Package: hvtiR (via r-universe)

October 22, 2026

Type Package

Title Install and Check the HVTI R Package Family

Version 1.3.2

Date 2026-10-22

Description Installs, updates and reports on the Cleveland Clinic Heart and Vascular Institute CORR group's family of R packages in a single step. Members are resolved from public GitHub repositories with 'pak', keeping installations aligned with the source where family releases land first. Provides a status table comparing installed versions and source commits against GitHub 'main', and a diagnostic report for environment problems.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports cli, utils

Suggests curl, gitcreds, knitr, pak, quarto, rstudioapi, testthat (>= 3.2.0), withr

VignetteBuilder quarto

SystemRequirements Quarto command line tool (<https://github.com/quarto-dev/quarto-cli>)

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

URL https://github.com/ehrlinger/hvtiR, https://ehrlinger.github.io/hvtiR/

BugReports https://github.com/ehrlinger/hvtiR/issues

Config/roxygen2/version 8.1.0

Repository https://ehrlinger.r-universe.dev

Date/Publication 2026-10-22 16:40:07 UTC

RemoteUrl https://github.com/ehrlinger/hvtiR

RemoteRef HEAD

RemoteSha 8e03679f900f389d23a6071d44d469ac441effde

Contents

doctor 2
install 3
members 3
setup 4
status 6
update 7

Index **8**

`doctor` *Diagnose an hvtiR installation*

Description

Reports the running R version against the strictest requirement in the package family, whether `pak` is installed, the platform, the configured package repositories (warning when one is a dated snapshot), and then the full member status table. When a remote check fails, reports the reason retained by `status()`. With `remote`, it also reads each member's `DESCRIPTION` from GitHub and names every dependency floor the repositories cannot meet, such as `varPro (>= 3.3.0)` against a snapshot that predates it. It also names any installed member whose dependency is missing from the library path, and warns when the study's `renv.lock` was made under a different R version than the running one. This is the report to run first when a member will not install.

Usage

```
doctor(remote = TRUE)
```

Arguments

`remote`	Consult GitHub for the latest versions, and the configured repositories for the dependencies members need? Passed through to `status()`.

Value

The `status()` data frame, invisibly. Called for the report it prints.

Examples

```
# Offline: environment checks plus what is installed
doctor(remote = FALSE)
```

install — *Install every hvtiR member*

Description

Installs all members from GitHub `main`, whether or not they are already present. This is the fresh-machine command; use `update()` to install only what is missing or out of date.

Usage

```
install(force = FALSE)
```

Arguments

`force`	Bypass the loaded-namespace guard. A package whose namespace is loaded cannot be safely overwritten; on Windows the write fails and leaves a broken library. Unsafe: restart R instead.

Details

Members are installed from GitHub rather than CRAN because GitHub is where family releases land first. CRAN is a downstream republication for members that are published there.

Value

The character vector of `"owner/repo"` specs passed to pak, invisibly.

Examples

```
## Not run:
install()

## End(Not run)
```

members — *Members of the hvtiR*

Description

The registry of R packages that make up the HVTI package family, together with the GitHub repository each one installs from.

Usage

```
members()
```

Details

Every member's package name currently matches its repository name, but the mapping is stored rather than derived. The two diverged until the 2026-08 renames – `hvtiRpropensity` lived in `ehrlinger/hvtiPropensityScores` and `TemporalHazard` in `ehrlinger/temporal_hazard` – and a derived repo would fail silently the next time a name moves.

Value

A data frame with one row per member and two character columns:

package The installed package name.

repo The GitHub repository, as `"owner/repo"`.

Examples

```
members()
```

`setup` *Set up a study's environment*

Description

The one command a study needs before analysis. In order: checks R, installs renv if it is missing, and turns off RStudio's `.RData` restore and save; runs `hvtiRutilities::study_setup()` with renv on, which cleans up template copies before renv when adopting; declares `hvtiR`, every member, the packages they need and the team's standard analysis packages in the study's `DESCRIPTION`; installs them into the study's library with pak and renv; snapshots; checks that every package's dependencies resolve there; and restarts into the project.

Usage

```
setup(study, study_tracker_id, root = NULL, adopt = FALSE, upgrade = TRUE, ...)
```

Arguments

`study`	The Study Tracker title.
`study_tracker_id`	The Study Tracker ID, an integer.
`root`	The study root. By default the study enclosing the working directory, or the working directory.
`adopt`	Set up an existing study, adding only what is missing. Adopting cleans up template copies unless `cleanup = FALSE` is passed.
`upgrade`	Whether the install may move versions the lockfile pins. The default, `TRUE`, moves them when the declared family needs newer ones: the guard lists each move (package, old and new version, `RemoteSha` changes) and the snapshot records the new versions. Pass `FALSE` for a study in revision, whose versions must not move: a locked package the install would move then stops the run and `renv.lock` is left unchanged.

`...` Passed to `hvtiRutilities::study_setup()`, such as `cleanup`, `template_dir` or `owner`. `renv` is not accepted: `setup()` always uses renv, so use `hvtiRutilities::study_setup()` directly for a study without it.

Details

renv runs in explicit mode: the lockfile records what the study's `DESCRIPTION` declares, not whatever the code happens to mention. A study that already has a `renv.lock` keeps its locked versions, and on first conversion every package that lockfile recorded is declared, so none drop out. They are restored, and only packages the lockfile lacks are installed. By default (`upgrade = TRUE`) it moves locked versions the declared family needs and says which; `upgrade = FALSE` keeps a study in revision on its locked versions.

The standard analysis packages are `tidyverse`, `haven`, `gtsummary`, `flextable`, `survival`, `survminer`, `randomForestSRC`, `varPro` and `pacman`, all from the configured repositories. A study needs no separate step to have them; anything beyond them is added with `hvtiRutilities::study_need()`.

`setup()` replaces packages in the study's library, so run it from a session started outside the study project, passing `root`. From inside the project it stops before installing anything, and names the command to run.

In the install, the study stops using renv's per-user cache (`renv::settings$use.cache(FALSE)`, kept in `renv/settings.json`, so it applies to everyone), links already in the library are copied in (`renv::isolate()`), and any link whose target the user cannot read is removed, as is pak's leftover `_cache` directory, and the package reinstalled. A shared library cannot depend on one user's home.

No personal GitHub token is needed. The family is installed with pak, which carries a shared token, into the study's library, and renv keeps the environment: it restores the lockfile first, then snapshots, and installs from the configured repositories the build-time (`LinkingTo`) dependencies pak skips. Restoring a package the lockfile pins on GitHub goes through renv, which has no shared token; a `GITHUB_PAT` (or a token in the git credential store) covers that, and `setup()` warns when there is none.

Safe to rerun: each step checks before it acts.

Value

The study status from `hvtiRutilities::study_setup()`, invisibly.

Examples

```
## Not run:
setup(
  study = "Study Tracker title", study_tracker_id = 1234L,
  root = "/studies/example"
)

## End(Not run)
```

status *Version status of every hvtiR member*

Description

Compares each member installed locally against the version and, when pak recorded its GitHub commit, the commit on the `main` branch of its repository. The returned table remains version-focused; commit provenance is an internal tie-breaker when versions match.

Usage

```
status(remote = TRUE)
```

Arguments

`remote`	Consult GitHub for the latest versions? When `FALSE`, no network request is made, `latest` is `NA` throughout, and installed members report `"ok-local"`.

Details

`hvtiR` itself is checked separately because it is not a family member. It remains outside the returned table, but the printed report says when the installer is behind and names the command that updates it.

The object is returned visibly and has a `print` method, so a bare call displays the table while `st <- status()` captures the data frame for scripting.

Value

A data frame of class `hvtiR_status`, one row per member, with character columns:

package The package name.

repo The GitHub repository it installs from.

installed Installed version, or `NA` if not installed.

latest Version on GitHub `main`, or `NA`.

status One of `"ok"`, `"stale"`, `"missing"`, `"ahead"`, `"unknown"` or `"ok-local"`. A member is also `"stale"` when its version matches `main` but its recorded GitHub commit does not.

Examples

```
# Offline: reports what is installed without contacting GitHub
status(remote = FALSE)
```

update *Update out-of-date hvtiR members*

Description

Installs only the members whose status is "missing" or "stale". When everything is current, reports that and installs nothing. Members whose version or commit could not be checked against GitHub are reported as unchecked rather than silently treated as current.

Usage

```
update(force = FALSE)
```

Arguments

force	Bypass the loaded-namespace guard. See install().

Details

The target set is expanded over in-family dependencies (see member_deps()) before installing, so a stale member's in-family dependency is sent to pak alongside it even when that dependency is already current. Without this, installing e.g. just hvtiRlifetables sends pak to CRAN to resolve its TemporalHazard import, where the required version may not exist.

hvtiR itself is checked by status() and reported here, but never installed. update() reuses that check; calling it means the installer's namespace is already loaded, so it cannot update itself. When the installer is behind, the report names pak::pak("ehrlinger/hvtiR") as the remedy.

If a required member is already loaded, update() warns and installs nothing. Restart R and run update() before those packages attach. The returned empty character vector carries a blocked attribute listing the loaded members, so callers can distinguish this from an up-to-date result.

Value

The character vector of "owner/repo" specs passed to pak, invisibly. Empty if nothing needed updating. If loaded members prevent installation, the empty vector has a blocked attribute with their names.

Examples

```
## Not run:
update()

## End(Not run)
```

Index

doctor, 2

install, 3
install(), *7*

members, 3

setup, 4
status, 6
status(), *2*, *7*

update, 7
update(), *3*