

# Week 2

Carlos Guadián

9/7/2021

```
{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)
```

## BASIC PROGRAMMING

Pipe expresa una secuencia de múltiples operaciones representadas con `%>%` por ejemplo para utilizar una función en otra

No ejecutar, es un ejemplo `ToothGrowth %>% filter(dose == 0.5 %>% arrange(len))`

## VECTORES

Type Description Example Logical True/False TRUE Integer Positive and negative whole values 3 Double Decimal values 101.175 Character String/character values “Coding”

un vector atómico sólo puede contener elementos del mismo tipo

Para crear un vector utilizar `c()`

Numérico `c(2.5, 48.5, 101.5)` Enteros `c(1L, 5L, 15L)` Caracteres `c("Sara" , "Lisa" , "Anna")`  
Lógico `c(TRUE, FALSE, TRUE)`

Podemos saber el tipo de clase de un vector con `typeof()` `typeof(c("a", "b"))` Si lo hemos asignado a una variable con `class()` `a <- c("a", "b") class(a)`

Se puede determinar el largo de un vector con `length()` `x <- c(33.5, 57.75, 120.05) length(x)`

Se puede comprobar si el vector es de una clase determinada utilizando `is.logical()`, `is.double()`, `is.integer()`, `is.character()` `x <- c(2L, 5L, 11L) is.integer(x)`

Nombres en los vectores Con `names()` se puede asignar nombres a cada uno de los elementos del vector `x <- c(1, 3, 5) names(x) <- c("a", "b", "c") x`

## LISTAS

Se generan con `list()` `list("a", 1L, 1.5, TRUE)`

Y pueden contener otras listas `list(list(list(1 , 3, 5)))`

Para determinar la clase de cada elemento de una lista usamos `str()` tanto a la lista `str(list("a", 1L, 1.5, TRUE))`

Como a la variable a la que hayamos asignado la lista `z <- list(list(list(1 , 3, 5))) str(z)`

Se puede poner nombre a los elementos de la lista cuando se crea `list('Chicago' = 1, 'New York' = 2, 'Los Angeles' = 3)`