

Nota2 - Intro

Santiago Casanova y Ernesto Barrios

Uso Básico

Ahora vamos a empezar a familiarizarnos con el ambiente de R y específicamente su sintaxis especial.

Declaración de variables

R tiene un operador especial para asignar valores que es diferente a otros lenguajes de programación. Además, las son flexibles y pueden pasar de contener un tipo de dato a otro sin problema. Por los mismo no es necesario especificar el tipo de dato como se hace en C o en Java. El operador de asignación es `<-`. También se puede asignar variables con `=` pero, por convención, `=` se reserva para operaciones dentro de funciones o paréntesis.

Vamos a asignar algunos valores a una serie de variables:

```
numeros <- 1234
numeros2 <- 1234.56
integer1 <- 4L
texto1 <- "ejemplo"
texto2 <- 'tambien se puede con comillas simples'
booleano <- T
booleano = TRUE
booleano2 = F
booleano2 = FALSE
```

```
typeof(numeros)
```

```
## [1] "double"
```

```
typeof(integer1)
```

```
## [1] "integer"
```

Como describimos, no se necesita especificar que tipo de dato queremos en cada variable ya que esto puede cambiar mas adelante. Hablando de tipos de datos, vamos a ver cuáles son las opciones que maneja R.

Tipos de Datos

1. Numérico: No hay diferencia inicialmente entre integer y double o float.
 - Si se quiere especificar solamente integer se puede escribir con una L al final. Ejemplo: 2L
2. Carácter: No hay diferencia entre string y character.
3. Booleano (lógico): Como vimos en el ejemplo anterior, se escribe TRUE o FALSE todo con mayúsculas, o bien, solo T o F.

Ausencia de datos

R tiene varias maneras de manejar la ausencia de datos dependiendo del tipo. Esto puede ser con un valor NA pero estos igualmente pueden ser más específicos. `NA_real_`, `NA_integer_`, `NA_character_` y `NA_complex_`

describen puntualmente el tipo de dato que falta pero en ultima instancia todos son tratados como NA por R. Por ejemplo:

```
NA
```

```
## [1] NA
```

```
NA_real_
```

```
## [1] NA
```

tienen la misma salida.

Además del NA, R reconoce NaN como **Not a Number** y es específico para cuando el resultado de una operación matemática resulta en algo imposible. Por ejemplo la división de 0 entre 0. Es diferente a NA porque no indica que falte un valor sino que el valor resultó en un no-numero.

```
print(0/0)
```

```
## [1] NaN
```

Por último tenemos el valor NULL que indica la ausencia de todo dato. Tanto así que se usa para desasignar variables. Por ejemplo:

```
var <- 'Ejemplo'  
print(var)
```

```
## [1] "Ejemplo"
```

```
var <- NULL  
print(var)
```

```
## NULL
```