    M   1.60   Danca       8014
Albert   19    H   1.76 Bowling       8034
Jana     23    M   1.70  Musica       8034
Josep    18    H   1.65 Lectura       8006
>
```

Notis que, a diferència de la funció `cbind()`, afegir directament columnes modifica el data frame.

4.6 Modificació de data frames

Hi ha alguns problemes on cal modificar parcialment els data frames, és a dir, cal modificar els valors d'algunes variables. En aquest cas, la funció de l'R que permet fer aquestes modificacions és

`transform()`

A continuació podem veure alguns exemples d'ús d'aquesta funció.

```
> df
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H   1.81  Futbol       8027
Anna     20    M   1.67 Lectura       8003
Maria    15    M   1.60   Danca       8014
Albert   19    H   1.76 Bowling       8034
Jana     23    M   1.70  Musica       8034
Josep    18    H   1.65 Lectura       8006
> transform(df, Alcada=Alcada*100)
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H    181  Futbol       8027
Anna     20    M    167 Lectura       8003
Maria    15    M    160   Danca       8014
Albert   19    H    176 Bowling       8034
Jana     23    M    170  Musica       8034
Josep    18    H    165 Lectura       8006
> transform(df,Alcada=Alcada%/%2.54) #passem cm a
   polzades
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H  71.26  Futbol       8027
Anna     20    M  65.75 Lectura       8003
Maria    15    M  62.99   Danca       8014
Albert   19    H  69.29 Bowling       8034
Jana     23    M  66.93  Musica       8034
Josep    18    H  64.96 Lectura       8006
> transform(df,Alcada=Alcada%/%12,CodiPostal=
   CodiPostal%%100) #passem polzades a peus, reduim cp
```

```
        Edat Sexe Alcada   Hobby CodiPostal
Joan      18    H   5.94  Futbol         27
Anna      20    M   5.48 Lectura          3
Maria     15    M   5.25   Danca         14
Albert    19    H   5.77 Bowling         34
Jana      23    M   5.58  Musica         34
Josep     18    H   5.41 Lectura          6
>
```

4.7 Ordenació de data frames

Els data frames poden ser ordenats segons una o diverses columnes. La funció que cal utilitzar és

```
order()
```

L'ordenació es fa de manera creixent o decreixent en funció del paràmetre `decreasing` que pren valor cert o fals. Per defecte pren valor `FALSE` i l'ordenació és fa segons valors creixents. Per exemple, hom pot ordenar el data frame `df` de més amunt segons la variable `Edat` i en ordre creixent com segueix

```
> df
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H   1.81  Futbol       8027
Anna     20    M   1.67 Lectura       8003
Maria    15    M   1.60   Danca       8014
Albert   19    H   1.76 Bowling       8034
Jana     23    M   1.70  Musica       8034
Josep    18    H   1.65 Lectura       8006
>
> df[order(df$Edat),]
       Edat Sexe Alcada   Hobby CodiPostal
Maria    15    M   1.60   Danca       8014
Joan     18    H   1.81  Futbol       8027
Josep    18    H   1.65 Lectura       8006
Albert   19    H   1.76 Bowling       8034
Anna     20    M   1.67 Lectura       8003
Jana     23    M   1.70  Musica       8034
>
```

Notis la coma que apareix al darrere de la columna seleccionada `$Edat),`. L'ordenació decreixent seria

```
> df
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H   1.81  Futbol       8027
Anna     20    M   1.67 Lectura       8003
Maria    15    M   1.60   Danca       8014
Albert   19    H   1.76 Bowling       8034
Jana     23    M   1.70  Musica       8034
Josep    18    H   1.65 Lectura       8006
>
```

```
> df[order(df$Edat, decreasing=TRUE),]
       Edat Sexe Alcada   Hobby CodiPostal
Jana     23    M   1.70  Musica        8034
Anna     20    M   1.67 Lectura        8003
Albert   19    H   1.76 Bowling        8034
Joan     18    H   1.81  Futbol        8027
Josep    18    H   1.65 Lectura        8006
Maria    15    M   1.60   Danca        8014
>
```

Una ordenació creixent simultàniament segons `Edat` i `Alcada` seria

```
> df
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H   1.81  Futbol        8027
Anna     20    M   1.67 Lectura        8003
Maria    15    M   1.60   Danca        8014
Albert   19    H   1.76 Bowling        8034
Jana     23    M   1.70  Musica        8034
Josep    18    H   1.65 Lectura        8006
> df[order(df$Edat, df$Alcada),]
       Edat Sexe Alcada   Hobby CodiPostal
Maria    15    M   1.60   Danca        8014
Josep    18    H   1.65 Lectura        8006
Joan     18    H   1.81  Futbol        8027
Albert   19    H   1.76 Bowling        8034
Anna     20    M   1.67 Lectura        8003
Jana     23    M   1.70  Musica        8034
>
```

També podríem afegir, alhora, una selecció per columnes i mostrar només un subrang d'elles, com per exemple:

```
> df
       Edat Sexe Alcada   Hobby CodiPostal
Joan     18    H   1.81  Futbol        8027
Anna     20    M   1.67 Lectura        8003
Maria    15    M   1.60   Danca        8014
Albert   19    H   1.76 Bowling        8034
Jana     23    M   1.70  Musica        8034
Josep    18    H   1.65 Lectura        8006
> df[order(df$Edat, df$Alcada), 2:4]
       Sexe Alcada   Hobby
Maria     M   1.60   Danca
Josep     H   1.65 Lectura
Joan      H   1.81  Futbol
Albert    H   1.76 Bowling
Anna      M   1.67 Lectura
Jana      M   1.70  Musica
>
```

4.8 Fusió de data frames

El llenguatge R permet construir data frames a partir de la fusió de dos data frames donats. La fusió es fa a partir d'una variable (columna) que els data frames tinguin en comú. El data frame resultant estarà compost per

1. La unió de les columnes dels dos data frames. Notis que això vol dir que no es repeteixen les que tinguin en comú.

2. Les files dels dos data frames tals que tenen el mateix índex de fila i el valor de la columna que tenen en comú els data frames són iguals.

La sintaxi de la funció és

```
merge(df1, df2)
```

Un exemple de data frame resultant de la fusió dels data frames `df1` i `df2` és

```
> df1
  Var1 Var2 Var3
1   A1   B4   C1
2   A2   B2   C2
3   A3   B1   C3
4   A4   B2   C1
5   A2   B7   C3
6   A4   B3   C4
7   A5   B6   C3
8   A6   B1   C3
>
> df2
  Var2 Var4
1   B1   D1
2   B2   D2
3   B3   D1
4   B4   D3
5   B5   D5
>
>
> merge(df1, df2)
>
  Var2 Var1 Var3 Var4
1   B1   A3   C3   D1
2   B1   A6   C3   D1
3   B2   A2   C2   D2
4   B2   A4   C1   D2
5   B3   A4   C4   D1
6   B4   A1   C1   D3
>
```

També podem fer servir el paràmetre `all` per indicar que el *merge* inclogui totes les files malgrat que hi hagi valors de la columna compartida que no siguin iguals. Per exemple

```
> df1
  Var1 Var2 Var3
1   A1   B4   C1
2   A2   B2   C2
3   A3   B1   C3
4   A4   B2   C1
5   A2   B7   C3
6   A4   B3   C4
7   A5   B6   C3
8   A6   B1   C3
>
> df2
>
  Var2 Var4
1   B1   D1
2   B2   D2
3   B3   D1
4   B4   D3
5   B5   D5
>
> merge(df1, df2, all = TRUE)
>
  Var2 Var1 Var3 Var4
1   B1   A3   C3   D1
2   B1   A6   C3   D1
3   B2   A2   C2   D2
4   B2   A4   C1   D2
5   B3   A4   C4   D1
6   B4   A1   C1   D3
7   B5 <NA> <NA>   D5
8   B6   A5   C3 <NA>
9   B7   A2   C3 <NA>
>
```

Notis que quan a les files on hi ha valors no compartits en la columna `Var2` prenen per valor `<NA>`.

Quan hi ha columnes (variables) que tenen valors comuns a dos data frames però els seus noms no coincideixin es pot aplicar l'operador `merge` fent servir els paràmetres `by.x` i `by.y` per indicar el nom de la variable a considerar tan a l'operador `x` com a l'operador `y`.

```
> df1
  Var1 Var2 Var3
1   A1   B4   C1
2   A2   B2   C2
3   A3   B1   C3
4   A4   B2   C1
5   A2   B7   C3
6   A4   B3   C4
7   A5   B6   C3
8   A6   B1   C3
```

```
> df2
>
  Dif2 Var4
1   B1   D1
2   B2   D2
3   B3   D1
4   B4   D3
5   B5   D5
>
> merge(df1, df2, by.x = "Var2", by.y = "Dif2")
>
  Var2 Var1 Var3 Var4
1   B1   A3   C3   D1
2   B1   A6   C3   D1
3   B2   A2   C2   D2
4   B2   A4   C1   D2
5   B3   A4   C4   D1
6   B4   A1   C1   D3
>
```