

# Package ‘SCIproj’

May 7, 2026

**Type** Package

**Title** Creates a Scientific Project Skeleton as an R Package

**Version** 1.0.1

**Description** Provides a template for new research projects structured as an R package-based research compendium. Everything - data, R scripts, custom functions and manuscript or reports - is contained within the same package to facilitate collaboration and promote reproducible research, following the FAIR principles.

**Language** en-US

**License** MIT + file LICENSE

**Depends** R(>= 4.1.0)

**Imports** usethis (>= 2.0.0), rstudioapi, cli

**Suggests** knitr, renv, rmarkdown, targets, testthat (>= 3.0.0), withr

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**RoxygenNote** 7.3.3

**Encoding** UTF-8

**URL** <https://github.com/saskiaotto/SCIproj>

**BugReports** <https://github.com/saskiaotto/SCIproj/issues>

**NeedsCompilation** no

**Author** Saskia Otto [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0001-7780-1322>>)

**Maintainer** Saskia Otto <saskia.a.otto@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-04-22 12:30:15 UTC

## Contents

create_proj	2
Index	5

---

 create\_proj

 Create new data analysis project as R package
 

---

## Description

Create all the scaffolding for a new project in a new directory. The scaffolding includes a README file, different folders to hold raw data, analyses, etc, a `_targets.R` pipeline template, `renv` dependency management, and a `CITATION.cff` file. Optionally, initialize a git repository and/or set up a private or public GitHub repo with GitHub Actions CI.

## Usage

```
create_proj(
  name,
  data_raw = TRUE,
  makefile = FALSE,
  testthat = FALSE,
  use_pipe = FALSE,
  add_license = NULL,
  license_holder = "Your name",
  orcid = NULL,
  use_git = TRUE,
  create_github_repo = FALSE,
  private_repo = TRUE,
  ci = "none",
  use_renv = TRUE,
  use_targets = TRUE,
  use_docker = FALSE,
  use_rproj = TRUE,
  setwd_to_proj = TRUE,
  open_proj = FALSE,
  verbose = FALSE
)
```

## Arguments

name	Character. Name of the new project. Could be a path, e.g. <code>"~/myRcode/newproj"</code> . A new folder will be created with that name. The directory name can contain underscores or hyphens (e.g. <code>"my_project"</code> ) - these will be automatically converted to dots for the internal R package name (e.g. <code>my.project</code> in <code>DESCRIPTION</code> ).
data_raw	Logical. If <code>TRUE</code> (default), adds a <code>data-raw/</code> folder for all the raw data files, including a <code>clean_data.R</code> script file for generating clean datasets, a <code>DATA_SOURCES.md</code> for documenting data provenance (source, license, download date, DOI), and a <code>README</code> .
makefile	Logical. If <code>TRUE</code> , adds a template <code>makefile.R</code> file to the project.

testthat	Logical. If TRUE, adds testthat infrastructure.
use_pipe	Logical. If TRUE, adds magrittr's pipe operator to the package. Default is FALSE since R >= 4.1.0 provides the native pipe operator  > which is recommended instead.
add_license	Character. Type of license to add. Options are "MIT", "GPL", "AGPL", "LGPL", "Apache", "CCBY", or "CC0". Default is NULL (no license added).
license_holder	Character. Name of the license holder / project author. Also used in CITATION.cff. Default is "Your name".
orcid	Character. ORCID iD of the project author (e.g. "0000-0001-7986-0308"). Used in CITATION.cff. Default is NULL.
use_git	Logical. If TRUE (default), initializes a local git repository.
create_github_repo	Logical. If TRUE, creates a GitHub repository. Note this requires some working infrastructure like git and a GITHUB_PAT. See <a href="https://usethis.r-lib.org/articles/articles/usethis-setup.html">https://usethis.r-lib.org/articles/articles/usethis-setup.html</a> .
private_repo	Logical. If TRUE (default), the GitHub repo will be private.
ci	Character. Type of continuous integration for your GitHub repository. Options are "none" (default) or "gh-actions" (uses GitHub Actions with R CMD check). Only used when create_github_repo = TRUE.
use_renv	Logical. If TRUE (default), initializes renv for dependency management. Creates a lockfile and .Rprofile bootstrap.
use_targets	Logical. If TRUE (default), adds a <code>_targets.R</code> template file for pipeline-based workflow management.
use_docker	Logical. If TRUE, adds a template Dockerfile for building a reproducible computational environment.
use_rproj	Logical. If TRUE (default), creates an RStudio project file (.Rproj). Set to FALSE when working exclusively in IDEs that do not use .Rproj files, such as Positron or VSCode. The .Rproj file is harmless in these IDEs but simply unused.
setwd_to_proj	Logical. If TRUE (default), sets the working directory to the new project after creation so you can continue working inside it. Ignored when open_proj = TRUE (the new session gets the project; the current session keeps its original WD). Set to FALSE for script-driven workflows where you want to stay in your current working directory - for example, when creating multiple projects in sequence.
open_proj	Logical. If TRUE, opens the newly created project in a new RStudio or Positron session (ignored in other IDEs). In that case, the current session keeps its original working directory, regardless of setwd_to_proj. Default is FALSE.
verbose	Logical. If TRUE, prints verbose output in the console while creating the new project. Default is FALSE.

## Details

SCIproj creates a research compendium that combines **structure** (where files live) with **workflow** (how analyses are reproduced). By default, it sets up:

- renv for dependency management (reproducible package versions)
- targets for pipeline-based workflow (automatic dependency tracking and caching)
- CITATION.cff for machine-readable citation metadata (FAIR)
- DATA\_SOURCES.md for data provenance documentation

Since R  $\geq$  4.1.0, the native pipe operator `|>` is available and recommended over the magrittr pipe `%>%`. The `use_pipe` parameter is therefore set to `FALSE` by default.

## Value

Invisibly returns the path to the new project directory.

## Examples

```
## Not run:
library("SCIProj")

# Minimal project (includes renv + targets by default)
create_proj("myproject")

# Names with underscores/hyphens are fine - the R package name is
# auto-cleaned (e.g. "baltic_cod" -> "baltic.cod" in DESCRIPTION)
create_proj("baltic_cod_analysis")

# Full-featured project
create_proj("my_research_project",
  add_license = "MIT",
  license_holder = "Jane Doe",
  orcid = "0000-0001-2345-6789",
  create_github_repo = TRUE,
  ci = "gh-actions"
)

# Minimal without workflow tools
create_proj("myproject", use_renv = FALSE, use_targets = FALSE)

## End(Not run)
```

# Index

create\_proj, [2](#)