Universidad
Carlos III de Madrid

Lenguaje R

José M. Valls Ferrán y Ricardo Aler Mur

Universidad
Carlos III de Madrid

Sesión 3: Dataframes

José M. Valls Ferrán y Ricardo Aler Mur

Data frames

- Es la estructura adecuada para almacenar tablas de datos, porque permiten combinar en una matriz distintos tipos de datos (números, cadenas, …)
- Recordar que las matrices sólo pueden almacenar el mismo tipo de datos
- Se puede considerar como una lista de elementos que son las columnas. Cada columna tiene el mismo número de elementos, que son las filas del dataframe
- Los data frames se crean normalmente con read.table () y read.csv(), leyendo los datos de ficheros
- Para ver y modificar los nombres de filas y columnas:
  - rownames()
  - colnames()

Data frames

- Creando un data frame (se hace por columnas)

```
mis.datos <-data.frame(nombre=c("luis","juan","antonio"),
            edad=c(25,30,29),  talla=c(1.75, 1.70, 1.83))
```

```
> mis.datos
   nombre edad talla
1    luis   25  1.75
2    juan   30  1.70
3 antonio   29  1.83
```

- Hay un paquete “datasets” que contiene muchos conjuntos de datos conocidos (iris, airquality, etc..)

Data frames

- Ejemplo: airquality (medidas de la calidad del aire en Nueva York)

```
> head(airquality)
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5   1
2    36     118  8.0   72     5   2
3    12     149 12.6   74     5   3
4    18     313 11.5   62     5   4
5    NA      NA 14.3   56     5   5
6    28      NA 14.9   66     5   6
```

Data frames

- **rownames() y colnames()** nos da los nombres de las filas y de las columnas
- **names()** nos dará los nombres de las columnas, porque es una lista de columnas

```
> colnames(airquality)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
> names(airquality)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

- **ncol() y nrow()** sirve para ver el número de columnas y de filas
- **length()** nos dará el número de columnas, porque realmente es una lista de cc

```
> ncol(airquality)
[1] 6
> nrow(airquality)
[1] 153
> length(airquality)
[1] 6
```

Data frames. Acceso

- Acceso por índice
  - Igual que con las matrices

```
> airquality[3,2]
[1] 149
> airquality[c(2,5),2]
[1] 118 NA
> airquality[c(2,3,5),2]
[1] 118 149 NA
```

  - Podemos usar el nombre de la columna (o fila) en el índice

```
> airquality["3","Solar.R"]
[1] 149
> airquality[1:5,"Solar.R"]
[1] 190 118 149 313 NA
> airquality[,"Solar.R"]
.....................
```

todos los elementos de la columna

Data frames. Acceso

- Acceso por nombre de columna a todos los elementos de la columna
  - También puede accederse con el símbolo $ y el nombre de la columna sin comillas (se hace igual en las listas)

```
> airquality$Solar.R
  [1] 190 118 149 313  NA  NA 299  9
 [31] 279 286 287 242 186 220 264 12
 [61] 138 269 248 236 101 175 314 27

> airquality$Solar.R[1:5]
[1] 190 118 149 313  NA
```

Todos los elementos de la columna. Realmente es el elemento Solar.R de la lista airquality. **Esta columna es un vector**

  - No puede sustituirse el nombre detrás de $ por una variable que contenga ese nombre:

```
> n<-"Solar.R"
> airquality[1:5,n]
[1] 190 118 149 313  NA
> airquality$n
NULL
```

Es válido. Con la misma expresión [1:5,n] podríamos acceder a diferentes columnas

Es **inválido**. Sólo puede usarse el literal sin comillas

Data frames. Acceso por subsetting

- Obtener las filas que tienen el Ozono a NA

```
> airquality[is.na(airquality[,"Ozone"]),]
    Ozone Solar.R Wind Temp Month Day
5      NA      NA 14.3   56     5   5
10     NA     194  8.6   69     5  10
25     NA      66 16.6   57     5  25
26     NA     266 14.9   58     5  26
27     NA      NA  8.0   57     5  27
```

  - También se puede hacer con $

```
> airquality[is.na(airquality$Ozone),]
    Ozone Solar.R Wind Temp Month Day
5      NA      NA 14.3   56     5   5
10     NA     194  8.6   69     5  10
25     NA      66 16.6   57     5  25
26     NA     266 14.9   58     5  26
27     NA      NA  8.0   57     5  27
32     NA     286  8.6   78     6   1
```

Data frames. Acceso por subsetting

- Hallar la media de las temperaturas en marzo

```
> temp.mayo <- airquality[airquality$Month==5,"Temp"]
> mean(temp.mayo)
[1] 65.54839
```

- Hallar la media del Ozono de todos los meses. Excluir los NA
  - Diciéndole a *mean()* que excluya los NA

```
> mean(airquality$Ozone, na.rm=T)
[1] 42.12931
```

  - Excluyendo previamente los NA

```
> oz <- airquality[!is.na(airquality$Ozone),"Ozone"]
> mean(oz)
[1] 42.12931
```

Leer data frames desde fichero

- Fichero con datos separados por espacios o tabuladores

"datos.txt"

```
454  3  2
23  34  4
12  3  5
14  7  8
```

```
m<-read.table("datos.txt")
```

```
> m
   V1 V2 V3
1 454  3  2
2  23 34  4
3  12  3  5
4  14  7  8
```

```
454,3,2
23,34,4
12,3,5
14,7,8
```

```
m<-read.csv("datos.txt", header=F)
```

Por defecto read.csv considera que hay cabecera y read.table que no hay

```
> m
   V1 V2 V3
1 454  3  2
2  23 34  4
3  12  3  5
4  14  7  8
```

Escribir data frames en un fichero

- write.table(m, “misdatos.txt”)

```
"V1" "V2" "V3"
"1" 454 3 2
"2" 23 34 4
"3" 12 3 5
"4" 14 7 8
```

- write.csv (m, “misdatos.csv”)

```
"","V1","V2","V3"
"1",454,3,2
"2",23,34,4
"3",12,3,5
"4",14,7,8
```

- write.table(m,"misdatos.txt", row.names=F, col.names=F)

```
454 3 2
23 34 4
12 3 5
14 7 8
```

- write.csv(m,"misdatos.txt", row.names=F, col.names=F)

  Warning message: In write.csv(m, "misdatos.txt", row.names = F, col.names = F) : attempt to set 'col.names' ignored

```
"V1","V2","V3"
454,3,2
23,34,4
12,3,5
14,7,8
```

Funciones adicionales. apply

- Función **apply(x, dim, func)**
  - Se aplica a matrices (matriz x)
  - Devuelve un vector de valores obtenidos al aplicar la función func a cada fila o columna de la matriz
  - Si dim=1, se aplica a las filas, devolviendo un vector de tantos elementos como filas.
  - Si dim=2, se aplica a las columnas, devolviendo un vector de tantos elementos como columnas
  - Cuidado: si se aplica a un data frame, previamente se convierte a matriz y produce el resultado
    - En el caso de que una columna sea character, todos se convierten a character y seguramente la función dará error (p. ej, las funciones numéricas)

Funciones adicionales. apply

- Ejemplos. Dada una matriz m
  - Obtener las sumas de las columnas

```
> apply(m,2,sum)
[1] 15 40 65
```

```
> m
     [,1] [,2] [,3]
[1,]    1    6   11
[2,]    2    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

  - Obtener las medias de las columnas

```
> apply(m,2,mean)
[1] 3 8 13
```

  - Obtener las sumas de las filas

```
> apply(m,1,sum)
[1] 18 21 24 27 30
```

  - Se podría aplicar a cualquier función que admita vectores (max, min, sd, etc...). Incluso podemos definirla nosotros sobre la marcha en el mismo apply. Ejemplo de función absurda pero válida:

```
> apply(m,1,f<-function(n){"hola"})
[1] "hola" "hola" "hola" "hola" "hola"
```

Funciones adicionales. apply

- Ejemplos. Si la matriz tiene NA
  - Obtener las sumas de las columnas

```
> m
     [,1] [,2] [,3]
[1,]    1    6   11
[2,]   NA    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```

```
> apply(m,2,sum)
[1] NA 40 65
```

  - Si queremos que la suma ignore los NA, se pone como un parámetro más de la función apply

```
> apply(m,2,sum, na.rm=T)
[1] 13 40 65
```

Funciones adicionales. apply

- Ejemplos. Si la matriz tiene NA
  - Obtener las sumas de las columnas

```
> apply(m,2,sum)
[1] NA 40 65
```

  - Si queremos que la suma ignore los NA

```
> apply(m,2,sum, na.rm=T)
[1] 13 40 65
```

```
> m
     [,1] [,2] [,3]
[1,]    1    6   11
[2,]   NA    7   12
[3,]    3    8   13
[4,]    4    9   14
[5,]    5   10   15
```