

Package ‘vecsets’

October 12, 2022

Type Package

Title Like Set Tools in 'Base' Package but Keeps Duplicate Elements

Version 1.3

Date 2021-03-08

Author Carl Witthoft

Maintainer Carl Witthoft <carl@witthoft.com>

Description The 'base' tools `union()` `intersect()`, etc., follow the algebraic definition that each element of a set must be unique. Since it's often helpful to compare all elements of two vectors, this toolset treats every element as unique for counting purposes. For ease of use, all functions in `vecsets` have an argument 'multiple' which, when set to `FALSE`, reverts them to the `base::sets` (alias for all the items) tools functionality.

License LGPL-3

Imports `pracma`

NeedsCompilation `no`

Repository CRAN

Date/Publication 2021-03-07 22:50:02 UTC

R topics documented:

<code>vecsets-package</code>	2
<code>vintersect</code>	2
<code>vperm</code>	3
<code>vsetdiff</code>	4
<code>vsetequal</code>	5
<code>vunion</code>	6

Index	7
--------------	---

vecsets-package	<i>An extension of the base tools such as 'intersect' which does not reduce to unique elements</i>
-----------------	--

Description

The base set-related tools follow the algebraic definition that each element of a set must be unique. Since it's often helpful to compare all elements of two vectors, this toolset treats every element as unique for counting purposes. For ease of use, all functions in `vecsets` have an argument `multiple` which, when set to `FALSE`, reverts them to the base set tools functionality.

Details

Package:	vecsets
Type:	Package
Version:	3.0
Date:	2021-03-08
License:	GPL-3

Author(s)

Carl Witthoft, with some code taken from Sven Hohenstein via Stack Overflow
 Maintainer: Carl Witthoft carl@witthoft.com

vintersect	<i>Perform intersection of two vectors, including counting repeated elements.</i>
------------	---

Description

Unlike the base `::intersect` function, if the vectors have repeated elements in common, the intersection returns as many of these elements as are in whichever vector has fewer of them.

Usage

```
vintersect(x, y, multiple = TRUE)
```

Arguments

<code>x</code>	A vector or an object which can be coerced to a vector
<code>y</code>	A vector or an object which can be coerced to a vector
<code>multiple</code>	Should repeated "multiple" items be returned? Default is <code>TRUE</code> ; if set to <code>FALSE</code> , <code>vintersect</code> acts like the base <code>::intersect</code> function.

Value

A vector of the elements in the intersection of the two vectors. If `multiple=FALSE` is set, only unique values are returned.

Author(s)

Carl Witthoft, with some code taken from Sven Hohenstein via Stack Overflow

See Also

[intersect](#), the CRAN package sets

Examples

```
x <- c(1:5,3,3,3,2,NA,NA)
y<- c(2:5,4,3,NA)
vintersect(x,y)
vintersect(x,y,multiple=FALSE)
intersect(x,y) #same as previous line
```

vperm	<i>Calculate all permutations of all combinations of a specified size from a data object.</i>
-------	---

Description

This function first uses `combn` to generate combinations of the desired size, then calculates all permutations of all said combinations.

Usage

```
vperm(x, m, FUN = NULL, ...)
```

Arguments

x	vector source for combinations, or integer n for <code>x <- seq_len(n)</code>
m	number of elements to choose in making the combinations
FUN	function to be applied to each combination; default <code>NULL</code> means the identity, i.e., to return the combination (vector of length m)
...	Additional arguments, if any, required for the function <code>FUN</code> . See Details.

Details

NA values are considered as valid elements and will be processed just as they are in `combn`. The input arguments are passed directly to `combn` but with one important exception. `combn`'s argument "simplify" is forced to "TRUE" inside this function so as to allow the permutations to be more easily generated. If the user includes `simplify = FALSE` in the ... input, it will be overwritten.

Value

An array within which each row contains one of the permutations.

Author(s)

Carl Witthoft, with some code taken from Sven Hohenstein via Stack Overflow

See Also

[intersect](#), the CRAN package [sets](#), [perms](#)

Examples

```
x <- c(1:5,3,3,3,2,NA,NA)
xp <- vperm(x,4) #large array
```

vsetdiff

Find all elements in first argument which are not in second argument.

Description

Finds all elements in first argument which are not in the second argument. Unlike the base `::setdiff` function, if the vectors have repeated elements in common, only the "excess" number of a given element are returned.

Usage

```
vsetdiff(x, y, multiple = TRUE)
```

Arguments

x	A vector or an object which can be coerced to a vector
y	A vector or an object which can be coerced to a vector
multiple	Should repeated "multiple" items be returned? Default is TRUE; if set to FALSE, <code>vintersect</code> acts like the base <code>::intersect</code> function.

Value

A vector of all elements in x which are not in y. If `multiple=FALSE` is set, only unique values are returned.

Author(s)

Carl Witthoft

See Also

[setdiff](#), the CRAN package [sets](#)

Examples

```
x <- c(1:5,3,3,3,2,NA,NA)
y<- c(2:5,4,3,NA)
vsetdiff(x,y)
vsetdiff(x,y,multiple=FALSE)
setdiff(x,y) # same as previous line
vsetdiff(y,x) #note the asymmetry
```

vsetequal	<i>Check whether two vectors contain exactly the same collection of elements.</i>
-----------	---

Description

Unlike the `base::setequal` function, if the vectors have repeated elements in common, the count of these elements is checked. As a result, vectors of different lengths will never be "equal."

Usage

```
vsetequal(x, y, multiple = TRUE)
```

Arguments

	<code>k</code>	A vector or an object which can be coerced to a vector
<code>y</code>		A vector or an object which can be coerced to a vector
<code>multiple</code>		Should repeated "multiple" items be returned? Default is TRUE; if set to FALSE, <code>vsetequal</code> acts like the <code>base::intersect</code> function.

Value

A logical value indicating equality or inequality. If `multiple=FALSE` is set, both input vectors are reduced to unique values before checking for equality.

Author(s)

Carl Witthoft

See Also

[setequal](#), the CRAN package sets

Examples

```
x <- c(1:5,3,3,3,2,NA,NA)
y<- c(1:5,4,3,NA)
vsetequal(x,y)
vsetequal(x,y,multiple=FALSE)
setequal(x,y) #same as previous line
```

vunion	<i>Returns the union of its inputs including repeated elements.</i>
--------	---

Description

The `base::union` function removes duplicates per algebraic set theory. `vunion` does not, and so returns as many duplicate elements as are in either input vector (not the sum of their inputs.) In short, `vunion` is the same as `vintersect(x,y) + vsetdiff(x,y) + vsetdiff(y,x)`.

Usage

```
vunion(x, y, multiple = TRUE)
```

Arguments

<code>x</code>	A vector or an object which can be coerced to a vector
<code>y</code>	A vector or an object which can be coerced to a vector
<code>multiple</code>	Should repeated "multiple" items be returned? Default is TRUE; if set to FALSE, <code>vunion</code> acts like the <code>base::union</code> function.

Value

A vector of the union of the two input vectors. If `multiple` is set to FALSE then the value returned is the same as `base::union`.

Author(s)

Carl Witthoft

See Also

[union](#), the CRAN package sets

Examples

```
x <- c(1:5,3,3,3,2,NA,NA)
y<- c(2:5,4,3,NA)
vunion(x,y)
vunion(x,y,multiple=FALSE)
union(x,y) #same as previous line
```

Index

`intersect`, [3](#), [4](#)

`perms`, [4](#)

`setdiff`, [4](#)

`setequal`, [5](#)

`union`, [6](#)

`vecsets` (`vecsets-package`), [2](#)

`vecsets-package`, [2](#)

`vintersect`, [2](#)

`vperm`, [3](#)

`vsetdiff`, [4](#)

`vsetequal`, [5](#)

`vunion`, [6](#)