mapvalues

Replace specified values with new values, in a vector or factor.

Item in x that match items from will be replaced by items in to, matched by position. For example, items in x that match the first element in from will be replaced by the first element of to.

Usage

```
Mapvalues (x, from, to, warn_missing = TRUE)
```

Arguments

x
    the factor or vector to modify
from
    a vector of the items to replace
to
    a vector of replacement values
warn_missing
    print a message if any of the old values are not actually present in x

Details

If x is a factor, the matching levels of the factor will be replaced with the new values.

The related revalue function works only on character vectors and factors, but this function works on vectors of any type and factors.

See Also

revalue to do the same thing but with a single named vector instead of two separate vectors.

```
x <- c("a", "b", "c")

mapvalues(x, c("a", "c"), c("A", "C"))

# Works on factors

y <- factor(c("a", "b", "c", "a"))

mapvalues(y, c("a", "c"), c("A", "C"))
```

```
# Works on numeric vectors

z <- c(1, 4, 5, 9)

mapvalues(z, from = c(1, 5, 9), to = c(10, 50, 90))
```

revalue

Replace specified values with new values, in a factor or character vector.

If x is a factor, the named levels of the factor will be replaced with the new values.

Usage

```
revalue(x, replace = NULL, warn_missing = TRUE)
```

Arguments

x

factor or character vector to modify

replace

named character vector, with new values as values, and old values as names.

warn_missing

print a message if any of the old values are not actually present in x

Details

This function works only on character vectors and factors, but the related mapvalues function works on vectors of any type and factors, and instead of a named vector specifying the original and replacement values, it takes two separate vectors

See Also

mapvalues to replace values with vectors of any type

```
x <- c("a", "b", "c")

revalue(x, c(a = "A", c = "C"))
revalue(x, c("a" = "A", "c" = "C"))

y <- factor(c("a", "b", "c", "a"))
revalue(y, c(a = "A", c = "C"))

# A sample factor to work with.
x <- factor(c("alpha","beta","gamma","alpha","beta"))
x
#> [1] alpha beta  gamma alpha beta
#> Levels: alpha beta gamma

levels(x)
#> [1] "alpha" "beta"  "gamma"
```

The easiest way is to use revalue() or mapvalues() from the plyr package:

```
library(plyr)
revalue(x, c("beta"="two", "gamma"="three"))
#> [1] alpha two   three alpha two
#> Levels: alpha two three

mapvalues(x, from = c("beta", "gamma"), to = c("two", "three"))
#> [1] alpha two   three alpha two
#> Levels: alpha two three
```