

# ¿Todo esto para hacer una tabla?

## Ejercicios

Julio 2019

### Introducción

En los ejercicios de la cuarta sesión vamos a trabajar sobre la creación de tablas 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<sup>1</sup>.
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, instalalo utilizando `install.packages("package")`<sup>2</sup>.
3. En la carpeta **data** encontrarás los ficheros de datos que vas a utilizar en esta práctica.

<sup>1</sup> Puedes usar el atajo `Ctrl + Enter` para ejecutar una línea de código en RStudio

<sup>2</sup> Usa comillas (" ") a la hora de instalar los paquetes con la función `install.packages()`

### A. Tablas de frecuencia y descriptivos

**A.1** Carga el archivo de datos `cis_oct17.sav` y haz la conversión necesaria a **factores**. Posteriormente utiliza la función `table()` para generar **tablas de frecuencias** de las variables `pol_pros`, `idv`, `sexo`. Repite las frecuencias utilizando la función `frq()`.

`frq(x, ..., sort.frq, weights, show.na, out, title, file)` Devuelve una tabla de frecuencias.

**A.2** Exporta las tablas que has generado con `frq()` en **A.1** a un **archivo externo** de tipo HTML empleando los argumentos `out` y `file`. Haz que **no se muestren** los valores perdidos NA y que las tablas estén organizadas de forma **descendente** según la frecuencia absoluta. Abre el archivo HTML que has generado.

**A.3** Genera **tablas de frecuencias** de las variables `idv` y `pol_pros` **segmentadas** por la variable `sexo`.

`descr(x, ..., weights, out)` Devuelve una tabla de estadísticos descriptivos.

**A.4** Adapta las variables `edad` y `valora_gob` para ser **incluidas en una tabla de estadísticos descriptivos**. Crea una tabla de estadísticos descriptivos para estas dos variables segmentada por la variable `sexo`. Almacena el objeto con el nombre `descriptivos` y extrae la **media de edad de los hombres** y la **media de edad de las mujeres**. Resta la media de edad de los hombres a la de las mujeres.

## B. Tablas de contingencia

**B.1** Utiliza la función `sel` del paquete `base` `table()` para crear una **tabla de contingencia** de `pol_pros` y `sexo`. Realiza la misma tabla utilizando la función `flat_table()` pero en vez de las frecuencias incluye los porcentajes de columna.

`flat_table(data, ..., margin)`  
Devuelve una tabla de contingencia.

**B.2** Utiliza la función `crosstab` del paquete `descr` para crear la **tabla de contingencia** del ejercicio anterior (`pol_pros` y `sexo`). La tabla debe contener las frecuencias y los porcentajes de columna. No olvides descartar el gráfico con la opción `plot = FALSE`.

`crosstab(dep, indep, weight, prop.r, prop.c, prop.t, chisq, total.r, total.c, plot)` Devuelve una tabla de contingencia.

**B.3** Añade a la tabla que has realizado anteriormente el **porcentaje de filas** y un test **chi-cuadrado**.

## C. Frecuencias y tablas de contingencia con *expss*

**C.1** Utiliza la función `fre()` del paquete `expss` para crear **tablas de frecuencias** de las variables `pol_pros`, `idv` y `sexo`.

`fre(x, weight)` Devuelve una tabla de frecuencias.

**C.2** Crea una **tabla de contingencia** de casos que cruce las variables `pol_pros` y `sexo` usando la función `cro()` del paquete `expss`. Cambia la etiqueta del total con el argumento `total_label` a "Totales".

`cro(cell_vars, col_vars, row_vars, weight, total_label, total_row_position)` Devuelve una tabla de contingencia con frecuencias.

**C.3** Reproduce la tabla del ejercicio anterior pero en vez de las frecuencias muestra los **porcentajes de columna**. Incluye una columna de total, para ello haz una lista de los elementos que quieres incluir en `col_vars` utilizando la función `total()` para designar la columna de totales.

`cro_cpct(cell_vars, col_vars, row_vars, weight, total_label, total_row_position)` Devuelve una tabla de contingencia con porcentajes de columna.

**C.4** Utiliza la función `cro_mean_sd_n()` para realizar una tabla resumen de la variable `edad` para los valores de `pol_pros`.

`cro_mean_sd_n(cell_vars, col_vars, row_vars, weight)` Devuelve una tabla de contingencia con estadísticos descriptivos.

## D. Tablas personalizadas con *expss*

**D.1** Crea una **tabla personalizada** que incluya las variables `pol_pros` y `idv` en filas. Las variables de columna serán `sexo` y `edad` en 3 grupos 18-34, 35-64 y 65 o más. El contenido de la tabla serán las frecuencias cruzadas.

`tab_cells(...)` Permite incluir las variables de fila.

`tab_cols(...)` Permite incluir las variables de columna.

`tab_stat_cases(total_label, total_statistic, label)` Añade las frecuencias de las variables determinadas por `tab_cells()` a la tabla.

**D.2** Ahora añade a la tabla anterior el **porcentaje de columna** y una columna de totales. Utiliza el argumento `stat_position = "inside_rows"` de la función `tab_pivot()`.

`tab_pivot(stat_position, stat_label)` Devuelve la tabla personalizada.

`tab_stat_cpct(total_label, total_statistic, label)` Añade los porcentajes de columna de las variables determinadas por `tab_cells()` a la tabla.

**D.3** Ahora vas a **añadir etiquetas** a la tabla que has generado en el ejercicio anterior. Por un lado vas a etiquetar las filas según se trate de los "Casos" o el "% columna". Además, la etiqueta de los totales debe figurar como "Totales".

**D.4** Utilizando el paquete `openxlsx` **exporta la tabla a MS Excel**.

Recuerda que en primer lugar tienes que crear un libro `createWorkbook()`, después añadir un hoja `addWorksheet()` y posteriormente escribir la tabla `xlwrite()`. Por último, guarda el libro utilizando la función `saveWorkbook()`.

`createWorkbook()` Crea un libro de tipo `xlsx`.

`addWorksheet(wb, sheetName)` Añade una hoja al libro `xlsx`.

`xl_write(obj, wb, sheet)` Escribe un objeto `expss` en un objeto `xlsx`.

`saveWorkbook(wb, file, overwrite)` Permite guardar el libro `xlsx`.