**Semana 2**

**08. Funciones**

En esta lección vamos a aprender cómo escribir funciones. Las funciones se crean con la función especial *function()* y quedan almacenadas como objetos. Primero ponemos el nombre de la función y la asignación a *fuction()*, estando dentro de esta última los parámetros y a continuación dentro de las llaves todas las operaciones que lleva a cabo. El valor que genera una función es el último que está en el cuerpo de la función. Las funciones tienen dos particularidades, que pueden ser pasadas como argumentos a otras funciones y que pueden ser anidadas, englobando unas dentro de otras. Pueden tener varios argumentos o parámetros (con argumentos omitidos), pudiendo ser argumentos formales que se incluyen en la definición de la función y pueden ser consultados con la función *formals()* cuando el ambiente está dentro de la línea de comandos de R, regresando una lista con todos los parámetros de la función.

En R no es necesario pasar toda la lista de comandos formales debido a que puede ser que un parámetro no se esté utilizando y puede ser que el parámetro ya tenga un valor por omisión.

Los argumentos en R pueden ser coincidentes posicionalmente y por nombre, lo que quiere decir que siempre que estén nombrados podemos cambiarle el orden con el que están definidos, aunque es un poco confuso. Es importante recalcar que si cambiamos el nombre no se debe hacer de manera seguida. Hay 3 tipos de argumentos coincidentes:

* Coincidencia exacta
* Coincidencia parcial
* Coincidencia por posición

La manera en la que se evalúan una función en R es una evaluación floja o *lazy evaluation*, lo que quiere decir es que el argumento no se evalúa en la función hasta que no sea necesario.

El argumento “…” (tres puntos) se usa cuando no se sabe *a priori* el número de argumentos que vamos a pasar a una función o si un número variable de argumentos se pasan a otra función. Las funciones genéricas también usan el argumento tres puntos, llevando a cabo su función sobre distintos tipos de objetos. La función *paste* concatena conjuntos de parámetros. Es importante destacar que una vez que se usa el parámetro tres puntos, cualquier parámetro que vaya después de él debe ser nombrado explícitamente y no puede ser emparejado parcialmente.