

Chapter 17: The logical class

Logical is a mode (and an implicit class) for vectors.

Section 17.1: Logical operators

There are two sorts of logical operators: those that accept and return vectors of any length (elementwise operators: `!`, `|`, `&`, `xor()`) and those that only evaluate the first element in each argument (`&&`, `||`). The second sort is primarily used as the `cond` argument to the `if` function.

Logical Operator	Meaning	Syntax
<code>!</code>	Not	<code>!x</code>
<code>&</code>	element-wise (vectorized) and	<code>x & y</code>
<code>&&</code>	and (single element only)	<code>x && y</code>
<code> </code>	element-wise (vectorized) or	<code>x y</code>
<code> </code>	or (single element only)	<code>x y</code>
<code>xor</code>	element-wise (vectorized) exclusive OR	<code>xor(x,y)</code>

Note that the `||` operator evaluates the left condition and if the left condition is TRUE the right side is never evaluated. This can save time if the first is the result of a complex operation. The `&&` operator will likewise return FALSE without evaluation of the second argument when the first element of the first argument is FALSE.

```
> x <- 5
> x > 6 || stop("X is too small")
Error: X is too small
> x > 3 || stop("X is too small")
[1] TRUE
```

To check whether a value is a logical you can use the `is.logical()` function.

Section 17.2: Coercion

To coerce a variable to a logical use the `as.logical()` function.

```
> x <- 2
> z <- x > 4
> z
[1] FALSE
> class(x)
[1] "numeric"
> as.logical(2)
[1] TRUE
```

When applying `as.numeric()` to a logical, a double will be returned. NA is a logical value and a logical operator with an NA will return NA if the outcome is ambiguous.

Section 17.3: Interpretation of NAs

See Missing values for details.

```
> TRUE & NA
[1] NA
> FALSE & NA
```

```
[1] FALSE
> TRUE || NA
[1] TRUE
> FALSE || NA
[1] NA
```
