Manipular datos en R (avanzado)

Ejercicios

Julio 2019

Introducción

En los ejercicios de la tercera sesión vamos a trabajar sobre la manipulación avanzada de datos en `R`. Antes de empezar:

1. **Abre el script** `sesion3_ejercicios.R`. **Limpia** el espacio de datos ejecutando `rm(list = ls())`. Con este comando eliminarás del espacio de trabajo todos los datos (objetos) que estén disponibles evitando posibles confusiones[1].
2. **Carga los paquetes** que necesitas para realizar la práctica, ejecutando las líneas de `library()`. En caso de que alguno de ellos no esté instalado, instalaló utilizando `install.packages("package")`[2].
3. En la carpeta **data** encontrarás los ficheros de datos que vas a utilizar en esta práctica.

[1] Puedes usar el atajo `Ctrl + Enter` para ejecutar una línea de código en `RStudio`

[2] Usa comillas (" ") a la hora de instalar los paquetes con la función `install.packages()`

A. Agrupar casos y resumir variables

`group_by(.data, ...)` Devuelve un data frame agrupado a partir de `....`

`summarise(.data, ...)` Devuelve un resumen de las columnas de un data frame a partir de `....`

A.1 Carga el archivo `cis_oct17.sav` que se encuentra en la carpeta `data`. Recuerda hacer la transformación de los **objetos de labelled a factores** con `as_factor()`. Crea un nuevo data frame agrupado, en el que la variable de **agrupación** sea `idv` y genera un resumen que sea la media de la variable `edad`. Imprime el objeto final. ¿Cuál es el problema?

A.2 Convierte la variable `edad` a **numérica** y crea la **agrupación** por `idv`. Realiza todas las transformaciones dentro del mismo `pipe`.

`na_if(x, y)` Devuelve un vector `x` en el que los valores `y` han sido transformados en `NA`.

A.3 Crea una **variable a nivel individual** que sea la media de `valora_gob` para cada estrato resultante de la combinación de `region` y `tamuni`. Guarda la nueva variable como `valora_gob_estrato`. Ten en cuenta que antes de agrupar tendrás que preparar la variable, especificando los valores perdidos y determinando el tipo de variable.

A.4 Añade dos variables al data frame `cis` agrupado por `region` y `tamuni`. La primera será `idv_cat`, que se refiere a la **categoría más frecuente** de la variable `idv` en cada estrato. Además crea la variable `idv_por`, que será el **porcentaje de la categoría más frecuente** en el estrato.

B. Combinar data frames

B.1 Carga dos ficheros, el primero es `escuelas1.csv` y el segundo es `escuelas2.xlsx`. Estos ficheros contienen casos (filas) del mismo conjunto, **combínalas** para crear un archivo único. Comprueba el resultado.

`bind_rows(...)` Devuelve un data frame en el que se han fusionado las filas de `...`.

B.2 Carga los ficheros `.sav` `cis_oct17_cols1.sav` (`cols1`) y `cis_oct17_cols2.sav` (`cols2`). Haz las **tansformaciones necesarias** y **combina** los dos data frames en uno utilizando `bind_cols()`.

`bind_cols(...)` Devuelve un data frame en el que se han fusionado las columna de `...`.

B.3 Ahora utiliza ula función `left_join()` para unir las columnas usando la columna `id`. Asigna el objeto al nombre `cis`. Determina cuáles son las filas presentes en **cols1** que no estaban en **cols2**, para lo que puedes utilizar la función `anti_join()`. ¿Existen filas presentes en **cols2** que no estén presentes en **cols1**?

`left_join(x, y, by)` Devuelve un data frame en el que se han fusionado las columna de `x` e `y` a partir de la columna clave `by`. Se utiliza `x` como referencia.

`anti_join(x, y, by)` Devuelve un data frame en el que se han fusionado las columna de `x` e `y` a partir de la columna clave `by`. Se utiliza `x` como referencia.

B.4 Realiza una unión completa de `cols1`y `cols2`. ¿Cuántos casos hay en total?

`full_join(x, y, by)` Devuelve un data frame en el que se han fusionado las columna de `x` e `y` a partir de la columna clave `by`. Se utiliza como referencia tanto `x` como `y`.

C. Cambiar el formato de los datos

C.1 Los datos `result_aut_partidos.RDS` están en **formato largo**. Carga los datos y conviértelos en **formato ancho** usando la función `spread()` de forma que cada columna corresponda a un partido diferente. Asigna el objeto resultante a `elec_wide`.

`spread(data, key, value)` Devuelve un data frame de tipo ancho a partir de uno largo.

C.2 Ahora transforma el objeto `cis` en **formato largo**, de forma que la primera columna corresponda con la variable `id`, la segunda sea una columna con el nombre de las variables llamada `var` y la tercera sean los valores (`vals`). Para ello utiliza la función `gather()` y asigna el resultado al nombre `cis_long`. Ordena los resultados por la variable `id` y comprueba la **estructura de los datos**. ¿Qué tipo de objeto es la variable `value`?

`gather(data, key, value)` Devuelve un data frame de tipo largo a partir de uno ancho.

C.3 A partir del data frame `cis_long` usa la función `spread()` para cambiar el **formato a ancho de nuevo**. Guarda el data frame resultante como `cis_wide`. Comprueba de qué tipo son las variables resultantes.

D. Funciones básicas

D.1 Crea una **función** (`to_factor`) que a partir de un vector lo transforme en factor e imprima los niveles del nuevo factor. Aplícalo a la variable `tamuni` del data frame `cis_wide` y comprueba el resultado.

`function(...){}` Devuelve un a función con los argumentos determinados por `...`.

`print(x)` Imprime `x`.

E. Repaso general a la manipulación de datos

E.1 Carga el data frame `autoesc_jun_2017.csv` y asignalo al nombre `auto` a continuación:

1. Seleccionar los **casos** que **corresponden al permiso B** en la variable `NOMBRE_PERMISO`. Explora los datos y fíjate en la relación entre la variable `NOMBRE_AUTOESCUELA` y `TIPO_EXAMEN`.

2. Agrupar por autoescuela (`NOMBRE_AUTOESCUELA`) y `TIPO_EXAMEN`. Calcular para cada autoescuela el **total** de NUM_APTOS (`aptos`), NUM_NO_APTOS (`no_aptos`), NUM_APTOS_1conv (`aptos_1`), NUM_APTOS_2conv (`aptos_2`).

3. Calcula el **total de presentados** (`total_pres`) en cada autoescuela, que será la suma de aptos y no aptos.

4. Cambia las etiquetas de la variable `TIPO_EXAMEN` de forma que `PRUEBA CONDUCCIÓN Y CIRCULACIÓN` sea `Práctico` y `PRUEBA TEÓRICA` sea `Teórico`.

5. Calcula el porcentaje de aptos a la primera (`p_aptos_conv1`) y aptos a la segunda en cada autoescuela (`p_aptos_conv2`) sobre el total de aptos.

6. Selecciona las variables `NOMBRE_AUTOESCUELA`, `TIPO_EXAMEN`, `p_aptos_conv1` y `p_aptos_conv2`.

7. Cambia el formato del data frame de ancho a largo, de forma que las variables `p_aptos_conv*` queden en dos columnas, la primera `convocatoria`, y la segunda `p_aptos`.

8. Ordena los datos según `NOMBRE_AUTOESCUELA`, `TIPO_EXAMEN` y `convocatoria`.

9. Explora el **data frame** final.