Package 'edm1'

June 20, 2024

Title Functions to perform new set operation on vectors and others functions bringing new features for base sets functions

Version 2.0.0.0

Description Provides functions to perform union and intersect operation on n vectors, to see the differences in term of elements between n vectors and to perform union operation keeping the number of original unique elements in the first vector...

License GPL (==3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Imports stringr,
stringi,
dplyr,
openxlsx

Contents

`intersect_all` *intersect_all*

Description

Allows to calculate the intersection between n vectors

Usage

```
intersect_all(...)
```

Arguments

...	is all the vector you want to calculate the intersection from

Examples

```
print(intersect_all(c(1:5), c(1, 2, 3, 6), c(1:4)))
```

```
[1] 1 2 3
```

---

see_diff *see_diff*

Description

Output the opposite of intersect(a, b). Already seen at: https://stackoverflow.com/questions/19797954/function-to-find-symmetric-difference-opposite-of-intersection-in-r

Usage

```
see_diff(vec1 = c(), vec2 = c())
```

Arguments

vec1	is the first vector
vec2	is the second vector

Examples

```
print(see_diff(c(1:7), c(4:12)))
```

```
[1] 1 2 3 8 9 10 11 12
```

---

see_diff_all *see_diff_all*

Description

Allow to perform the opposite of intersect function to n vectors.

Usage

```
see_diff_all(...)
```

Arguments

...	are all the input vectors

Examples

```
vec1 <- c(3:6)
vec2 <- c(1:8)
vec3 <- c(12:16)

print(see_diff_all(vec1, vec2))

[1] 1 2 7 8

print(see_diff_all(vec1, vec2, vec3))

[1]  3  4  5  6  1  2  7  8 12 13 14 15 16
```

---

union_all *union_all*

Description

Allow to perform a union function to n vectors.

Usage

```
union_all(...)
```

Arguments

`...`	are all the input vectors

Examples

```
print(union_all(c(1, 2), c(3, 4), c(1:8)))

[1] 1 2 3 4 5 6 7 8

print(union_all(c(1, 2), c(3, 4), c(7:8)))

[1] 1 2 3 4 7 8
```

---

union_keep *union_keep*

Description

Performs a union operation keeping the number of elements of all input vectors, see examples

Usage

```
union_keep(...)
```

Arguments

...	are all the input vectors

Examples

```
print(union_keep(c("a", "ee", "ee"), c("p", "p", "a", "i"), c("a", "a", "z")))

[1] "a"  "ee" "ee" "p"  "p"  "i"  "z"

print(union_keep(c("a", "ee", "ee"), c("p", "p", "a", "i")))

[1] "a"  "ee" "ee" "p"  "p"  "i"
```

Index