

Package ‘reproj’

July 10, 2026

Type Package

Title Coordinate System Transformations for Generic Map Data

Version 0.8.0

Description Transform coordinates from a specified source to a specified target map projection. This uses the 'PROJ' library directly, via the 'PROJ' package. The 'reproj()' function is generic, methods may be added to remove the need for an explicit source definition. This is for use primarily to transform generic data formats and direct leverage of the underlying 'PROJ' library. (There are transformations that aren't possible with 'PROJ' and that are provided by the 'GDAL' library, a limitation which users of this package should be aware of.) The 'PROJ' library is available at <https://proj.org/>.

License GPL-3

Depends R (>= 3.2.5)

Imports crsmeta (>= 0.3.0), PROJ (>= 0.7.0), wk

Suggests testthat, covr

Encoding UTF-8

SystemRequirements PROJ (>= 6.3.1)

BugReports <https://github.com/hypertidy/reproj/issues>

URL <https://github.com/hypertidy/reproj>,
<https://hypertidy.github.io/reproj/>

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Michael D. Sumner [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2471-7511>)

Maintainer Michael D. Sumner <mdsumner@gmail.com>

Repository CRAN

Date/Publication 2026-07-10 12:00:02 UTC

Contents

reproj.sc	2
reproj_extent	4
Index	5

reproj.sc	<i>Reproject coordinates.</i>
-----------	-------------------------------

Description

Reproject coordinates from a matrix or data frame by explicitly specifying the 'source' and 'target' projections.

Usage

```
## S3 method for class 'sc'
reproj(x, target = NULL, ..., source = NULL)

## S3 method for class 'mesh3d'
reproj(x, target, ..., source = NULL)

## S3 method for class 'quadmesh'
reproj(x, target, ..., source = NULL)

## S3 method for class 'triangmesh'
reproj(x, target, ..., source = NULL)

reproj(x, target, ..., source = NULL, four = FALSE)

## S3 method for class 'matrix'
reproj(x, target, ..., source = NULL, four = FALSE)

## S3 method for class 'data.frame'
reproj(x, target, ..., source = NULL, four = FALSE)

reproj_xy(x, target, ..., source = NULL)

reproj_xyz(x, target, ..., source = NULL)
```

Arguments

x	coordinates
target	target specification (PROJ.4 string or epsg code)
...	arguments passed to PROJ::proj_trans()
source	source specification (PROJ.4 string or epsg code)
four	defunct, was 'if TRUE return four columns xyzt', use the shape of the input instead

Details

The transformation engine is the PROJ package, a wrapper around the PROJ library, so that we can simply give coordinates in data frame or matrix form, with a source projection and a target projection.

reproj accepts a wide variety of source and target strings, not just "proj4string", and we are completely subject to the rules and behaviours of the PROJ library. We always assume "visualization order", i.e. longitude then latitude, easting then northing (as X, Y).

The basic function `reproj()` takes input in generic form (matrix or data frame) and returns a matrix with columns matching the coordinate columns of the input (2, 3, or 4 of x, y, z, m), by transforming from map projection specified by the source argument to that specified by the target argument. Only column order is respected, column names are ignored.

This model of working also allows adding methods for specific data formats that already carry a suitable source projection string. Currently we support types from the silicate and quadmesh and rgl packages, and only the target string need be specified.

This model has obvious flexibility, for packages to import the generic and call it with the correct source (from the data format) and the target from user, or process controlled mechanism.

The source argument must be named, and if it is not present a light check is made that the source data could be "longitude/latitude" and transformation to target is applied (this can be controlled by setting options).

Functions `reproj_xy()` and `reproj_xyz()` are helpers for `reproj()` and always return 2- or 3-column matrix respectively.

Note that any integer input for source or target will be formatted to a character string like "EPSG:<integer_code>" as a simple convenience. Note that there are other authorities besides EPSG, so the pattern "AUTH:code" is a general one and you should really be explicit.

Other R packages for transforming coordinates are geared toward data that's in a particular format. It's true that only GDAL provides the full gamut of available geographic map projections, but this leaves a huge variety of workflows and applications that don't need that level of functionality.

Value

numeric matrix of the transformed coordinates, either 2, 3, or 4 columns depending on the shape of the input. Use `reproj_xy()` or `reproj_xyz()` for those specific 2- and 3-column cases.

Dependencies

The **PROJ** package ($\geq 0.7.0$) does all transformation work, wrapping the PROJ library ($\geq 6.3.1$).

Global options

Assuming longitude/latitude input:

The behaviour is controlled by user-settable options which on start up are `reproj.assume.longlat = TRUE` and `reproj.default.longlat = "OGC:CRS84"`.

If the option `reproj.assume.longlat` is set to `FALSE` then the source argument must be named explicitly, i.e. `reproj(xy, t_srs, source = s_srs)`, this is to help catch mistakes being made.

If the option `reproj.assume.longlat` is set to `TRUE` and the input data appear to be sensible longitude/latitude values, then the value of `reproj.default.longlat` is used as the assumed source projection.

Examples

```
reproj(cbind(147, -42), target = "+proj=laea +datum=WGS84",
      source = getOption("reproj.default.longlat"))
```

reproj_extent	<i>Reproject extent</i>
---------------	-------------------------

Description

A four figure extent (`xmin`, `xmax`, `ymin`, `ymax`) is used to approximate the boundary of its reprojected version by interpolating new vertices along each edge.

Usage

```
reproj_extent(extent, target, ..., source = NULL, dimension = c(64, 64))
```

Arguments

extent	a four element vector of extent <code>c(xmin, xmax, ymin, ymax)</code> , or matrix with 4 columns
target	target specification (PROJ.4 string or epsg code)
...	arguments passed to PROJ::proj_trans()
source	source specification (PROJ.4 string or epsg code)
dimension	a 2 element integer to give the discretization within each extent (defaults to 64x64)

Details

This is a simple version of what GDAL's 'SuggestedWarpOutput' does, and similar functions like the raster package 'projectExtent()'.
Internal functions unpack the various stages, and might be exposed in future. These stages are

1. interpolate around the boundary with correct ordering (can be used as a polygon or line)
2. reproject the interpolated boundary
3. summarize the interpolated boundary to the new extent

Value

four value extent `c(xmin, xmax, ymin, ymax)` or a matrix with four columns (matching the input)

Examples

```
reproj_extent(c(0, 10, 0, 20), "+proj=laea", source = "+proj=longlat")
```

Index

PROJ::proj_trans(), [2](#), [4](#)
reproj(reproj.sc), [2](#)
reproj(), [3](#)
reproj.sc, [2](#)
reproj_extent, [4](#)
reproj_xy(reproj.sc), [2](#)
reproj_xy(), [3](#)
reproj_xyz(reproj.sc), [2](#)
reproj_xyz(), [3](#)