DATA FRAME

Ramon Ceballos

22/1/2021

VARIAR DATOS DE UN DATA FRAME

1. Cambiando los tipos de datos

Para cambiar los tipos de datos de un vector (la columna) se emplean las siguientes funciones.

- `as.character`: para transformar todos los datos de un objeto en palabras
- `as.integer`: para transformar todos los datos de un objeto a números enteros
- `as.numeric`: para transformar todos los datos de un objeto a números reales

2. Sub-data Frame

Cuando se realiza un Sub-data frame, hereda los factores contenidos en el data frame original pese a que no aprezcan en el sub-data frame obtenido.

Aunque una variable no aparezca en el trozo extraído, si no queremos que se guarde, se podrían borrar los niveles sobrantes de un factor redefiniendo el data frame con la función **droplevels()**.

- `droplevels(d.f)`: para borrar los niveles sobrantes de todos los factores, ya que las columnas que son factores heredan en los sub-data frames todos los niveles del factor original, aunque no aparezcan en el trozo que hemos extraído

```
gender = c("H", "M", "M", "M", "H")
age =    c( 23,  45,  20,  30,  18)
family = c(  2,   3,   4,   2,   5)
df5 = data.frame(genero = gender, edad = age, familia = family, stringsAsFactors = TRUE)
df5[df5$genero=="M", ] -> df_m
str(df_m)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ genero : Factor w/ 2 levels "H","M": 2 2 2
##  $ edad   : num  45 20 30
##  $ familia: num  3 4 2
```

```
df_m = droplevels(df_m)
str(df_m)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ genero : Factor w/ 1 level "M": 1 1 1
##  $ edad   : num  45 20 30
##  $ familia: num  3 4 2
```

Como intro a la librería **Tidyverse** tenemos a la función **select(df, parámetros)**.

- `select(d.f, parámetros)`: para especificar que queremos extraer de un data frame
  - `starts_with("x")`: extrae del data frame las variables cuyo nombre empieza con la palabra "x"
  - `ends_with("x")`: extrae del data frame las variables cuyo nombre termina con la palabra "x"
  - `contains("x")`: extrae del data frame las variables cuyo nombre contiene la palabra "x"
  - Se necesita el paquete `dplyr` o mejor aún `tidyverse`

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#Crear sub-data frame con las columnas que empiezan por "Petal"
iris_petal = select(iris, starts_with("Petal"))
head(iris_petal)
```

```
##   Petal.Length Petal.Width
## 1          1.4         0.2
## 2          1.4         0.2
## 3          1.3         0.2
## 4          1.5         0.2
## 5          1.4         0.2
## 6          1.7         0.4
```

```
#Crea sub-data frame con las columnas que acaban por "Length"
iris_length = select(iris, ends_with("Length"))
head(iris_length)
```

```
##   Sepal.Length Petal.Length
## 1          5.1          1.4
## 2          4.9          1.4
## 3          4.7          1.3
## 4          4.6          1.5
## 5          5.0          1.4
## 6          5.4          1.7
```

La sintaxis de **subset()** que sirve para extraer una subtabla a partir del data frame dado.

- `subset(d.f,condición,select = columnas)`: para extraer del data frame las filas que cumplen la condición y las columnas especificadas
  - Si queremos todas las filas, no hay que especificar ninguna condición

- Si queremos todas las columnas, no hace especificar el parámetro `select`
- Las variables en la condición se especifican con su nombre, sin añadir antes el nombre del data frame

```
subset(iris, Species == "versicolor", select = c(1,3)) -> versicolor

#Para que los identificadores no guarden la información del data frame hay que renombrarlos
#Se redimensionan desde 1 hasta numero de filas total (nrow)
rownames(versicolor) = 1:nrow(versicolor)

head(versicolor, 5)
```

```
##   Sepal.Length Petal.Length
## 1          7.0          4.7
## 2          6.4          4.5
## 3          6.9          4.9
## 4          5.5          4.0
## 5          6.5          4.6
```

```
str(versicolor)
```

```
## 'data.frame':    50 obs. of  2 variables:
##  $ Sepal.Length: num  7 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 ...
##  $ Petal.Length: num  4.7 4.5 4.9 4 4.6 4.5 4.7 3.3 4.6 3.9 ...
```