Capítol 4

Data Frames

Un **data frame** *df* és un objecte **estructurat**, de dues dimensions, on les columnes són dades d'un mateix tipus (de les columnes se'n diuen *variables*) i les files són un conjunt de dades de diferent tipus (de les files se'n diuen *individus*). 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 $\forall i \in \{1, \dots, F\}\ \forall j \in \{1, \dots, C\}$, $df_{i,j} \in T_j$. Dit d'altra manera, al data frame *df* li podem associat la dimensió $F \times C$. En un data frame, cada fila i és un objecte de tipus list (no homogeni) mentre que cada columna j és un objecte de tipus vector (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 ens permet representar les dades recollides mitjançant l'aplicació de 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:

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

Veient aquest exemple com 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 posible resposta a la enquesta. Com hem dit abans, cada fila pot ser vista com un list, com ara $\langle 23, M, 1.70, Música \rangle$ i cada columna com a un vector, per exemple el vector de seqüencia de caràcters $\langle Fútbol, Lectura, Dança, Bowling, Música, Lectura \rangle$.

4.1 Creació de data frames al llenguatge R

Per a construir un data frame, usarem l'operador següent:

1. `data.frame()`

   Aquest constructor rep com a paràmetres les columnes del data frame a crear. Com en el cas dels list, podem donar nom a aquestes columnes (variables) del data frame, per tant podem construir un data frame fent:

   $$data.frame(nomcol_1 = valcol_1, \ldots, nomcol_k = valcol_k, stringsAsFactors = FALSE)$$

```
1 > df <- data.frame(Nom=c("Maria","Kim","Pep"),Edat
    =c(18,20,23), stringsAsFactors=FALSE)
2 > df
3       Nom Edat
4 1  Maria   18
5 2    Kim   20
6 3    Pep   23
7 >
```

   Però a diferència del list, en aquest cas també podem aprofitar-nos del nom de les variables de R per a donar nom als camps del data frame. Així doncs podem fer el següent:

```
1 > Edat <- c(18,20,15,19,23,18)
2 > Sexe <- c("H","M","M","H","M","H")
3 > Alcada <- c(1.81,1.67,1.60,1.76,1.70,1.65)
4 > Hobby <- c("Futbol","Lectura","Danca","Bowling",
    "Musica","Lectura")
5 > 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 paràmetre $stringsAsFactors$ indica a l'R com s'han de tractar les cadenes de caràcters. Si no li posem $stringsAsFactors = FALSE$, per defecte, tindrà el valor $TRUE$ 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 necesitem crear data frames amb columnes on hi hagi cadenes de caràcters hem de posar $stringsAsFactors = FALSE$.

També podem crear un data frame buit.

```
> df1 <- data.frame()
> df1
data frame with 0 columns and 0 rows
>
```

2. Generalment els data frames són objectes grans i ens interessarà construir-los a partir dels dades d'un fitxer CSV (usualment les dades estan separades per un espai). En aquest cas, fem servir un operador de lectura de fitxers `read.table` que s'utilitza com segueix:

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

   El paràmetre $header = TRUE$ es fa servir per dir que la primera línia conté les capçaleres del data frame (els noms de les variables). Si no es posa, per defecte es $header = FALSE$.

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

4.2 Fent servir operadors del tipus list

Ja hem explicat que un data frame es pot veure com un list, per tant podrem fer servir tots els operadors que hem vist al capítol anterior.

1. `names()`

```
> names(df)
[1] "Edat"   "Sexe"   "Alcada" "Hobby"
>
```

2. Accés a cada columna:

```
> df[[1]]
[1] 18 20 15 19 23 18
> df$Edat
[1] 18 20 15 19 23 18
> df[["Hobby"]]
[1] "Futbol"  "Lectura" "Danca"   "Bowling" "
    Musica"  "Lectura"
>
```

4.3 Fent servir operadors del tipus matriu

Però com ja hem dit un data frame també es pot veure com una matriu, per tant també podem fer servir operadors del tipus matriu i accedir als elements de forma matricial.

1. `nrow()`, `ncol()` i `dim()`

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

Important En particular, quan fem servir operadors d'indexat, hem de vigilar que l'avaluació de l'expressió corresponent a l'índex es trobi al rang de valors adequats. En general hem de tenir molta cura amb el fet que les funcions utilitzades tinguin un valor definit per als paràmetres. Seguidament podem veure un exemple de problemes d'aquest tipus.

```
> df1
data frame with 0 columns and 0 rows
> nrow(df1)
[1] 0
> ncol(df1)
[1] 0
> ndim(df1)
Error: no se pudo encontrar la funcion "ndim"
```

2. **Accés a una columna sencera**.

   Recodem que cada columna és un vector, per tant, un objecte homogeni.

```
> 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
> df[,2]
[1] "H" "M" "M" "H" "M" "H"
> df[,3]
[1] 1.81 1.67 1.60 1.76 1.70 1.65
>
```

3. **Accés a una fila sencera**.

   Recodem que cada fila és un list per tant, un objecte heterogeni.

```
> 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
> df[2,]
  Edat Sexe Alcada   Hobby
2   20    M   1.67 Lectura
> df[4,]
  Edat Sexe Alcada   Hobby
4   19    H   1.76 Bowling
```

4. **Accés a un component d'un data frame** (accés matricial):

```
> 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
> df[1,3]
[1] 1.81
> df[3,4]
[1] "Danca"
> df[2,5]
```

```
NULL
> df[5,6]
NULL
>
```

Noteu que quan tractem d'accedir a elements que no estan definits al data frame, l'operador ens torna $NULL$, que no és un objecte vàlid. Això vol dir que s'ha de tenir molta cura quan es fa servir l'indexat.

4.4 Subdata frames

1. De la mateixa manera que de les matrius podíem extreure submatrius, dels data frames podem extreure subdata frames. Els operands a fer servir són els mateixos que fèiem servir per extreure submatrius.

```
> 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
> df[2:4,]
  Edat Sexe Alcada   Hobby
2   20    M   1.67 Lectura
3   15    M   1.60   Danca
4   19    H   1.76 Bowling
> df[,1:3]
  Edat Sexe Alcada
1   18    H   1.81
2   20    M   1.67
3   15    M   1.60
4   19    H   1.76
5   23    M   1.70
6   18    H   1.65
> df[2:4,1:3]
  Edat Sexe Alcada
2   20    M   1.67
3   15    M   1.60
4   19    H   1.76
> df[2:4,3]
[1] 1.67 1.60 1.76  #Noteu que en aquest cas el
    resultat es un vector
```

Com hem vist, l'accés a les dades d'una única columna del data frame ens retorna aquestes dades en un vector. Si el que volem és que ens retorni un data frame amb una columna en lloc d'un vector el que cal és afegir el paràmetre $drop = FALSE$ en l'accés com segueix:

```
> df[2:4,3, drop=FALSE]
```

```
  Alcada
2   1.67
3   1.60
4   1.76
>
```

2. Filtres
   Moltes vegades el que volem és filtrar els data frames amb una condició lògica repecte d'una columna. És a dir, extreure un subdata frame que satisfaci una condició. Sigui D un data frame, la sentència seria:

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

```
> 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
> df[df$Alcada < 1.70,]
  Edat Sexe Alcada   Hobby
2   20    M   1.67 Lectura
3   15    M   1.60   Danca
6   18    H   1.65 Lectura
> df[df$Alcada > 1.70, 2:4]
  Sexe Alcada   Hobby
1    H   1.81  Futbol
4    H   1.76 Bowling
>
```

3. Selecció de files.
   També es pot fer una selecció utilitzant la funció `subset`, que permet fer un subdata frame amb el mateix nombre de columnes però només aquelles files que compleixin la condició:

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

```
> 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
> subset(df, Sexe == "H")
  Edat Sexe Alcada   Hobby
1   18    H   1.81  Futbol
4   19    H   1.76 Bowling
6   18    H   1.65 Lectura
```

```
> subset(df, Sexe == "H" & Alcada >= 1.65)
  Edat Sexe Alcada   Hobby
1   18    H   1.81  Futbol
4   19    H   1.76 Bowling
6   18    H   1.65 Lectura
>
```

4.5 Extensió de data frames

Els data frame es poden extendre afegint tant files com columnes fent servir els següents operadors.

1. `rbind()` per afegir una fila, és a dir un individu.

   Per a poder afegir un individu (una fila) al data frame, com que les dades a afegir són heterogènies (de diferent tipus), el que s'ha d'afegir és un altre data frame o un list. Aquest nou data frame (o list) ha de tenir necessàriament el mateix nombre de columnes que el data frame que es vol augmentar.

```
> 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
> rbind(df, data.frame(Edat=22,Sexe="M",Alcada=1
    .70,Hobby="Tenis",stringsAsFactors=FALSE))
  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
7   22    M   1.70   Tenis
>
```

   Fixeu-vos que, afegint data frame, cal tenir cura de posar l'opció stringsAsFactors a FALSE si algun dels camps del data frame és una cadena de caràcters. Ara veurem com afegir files mitjançant un list.

```
> 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
> NuevaFila <- list(Edat=22, Sexe="M", Alcada=1.70
   , Hobby="Tenis")
> rbind(df, NuevaFila)
  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
7   22    M   1.70   Tenis
>
```

Addicionalment, podem afegir més d'una fila, és a dir, combinar data frames:

```
> 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
> 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
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
7   17    H   1.70 Bowling
8   21    M   1.71   Danca
9   15    H   1.65  Musica
>
```

2. `cbind()` per afegir una nova columna (variable).

   Per a poder afegir una variable (una columna) al data frame, les dades poden estar guardades directament en un vector, perquè són homogènies

(del mateix tipus). L'única restricció que tenim en aquest cas és que el vector ha de tenir necessàriament el mateix nombre d'elements que files té el 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
1   18    H   1.81  Futbol     Paris
2   20    M   1.67 Lectura Barcelona
3   15    M   1.60   Danca Barcelona
4   19    H   1.76 Bowling      Roma
5   23    M   1.70  Musica   Caracas
6   18    H   1.65 Lectura Barcelona
7   22    M   1.70   Tenis     Paris
>
```

3. També podem afegir una columna tal i com fèiem en el list, és a dir fent servir directament l'operador "$":

```
> 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
> df$CodiPostal <- c
   (08027,08003,08014,08034,08034,08006)
> df
  Edat Sexe Alcada   Hobby CodiPostal
1   18    H   1.81  Futbol       8027
2   20    M   1.67 Lectura       8003
3   15    M   1.60   Danca       8014
4   19    H   1.76 Bowling       8034
5   23    M   1.70  Musica       8034
6   18    H   1.65 Lectura       8006
>
```

Noteu que, a diferència de la funció `cbind()`, aquesta manera directa d'afegir columnes modifica directament el data frame.

4.6 Modificació de data frames

Hi ha alguns problemes on podem necessitar modificar parcialment els data frames. És a dir, podem necessitar modificar els valors d'algunes variables. En aquest cas, podem fer servir una funció de l'R que ens permet fer aquestes

modificacions, `transform()`. A continuació podem veure alguns exemples d'ús d'aquesta funció.

```
> df
  Edat Sexe Alcada   Hobby CodiPostal
1   18    H   1.81  Futbol       8027
2   20    M   1.67 Lectura       8003
3   15    M   1.60   Danca       8014
4   19    H   1.76 Bowling       8034
5   23    M   1.70  Musica       8034
6   18    H   1.65 Lectura       8006
> transform(df, Alcada=Alcada*100)
  Edat Sexe Alcada   Hobby CodiPostal
1   18    H    181  Futbol       8027
2   20    M    167 Lectura       8003
3   15    M    160   Danca       8014
4   19    H    176 Bowling       8034
5   23    M    170  Musica       8034
6   18    H    165 Lectura       8006
> transform(df, Alcada=Alcada%/%2.54)  # passem cm a
   polzades
   Edat Sexe Alcada   Hobby CodiPostal
1    18    H  71.26  Futbol       8027
2    20    M  65.75 Lectura       8003
3    15    M  62.99   Danca       8014
4    19    H  69.29 Bowling       8034
5    23    M  66.93  Musica       8034
6    18    H  64.96 Lectura       8006
> transform(df, Alcada=Alcada%/%12, CodiPostal=
   CodiPostal%%100) # passem polzades a peus i reduim
   cp
   Edat Sexe Alcada   Hobby CodiPostal
1    18    H   5.94  Futbol         27
2    20    M   5.48 Lectura          3
3    15    M   5.25   Danca         14
4    19    H   5.77 Bowling         34
5    23    M   5.58  Musica         34
6    18    H   5.41 Lectura          6
>
```

4.7 Ordenació de data frames

Els data frames poden ser ordenats per una o més d'una columna. Per exemple, podem necessitar ordenar el data frame `df` segons la variable `Edat`, en ordre creixent. Per a fer això cal fer servir l'operador `order` con segueix:

```
> df
  Edat Sexe Alcada   Hobby CodiPostal
1   18    H   1.81  Futbol       8027
2   20    M   1.67 Lectura       8003
```

```
3   15    M   1.60   Danca       8014
4   19    H   1.76 Bowling       8034
5   23    M   1.70  Musica       8034
6   18    H   1.65 Lectura       8006
> df[order(df$Edat),]
  Edat Sexe Alcada   Hobby CodiPostal
3   15    M   1.60   Danca       8014
1   18    H   1.81  Futbol       8027
6   18    H   1.65 Lectura       8006
4   19    H   1.76 Bowling       8034
2   20    M   1.67 Lectura       8003
5   23    M   1.70  Musica       8034
>
```

També podem ordenar el data frame en ordre decreixent posant el flag `decreasing` a TRUE en la funció `order` (per defecte és FALSE):

```
> df
  Edat Sexe Alcada   Hobby CodiPostal
1   18    H   1.81  Futbol       8027
2   20    M   1.67 Lectura       8003
3   15    M   1.60   Danca       8014
4   19    H   1.76 Bowling       8034
5   23    M   1.70  Musica       8034
6   18    H   1.65 Lectura       8006
> df[order(df$Edat, decreasing=TRUE),]
  Edat Sexe Alcada   Hobby CodiPostal
5   23    M   1.70  Musica       8034
2   20    M   1.67 Lectura       8003
4   19    H   1.76 Bowling       8034
1   18    H   1.81  Futbol       8027
6   18    H   1.65 Lectura       8006
3   15    M   1.60   Danca       8014
>
```

I ara, creixent per `Edat` i per `Alcada`

```
> df
  Edat Sexe Alcada   Hobby CodiPostal
1   18    H   1.81  Futbol       8027
2   20    M   1.67 Lectura       8003
3   15    M   1.60   Danca       8014
4   19    H   1.76 Bowling       8034
5   23    M   1.70  Musica       8034
6   18    H   1.65 Lectura       8006
> df[order(df$Edat, df$Alcada),]
  Edat Sexe Alcada   Hobby CodiPostal
3   15    M   1.60   Danca       8014
6   18    H   1.65 Lectura       8006
1   18    H   1.81  Futbol       8027
4   19    H   1.76 Bowling       8034
2   20    M   1.67 Lectura       8003
```

```
5   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 CodigoPostal
1   18    H   1.81  Futbol         8027
2   20    M   1.67 Lectura         8003
3   15    M   1.60   Danca         8014
4   19    H   1.76 Bowling         8034
5   23    M   1.70  Musica         8034
6   18    H   1.65 Lectura         8006
> df[order(df$Edat, df$Alcada), 2:4]
  Sexe Alcada   Hobby
3    M   1.60   Danca
6    H   1.65 Lectura
1    H   1.81  Futbol
4    H   1.76 Bowling
2    M   1.67 Lectura
5    M   1.70  Musica
>
```

4.8 Fusió de data frames

`merge(x,y)`: A l'R és possible construir data frames a partir de la fusió de dos data frames `x` i `y`. Els data frames es combinen segons els valors d'una variable (columna) comú. El data frame resultant de fusionar dos data frames tindrà les columnes dels dos data frames -sense repetir les que siguin comuns- i només les files d'ambdós data frames on els valors de la columna en comú es correspon en els dos data frames:

```
> D1
  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
>
> D2
>
  Var2 Var4
1   B1   D1
2   B2   D2
3   B3   D1
```

```
4   B4   D3
5   B5   D5
>
> merge(D1,D2)
>
  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 volem el resultat del `merge` amb totes les files malgrat que hi hagi valors de la columna compartida que no siguin comuns. Fixeu-vos que a les files on hi ha valors no compartits en la columna `Var2` es coloca `<NA>` en els espais on no existeix valor a colůlocar:

```
> D1
  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
>
> D2
>
  Var2 Var4
1   B1   D1
2   B2   D2
3   B3   D1
4   B4   D3
5   B5   D5
>
> merge(D1,D2,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>
```

```
33 >
```

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`.

```
1  > D1
2    Var1 Var2 Var3
3  1   A1   B4   C1
4  2   A2   B2   C2
5  3   A3   B1   C3
6  4   A4   B2   C1
7  5   A2   B7   C3
8  6   A4   B3   C4
9  7   A5   B6   C3
10 8   A6   B1   C3
11 >
12 > D2
13 >
14   Dif2 Var4
15 1   B1   D1
16 2   B2   D2
17 3   B3   D1
18 4   B4   D3
19 5   B5   D5
20 >
21 > merge(D1,D2,by.x = "Var2",by.y = "Dif2")
22 >
23   Var2 Var1 Var3 Var4
24 1   B1   A3   C3   D1
25 2   B1   A6   C3   D1
26 3   B2   A2   C2   D2
27 4   B2   A4   C1   D2
28 5   B3   A4   C4   D1
29 6   B4   A1   C1   D3
30 >
```