

# Package: sourcemodify (via r-universe)

May 26, 2026

**Type** Package

**Title** Helper functions to modify source code of R scripts

**Version** 0.1.0

**Author** Sebastian Kranz

**Maintainer** Sebastian Kranz <sebastian.kranz@uni-ulm.de>

**Description** Helper functions to modify source code of R scripts

**License** GPL >= 2.0

**Encoding** UTF-8

**LazyData** true

**Remotes** skranz/stringtools

**Depends** stringi, restorepoint, stringtools, dplyr

**RoxygenNote** 7.1.1

**Config/pak/sysreqs** libicu-dev

**Repository** <https://skranz.r-universe.dev>

**Date/Publication** 2026-03-27 06:15:15 UTC

**RemoteUrl** <https://github.com/repboxr/sourcemodify>

**RemoteRef** main

**RemoteSha** 9a7752de771856c0ddb5e9ed24e2a30f87af2376

## Contents

getAugmentedParseData . . . . .	2
somo_calls_info . . . . .	2
somo_change_assign_rhs . . . . .	3
somo_change_calls_fun_name . . . . .	3
somo_find_fun_arg . . . . .	4
somo_find_fun_call . . . . .	4
somo_found . . . . .	4
somo_init . . . . .	5
somo_make_code . . . . .	5

somo_make_mod_df . . . . .	5
somo_make_pd_code . . . . .	6
somo_surround_calls . . . . .	6
somo_surround_calls_arg . . . . .	7

<b>Index</b>	<b>8</b>
--------------	----------

---

`getAugmentedParseData` *Key internal function fo sourcemodify Returns parsed code as a tibble with detailed information where each subexpression is found in the code*

---

### Description

Key internal function fo sourcemodify Returns parsed code as a tibble with detailed information where each subexpression is found in the code

### Usage

```
getAugmentedParseData(
  code,
  calls = parse(text = code, keep.source = TRUE),
  line.starts = find.line.starts(code),
  add_funid = TRUE
)
```

---

`somo_calls_info` *A simplified interface to `somo_find_fun_arg`*

---

### Description

Only works with named arguments.

### Usage

```
somo_calls_info(
  somo,
  fun,
  arg_names,
  arg_pos = NULL,
  arg_def = NULL,
  call_pd_rows = NULL,
  arg_prefix = ""
)
```

---

somo\_change\_assign\_rhs

*Changes the rhs of variable assignments*


---

### Description

Take a look at `somo$as_df` to find the ids of the assignment that shall be changed. To get the new code call afterward `somo_make_code`

### Usage

```
somo_change_assign_rhs(
  somo,
  id,
  new,
  new_df = data.frame(parent = id, new = new)
)
```

### Arguments

somo	The somo object generated with <code>somo_init</code> .
id	The ids of the assignment as specified in <code>somo\$as_df</code>
new	New texts for the right hand sides

### Value

The modified somo object

---

somo\_change\_calls\_fun\_name

*Change the name of a called function without adapting any arguments*


---

### Description

If the function call directly

### Usage

```
somo_change_calls_fun_name(somo, fun, new, ignore_if_pkg = TRUE)
```

### Arguments

somo	The somo object generated with <code>somo_init</code> .
fun	Function name that shall be changed in calls
new	New function name to be called
ignore_if_pkg	If TRUE don't change calls of the form <code>pkg::fun()</code> . If FALSE a call like <code>pkg::fun()</code> will be replaced by just <code>new()</code> .

**Value**

The modified somo object

---

somo_find_fun_arg	<i>Find the text of the function argument in all calls to fun</i>
-------------------	---

---

**Description**

Find the text of the function argument in all calls to fun

**Usage**

```
somo_find_fun_arg(
  somo,
  fun,
  arg_name = NULL,
  arg_pos = NULL,
  arg_def = NULL,
  call_pd_rows = NULL
)
```

---

somo_find_fun_call	<i>Find the text of the function argument in all calls to fun</i>
--------------------	---

---

**Description**

Find the text of the function argument in all calls to fun

**Usage**

```
somo_find_fun_call(somo, fun)
```

---

somo_found	<i>Returns the results from the last somo_find_* call as a tibble</i>
------------	---

---

**Description**

Returns the results from the last somo\_find\_\* call as a tibble

**Usage**

```
somo_found(somo)
```

---

somo_init	<i>Init a new source modifier object</i>
-----------	--

---

**Description**

Init a new source modifier object

**Usage**

```
somo_init(
  code = NULL,
  files = NULL,
  start.size = 200,
  encoding = "UTF-8",
  add_funid = TRUE
)
```

**Arguments**

code	The R source code as text file
add_funid	Shall for each expression be determined inside which function it is? Computation sometimes be quite time intensive.

---

somo_make_code	<i>Returns the modified code of a somo object</i>
----------------	---

---

**Description**

Returns the modified code of a somo object

**Usage**

```
somo_make_code(somo)
```

---

somo_make_mod_df	<i>Generate a data frame that contains information for all code modifications</i>
------------------	---

---

**Description**

Generate a data frame that contains information for all code modifications

**Usage**

```
somo_make_mod_df(somo)
```

---

somo_make_pd_code	<i>Computes for the specified rows in somo\$pd the modified code after replacements have been performed.</i>
-------------------	--

---

**Description**

WARNING: Works currently only for non-nested replacements

**Usage**

```
somo_make_pd_code(somo, pd_rows = seq_len(NROW(somo$pd)))
```

**Arguments**

somo	A somo object
pd_rows	An integer vector of rows of somo\$pd for which the modified code shall be computed.

**Value**

A character vector with resulting code for each row in pd specified by pd\_rows

---

somo_surround_calls	<i>Surround calls to the specified functions</i>
---------------------	--

---

**Description**

Will by typically used to wrap a function call into another function call that can analyse or modify the result.

**Usage**

```
somo_surround_calls(somo, funs = NULL, pre, post, call_pd_rows = NULL)
```

**Arguments**

somo	The somo object generated with somo_init.
funs	One or several function names.
pre	Text to be added before the function call.
post	Text to be added the function call.

**Value**

The modified somo object

---

somo\_surround\_calls\_arg

*Surround an argument of specific function*


---

### Description

Will by typically used to wrap the argument (like a filename) of function call into another function call that can analyse or modify the argument.

### Usage

```
somo_surround_calls_arg(
    somo,
    fun = NULL,
    arg_name = NULL,
    arg_pos = NULL,
    pre,
    post,
    arg_def = NULL,
    call_pd_rows = NULL
)
```

### Arguments

somo	The somo object generated with somo_init.
fun	Name of the function whose argument shall be wrapped
arg_name	The name of the argument (must be given if arg_pos is NULL)
arg_pos	Position of the argument (must be given if arg_name is NULL)
pre	Text to be added before the function call.
post	Text to be added the function call.
arg_def	Names of all function arguments in the correct order. The result of calling names(as.list(args(fun))). This argument is optional, but without it matching may fail.

### Value

The modified somo object

# Index

[getAugmentedParseData](#), [2](#)

[somo\\_calls\\_info](#), [2](#)

[somo\\_change\\_assign\\_rhs](#), [3](#)

[somo\\_change\\_calls\\_fun\\_name](#), [3](#)

[somo\\_find\\_fun\\_arg](#), [4](#)

[somo\\_find\\_fun\\_call](#), [4](#)

[somo\\_found](#), [4](#)

[somo\\_init](#), [5](#)

[somo\\_make\\_code](#), [5](#)

[somo\\_make\\_mod\\_df](#), [5](#)

[somo\\_make\\_pd\\_code](#), [6](#)

[somo\\_surround\\_calls](#), [6](#)

[somo\\_surround\\_calls\\_arg](#), [7](#)