

Package: stanflow (via r-universe)

September 30, 2026

Title A Mildly Opinionated Stan Bayesian Workflow

Version 0.2.0

Date 2026-01-24

Description Streamlines access to the Stan ecosystem by attaching core workflow packages, surfacing conflicts, and providing helpers to set up Stan interfaces with sensible defaults. Facilitates the Bayesian Workflow as described in Gelman et al. (2020) <<https://arxiv.org/abs/2011.01808>>.

License BSD_3_clause + file LICENSE

URL <https://stanflow.visruth.com/>

BugReports <https://github.com/VisruthSK/stanflow/issues/>

Depends R (>= 4.2.0)

Imports ascribe (>= 0.2.0), bayesplot, cli, loo, posterior, projpred, shinystan, withr

Suggests brms, cmdstanr, knitr, quarto, rstan, rstanarm, testthat (>= 3.0.0)

VignetteBuilder quarto

Additional_repositories <https://community.r-multiverse.org>,
<https://stan-dev.r-universe.dev>

Config/roxygen2/version 8.1.0

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/pak/sysreqs cmake libglib-dev make libicu-dev libuv1-dev libxml2-dev zlib1g-dev

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

Date/Publication 2026-08-31 18:45:08 UTC

RemoteUrl <https://github.com/VisruthSK/stanflow>

RemoteRef HEAD

RemoteSha 2dd443214f1033fdf174eb0ba52e03586c669543

Contents

flow_check	2
setup_brms	3
setup_cmdstanr	3
setup_interface	5
setup_rstan	6
setup_rstanarm	7
stan_cite	8
stan_logo	9
stan_repos	9
stanflow_conflicts	10
stanflow_deps	11
stanflow_logo	12
stanflow_update	12
Index	14

flow_check	<i>Print stanflow status and conflicts</i>
------------	--

Description

Print a consolidated status report showing attached packages, available interfaces, and any conflicts.

Usage

```
flow_check(check_updates = FALSE, only = NULL)
```

Arguments

- check_updates If TRUE, checks for stable updates to stanflow packages.
- only Set this to a character vector to restrict to conflicts only between the provided packages and loaded stanflow packages.

Value

Invisibly returns the character vector that was printed.

Examples

```
flow_check()
```

 setup_brms

Setup brms

Description

Configures brms to use available cores and sets the backend. Prefer `setup_interface()` for user-facing setup since it performs argument validation and defaults; `setup_brms()` assumes inputs are already checked.

Usage

```
setup_brms(quiet, brms_backend, cores, dry_run = FALSE)
```

Arguments

<code>quiet</code>	Logical. If TRUE, suppresses status messages. This cannot suppress cmdstan messages.
<code>brms_backend</code>	Character. The brms backend to use. Defaults to <code>getOption("brms.backend", "cmdstanr")</code> and must be one of <code>c("cmdstanr", "rstan")</code> .
<code>cores</code>	Integer. Number of cores to use. Defaults to <code>getOption("mc.cores")</code> . You must set <code>options(mc.cores = ...)</code> or pass <code>cores</code> explicitly.
<code>dry_run</code>	Logical. If TRUE, previews mutating setup actions without installing, attaching, changing options, or prompting. Dry-run output is shown even when <code>quiet = TRUE</code> .

Value

Returns NULL invisibly.

Examples

```
## Not run:
setup_brms(quiet = TRUE, brms_backend = "cmdstanr", cores = 2)

## End(Not run)
```

 setup_cmdstanr

Setup cmdstanr and CmdStan

Description

Checks the C++ toolchain, locates CmdStan, and installs or upgrades CmdStan if needed. Prefer `setup_interface()` for user-facing setup since it performs argument validation and defaults; `setup_cmdstanr()` assumes inputs are already checked.

Usage

```
setup_cmdstanr(  
  quiet,  
  force,  
  reinstall = FALSE,  
  check_updates = FALSE,  
  cores,  
  dry_run = FALSE  
)
```

Arguments

quiet	Logical. If TRUE, suppresses status messages. This cannot suppress cmdstan messages.
force	Logical. If TRUE, allows installation in non-interactive sessions.
reinstall	Logical. If TRUE, forces re-installation.
check_updates	Logical. If TRUE, checks for CmdStan updates.
cores	Integer. Number of cores to use. Defaults to <code>getOption("mc.cores")</code> . You must set <code>options(mc.cores = ...)</code> or pass cores explicitly.
dry_run	Logical. If TRUE, previews mutating setup actions without installing, attaching, changing options, or prompting. Dry-run output is shown even when <code>quiet = TRUE</code> .

Value

Returns TRUE invisibly when no install/upgrade is needed. Otherwise, returns NULL invisibly after installation.

Examples

```
## Not run:  
setup_cmdstanr(  
  quiet = TRUE,  
  force = TRUE,  
  reinstall = FALSE,  
  check_updates = FALSE,  
  cores = 2  
)  
  
## End(Not run)
```

setup_interface	<i>Setup and Load Stan Interfaces</i>
-----------------	---------------------------------------

Description

This function ensures specific Stan interfaces are installed, configured, and loaded. It handles package installation (from R-multiverse/CRAN (stable) or Stan universe (dev)) and performs necessary one-time setup (like installing CmdStan).

Usage

```
setup_interface(
  interface = c("brms", "cmdstanr", "rstan", "rstanarm"),
  cores = getOption("mc.cores"),
  quiet = getOption("stanflow.quiet", FALSE),
  force = FALSE,
  reinstall = FALSE,
  check_updates = FALSE,
  dev = FALSE,
  brms_backend = c("cmdstanr", "rstan"),
  rstan_auto_write = TRUE,
  dry_run = FALSE
)
```

Arguments

interface	A character vector. Select at least one of: "brms", "cmdstanr", "rstan", "rstanarm".
cores	Integer. Number of cores to use. Defaults to <code>getOption("mc.cores")</code> . You must set <code>options(mc.cores = ...)</code> or pass cores explicitly.
quiet	Logical. If TRUE, suppresses status messages. This cannot suppress cmdstan messages.
force	Logical. If TRUE, allows installation in non-interactive sessions.
reinstall	Logical. If TRUE, forces re-installation.
check_updates	Logical. If TRUE, checks for CmdStan updates.
dev	Logical. If FALSE (default), installs stable releases from R-multiverse or CRAN. If TRUE, installs development versions from Stan R-universe.
brms_backend	Character. The brms backend to use. Defaults to <code>getOption("brms.backend", "cmdstanr")</code> and must be one of <code>c("cmdstanr", "rstan")</code> .
rstan_auto_write	Logical. If TRUE (default), sets <code>rstan::rstan_options(auto_write = TRUE)</code>
dry_run	Logical. If TRUE, previews mutating setup actions without installing, attaching, changing options, or prompting. Dry-run output is shown even when <code>quiet = TRUE</code> .

Details

The setup functions are exported (e.g., `setup_brms()`) for transparency. Each function has some side effects, mainly setting `mc.cores`, see the function for specifics.

Value

Returns attached package names invisibly. With `dry_run = TRUE`, returns the package names that would be attached.

Examples

```
## Not run:
options(mc.cores = 2)
setup_interface("cmdstanr", quiet = TRUE)
setup_interface(
  c("brms", "cmdstanr"),
  brms_backend = "cmdstanr",
  quiet = TRUE
)

## End(Not run)
```

setup_rstan

Setup rstan

Description

Configures `rstan` to use available cores and write compiled models to disk. Prefer `setup_interface()` for user-facing setup since it performs argument validation and defaults; `setup_rstan()` assumes inputs are already checked.

Usage

```
setup_rstan(quiet, cores, rstan_auto_write, dry_run = FALSE)
```

Arguments

<code>quiet</code>	Logical. If <code>TRUE</code> , suppresses status messages. This cannot suppress <code>cmdstan</code> messages.
<code>cores</code>	Integer. Number of cores to use. Defaults to <code>getOption("mc.cores")</code> . You must set <code>options(mc.cores = ...)</code> or pass <code>cores</code> explicitly.
<code>rstan_auto_write</code>	Logical. If <code>TRUE</code> (default), sets <code>rstan::rstan_options(auto_write = TRUE)</code>
<code>dry_run</code>	Logical. If <code>TRUE</code> , previews mutating setup actions without installing, attaching, changing options, or prompting. Dry-run output is shown even when <code>quiet = TRUE</code> .

Value

Returns NULL invisibly.

Examples

```
## Not run:
setup_rstan(quiet = TRUE, cores = 2, rstan_auto_write = TRUE)

## End(Not run)
```

setup_rstanarm	<i>Setup rstanarm</i>
----------------	-----------------------

Description

Configures rstanarm to use available cores. Prefer setup_interface() for user-facing setup since it performs argument validation and defaults; setup_rstanarm() assumes inputs are already checked.

Usage

```
setup_rstanarm(quiet, cores, dry_run = FALSE)
```

Arguments

quiet	Logical. If TRUE, suppresses status messages. This cannot suppress cmdstan messages.
cores	Integer. Number of cores to use. Defaults to getOption("mc.cores"). You must set options(mc.cores = ...) or pass cores explicitly.
dry_run	Logical. If TRUE, previews mutating setup actions without installing, attaching, changing options, or prompting. Dry-run output is shown even when quiet = TRUE.

Value

Returns NULL invisibly.

Examples

```
## Not run:
setup_rstanarm(quiet = TRUE, cores = 2)

## End(Not run)
```

stan_cite

*Cite Stan packages in a project/files***Description**

stan_cite() finds Stan packages and functions used in a project, then returns their citations as BibTeX or bibentry records.

Usage

```
stan_cite(
  path = ".",
  strict = FALSE,
  format = c("bibtex", "bibentry"),
  skip_dirs = ascribe::scan_skip_dirs(),
  ignore_unqualified_functions = ascribe::stdlib_funs(),
  use_knitr = FALSE,
  quiet = getOption("stanflow.quiet", FALSE)
)
```

Arguments

path	A single project directory (searched recursively) or a vector of files (.R/.Rmd/.qmd).
strict	If FALSE (default), warn on ambiguous function calls whose origin cannot be determined exactly. If TRUE, abort on ambiguous calls.
format	One of "bibtex" or "bibentry".
skip_dirs	Character vector of directory names to skip when scanning a directory. Defaults to scan_skip_dirs().
ignore_unqualified_functions	Defaults to exports from base R packages listed in stdlib_funs(). Character vector of function names to ignore when attributing (unqualified) calls. Calls like pkg::fun() will NOT be ignored even if fun is in ignore_unqualified_functions, since they are namespaced.
use_knitr	Logical. If TRUE, parse .Rmd and .qmd files with knitr::purl(), which resolves knitr features the in-house parser ignores, such as child documents. It also comments out eval=FALSE and purl=FALSE chunks, so usage in them goes unrecorded. Defaults to FALSE.
quiet	Logical. If TRUE, suppresses status messages. Defaults to FALSE.

Value

A BibTeX character vector or a bibentry object.

Examples

```

path <- tempfile(fileext = ".R")
writeLines(
  c(
    "# one messy analysis file",
    "library(posterior)",
    "requireNamespace(\"loo\")",
    "draws <- as_draws(list(mu = rnorm(10)))",
    "posterior::rhat(draws)",
    "loo::loo(matrix(1))"
  ),
  path
)

stan_cite(path, quiet = TRUE)
stan_cite(path, format = "bibentry", quiet = TRUE)
unlink(path)

```

stan_logo	<i>Print the Stan ASCII art logo</i>
-----------	--------------------------------------

Description

Print the Stan ASCII art logo

Usage

```
stan_logo()
```

Value

Invisibly returns the character vector that was printed.

Examples

```
stan_logo()
```

stan_repos	<i>Stan package repositories</i>
------------	----------------------------------

Description

Stan package repositories

Usage

```
stan_repos(dev = FALSE)
```

Arguments

`dev` Include the development r-universe repo—don't use this unless you need the latest commits.

Value

Character vector of repository URLs.

Examples

```
stan_repos()
stan_repos(dev = TRUE)
```

stanflow_conflicts	<i>Conflicts between stanflow and other packages</i>
--------------------	--

Description

List conflicts between stanflow packages and other attached packages.

Usage

```
stanflow_conflicts(only = NULL)

## S3 method for class 'stanflow_conflicts'
print(x, ...)
```

Arguments

`only` Defaults to NULL. Set this to a character vector to restrict to conflicts only between the provided packages and loaded stanflow packages.

`x` A stanflow_conflicts object, usually from [stanflow_conflicts\(\)](#).

`...` Unused. Included for consistency with [base::print\(\)](#).

Details

There are several conflicts that are deliberately ignored: `diag`, `drop`, `match`, `%in%`, `mad`, `sd`, and `var` from posterior.

Adapted from [tidyverse::tidyverse_conflicts\(\)](#) for stanflow's package set.

Value

Invisibly returns `x`.

Examples

```
stanflow_conflicts()
stanflow_conflicts(c("base"))
conflicts <- stanflow_conflicts()
print(conflicts)
```

stanflow_deps	<i>List all stanflow dependencies</i>
---------------	---------------------------------------

Description

Returns a data frame of Stan workflow packages and their local/remote versions. When `check_updates = FALSE`, remote versions are not queried and the `remote` and `behind` columns are NA and FALSE, respectively.

Adapted from `tidyverse::tidyverse_deps()`.

Usage

```
stanflow_deps(recursive = FALSE, dev = FALSE, check_updates = TRUE)
```

Arguments

<code>recursive</code>	If TRUE, will also list dependencies of dependencies. When <code>check_updates = TRUE</code> , the recursive traversal follows only "strong" dependencies (Depends/Imports/LinkingTo), so Suggests are not expanded recursively.
<code>dev</code>	If FALSE (default), checks for updates in the R-multiverse or CRAN (stable releases). If TRUE, checks the Stan R-universe (dev versions). This is only cogent for Stan packages, and cannot compare two dev versions.
<code>check_updates</code>	Logical. If FALSE, skips checking for remote versions and only reports locally installed package versions.

Value

A data frame with columns:

package Package name.

remote Repository version (character, NA when not queried).

local Installed version (character, "0" if not installed).

behind Logical; TRUE when remote is newer than local.

Examples

```
## Not run:
# Full dependency check with remote versions
stanflow_deps(recursive = TRUE)

# Local-only inventory (fast, no network)
stanflow_deps(check_updates = FALSE)

## End(Not run)
```

stanflow_logo	<i>Print the stanflow ASCII art logo</i>
---------------	--

Description

Adapted from `tidyverse::tidyverse_logo()`.

Usage

```
stanflow_logo()
```

Value

Invisibly returns the character vector that was printed.

Examples

```
stanflow_logo()
```

stanflow_update	<i>Update stanflow packages</i>
-----------------	---------------------------------

Description

Checks for outdated Stan workflow packages and installs updates. This function requires an interactive R session for installation unless `dry_run = TRUE`. Dry runs check repositories and preview installs without prompting or installing packages. Adapted from `tidyverse::tidyverse_update()`.

Usage

```
stanflow_update(recursive = FALSE, dev = FALSE, dry_run = FALSE)
```

Arguments

recursive	If TRUE, will also list dependencies of dependencies. When check_updates = TRUE, the recursive traversal follows only "strong" dependencies (Depends/Imports/LinkingTo), so Suggests are not expanded recursively.
dev	If FALSE (default), checks for updates in the R-multiverse or CRAN (stable releases). If TRUE, checks the Stan R-universe (dev versions). This is only cogent for Stan packages, and cannot compare two dev versions.
dry_run	Logical. If TRUE, previews update steps without installing packages or prompting.

Value

Invisibly returns a data frame of outdated packages (same columns as [stanflow_deps](#)). Returns NULL invisibly when no updates are needed.

Examples

```
## Not run:  
# Update direct dependencies only  
stanflow_update()  
  
# Update full dependency tree (including suggests)  
stanflow_update(recursive = TRUE)  
  
## End(Not run)
```

Index

`base::print()`, [10](#)

`flow_check`, [2](#)

`print.stanflow_conflicts`
 (`stanflow_conflicts`), [10](#)

`setup_brms`, [3](#)

`setup_cmdstanr`, [3](#)

`setup_interface`, [5](#)

`setup_rstan`, [6](#)

`setup_rstanarm`, [7](#)

`stan_cite`, [8](#)

`stan_logo`, [9](#)

`stan_repos`, [9](#)

`stanflow_conflicts`, [10](#)

`stanflow_conflicts()`, [10](#)

`stanflow_deps`, [11](#), [13](#)

`stanflow_logo`, [12](#)

`stanflow_update`, [12](#)

`tidyverse::tidyverse_conflicts()`, [10](#)

`tidyverse::tidyverse_deps()`, [11](#)

`tidyverse::tidyverse_logo()`, [12](#)

`tidyverse::tidyverse_update()`, [12](#)