Package: rgeowheels (via r-universe)

May 8, 2026

Type Package

Title Convenient Access to 'Python' 'Wheel' Packages Prepared for 'Windows'

Version 0.1.0

Author Andrew G. Brown

Maintainer Andrew G. Brown <brown.andrewg@gmail.com>

Description Downloads pre-compiled 'Windows' 'wheel' files (.whl) for 'Python' geospatial packages from <https://github.com/cgohlke/geospatial-wheels>. These are unofficial binary installers for some geospatial libraries prepared by Christoph Gohlke. Wheels for various packages, 'Python' versions, and architectures are made available via 'GitHub' releases, providing the ability to revert to prior versions when needed. Automatically detects and matches Python versions, with support for virtual environments and conda environments.

Imports tools, utils, jsonlite, httr

Suggests tinytest, reticulate, knitr, rmarkdown

Encoding UTF-8

Language en-US

License CC0

URL https://github.com/brownag/rgeowheels, https://humus.rocks/rgeowheels/

BugReports https://github.com/brownag/rgeowheels/issues

LazyData true

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Config/pak/sysreqs libssl-dev

Repository https://ncss-tech.r-universe.dev

Date/Publication 2025-11-09 00:43:48 UTC

RemoteUrl https://github.com/brownag/rgeowheels

RemoteRef HEAD

RemoteSha 4c6449f6bba74672cd6f5c4ee4b916dfceb7c216

Contents

`detect_python_envs` *Detect Available Python Environments*

Description

Scan the system for available Python environments including virtual environments, conda environments, and system Python.

Usage

```
detect_python_envs(include_system = TRUE, project_root = getwd())
```

Arguments

`include_system`	Include system Python in results? Default: `TRUE`
`project_root`	Directory to scan for project-local virtual environments. Default: current working directory

Details

Scans for virtual environments in the following project-local directories (in order): `.venv`, `venv`, `.virtualenv`, `env`

Also detects active virtual environment via `VIRTUAL_ENV` environment variable and active conda environment via `CONDA_DEFAULT_ENV`.

Value

A *data.frame* with columns: `type` (venv/conda/system), `path`, `version`, `active`

Examples

```
## Not run:
  detect_python_envs()
  detect_python_envs(project_root = "/path/to/project")

## End(Not run)
```

`detect_python_version` *Detect Python Version*

Description

Extract the major.minor Python version from a Python binary.

Usage

```
detect_python_version(python = get_rgeowheels_python())
```

Arguments

`python`	Path to Python executable. Default: `get_rgeowheels_python()`

Value

character Python version in major.minor format (e.g., "3.11")

Examples

```
## Not run:
  detect_python_version()
  detect_python_version("/path/to/venv/bin/python")

## End(Not run)
```

`install_wheel` *Install Python Wheels From 'geospatial-wheels' Repository*

Description

Install Python Wheels From 'geospatial-wheels' Repository

Usage

```
install_wheel(
  package,
  version = "latest",
  pyversion = "latest",
  architecture = "win_amd64",
  python = get_rgeowheels_python(),
  destdir = tempdir(),
  url_only = FALSE,
  download_only = FALSE
)
```

Arguments

package	Python package name to install. e.g. "rasterio"
version	Python package version to install. Default "latest" determines latest version available from asset list (considers pyversion if set).
pyversion	Python version to install package for. Default "latest" determines latest version available from asset list. Use "auto" to detect the Python version from the specified Python binary.
architecture	Target architecture for the wheel to install. Default "win_amd64", alternatives include "win_arm64" and "win32".
python	Path to Python executable to use for install. Default: get_rgeowheels_python()
destdir	Destination directory for downloaded wheel file. Default: tempdir()
url_only	Return the URL of the .whl file without downloading? Default: FALSE
download_only	Download .whl file without attempting install? Default: FALSE

Value

Called for side effects (download and install a Python wheel). Returns *character* containing path to .whl file when url_only=TRUE or download_only=TRUE.

---

list_rgeowheels_assets

List assets available from "geospatial-wheels" repository

Description

List assets available from "geospatial-wheels" repository

Usage

```
list_rgeowheels_assets(
  release = NULL,
  update_cache = FALSE,
  check_freshness = FALSE
)
```

Arguments

`release`	Specify custom release to list assets for. Default: `NULL`
`update_cache`	Force update of wheel download index? Default: `FALSE`
`check_freshness`	Check if cached data is from the latest release? Default: `FALSE`

Value

A *data.frame* containing `package`, `version`, `pyversion`, `architecture` and other metadata about each asset in a release.

`refresh_rgeowheels_cache` *Refresh rgeowheels cache*

Description

Force update of the cached wheel download index from the latest GitHub release.

Usage

```
refresh_rgeowheels_cache()
```

Value

Called for side effects. Updates the local cache with the latest available wheels.

`set_rgeowheels_python` *Get or Set Python Path*

Description

Set the path the Python binary used to run installation commands. May be a system or virtual/conda environment.

Usage

```
set_rgeowheels_python(x)

get_rgeowheels_python()
```

Arguments

`x`	Path to `python` or `python3` binary.

Value

character Value of option `"rgeowheels.python"`, or, if set, the value of the system environment variable `"R_RGEOWHEELS_PYTHON"`. If neither are set, then the result of `Sys.which("python")` (or `Sys.which("python3")` if the former fails).

Index