|  |  |
| --- | --- |
| unique {base} | R Documentation |

**Extract Unique Elements**

**Description**

unique returns a vector, data frame or array like x but with duplicate elements/rows removed.

**Usage**

unique(x, incomparables = FALSE, ...)

## Default S3 method:

unique(x, incomparables = FALSE, fromLast = FALSE,

nmax = NA, ...)

## S3 method for class 'matrix'

unique(x, incomparables = FALSE, MARGIN = 1,

fromLast = FALSE, ...)

## S3 method for class 'array'

unique(x, incomparables = FALSE, MARGIN = 1,

fromLast = FALSE, ...)

**Arguments**

|  |  |
| --- | --- |
| x | a vector or a data frame or an array or NULL. |
| incomparables | a vector of values that cannot be compared. FALSE is a special value, meaning that all values can be compared, and may be the only value accepted for methods other than the default. It will be coerced internally to the same type as x. |
| fromLast | logical indicating if duplication should be considered from the last, i.e., the last (or rightmost) of identical elements will be kept. This only matters for [names](http://127.0.0.1:14695/library/base/help/names) or [dimnames](http://127.0.0.1:14695/library/base/help/dimnames). |
| nmax | the maximum number of unique items expected (greater than one). See [duplicated](http://127.0.0.1:14695/library/base/help/duplicated). |
| ... | arguments for particular methods. |
| MARGIN | the array margin to be held fixed: a single integer. |

**Details**

This is a generic function with methods for vectors, data frames and arrays (including matrices).

The array method calculates for each element of the dimension specified by MARGIN if the remaining dimensions are identical to those for an earlier element (in row-major order). This would most commonly be used for matrices to find unique rows (the default) or columns (with MARGIN = 2).

Note that unlike the Unix command uniq this omits *duplicated* and not just *repeated* elements/rows. That is, an element is omitted if it is equal to any previous element and not just if it is equal the immediately previous one. (For the latter, see [rle](http://127.0.0.1:14695/library/base/help/rle)).

Missing values are regarded as equal, but NaN is not equal to NA\_real\_. Character strings are regarded as equal if they are in different encodings but would agree when translated to UTF-8.

Values in incomparables will never be marked as duplicated. This is intended to be used for a fairly small set of values and will not be efficient for a very large set.

When used on a data frame with more than one column, or an array or matrix when comparing dimensions of length greater than one, this tests for identity of character representations. This will catch people who unwisely rely on exact equality of floating-point numbers!

Character strings will be compared as byte sequences if any input is marked as "bytes" (see [Encoding](http://127.0.0.1:14695/library/base/help/Encoding)).

**Value**

For a vector, an object of the same type of x, but with only one copy of each duplicated element. No attributes are copied (so the result has no names).

For a data frame, a data frame is returned with the same columns but possibly fewer rows (and with row names from the first occurrences of the unique rows).

A matrix or array is subsetted by [, drop = FALSE], so dimensions and dimnames are copied appropriately, and the result always has the same number of dimensions as x.

**Warning**

Using this for lists is potentially slow, especially if the elements are not atomic vectors (see [vector](http://127.0.0.1:14695/library/base/help/vector)) or differ only in their attributes. In the worst case it is *O(n^2)*.

**References**

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

**See Also**

[duplicated](http://127.0.0.1:14695/library/base/help/duplicated) which gives the indices of duplicated elements.

[rle](http://127.0.0.1:14695/library/base/help/rle) which is the equivalent of the Unix uniq -c command.

**Examples**

x <- c(3:5, 11:8, 8 + 0:5)

(ux <- unique(x))

(u2 <- unique(x, fromLast = TRUE)) # different order

stopifnot(identical(sort(ux), sort(u2)))

length(unique(sample(100, 100, replace = TRUE)))

## approximately 100(1 - 1/e) = 63.21

unique(iris)