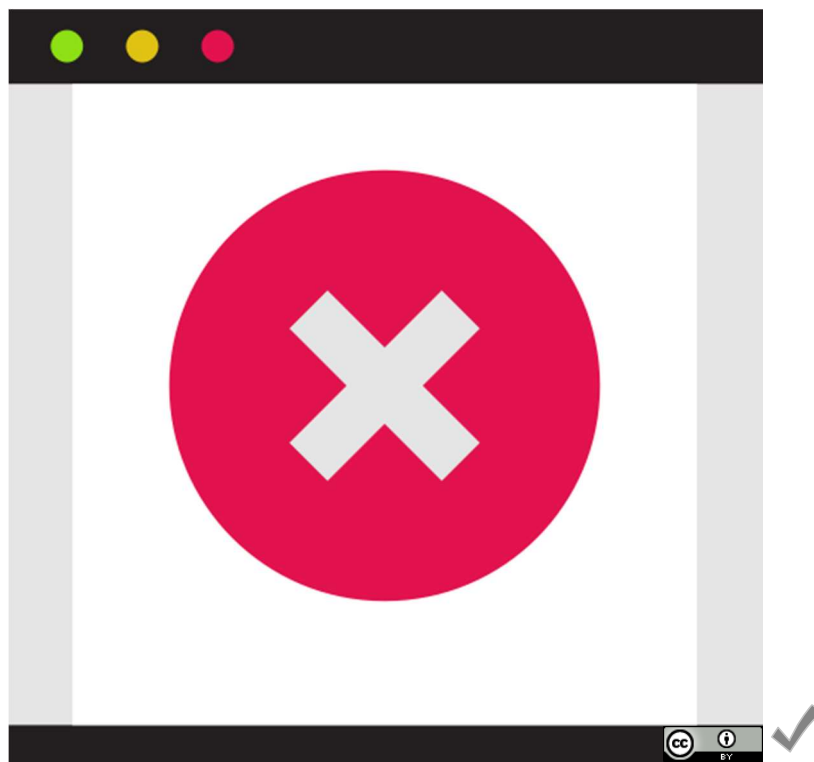


# Fallos en operaciones

## aritméticas



¿No se te ocurre por qué no obtienes los resultados esperados en los ejercicios que te mandan? ¿Te confundes constantemente con las operaciones en este lenguaje de programación? Aquí encontrarás los errores más comunes en operaciones aritméticas.

- No podemos realizar operaciones aritméticas con **variables no numéricas**, incluso aunque estas contengan números.

```
> a=35
> b='14'
> a+b
Error in a + b : argumento no-numérico para operador binario
```

Si intentamos operar con una variable de caracteres, surgirá un **error** tipo 'argumento no numérico para operador binario'.

\* En caso de que no recordemos de qué tipo eran nuestras variables, podemos comprobarlo con el comando 'is.str()' \*

- No se puede **multiplicar** empleando 'x'. Debemos utilizar el signo '\*'.

```
> 3x8
Error: unexpected symbol in "3x8"
> 3*8
[1] 24
```

El **error** será del tipo '*unexpected symbol*'.

- R emplea el orden de prelación matemático (potencias, productos y divisiones, sumas y restas), por lo que no podemos olvidar el **uso de paréntesis** en caso de querer modificar esta ordenación.

```
> #Si queremos dividir una suma entre otra:
> 3+4/3+4
[1] 8.333333
> (3+4)/(3+4)
[1] 1
```

**Atención:** Se trata de un error muy común, pero R no nos lo va a notificar ya que desconoce la operación que deseamos realizar.

- Los **signos '>' y '<'** se escriben antes del signo '=' en 'mayor o igual que' y 'menor o igual que'.

```
> 9=<8
Error: inesperado '<' in "9=<"
> 9<=8
[1] FALSE
```

Si escribimos '=<' o '=>', R nos avisa de que hay un **símbolo inesperado** (debemos revisar lo escrito antes de él).

- En R 'log(x)' es un logaritmo neperiano. Para calcular un **logaritmo en base 10** debemos escribir 'log10(x)'.

```
> log(10)
[1] 2.302585
> log10(10)
[1] 1
```

**Atención:** Se trata de un error muy común, pero R no nos lo va a notificar ya que desconoce la operación que deseamos realizar.