Package 'rformat'

May 9, 2026

Type Package

Title Base R Code Formatter

Version 0.1.0

Description A minimal R code formatter following base R style conventions.
Formats R code with consistent spacing, indentation, and structure.

License GPL-3

URL https://github.com/cornball-ai/rformat

BugReports https://github.com/cornball-ai/rformat/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports Rcpp

LinkingTo Rcpp

Suggests tinytest, simplermarkdown

VignetteBuilder simplermarkdown

NeedsCompilation yes

Author Troy Hernandez [aut, cre] (ORCID:
<https://orcid.org/0009-0005-4248-604X>),
Dirk Eddelbuettel [ctb] (ORCID:
<https://orcid.org/0000-0001-6419-907X>)

Maintainer Troy Hernandez <troy@cornball.ai>

Repository CRAN

Date/Publication 2026-03-09 16:30:09 UTC

Contents

`rformat` *Format R Code*

Description

Format R code string according to base R style conventions.

Usage

```
rformat(code, indent = 4L, line_limit = 80L, wrap = "paren",
        brace_style = "kr", control_braces = FALSE, expand_if = FALSE,
        else_same_line = TRUE, function_space = FALSE, join_else = TRUE)
```

Arguments

`code`	Character string of R code to format.
`indent`	Indentation per level: integer for spaces (default 4), or character string for literal indent (e.g., ‘"\t\t"‘ for vintage R Core style).
`line_limit`	Maximum line length before wrapping (default 80).
`wrap`	Continuation style for long function signatures: ‘"paren"‘ (default) aligns to opening parenthesis, ‘"fixed"‘ uses 8-space indent.
`brace_style`	Brace placement for function definitions: ‘"kr"‘ (default) puts opening brace on same line as ‘) {‘, ‘"allman"‘ puts it on a new line.
`control_braces`	If TRUE, add braces to bare one-line control flow bodies (e.g., ‘if (x) y‘ becomes ‘if (x) { y }‘). Default FALSE matches R Core source code where 59% of control flow bodies are bare.
`expand_if`	Expand inline if-else to multi-line (default FALSE).
`else_same_line`	If TRUE (default), repair top-level ‘}\nelse‘ (which is a parse error in R) by joining to ‘} else‘ before formatting. When FALSE, unparseable input is returned unchanged with a warning.
`function_space`	If TRUE, add space before ‘(‘ in function definitions: ‘function (x)‘ instead of ‘function(x)‘. Default FALSE matches 96% of R Core source code.
`join_else`	If TRUE (default), move ‘else‘ to the same line as the preceding ‘}‘: ‘} else {‘. Matches R Core source code where 70% use same-line else. When FALSE, ‘}\nelse‘ on separate lines is preserved.

Value

Formatted code as a character string.

Examples

```
# Basic formatting: spacing around operators
rformat("x<-1+2")

# Add braces to bare control-flow bodies
rformat("if(x>0) y<-1", control_braces = TRUE)

# Expand inline if-else to multi-line
rformat("x <- if (a) b else c", expand_if = TRUE)

# Wrap long function signatures (default: paren-aligned)
long_sig <- paste0(
    "f <- function(alpha, beta, gamma, delta, ",
    "epsilon, zeta, eta) {\n    1\n}")
cat(rformat(long_sig), sep = "\n")

# Wrap with fixed 8-space continuation indent
cat(rformat(long_sig, wrap = "fixed"), sep = "\n")

# Allman brace style
rformat("f <- function(x) { x }", brace_style = "allman")
```

`rformat_dir` *Format R Files in Directory*

Description

Format all R files in a directory.

Usage

```
rformat_dir(path = ".", recursive = TRUE, dry_run = FALSE, indent = 4L,
            line_limit = 80L, wrap = "paren", brace_style = "kr",
            control_braces = FALSE, expand_if = FALSE, else_same_line = TRUE,
            function_space = FALSE, join_else = TRUE)
```

Arguments

`path`	Path to directory.
`recursive`	If TRUE, process subdirectories.
`dry_run`	If TRUE, report changes without writing.
`indent`	Indentation per level: integer for spaces (default 4), or character string for literal indent (e.g., ‘"\t\t"‘ for vintage R Core style).
`line_limit`	Maximum line length before wrapping (default 80).
`wrap`	Continuation style for long function signatures: ‘"paren"‘ (default) aligns to opening parenthesis, ‘"fixed"‘ uses 8-space indent.

brace_style	Brace placement for function definitions: ‘"kr"‘ (default) puts opening brace on same line as ‘) {‘, ‘"allman"‘ puts it on a new line.
control_braces	If TRUE, add braces to bare one-line control flow bodies. Default FALSE matches R Core majority style.
expand_if	Expand inline if-else to multi-line (default FALSE).
else_same_line	If TRUE (default), repair top-level ‘}\nelse‘ (which is a parse error in R) by joining to ‘} else‘ before formatting.
function_space	If TRUE, add space before ‘(‘ in function definitions: ‘function (x)‘ instead of ‘function(x)‘. Default FALSE matches 96% of R Core source code.
join_else	If TRUE (default), move ‘else‘ to the same line as the preceding ‘}‘.

Value

Invisibly returns vector of modified file paths.

Examples

```
# Format all R files in a directory (dry run)
d <- tempfile()
dir.create(d)
writeLines("x<-1", file.path(d, "test.R"))
rformat_dir(d, dry_run = TRUE)

# Format and overwrite
rformat_dir(d)
unlink(d, recursive = TRUE)
```

---

rformat_file *Format R File*

Description

Format an R file in place or write to a new file.

Usage

```
rformat_file(path, output = NULL, dry_run = FALSE, indent = 4L,
             line_limit = 80L, wrap = "paren", brace_style = "kr",
             control_braces = FALSE, expand_if = FALSE, else_same_line = TRUE,
             function_space = FALSE, join_else = TRUE)
```

Arguments

path	Path to R file.
output	Optional output path. If NULL, overwrites input file.
dry_run	If TRUE, return formatted code without writing.
indent	Indentation per level: integer for spaces (default 4), or character string for literal indent (e.g., ‘"\t\t"‘ for vintage R Core style).
line_limit	Maximum line length before wrapping (default 80).
wrap	Continuation style for long function signatures: ‘"paren"‘ (default) aligns to opening parenthesis, ‘"fixed"‘ uses 8-space indent.
brace_style	Brace placement for function definitions: ‘"kr"‘ (default) puts opening brace on same line as ‘) {‘, ‘"allman"‘ puts it on a new line.
control_braces	If TRUE, add braces to bare one-line control flow bodies. Default FALSE matches R Core majority style.
expand_if	Expand inline if-else to multi-line (default FALSE).
else_same_line	If TRUE (default), repair top-level ‘}\nelse‘ (which is a parse error in R) by joining to ‘} else‘ before formatting.
function_space	If TRUE, add space before ‘(‘ in function definitions: ‘function (x)‘ instead of ‘function(x)‘. Default FALSE matches 96% of R Core source code.
join_else	If TRUE (default), move ‘else‘ to the same line as the preceding ‘}‘.

Value

Invisibly returns formatted code.

Examples

```
# Format a file (dry run to see result without writing)
f <- tempfile(fileext = ".R")
writeLines("x<-1+2", f)
rformat_file(f, dry_run = TRUE)

# Format and overwrite
rformat_file(f)
readLines(f)
unlink(f)
```

Index