Data Frames

HR Analytics: Teoría y Práctica

http://pablohaya.com/contact

09/2018

Data Frame

Un *data frame* es una tabla de datos formada por columnas que pueden ser distinto tipo. Cada columna tiene asociado un nombre. La función `data.frame()` permite especifar los vectores que formarán las columnas, y el nombre de cada una de ellas.

```
df <- data.frame(nombre=c("Pedro", "Ana", "Mara", "Juan"),
                 edad=c(34, 35, 23, 54),
                 genero=(c("M","F","F", "M")))
df
```

```
##   nombre edad genero
## 1  Pedro   34      M
## 2    Ana   35      F
## 3   Mara   23      F
## 4   Juan   54      M
```

Como ya hemos visto, se puede utilizar la función `str()` para obtener un resumen del contenido almacenado.

```
str(df)
```

```
## 'data.frame':    4 obs. of  3 variables:
##  $ nombre: Factor w/ 4 levels "Ana","Juan","Mara",..: 4 1 3
##  $ edad  : num  34 35 23 54
##  $ genero: Factor w/ 2 levels "F","M": 2 1 1 2
```

También se puede emplear la función `dim()` si sólo se necesita saber el número de filas y columnas que tiene la tabla.

```
dim(df)
```

```
## [1] 4 3
```

Por defecto, las cadenas de caracteres se convierten en factores.

Si se quiere deshabilitar este comportamiento es preciso emplear el parámetro `stringAsFactors=FALSE`, e indicar explícitamente que columnas son factores.

```
df <- data.frame(nombre=c("Pedro", "Ana", "Mara", "Juan"),
                 edad=c(34, 35, 23, 54),
                 genero=as.factor(c("M","F","F", "M")),
                 stringsAsFactors = FALSE)
str(df)
```

```
## 'data.frame':    4 obs. of  3 variables:
##  $ nombre: chr  "Pedro" "Ana" "Mara" "Juan"
##  $ edad  : num  34 35 23 54
##  $ genero: Factor w/ 2 levels "F","M": 2 1 1 2
```

Es habitual que las tablas que se almacenen contenga mucha información, de manera que visualizar todo el contenido, aunque sea de manera parcial, puede ser tedioso.

Las funciones `head()` y `tail()` permiten inspeccionar los primeros y últimos elementos de la tabla.

```
data(USArrests)
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## Alabama      13.2     236       58 21.2
## Alaska       10.0     263       48 44.5
## Arizona       8.1     294       80 31.0
## Arkansas      8.8     190       50 19.5
## California    9.0     276       91 40.6
## Colorado      7.9     204       78 38.7
```

```r
tail(USArrests)
```

```
##                Murder Assault UrbanPop Rape
## Vermont           2.2      48        32 11.2
## Virginia          8.5     156        63 20.7
## Washington        4.0     145        73 26.2
## West Virginia     5.7      81        39  9.3
## Wisconsin         2.6      53        66 10.8
## Wyoming           6.8     161        60 15.6
```

```r
dim(USArrests)
```

```
## [1] 50  4
```

Indexado

Un *data frame* se indexa de manera similar a un vector añadiendo un índice más poder referenciar tanto la fila como la columna. El siguiente ejemplo obtiene el valor de la celda que identifica por la fila 1 y la columa 2.

```
df[1,2]
```

```
## [1] 34
```

De la misma manera que los vectores se pueden seleccionar varios elementos con la notación i:j. En el caso de las tablas, se pueden seleccionar varias filas, varias columnas o una subconjunto de ambas.

```
df[1:2, 1:3]
```

```
##   nombre edad genero
## 1  Pedro   34      M
## 2    Ana   35      F
```

Para seleccionar todas las filas, o todas las columnas basta con omitir el índice correspondiente y dejarlo vacío. Así, la siguiente expresión es equivalente a la que se empleo en la anterior transparencia.

```
# mismo resultado que df[1:2,1:3]
df[1:2,]
```

```
##   nombre edad genero
## 1  Pedro   34      M
## 2    Ana   35      F
```

Lo mismo pero seleccionando dos columnas:

```
df[,2:3]
```

```
##   edad genero
## 1   34      M
## 2   35      F
## 3   23      F
## 4   54      M
```

También es posible emplear un vector para indicar la selección de filas y/o columnas. Esto permite escoger filas/columnas no consecutivas.

```
df[c(1,4), 1]
```

```
## [1] "Pedro" "Juan"
```

En el caso de la columnas, se puede sustituir las posiciones por los nombres de las mismas.

```
df[c(1,4), c("edad", "genero")]
```

```
##   edad genero
## 1   34      M
## 4   54      M
```

Si sólo queremos acceder a una columna es más sencillo empleado la siguiente notación (`$nombre_columna`).

```
df$edad
```

```
## [1] 34 35 23 54
```

Anteriormente empleamos las posiciones de las filas para seleccionar aquellas que quería quedarnos. También se puede emplear un vector con valores booleanos.

```
df[c(FALSE, TRUE, TRUE, FALSE), ]
```

```
##   nombre edad genero
## 2    Ana   35      F
## 3   Mara   23      F
```

Como la comparación de un vector con un valor hace que obtengamos el valor de la comparación para elemento del vector, podemos emplearlo para seleccionar filas en función de un valor.

```
df$genero == "F"
```

```
## [1] FALSE  TRUE  TRUE FALSE
```

```
# df$genero == "F" equivalente a
# df[c(FALSE, TRUE, TRUE, FALSE), ]
# devuelve todas las filas cuyo genero sea "F"
df[df$genero == "F", ]
```

```
##   nombre edad genero
## 2    Ana   35      F
## 3   Mara   23      F
```

Ordenación

Se puede cambiar el orden de las filas incluyendo las nuevas posiciones en un vector que indexe la tabla. El siguiente ejemplo invierte el orden de la tabla.

```
df[c(4, 3, 2, 1),]
```

```
##   nombre edad genero
## 4   Juan   54      M
## 3   Mara   23      F
## 2    Ana   35      F
## 1  Pedro   34      M
```

La función `order()` devuelve las posiciones de los elementos para que estén ordenados de mayor a menor.

```
df$edad
```

```
## [1] 34 35 23 54
```

```
order(df$edad)
```

```
## [1] 3 1 2 4
```

El resultado de la función `order()` se puede emplear para ordenar las filas de una tabla tomando como criterio de ordenación una columna.

```
df[order(df$edad),]
```

```
##   nombre edad genero
## 3   Mara   23      F
## 1  Pedro   34      M
## 2    Ana   35      F
## 4   Juan   54      M
```

El parámetro `decreasing` permite establecer el dirección de la ordenación.

```
df[order(df$edad, decreasing = FALSE),]
```

```
##    nombre edad genero
## 3    Mara   23      F
## 1   Pedro   34      M
## 2     Ana   35      F
## 4    Juan   54      M
```