

# Package ‘formulops’

January 22, 2025

**Title** Mathematical Operations on R Formula

**Version** 0.5.1

**Description** Perform mathematical operations on R formula (add, subtract, multiply, etc.) and substitute parts of formula.

**Depends** R (>= 3.5)

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Suggests** covr, spelling, testthat (>= 3.0.0)

**URL** <https://github.com/humanpred/formulops>

**BugReports** <https://github.com/humanpred/formulops/issues>

**Config/testthat/edition** 3

**Language** en-US

**NeedsCompilation** no

**Author** Bill Denney [aut, cre] (<<https://orcid.org/0000-0002-5759-428X>>)

**Maintainer** Bill Denney <[wdenney@humanpredictions.com](mailto:wdenney@humanpredictions.com)>

**Repository** CRAN

**Date/Publication** 2025-01-22 19:10:02 UTC

## Contents

formula.substituting_formula . . . . .	2
formula_side . . . . .	2
modify_formula . . . . .	3
op_formula . . . . .	4
simplify_parens . . . . .	5
substituting_formula . . . . .	6

<b>Index</b>	<b>7</b>
--------------	----------

---

formula.substituting\_formula

*Convert a substituting\_formula object into a regular formula.*

---

### Description

Convert a substituting\_formula object into a regular formula.

### Usage

```
## S3 method for class 'substituting_formula'
formula(x, ...)
```

### Arguments

x	the substituting_formula object
...	Ignored

### Value

A formula with values substituted.

---

formula\_side

*Extract formula parts*

---

### Description

Extract formula parts

### Usage

```
get_lhs(x)
get_rhs(x)
```

### Arguments

x	A formula (or something that can be coerced to a formula) to extract a part from
---	--

### Value

The requested part of the formula as a name or call or NULL if it does not exist.

### Functions

- `get_lhs()`: Extract the left hand side (NULL for one-sided formula).
- `get_rhs()`: Extract the right hand side.

---

modify_formula	<i>Modify a formula by finding some part of it and replacing it with a new value.</i>
----------------	---

---

**Description**

Modify a formula by finding some part of it and replacing it with a new value.

**Usage**

```
modify_formula(formula, find, replace, add_parens = FALSE)
```

**Arguments**

formula	The formula to modify (may also be a call)
find	A call or name (or list thereof) to search for within the formula
replace	A call or name (or list thereof) to replace the find values
add_parens	Add parentheses if replace is not a name or if it is not already something in parentheses?

**Details**

Replacement occurs at the first match, so if the replacement list would modify something in the find list, that change will not occur (make two calls to the function for that effect). See the "Replacement is not sequential" examples below.

A special call can be used to expand a formula. If an expansion of arguments is desired to change a single function argument to multiple arguments, 'formulops\_expand()' can be used. (See the examples.)

**Value**

formula modified

**Examples**

```
modify_formula(a~b, find=quote(a), replace=quote(c))
modify_formula(a~b, find=quote(a), replace=quote(c+d))
modify_formula(a~b/c, find=quote(b/c), replace=quote(d))
# Replacement is not sequential
modify_formula(a~b/c, find=list(quote(b/c), quote(d)), replace=list(quote(d), quote(e)))
modify_formula(a~b/c+d, find=list(quote(b/c), quote(d)), replace=list(quote(d), quote(e)))
# Expanding arguments to functions is possible
modify_formula(a~b(c), find=quote(c), replace=quote(formulops_expand(d, e)))
```

---

op_formula	<i>Perform a mathematical operation on two formula</i>
------------	--

---

**Description**

Perform a mathematical operation on two formula

**Usage**

```
op_formula(op, e1, e2)

multiply_formula(e1, e2)

divide_formula(e1, e2)

add_formula(e1, e2)

subtract_formula(e1, e2)

## S3 method for class 'formula'
Ops(e1, e2)

## S3 method for class 'formula'
Math(x, ...)
```

**Arguments**

op	The operation to perform either as a name or something that can be coerced into a name.
e1, e2, x	The formulae to operate on
...	Ignored.

**Details**

The method for combination depends if the two formula are one- or two-sided.

If both formula are one-sided, the right hand side (RHS) of both are added together with additional parentheses added, if parentheses appear to be needed. If both formula are two-sided, the left hand side (LHS) and RHS are separately added. If one formula is one-sided and the other is two-sided, the LHS is selected from the two-sided formula and the RHS follows rules as though two one-sided formula were added.

`multiply_formula` Multiply two formula (identical to  $(a \sim b) * (c \sim d)$ )

`divide_formula` Divide two formula (identical to  $(a \sim b) / (c \sim d)$ )

`add_formula` Add two formula (identical to  $(a \sim b) + (c \sim d)$ )

`subtract_formula` Multiply two formula (identical to  $(a \sim b) - (c \sim d)$ )

`Ops.formula` Supports generic binary operators and a couple of unary operators (see `?Ops`).

`Math.formula` Supports generic unary operators (see `?Math`).

**Value**

e1 and e2 combined by the operation with the environment from e1. See Details.

**Examples**

```
op_formula("+", a~b, c~d)
op_formula("+", a~b, ~d)
op_formula("+", ~b, c~d)
op_formula("+", ~b, ~d)
op_formula("-", a~b)
op_formula("-", -a~b) # Dumb, but accurate
op_formula("-", -a~b, c~d) # Dumb, but accurate

log(a~b)
```

---

simplify_parens	<i>Remove extraneous parentheses from a formula.</i>
-----------------	--

---

**Description**

Remove extraneous parentheses from a formula.

**Usage**

```
simplify_parens(x)
```

**Arguments**

x                    The formula (or call) to simplify

**Value**

The simplified formula

**Examples**

```
simplify_parens(((a))~((b+c)))
```

---

substituting\_formula *A substituting formula helps clarify a formula where the parameters are more simply described in separate formulae.*

---

### Description

A substituting formula helps clarify a formula where the parameters are more simply described in separate formulae.

### Usage

```
substituting_formula(x, ...)  
as_substituting_formula(x, substitutions)
```

### Arguments

x	The base formula
...	Supporting formula of the form $x_1 \sim x_2 + x_3 * x_4 \dots$
substitutions	A list of supporting formula.

### Details

Formula are substituted in order. Substitutions may not have the same left hand side.

### Value

A substituting\_formula object which may be coerced into a single formula with an `as.formula()` method or printed as a list of formulae.

### Functions

- `as_substituting_formula()`: Generate and check substituting\_formula

### Examples

```
foo <- substituting_formula(y~x1+x2, x1~x3*x4, x2~x5/x6+x7)  
as.formula(foo)
```

# Index

`add_formula (op_formula)`, 4  
`as_substituting_formula`  
    (`substituting_formula`), 6

`divide_formula (op_formula)`, 4

`formula.substituting_formula`, 2  
`formula_side`, 2

`get_lhs (formula_side)`, 2  
`get_rhs (formula_side)`, 2

`Math.formula (op_formula)`, 4  
`modify_formula`, 3  
`multiply_formula (op_formula)`, 4

`op_formula`, 4  
`Ops.formula (op_formula)`, 4

`simplify_parens`, 5  
`substituting_formula`, 6  
`subtract_formula (op_formula)`, 4