

Package: whatarelief (via r-universe)

August 24, 2024

Title Get topography elevation and online imagery data

Version 0.0.1.9012

Description Obtain elevation data, topography relief for any region on the Earth. Topography and bathymetry data is supported by default. Sensible defaults exist for usage, with a matrix of entire planet topography (and bathymetry) returned. The geographic extent can be modified (from whole-planet) to a simple region in longitude/latitude by 'xmin,xmax,ymin,ymax' range, or by specifying a grid exactly with extent, dimension, projection in generic or spatial formats ('terra' or 'raster'). Online sources for data are used, 'GEBCO' General Bathymetric Chart of the Oceans (GEBCO) as a background (to ~500m resolution), and Copernicus GLO-30 Digital Elevation Model for higher resolution (to ~30m resolution). Custom source/s of topography may be input to override defaults, links to file/s or URLs as required.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.0

Imports sds, readr, vapour (>= 0.9.3.9003)

Suggests terra, raster, knitr, rmarkdown, ximage, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/hypertidy/whatarelief>

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

Remotes hypertidy/ximage, hypertidy/vapour, hypertidy/sds

Config/testthat/edition 3

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

RemoteUrl <https://github.com/hypertidy/whatarelief>

RemoteRef HEAD

RemoteSha 4520ba7549c7f3f05b21775d2f3c1405c395923e

Contents

coastline	2
elevation	3
imagery	4
raad_source	5

Index	7
--------------	----------

coastline	<i>Get coastline data</i>
-----------	---------------------------

Description

Get a coastline for any region on Earth.

Usage

```
coastline(extent = NULL, ..., dimension = c(1024, 1024), projection = NULL)
```

Arguments

extent	extent xmin, xmax, ymin, ymax
...	arguments passed to [elevation()]
dimension	optional size of matrix to contour (a default is used)
projection	provide this for custom extents in a given projection

Details

This uses the [elevation()] function to get a coastline by contouring elevation data at 0. (This might return nothing.)

For 'extent' not provided it will be derived from a plot extent if one exists. It's the user's job to ensure the projection is appropriate (we don't have a graphics mechanism to record a current projection string.)

If no device is open and no extent given, a coastline for longlat is given.

Value

matrix of coordinates in longlat (or whatever projection is specified)

Examples

```
plot(coastline(extent = c(-180, 180, -90, 90)), type = "l")
```

elevation	<i>Get digital elevation data</i>
-----------	-----------------------------------

Description

Read elevation data for any region on Earth.

Usage

```
elevation(
  extent = c(-180, 180, -90, 90),
  ...,
  dimension = NULL,
  projection = NULL,
  resample = "bilinear",
  source = NULL,
  threshold = 0.5
)
```

Arguments

extent	a numeric vector of xmin,xmax,ymin,ymax or a terra or raster rast object
...	arguments passed to 'vapour::vapour_warp_raster'
dimension	optional output size in ncol,nrow (x,y)
projection	optional coordinate reference system to use (map projection), longlat is assumed
resample	resampling algorithm for the GDAL warper, "bilinear" by default
source	a GDAL raster source, to override the inbuild GEBCO + SRTM (in future we might patch in local source)
threshold	a size in degrees above which no SRTM data is queried (about)

Details

Originally using GEBCO 2019 as a background, and SRTM 30m resolution for regions that fit approximately within the size of an SRTM tile (these are 1 degree wide).

Note that data is streamed into memory, so don't make the dimensions of the 'x' target raster too big.

To use these data, please attribute the use of GEBCO to the GEBCO Compilation Group (2021) GEBCO 2021 Grid (doi:10.5285/c6612cbe-50b3-0cff-e053-6c86abc09f8f), and the use of Copernicus GLO30 to European Space Agency, Sinergise (2021). Copernicus Global Digital Elevation Model. Distributed by OpenTopography. <https://doi.org/10.5069/G9028PQB>.

Value

a matrix, or depending on format of 'extent' a terra rast or raster object with elevation data

Examples

```

image(elevation(), useRaster = TRUE)
image(elevation(c(100, 150, -60, -20)), useRaster = TRUE)
elevation(terra::rast())
elevation(raster::raster())

elevation(raster::raster(raster::extent(80, 120, -60, -40), res = 0.25, crs = "OGC:CRS84"))

elevation(terra::rast(terra::ext(c(-1, 1, -1, 1) * 15e3), nrows = 256, ncols = 256,
  crs = "+proj=laea +lat_0=44.6371 +lon_0=-63.5923"))

```

imagery

Get imagery data

Description

Read imagery data for any region on Earth.

Usage

```

imagery(
  extent = c(-180, 180, -90, 90),
  ...,
  dimension = NULL,
  projection = "OGC:CRS84",
  resample = "near",
  source = NULL
)

streetmap(
  extent = c(-180, 180, -90, 90),
  ...,
  dimension = NULL,
  projection = "OGC:CRS84",
  resample = "cubic",
  source = NULL
)

satbox(
  extent = c(-180, 180, -90, 90),
  ...,
  dimension = NULL,
  projection = "OGC:CRS84",
  resample = "near",
  source = NULL
)

```

Arguments

extent	a numeric vector of xmin,xmax,ymin,ymax or a terra or raster rast object
...	arguments passed to 'vapour::vapour_warp_raster'
dimension	optional output size in ncol,nrow (x,y)
projection	optional coordinate reference system to use (map projection), longlat is assumed
resample	resampling algorithm for the GDAL warper, "near" by default
source	a GDAL raster source, to override the inbuilt source/s

Details

Note that data is streamed into memory, so don't make the dimensions of the 'x' target raster too big.

By default we're expecting 3 bands of bytes, but if only one is available that is used instead.

To use these data, please attribute the image provider, or otherwise use your own sources.

'imagery()' defaults to Virtual Earth, 'streetmap()' to OSM, and 'satbox()' to Mapbox Satellite.

To use 'satbox()' you must set an environment variable such as 'MAPBOX_API_KEY' to your Mabox token.

Value

a matrix, or depending on format of 'extent' a terra rast or raster object with elevation data

Examples

```
plt <- function(x) {
  nr <- nrow(x)
  nc <- ncol(x)
  plot(NA, xlim = c(0, nc), ylim = c(0, nr), asp = 1);
  rasterImage(x, 0, 0, nc, nr)
}

plt(imagery())
plt(imagery(c(100, 150, -60, -20), projection = "OGC:CRS84"))
## can't do yet
#plt(imagery(terra::rast()))
#imagery(raster::raster())
```

 raad_source

Data sources raadtools knows about

Description

Currently just 'nsidc_25km_seaice' is available.

Usage

```
raad_source(id = "nsidc_25km_seaice", ...)
```

Arguments

id	raadtools id of data sourced
...	unused

Details

Obtains a listing of GDAL-read URLs indexed by raadtools.

I think the early URLs are wrong ... but later ones work, see the README

Value

data frame of dates and vrt dsn (names vary, this is WIP)

Examples

```
raad_source("nsidc_25km_seaice")
```

Index

coastline, 2

elevation, 3

imagery, 4

raad_source, 5

satbox (imagery), 4

streetmap (imagery), 4