Package 'set'

February 22, 2020

Type Package

Title Set Operation

Version 1.1

Author Zhi Jin, Jing Zhang

Maintainer Zhi Jin <nalanchongxuan@163.com>

Description More easy to get intersection, union or complementary set and combinations.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports do

URL https://github.com/yikeshu0611/set

BugReports https://github.com/yikeshu0611/set/issues

NeedsCompilation no

Repository CRAN

Date/Publication 2020-02-22 17:30:02 UTC

R topics documented:

---

and   *Get Intersection Set for Sets*

Description

Get intersection set for sets.

Usage

```
and(...)
```

Arguments

`...`   see argument x in [`toVector`](#)

Value

intersection elements

Examples

```
A <- c("a","b","c")
B <- c("a","b","c","d")
C <- c("a","e","h")
and(A, B)
and(A, B, C)
```

---

and2   *Get Intersection Set for Two Sets*

Description

Get intersection set for two sets, which can be numbers, characters, vectors even dataframe, matrix or list.

Usage

```
a %and% b

a %a% b

a %A% b

"&"(a, b)
```

Arguments

a	see argument x in [toVector](#)
b	see argument x in [toVector](#)

Value

intersection set

Examples

```
A <- c("a","b","c")
B <- c("a","b","c","d")
C <- c("a","e","h")
A %and% B
A %and% B %and% C
```

---

combination   *Combination of Characters or Vectors*

Description

Combination of characters or vectors.

Usage

```
combination(...)
```

Arguments

...	one or more vectors

Value

binary combination

Examples

```
A <- c("a","b","c")
combination(A)

B <- c("a","b","c","d")
C <- c("a","e","h")
D <- c("a","b","e")
E <- c("a","c")
combination(A,B)
combination(A,B,C)
combination(A,B,C,D)
combination(A,B,C,D,E)
```

---

is.sub   *Judge Subset*

Description

Whether data set a is a subject of data set A.

Usage

```
is.sub(a, A)
```

Arguments

a   see argument x in [toVector](#)

A   see argument x in [toVector](#)

Value

logical result.

Examples

```
a <- c("a","b","c")
A <- c("a","b","c","d")
is.sub(a,A)
```

---

not   *Get Elements only Existed in Dataset a*

Description

Get elements only existed in dataset a.

Usage

```
not(...)
```

Arguments

...   see argument x in [toVector](#)

Value

elements only existed in dataset a

Examples

```
A <- c("a","b","c")
B <- c("a","b","c","d")
not(B, A)
```

---

not2   *Get Elements only Existed in Dataset a*

Description

Get elements only existed in dataset a.

Usage

```
a %not% b

a %n% b

a %N% b

"/"(a, b)
```

Arguments

a	see argument x in `toVector`
b	see argument x in `toVector`

Value

elements only existed in dataset a

Examples

```
A <- c("a","b","c")
B <- c("a","b","c","d")
B %not% A
```

---

or   *Get Union Set for Sets*

Description

Get union set for sets.

Usage

```
or(...)
```

Arguments

...   see argument x in [`toVector`](#)

Value

union elements

Examples

```
A <- c("a","b","c")
B <- c("a","b","c","d")
C <- c("a","e","h")
or(A, B)
or(A, B, C)
```

---

or2   *Get Union Set for Two Sets*

Description

Get union set for two sets.

Usage

```
a %or% b

a %r% b

a %R% b

"|"(a, b)
```

Arguments

a	see argument x in [toVector](#)
b	see argument x in [toVector](#)

Value

union set

Examples

```
A <- c("a","b","c")
B <- c("a","b","c","d")
C <- c("a","e","h")
A %or% B
A %and% B %or% C
```

---

toVector   *Convet to Character*

Description

Convert dataframe, matrix, list, array or vector to character vector.

Usage

```
toVector(x)
```

Arguments

x	can be vector, dataframe, matrix, list, array

Value

a character vector

Examples

```
df=data.frame(a=c(1,2,3))
toVector(df)
```

Index