

Package: scilintr (via r-universe)

June 13, 2026

Title Scientific Code Lint for R Analyses

Version 0.1.1

Description Static analysis for R scientific data analysis code. Flags patterns that often correspond to hidden scientific commitments -- silent error swallowing, smuggled defaults, label leakage in selection-stage code, magic-eps floors in 'BIC' formulas, and shadow-overwrite of sourced helpers. Designed for agentic coding workflows; high recall over precision; structured 'ANALYSIS_OK' waivers as the audit trail.

License MIT + file LICENSE

URL <https://github.com/arjunrajlaboratory/scilintr>

BugReports <https://github.com/arjunrajlaboratory/scilintr/issues>

Encoding UTF-8

Depends R (>= 4.1)

Imports lintr (>= 3.0.0), xml2, xmlparsedata, yaml

Suggests testthat (>= 3.0.0), roxygen2

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Arjun Raj [aut, cre]

Maintainer Arjun Raj <arjunraj@seas.upenn.edu>

Config/pak/sysreqs libxml2-dev

Repository <https://arjunrajlab.r-universe.dev>

Date/Publication 2026-06-12 11:20:02 UTC

RemoteUrl <https://github.com/cran/scilintr>

RemoteRef HEAD

RemoteSha 59a452144a606930bd652bce821c042fd47bfa20

Contents

lint_file	2
lint_project	3
main	3
Index	5

lint_file	<i>Lint a single R file.</i>
-----------	------------------------------

Description

Runs every registered per-file linter against path, converts the resulting `lintr::Lint` objects to `scilintr_finding` records, and applies the `ANALYSIS_OK[...]` waiver filter.

Usage

```
lint_file(path, config = NULL)
```

Arguments

path	Path to a .R file.
config	Optional configuration list (loaded from <code>.scilintr.yml</code>).

Value

A list of `scilintr_finding` records.

Examples

```
# Lint a tiny self-contained file in the session temp directory.
tmp <- tempfile(fileext = ".R")
writeLines("x <- 1", tmp)
findings <- lint_file(tmp)
length(findings)
unlink(tmp)
```

lint_project	<i>Lint an entire project directory.</i>
--------------	--

Description

Walks every .R file, runs the per-file linters, then builds the project index and runs the cross-file rules against it.

Usage

```
lint_project(root = ".", config = NULL)
```

Arguments

root	Project root directory.
config	Optional configuration list.

Value

A list of `scilintr_finding` records aggregated across files.

Examples

```
# Build a throwaway one-file project inside the session temp directory.
proj <- file.path(tempdir(), "scilintr-demo")
dir.create(proj, showWarnings = FALSE)
writeLines("y <- 2", file.path(proj, "analysis.R"))
findings <- lint_project(proj)
length(findings)
unlink(proj, recursive = TRUE)
```

main	<i>Main CLI entry point.</i>
------	------------------------------

Description

Invoked from `inst/bin/scilintr` or `Rscript -e 'scilintr::main()'`.

Usage

```
main(args = commandArgs(trailingOnly = TRUE))
```

Arguments

args	Character vector of command-line arguments.
------	---

Value

Invisibly returns `0L` when no findings are reported and `1L` otherwise. Called for its side effect of printing findings.

Examples

```
# Run the CLI entry point over a throwaway project in tempdir().
proj <- file.path(tempdir(), "scilintr-cli-demo")
dir.create(proj, showWarnings = FALSE)
writeLines("z <- 3", file.path(proj, "analysis.R"))
main(proj)
unlink(proj, recursive = TRUE)
```

Index

lint_file, 2
lint_project, 3
main, 3