Package ‘PROJ’

October 12, 2022

Title  Generic Coordinate System Transformations Using ‘PROJ’
Description  Currently non-operational, a harmless wrapper to allow package ‘reproj’ to install and
function while relying on the ‘proj4’ package.
Depends  R (>= 3.0.2)
License  GPL-3
Encoding  UTF-8
LazyData  true
Suggests  testthat (>= 2.1.0), spelling, knitr, rmarkdown
URL  https://github.com/hypertidy/PROJ
BugReports  https://github.com/hypertidy/PROJ/issues
RoxygenNote  7.1.1
Language  en-US
VignetteBuilder  knitr
NeedsCompilation  yes
Author  Michael D. Sumner [aut, cre] (<https://orcid.org/0000-0002-2471-7511>),
        Jeroen Ooms [ctb] (provided PROJ library support on Windows, and
        assistance with Windows configuration),
        Simon Urbanek [cph, ctb] (wrote original code versions for PROJ version
        6),
        Dewey Dunnington [ctb] (key code contributions, and provided libproj to
        improve things a lot)
Maintainer  Michael D. Sumner <mdsumner@gmail.com>
Repository  CRAN
Date/Publication  2020-10-19 17:20:10 UTC
```
ok_proj6

R topics documented:

ok_proj6 .............................. 2
proj_crs_text ......................... 3
proj_trans ............................ 4
xymap .................................. 5

Index

ok_proj6  Is 'PROJ library >= 6' available

Description

Test for availability of 'PROJ' system library version 6 or higher.

Usage

ok_proj6()

Details

On unix-alikes, this function is run in .onLoad() to check that version 6 functionality is available. On Windows, the load process sets the data file location with the version 6 API, and that is used as a test instead.

If 'PROJ' library version 6 is not available, the package still compiles and installs but is not functional.

The lack of function can be simulated by setting options(reproj.mock.noproj6 = TRUE), designed for use with the reproj package.

Value

logical, TRUE if the system library 'PROJ >= 6'

Examples

ok_proj6()
```
proj_crs_text

Generate a projection string.

Description

Input any accepted format of 'PROJ' coordinate reference system specification. Return value is a string in the requested format.

Usage

proj_crs_text(source, format = 0L)

Arguments

source input projection specification one of ('PROJ4', 'WKT2', 'EPSG', 'PROJJSON', ... see the library documentation link in Details)

format integer, 0 for 'WKT', 1 for 'PROJ'

Details

This function requires PROJ version 6.0 or higher to be useful. If not, this function simply returns 'NA'.

See the library documentation for details on input and output formats.

Some nuances of the format are not available, currently we use formats '0: PJ_WKT2_2018' '1: PJ_PROJ_5', '2: PROJJSON'.

Some formats are hard to read, such as WKT so for easy reading use cat().

Value

character string in requested format

Examples

# all examples are disabled
#cat(proj_crs_text("EPSG:4326", format = 0L))
#proj_crs_text("EPSG:4326", format = 1L)
#south55 <- "+proj=utm +zone=55 +south +ellps=GRS80 +units=m +no_defs +type=crs"
#proj_crs_text(proj_crs_text(south55), 1L)
proj_trans

Transform a set of coordinates with 'PROJ'

Description
A raw interface to 'proj_trans' in 'PROJ => 6', if it is available.

Usage

proj_trans(x, target, ..., source = NULL, z_ = NULL, t_ = NULL)

proj_trans_generic(x, target, ..., source = NULL, z_ = 0, t_ = 0)

Arguments

x  input coordinates (x,y, list or matrix see z_ and t_)
target  projection for output coordinates
...  ignored
source  projection of input coordinates (must be named)
z_  optional z coordinate vector
t_  optional t coordinate vector

Details

'proj_trans_generic()' and 'proj_trans()' have the same arguments, but differ in the default values of z_ and t_, 0 or NULL. 'proj_trans_generic()' always returns a list for 4 elements, 'proj_trans()' will return 2 or 4 depending on the input.

'proj_trans_generic()' is a misnomer in that 'proj_trans' is the function from the PROJ library that is now used.

Input 'x' is assumed to be 2-columns of "x", then "y" coordinates. If "z" or "t" is required pass these in as named vectors with "z_" and "t_". For simplifying reasons z_ and t_ must always match the length of x y. Both default to 0, and are automatically recycled to the number of rows in x so it's pretty flexible.

Values that are detected out of bounds by library PROJ are allowed, we return Inf in this case, rather than the error "tolerance condition error".

Value

list of transformed coordinates, with 4-elements x_, y_, z_, t_

References

see the PROJ library documentation for details on the underlying functionality.
Examples

# proj_trans(cbind(147, -42), "+proj=laea", source = "epsg:4326")
# proj_trans(cbind(147, -42), z_ = -2, "+proj=laea", source = "epsg:4326")
# proj_trans(cbind(147, -42), z_ = -2, t_ = 1, "+proj=laea", source = "epsg:4326")

Description

A copy of the xymap data set from the quadmesh package.

Details

A matrix of longitude/latitude values of the world coastline.
Index

ok_proj6, 2
proj_crs_text, 3
proj_trans, 4
proj_trans_generic (proj_trans), 4
xymap, 5