

Package ‘ggOceanMaps’

July 22, 2025

Type Package

Title Plot Data on Oceanographic Maps using 'ggplot2'

Version 2.2.0

Date 2024-01-15

URL <https://mikkovihtakari.github.io/ggOceanMaps/>

BugReports <https://github.com/MikkoVihtakari/ggOceanMaps/issues>

Description Allows plotting data on bathymetric maps using 'ggplot2'. Plotting oceanographic spatial data is made as simple as feasible, but also flexible for custom modifications. Data that contain geographic information from anywhere around the globe can be plotted on maps generated by the `basemap()` or `qmap()` functions using 'ggplot2' layers separated by the '+' operator. The package uses spatial shape- ('sf') and raster ('stars') files, geospatial packages for R to manipulate, and the 'ggplot2' package to plot these files. The package ships with low-resolution spatial data files and higher resolution files for detailed maps are stored in the 'ggOceanMapsLargeData' repository on GitHub and downloaded automatically when needed.

Depends R (>= 3.5.0), ggplot2

Imports sf, stars, methods, utils, smoothr, units

Suggests ggspatial, cowplot, knitr, rmarkdown, scales, ggnewscale

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

LazyData true

LazyDataCompression xz

NeedsCompilation no

Author Mikko Vihtakari [aut, cre] (affiliation: Institute of Marine Research,
ORCID: <<https://orcid.org/0000-0003-0371-4319>>),
Roger Bivand [ctb],
Hadley Wickham [ctb]

Maintainer Mikko Vihtakari <mikko.vihtakari@hi.no>
Repository CRAN
Date/Publication 2024-01-15 13:10:05 UTC

Contents

basemap	2
dd_land	9
dd_rbathy	10
dist2land	10
fdir_main_areas	12
fdir_sub_areas	13
geonorge_bathymetry	14
get_depth	15
ices_areas	16
qmap	17
raster_bathymetry	20
reorder_layers	22
shapefile_list	23
theme_map	24
transform_coord	24
vector_bathymetry	26
Index	28

basemap	Create a ggplot2 basemap for plotting variables
---------	---

Description

Creates a ggplot2 basemap for further plotting of data.

Usage

```
basemap(  
  x = NULL,  
  limits = NULL,  
  data = NULL,  
  shapefiles = NULL,  
  crs = NULL,  
  bathymetry = FALSE,  
  glaciers = FALSE,  
  rotate = FALSE,  
  legends = TRUE,  
  legend.position = "right",  
  lon.interval = NULL,
```

```

    lat.interval = NULL,
    bathy.style = NULL,
    downsample = 0,
    bathy.border.col = NA,
    bathy.size = 0.1,
    bathy.alpha = 1,
    land.col = "grey60",
    land.border.col = "black",
    land.size = 0.1,
    gla.col = "grey95",
    gla.border.col = "black",
    gla.size = 0.1,
    grid.col = "grey70",
    grid.size = 0.1,
    base_size = 11,
    projection.grid = FALSE,
    expand.factor = 1,
    verbose = FALSE
  )

```

Arguments

x	The limit type (limits, data, or shapefiles) is automatically recognized from the class of this argument.
limits	Map limits. One of the following: • numeric vector of length 4: The first element defines the start longitude, the second element the end longitude (counter-clockwise), the third element the minimum latitude, and the fourth element the maximum latitude of the bounding box. Also accepts <code>sf::st_bbox</code> type named vectors with limits in any order. The coordinates can be given as decimal degrees or coordinate units for shapefiles used by a projected map. Produces a rectangular map. Latitude limits not given in min-max order are automatically ordered to respect this requirement. • single integer between 30 and 88 or -88 and -30 produces a polar map for the Arctic or Antarctic, respectively. Can be omitted if data or shapefiles are defined.
data	A data frame, sp, or <code>sf</code> shape containing longitude and latitude coordinates. If a data frame, the coordinates have to be given in decimal degrees. The limits are extracted from these coordinates and produce a rectangular map. Suited for situations where a certain dataset is plotted on a map. The function attempts to guess the correct columns and it is advised to use intuitive column names for longitude (such as "lon", "long", or "longitude") and latitude ("lat", "latitude") columns. Can be omitted if limits or shapefiles are defined.
shapefiles	Either a list containing shapefile information or a character argument referring to a name of pre-made shapefiles in <code>shapefile_list</code> . This name is partially matched. Can be omitted if limits or data is defined as decimal degrees.

<code>crs</code>	Coordinate reference system (CRS) for the map. If NULL (default), the CRS is selected automatically based on <code>limits</code> , <code>data</code> , or <code>shapefiles</code> . Passed to <code>st_crs</code> . Typically integers giving the EPSG code are the easiest. Cannot be used simultaneously with <code>rotate</code> .
<code>bathymetry</code>	Logical indicating whether bathymetry should be added to the map. Functions together with <code>bathy.style</code> . See Details.
<code>glaciers</code>	Logical indicating whether glaciers and ice sheets should be added to the map.
<code>rotate</code>	Logical indicating whether the projected maps should be rotated to point towards the pole relative to the mid-longitude limit.
<code>legends</code>	Logical indicating whether the legend for bathymetry should be shown.
<code>legend.position</code>	The position for ggplot2 legend. See the argument with the same name in theme .
<code>lon.interval, lat.interval</code>	Numeric value specifying the interval of longitude and latitude grids. NULL finds reasonable defaults depending on <code>limits</code> .
<code>bathy.style</code>	Character (plots bathymetry; list of alternatives in Details) or NULL ("raster_binned_blues" if <code>bathymetry = TRUE</code>) defining the bathymetry style. Partially matched, can be abbreviated, and used to control bathymetry plotting together with <code>bathymetry</code> . See Details.
<code>downsample</code>	Integer defining the downsampling rate for raster bathymetries. A value of 0 (default) does not downsample, 1 skips every second row, 2 every second and third. See geom_stars
<code>bathy.alpha</code>	Transparency parameter for the bathymetry fill color. See scale_alpha .
<code>land.col, gla.col, grid.col</code>	Character code specifying the color of land, glaciers, and grid lines, respectively. Use NA to remove the grid lines.
<code>land.border.col, gla.border.col, bathy.border.col</code>	Character code specifying the color of the border line for land, glacier, and bathymetry shapes.
<code>land.size, gla.size, bathy.size, grid.size</code>	Numeric value specifying the width of the border line land, glacier and bathymetry shapes as well as the grid lines, respectively. Use the LS function for a specific width in pt. See Details.
<code>base_size</code>	Base size parameter for ggplot. See ggtheme .
<code>projection.grid</code>	Logical indicating whether the coordinate grid should show projected coordinates instead of decimal degree values. Useful to define limits for large maps in polar regions.
<code>expand.factor</code>	Expansion factor for map limits. Can be used to zoom in (decrease the value under 1) and out (increase the value over 1) automatically (data) limited maps. Defaults to 1, which means that outermost data points are located at the boundaries of the plotting region.
<code>verbose</code>	Logical indicating whether information about the projection and guessed column names should be returned as messages. Set to FALSE to make the function silent.

Details

The function uses [ggplot2](#), [sf](#), [stars](#) and spatial files to plot maps of the world's oceans.

Limits

If the limits are in decimal degrees, the longitude limits (`[1:2]`) specify the start and end segments of corresponding angular lines that should reside inside the map area. The longitude limits are defined **counter-clockwise**. The latitude limits `[3:4]` define the parallels that should reside inside the limited region given the longitude segments. Note that the actual limited region becomes wider than the polygon defined by the coordinates (shown in Examples). Using data to limit the map, making the points barely fit into the map. The `expand.factor` argument can be used to adjust the space between map borders and points. If the limits are given as projected coordinates or as decimal degrees for maps with $-60 < \text{latitude} < 60$, limit elements represent lines encompassing the map area in cartesian space.

Projections

If the shapefiles are not specified, the function uses either the `limits` or `data` arguments to decide which projection to use. Up-to-date conditions are defined in [define_shapefiles](#) and [shapefile_list](#) functions. At the time of writing, the function uses three different projections (given as [EPSG codes](#))

- **3995** WGS 84 / Arctic Polar Stereographic. Called "ArcticStereographic". For max latitude (`limits[4]`) ≥ 60 (if min latitude (`limits[3]`) ≥ 30), and single integer latitudes ≥ 30 and ≤ 89 .
- **3031** WGS 84 / Antarctic Polar Stereographic. Called "AntarcticStereographic". For max latitude (`limits[4]`) ≤ -60 (if min latitude (`limits[3]`) ≤ -30), and single integer latitudes ≤ -30 and ≥ -89 .
- **4326** WGS 84 / World Geodetic System 1984, used in GPS. Called "DecimalDegree". For min latitude (`limits[3]`) < 30 or > -30 , max latitude (`limits[4]`) < 60 or > -60 , and single integer latitudes < 30 and > -30 .

The `rotate` argument changes the pre-defined projection such that mid-longitude point in the map points northward.

The `crs` argument can be used to define the projection, which can be useful when plotting, for instance, model data that are difficult to transform into another projection.

Bathymetry

Bathymetry can be plotted by simply specifying `bathymetry = TRUE` or `bathy.style` (you won't need to specify both any longer). The former uses a low-resolution raster file shipped with `ggOceanMaps`. The package contains an option to plot higher resolution bathymetries than the default binned blue alternative (`bathy.style = "raster_binned_blues"`). These bathymetries can be accessed by specifying the `bathy.style` argument and require a download from [ggOceanMapsLargeData](#) or other online repositories. The `bathy.style` character argument consists of three parts separated by a `_`. The first part gives the type: `raster`, `poly(gon)`, or `contour`. The two latter ones use vector data. The second part gives the resolution: `binned`, `continuous` or `user`. The `continuous` and `user` options cannot be used for vector data. The `user` option accepts any raster file that can be opened using [read_stars](#). The path to the file has to be stored in `ggOceanMaps.userpath` [option](#) (e.g. `options(ggOceanMaps.userpath = "PATH_TO_THE_FILE")`) (you can set this in `.Rprofile` to avoid having to type it every time). The last part defines the color: `blues` or `grays`. These options can be abbreviated by specifying the first letter of each part. Gray contour lines are an exception to the

rule above and can be plotted using `bathy.style = "contour_gray"`. Future versions may contain a combination of raster and gray contours, but these have not been implemented yet. Currently implemented `bathy.style` alternatives are:

- **NULL (default)**. Bathymetry style is searched from `getOption("ggOceanMaps.bathy.style")`. If not found, "raster_binned_blues" is used.
- "raster_binned_blues" or "rbb" plots binned raster bathymetry filled with different shades of blue. Does not require a download.
- "raster_binned_grays" or "rbg" the same than above but uses different shades of gray.
- "raster_continuous_blues" or "rcb" plots continuous raster bathymetry filled with different shades of blue. More detailed and visually more appealing than the binned bathymetry. Recommended. Requires a download.
- "raster_continuous_grays" or "rcg" the same than above but uses different shades of gray.
- "raster_user_blues" or "rub" plots continuous raster bathymetry filled with different shades of blue from `getOption("ggOceanMaps.user.path")`. Any file supported by [read_stars](#) should work. The file has to be placed into the location specified by the path. Experimental feature. Has been tested using **ETOPO 60 arc-second** and **GEBCO 15 arc-second** grids. Please report any bugs you find.
- "raster_user_grays" or "rug" the same than above but uses different shades of gray.
- "poly_binned_blues", "poly_blues", "pbb" or "pb" plots polygon bathymetry filled with different shades of blue. Default in the versions older than 2.0 of ggOceanMaps. Requires a download.
- "poly_binned_grays", "poly_grays", "pbg" or "pg" same than above but uses different shades of gray.
- "contour_binned_blues", "contour_blues", "cbb" or "cb" contour lines with different shades of blue. Requires a download.
- "contour_gray", "contour_gray" or "cg" plots gray contour lines. Requires a download.

The default can be changed by setting the `ggOceanMaps.bathy.style` option. `options(ggOceanMaps.bathy.style = "poly_blues")` would make the style similar to older pre-2.0 versions of ggOceanMaps.

Pre-made shapefiles

If the limits are not defined as decimal degrees (any longitude outside the range [-180, 180] or latitude [-90, 90]), the function will ask to specify shapefiles. The shapefiles can be defined by partially matching the names of the pre-made shapefiles in [shapefile_list](#) (e.g. "Ar" would be enough for "ArcticStereographic") or by specifying custom shapefiles.

Custom shapefiles

Custom shapefiles have to be a named list containing at least the following elements:

- **land** Object name of the [spatial polygon](#) containing land. Required.
- **glacier** Object name of the [spatial polygon](#) containing glaciers. Not required if glaciers are not needed.
- **bathy** Object name of the [spatial polygon](#) or [raster](#) containing bathymetry data. Not required if bathymetry is not needed.

See Examples.

Line width and font size

The line size aesthetics in [ggplot2](#) generates approximately 2.13 wider lines measured in pt than the given values. If you want a specific line width in pt, use the internal function [LS](#) to convert the desired line width to the ggplot2 equivalent. A similar function is also available for font sizes ([FS](#)).

Value

Returns a [ggplot](#) map, which can be assigned to an object and modified as any ggplot object.

Author(s)

Mikko Vihtakari

References

Note that if you use this function to generate maps for a publication, it is advised to cite the underlying data. The spatial data used by this function have been acquired from the following sources:

- **Land polygons.** [Natural Earth Data](#) 1:10m Physical Vectors with the Land and Minor Island datasets combined. Distributed under the [CC Public Domain license \(terms of use\)](#).
- **Glacier polygons.** [Natural Earth Data](#) 1:10m Physical Vectors with the Glaciated Areas and Antarctic Ice Shelves datasets combined. Distributed under the [CC Public Domain license \(terms of use\)](#)
- **Bathymetry.** [NOAA National Centers for Environmental Information](#). 2022: ETOPO 2022 15 Arc-Second Global Relief Model. NOAA National Centers for Environmental Information. doi:10.25921/fd45gt74. Distributed under the [U.S. Government Work license](#).

See Also

[ggplot](#)

Other basemap functions: [qmap\(\)](#), [shapefile_list\(\)](#), [transform_coord\(\)](#)

Examples

```
# The easiest way to produce a map is to use the limits
# argument and decimal degrees:

basemap(limits = 60) # synonym to basemap(60)

# Bathymetry can be added using the respective argument:
basemap(limits = -60, bathymetry = TRUE)

## Not run:
# Glaciers require a download in the new version:
basemap(limits = -60, glaciers = TRUE, shapefiles = "Arctic")

## End(Not run)

# The easiest way to add data on the maps is to use the ggspatial functions:
```

```

dt <- data.frame(lon = c(-150, 150), lat = c(60, 90))
if(requireNamespace("ggspatial", quietly = TRUE)) {
  basemap(data = dt, bathymetry = TRUE) +
    ggspatial::geom_spatial_point(data = dt, aes(x = lon, y = lat),
      color = "red")
}
## Not run:
# Note that writing out data = dt is required because there are multiple
# underlying ggplot layers plotted already:
basemap(data = dt) +
  ggspatial::geom_spatial_point(dt, aes(x = lon, y = lat), color = "red")
#> Error: `mapping` must be created by `aes()`

## End(Not run)

# If you want to use native ggplot commands, you need to transform your data
# to the projection used by the map:
dt <- transform_coord(dt, bind = TRUE)

basemap(data = dt) +
  geom_point(data = dt, aes(x = lon.proj, y = lat.proj), color = "red")

# The limits argument of length 4 plots a map anywhere in the world:
basemap(limits = c(100, 160, -20, 30), bathymetry = TRUE)

# The limits are further expanded when using the data argument:

dt <- data.frame(lon = c(-160, 160, 160, -160), lat = c(80, 80, 60, 60))

if(requireNamespace("ggspatial", quietly = TRUE)) {
  basemap(data = dt) +
    ggspatial::geom_spatial_polygon(data = dt, aes(x = lon, y = lat),
      fill = NA, color = "red")

# Rotate:
basemap(data = dt, rotate = TRUE) +
  ggspatial::geom_spatial_polygon(data = dt, aes(x = lon, y = lat),
    fill = NA, color = "red")
}

# Alternative:
basemap(data = dt, rotate = TRUE) +
  geom_polygon(data = transform_coord(dt, rotate = TRUE),
    aes(x = lon, y = lat), fill = NA, color = "red")

## To find UTM coordinates to limit a polar map:
basemap(limits = 60, projection.grid = TRUE)

## Not run:
# (Arctic shapes require a download in 2.0)
basemap(limits = c(2.5e4, -2.5e6, 2e6, -2.5e5), shapefiles = "Arctic")

# Using custom shapefiles (requires download):

```



```

data(bs_shapes, package = "ggOceanMapsData")
basemap(shapefiles = list(land = bs_land))#'

# Premade shapefiles from ggOceanMapsLargeData (requires download):
basemap("BarentsSea", bathymetry = TRUE)

## End(Not run)

# grid.col = NA removes grid lines, rotate = TRUE rotates northwards:
basemap(limits = c(-180, -140, 50, 70), grid.col = NA, rotate = TRUE)

# Rename axis labels

basemap(limits = c(-140, -105, 20, 40), bathymetry = TRUE) + xlab("Lat")

# Remove axis labels

basemap(limits = c(0, 60, 68, 80)) + labs(x = NULL, y = NULL)

basemap(limits = c(0, 60, 68, 80), rotate = TRUE) +
  theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank()
  )

```

dd_land

*Decimal degree land shapes***Description**

Decimal degree land shapes

Usage

dd_land

Format

[Simple feature collection](#) land shapes in decimal degrees (EPSG:4326). Obtained from Natural Earth Data (10m vectors). Includes the islands dataset.

Source

[Natural Earth Data](#)

See Also

Other mapfiles: [dd_rbathy](#)

dd_rbathy	<i>Decimal degree bathymetry</i>
-----------	----------------------------------

Description

Decimal degree bathymetry

Usage

dd_rbathy

Format

[Raster](#) bathymetry in decimal degrees (EPSG:4326). Downsampled from ETOPO 60 arc-second grid.

Source

NOAA National Centers for Environmental Information. 2022: ETOPO 2022 15 Arc-Second Global Relief Model. NOAA National Centers for Environmental Information. [doi:10.25921/fd45-gt74](https://doi.org/10.25921/fd45-gt74)

See Also

Other mapfiles: [dd_land](#)

dist2land	<i>Calculate distance to the closest land for coordinates</i>
-----------	---

Description

Calculates the closest distance to land for coordinates in a data frame

Usage

```
dist2land(  
  data,  
  lon = NULL,  
  lat = NULL,  
  shapefile = "DecimalDegree",  
  proj.in = 4326,  
  bind = TRUE,  
  dist.col = "ldist",  
  binary = FALSE,  
  verbose = TRUE  
)
```

Arguments

<code>data</code>	Data frame or sf object containing geographic coordinates.
<code>lon, lat</code>	Either the names of the longitude and latitude columns in data or NULL to guess the longitude and/or latitude columns in data.
<code>shapefile</code>	Land shape to which distances should be calculated. Either a character argument referring to a name of pre-made shapefiles in shapefile_list , a single sf or sp polygons object or NULL to enable automatic definition of the land shapes based on data. Set to "DecimalDegree" by default which enables great circle distances using s2 features assuming a spherical Earth (as a contrast to earlier versions of the function which used flat Earth).
<code>proj.in</code>	coordinate reference system of data.
<code>bind</code>	Logical indicating whether <code>x</code> should be returned with the distances (TRUE, default) or should the distances be returned as vector (FALSE).
<code>dist.col</code>	The name of the distance column, if <code>bind = TRUE</code> . Defaults to "ldist".
<code>binary</code>	Logical indicating whether binary (TRUE = the position is in the ocean, FALSE = the position is on land) should be returned instead of distances. Speeds up the function considerably.
<code>verbose</code>	Logical indicating whether information about the process should be returned as messages. Set to FALSE to make the function silent.

Details

The function calculates great circle spherical distances using the [st_distance](#) function by default. The function can be slow for large datasets. If you only want to use the function to remove (wrong) observations reported on land, set the `binary` argument to TRUE. This speeds up the calculations by a factor of ten.

Value

Returns a vector if `bind = FALSE`, otherwise a data frame. The distances are given in a new column defined by the `dist.col` argument. The distances are **kilometers** if `binary = FALSE`, otherwise logical (TRUE = the position is in the ocean, FALSE = the position is on land).

Author(s)

Mikko Vihtakari

Examples

```
# Simple example:
dt <- data.frame(lon = seq(-20, 80, length.out = 41), lat = 50:90)
dt <- dist2land(dt, verbose = FALSE)

qmap(dt, color = ldist) + scale_color_viridis_c()

# Datasets covering the entire Earth seem to work now, except 0,0 lon/lat point
lon = deg_to_dd(seq(0,360,30)); lat = c(80,50,20,0,-20,-50,-80)
```

```

dt <- data.frame(
  lon = rep(lon, length(lat)), lat = rep(lat, each = length(lon)))

qmap(dist2land(dt, verbose = FALSE), color = ldist) +
  scale_color_viridis_c()

## Not run:
dt <- data.frame(
  lon = deg_to_dd(seq(0,360,length.out = 1e3)),
  lat = rep(60, 1000))

# The distance calculation is slow for large datasets
system.time(dist2land(dt))
# user system elapsed
# 12.677 0.146 12.849

# binary = TRUE speeds the function up
system.time(dist2land(dt, binary = TRUE))
# user system elapsed
# 1.239 0.120 1.369

## End(Not run)

```

fdir_main_areas

Major fisheries areas (hovedomraade) of Norway

Description

Major fisheries areas (hovedomraade) of Norway

Usage

```
fdir_main_areas
```

Format

[sf object](#) containing major fishing zones defined by the Norwegian Directorate of Fisheries. Contains also Northwest Atlantic Fisheries Organization's divisions where Norwegian vessels tend to fish.

Source

Norwegian Directorate of Fisheries and Northwest Atlantic Fisheries Organization

See Also

Other datasets: [fdir_sub_areas](#), [ices_areas](#)

Examples

```
if(requireNamespace("ggspatial")) {  
  
  basemap(fdir_main_areas) +  
  ggspatial::annotation_spatial(fdir_main_areas, fill = NA)  
  
}
```

fdir_sub_areas	<i>Norwegian sub-areas (lokasjon) for commercial fishing</i>
----------------	--

Description

Norwegian sub-areas (lokasjon) for commercial fishing

Usage

```
fdir_sub_areas
```

Format

[sf object](#) containing major fishing zones defined by the Norwegian Directorate of Fisheries.

Source

[Norwegian Directorate of Fisheries](#)

See Also

Other datasets: [fdir_main_areas](#), [ices_areas](#)

Examples

```
if(requireNamespace("ggspatial")) {  
  
  basemap(fdir_sub_areas) +  
  ggspatial::annotation_spatial(fdir_sub_areas, fill = NA)  
  
}
```

geonorge_bathymetry	<i>Open Geonorge bathymetry shapefiles</i>
---------------------	--

Description

Opens and formats Geonorge bathymetry shapefiles ready for plotting in ggOceanMaps

Usage

```
geonorge_bathymetry(filepath, layer = NULL, verbose = FALSE)
```

Arguments

filepath	Character string defining the path to the .gml file. Must contain the file extension.
layer	Character string defining the layer containing depth information. If NULL assumed to be "dybdeareal".
verbose	Logical indicating whether information the reading process should be returned.

Details

You can download the bathymetry polygon shapefiles from [Geonorge](#). Download the file in GLM format.

Value

An [sf](#) object containing the depth polygons. Uses same projection than bathy (see [CRS](#)).

Author(s)

Mikko Vihtakari

See Also

Other create shapefiles: [clip_shapefile\(\)](#), [raster_bathymetry\(\)](#), [vector_bathymetry\(\)](#)

get_depth

*Extract depth for coordinates from a raster bathymetry dataset***Description**

Extracts depth from [basemap](#) bathymetry raster dataset for coordinates in a data frame

Usage

```
get_depth(
  data,
  bathy.style = "raster_continuous",
  lon = NULL,
  lat = NULL,
  shapefile = "DecimalDegree",
  proj.in = 4326,
  bind = TRUE,
  depth.col = "depth",
  verbose = FALSE
)
```

Arguments

data	Data frame or sf object containing geographic coordinates.
bathy.style	Character defining the basemap bathymetry raster which should be used for the depth extraction. Valid alternatives: "raster_binned" (or "rb"), "raster_continuous" (or "rc"; default), or "raster_user" (or "ru").
lon, lat	Either the names of the longitude and latitude columns in data or NULL to guess the longitude and/or latitude columns in data.
shapefile	Land shape to which distances should be calculated. Either a character argument referring to a name of pre-made shapefiles in shapefile_list , a single sf or sp polygons object or NULL to enable automatic definition of the land shapes based on data. Set to "DecimalDegree" by default which enables great circle distances using s2 features assuming a spherical Earth (as a contrast to earlier versions of the function which used flat Earth).
proj.in	coordinate reference system of data.
bind	Logical indicating whether x should be returned with the distances (TRUE, default) or should the distances be returned as vector (FALSE).
depth.col	The name of the depth column, if bind = TRUE. Defaults to "depth".
verbose	Logical indicating whether information about the process should be returned as messages. Set to FALSE to make the function silent.

Details

Uses the [st_extract](#) function to extract values from [basemap](#) bathymetry raster grids. Does not work for vector bathymetries.

Value

Returns a vector if `bind = FALSE`, otherwise a data frame. The depths are given in a new column defined by the `dist.col` argument. The distances are **kilometers**. NA distance means that the position is on land.

Author(s)

Mikko Vihtakari

Examples

```
## Not run:
dt <- data.frame(lon = seq(-20, 80, length.out = 41), lat = 50:90)
dt <- get_depth(dt)
qmap(dt, color = depth) + scale_color_viridis_c()

## End(Not run)
```

ices_areas	<i>ICES Advisory Areas</i>
------------	----------------------------

Description

ICES Advisory Areas

Usage

```
ices_areas
```

Format

[sf object](#) containing ICES Advisory Areas.

Source

[International Council for the Exploration of the Sea](#)

See Also

Other datasets: [fdir_main_areas](#), [fdir_sub_areas](#)

Examples

```
if(requireNamespace("ggspatial")) {

  basemap(ices_areas) +
  ggspatial::annotation_spatial(ices_areas, fill = NA)

}
```

qmap*Quick map*

Description

qmap is a shortcut similar to ggplot2's [qplot](#) designed to quickly plot data with a limited range of options.

Usage

```
qmap(  
  data,  
  ...,  
  x = NULL,  
  y = NULL,  
  geom = "point",  
  limits = NULL,  
  shapefiles = NULL,  
  crs = NULL,  
  bathymetry = FALSE,  
  glaciers = FALSE,  
  rotate = FALSE,  
  legends = TRUE,  
  legend.position = "right",  
  lon.interval = NULL,  
  lat.interval = NULL,  
  bathy.style = NULL,  
  downsample = 0,  
  bathy.border.col = NA,  
  bathy.size = 0.1,  
  bathy.alpha = 1,  
  land.col = "grey60",  
  land.border.col = "black",  
  land.size = 0.1,  
  gla.col = "grey95",  
  gla.border.col = "black",  
  gla.size = 0.1,  
  grid.col = "grey70",  
  grid.size = 0.1,  
  base_size = 11,  
  projection.grid = FALSE,  
  expand.factor = 1.1,  
  verbose = FALSE  
)
```

Arguments

data	Data frame to use.
------	--------------------

x, y, ...	Aesthetics passed into each layer. Longitude and latitude columns are automatically recognized using the guess_coordinate_columns function.
geom	Character argument specifying geom(s) to draw. Defaults to "point". Other alternatives are "text" and "label". The "text" option can also be triggered by simply mapping a variable to label (see Examples).
limits	Map limits. One of the following: • numeric vector of length 4: The first element defines the start longitude, the second element the end longitude (counter-clockwise), the third element the minimum latitude, and the fourth element the maximum latitude of the bounding box. Also accepts sf::st_bbox type named vectors with limits in any order. The coordinates can be given as decimal degrees or coordinate units for shapefiles used by a projected map. Produces a rectangular map. Latitude limits not given in min-max order are automatically ordered to respect this requirement. • single integer between 30 and 88 or -88 and -30 produces a polar map for the Arctic or Antarctic, respectively. Can be omitted if data or shapefiles are defined.
shapefiles	Either a list containing shapefile information or a character argument referring to a name of pre-made shapefiles in shapefile_list . This name is partially matched. Can be omitted if limits or data is defined as decimal degrees.
crs	Coordinate reference system (CRS) for the map. If NULL (default), the CRS is selected automatically based on limits, data, or shapefiles. Passed to st_crs . Typically integers giving the EPSG code are the easiest. Cannot be used simultaneously with rotate.
bathymetry	Logical indicating whether bathymetry should be added to the map. Functions together with bathy.style. See Details.
glaciers	Logical indicating whether glaciers and ice sheets should be added to the map.
rotate	Logical indicating whether the projected maps should be rotated to point towards the pole relative to the mid-longitude limit.
legends	Logical indicating whether the legend for bathymetry should be shown.
legend.position	The position for ggplot2 legend. See the argument with the same name in theme .
lon.interval, lat.interval	Numeric value specifying the interval of longitude and latitude grids. NULL finds reasonable defaults depending on limits.
bathy.style	Character (plots bathymetry; list of alternatives in Details) or NULL ("raster_binned_blues" if bathymetry = TRUE) defining the bathymetry style. Partially matched, can be abbreviated, and used to control bathymetry plotting together with bathymetry. See Details.
downsample	Integer defining the downsampling rate for raster bathymetries. A value of 0 (default) does not downsample, 1 skips every second row, 2 every second and third. See geom_stars
bathy.alpha	Transparency parameter for the bathymetry fill color. See scale_alpha .

<code>land.col, gla.col, grid.col</code>	Character code specifying the color of land, glaciers, and grid lines, respectively. Use NA to remove the grid lines.
<code>land.border.col, gla.border.col, bathy.border.col</code>	Character code specifying the color of the border line for land, glacier, and bathymetry shapes.
<code>land.size, gla.size, bathy.size, grid.size</code>	Numeric value specifying the width of the border line land, glacier and bathymetry shapes as well as the grid lines, respectively. Use the LS function for a specific width in pt. See Details.
<code>base_size</code>	Base size parameter for ggplot. See ggtheme .
<code>projection.grid</code>	Logical indicating whether the coordinate grid should show projected coordinates instead of decimal degree values. Useful to define limits for large maps in polar regions.
<code>expand.factor</code>	Expansion factor for map limits. Can be used to zoom in (decrease the value under 1) and out (increase the value over 1) automatically (data) limited maps. Defaults to 1, which means that outermost data points are located at the boundaries of the plotting region.
<code>verbose</code>	Logical indicating whether information about the projection and guessed column names should be returned as messages. Set to FALSE to make the function silent.

Value

Returns a [ggplot](#) map, which can be assigned to an object and modified as any ggplot object.

Author(s)

Mikko Vihtakari

See Also

Other basemap functions: [basemap\(\)](#), [shapefile_list\(\)](#), [transform_coord\(\)](#)

Examples

```
dt <- data.frame(lon = c(-100, -80, -60), lat = c(10, 25, 40), var = c("a", "a", "b"))

# Quickly see position of data
qmap(dt)

# Set color
qmap(dt, color = I("blue"))

# Map color to a variable
qmap(dt, color = var)
```

```
# Map text to a variable
qmap(dt, label = var)

# All basemap arguments work in qmap()
dt <- data.frame(lon = c(-80, -80, -50, -50), lat = c(65, 80, 80, 65))
qmap(dt, rotate = TRUE)
```

raster_bathymetry	<i>Simplify a bathymetry raster ready for vectorization</i>
-------------------	---

Description

Simplifies bathymetry raster ready for the [vector_bathymetry](#) function. Warning: processing may take a long time if the bathymetry raster is large.

Usage

```
raster_bathymetry(
  bathy,
  depths,
  proj.out = NULL,
  proj.bathy = NULL,
  boundary = NULL,
  warp = FALSE,
  estimate.land = FALSE,
  downsample = NULL,
  verbose = TRUE
)
```

Arguments

bathy	A stars object or a string giving the path to a bathymetry NetCDF or grd file
depths	Numeric vector giving the cut points for depth contours (see cut). If NULL, no depth aggregation will be made. This option is suitable for raster bathymetries passed directly to basemap.
proj.out	A character string specifying the coordinate reference system (CRS) argument for the output. See st_crs and proj.org . If NULL, the projection is retrieved from bathy and the output will not be reprojected saving processing time (since proj.out and proj.bathy would match).
proj.bathy	A character string specifying the CRS for the input (bathy). Only required if bathy lacks CRS information. If NULL, "EPSG:4326" is assumed.
boundary	A st_polygon object, text string defining the file path to a spatial polygon, bounding box , or a numeric vector of length 4 giving the boundaries for which bathy should be cut to. Should be given as decimal degrees . If unnamed numeric vector, the first element defines the minimum longitude, the second element the maximum longitude, the third element the minimum latitude and the

	fourth element the maximum latitude of the bounding box. You can also use the sf bounding box format as named vector. Use NULL not to cut bathy.
warp	Logical indicating whether the resulting grid should be resampled to a new CRS if <code>proj.out != proj.bathy</code> using the <code>st_warp</code> function. A time-consuming operation, but necessary when CRS changes in raster bathymetries. Not required if the next step is to vectorise the bathymetry.
estimate.land	Logical indicating whether to include land to the output. Can be used in the following <code>vector_bathymetry</code> step to estimate land polygons.
downsample	An integer defining how many rows in bathy should be skipped to reduce the size (and resolution). 1 skips every second row, 2 every second and third. See <code>st_downsample</code> . Set to NULL (default) to skip downsampling.
verbose	Logical indicating whether information about progress and guessed projection should be returned. Set to FALSE to make the function silent.

Details

You can use **GEBCO**, **IBCAO**, **ETOPO** bathymetry grids downloaded from respective sources as the bathy argument. The bathymetry grids read from files must be in any format read by `read_stars`. Alternatively use the `marmap::getNOAA.bathy` function to download ETOPO1 bathymetry and convert it to a raster object using the `marmap::as.raster` function.

Note that the size of the output is heavily influenced by the number of depth contours (depths) as well as the resolution of bathy and choice of `downsample`. To make the `vector_bathymetry` function and consequent plotting faster, limiting the details of the bathymetry raster may be desirable.

Value

A list with a `stars` object the containing projected bathymetry defined by the `proj.out` argument and a data frame of depth intervals.

Author(s)

Mikko Vihtakari

References

GEBCO Compilation Group (2019) GEBCO 2019 15-arcsecond grid (doi:10.5285/836f016a-33be-6ddc-e053-6c86abc0788e). URL: https://www.gebco.net/data_and_products/gridded_bathymetry_data/gebco_2019/gebco_2019_info.html. NOAA National Centers for Environmental Information. 2022: ETOPO 2022 15 Arc-Second Global Relief Model. NOAA National Centers for Environmental Information. doi:10.25921/fd45gt74.

See Also

Other create shapefiles: `clip_shapefile()`, `geonorge_bathymetry()`, `vector_bathymetry()`

reorder_layers	<i>Move basemap land, glacier and grid layers on top of other ggplot layers</i>
----------------	---

Description

Moves existing land, glacier and grid layers on top of other layers. Useful for hiding region polygons under land.

Usage

```
reorder_layers(p)
```

Arguments

p ggplot object from the [basemap](#) function.

Details

This function has not been tested properly yet and is likely to contain bugs.

Value

Returns a ggplot object with land, glacier and grid layers on top.

Author(s)

Mikko Vihtakari

See Also

Other customize shapefiles: [auto_limits\(\)](#), [theme_map\(\)](#)

Examples

```
if(requireNamespace("ggspatial", quietly = TRUE)) {  
  
  data("ices_areas")  
  p <- basemap(c(-20, 15, 50, 70)) +  
    ggspatial::annotation_spatial(ices_areas, aes(fill = Area_Full), show.legend = FALSE)  
  
  # Polygons on top of land  
  p  
  
  # Move land on top  
  reorder_layers(p)  
  
}
```

shapefile_list	<i>A list of pre-made shapefiles for basemap</i>
----------------	--

Description

Lists available pre-made shapefiles for plotting in the [basemap](#) function. Gives also instructions how to make custom ones.

Usage

```
shapefile_list(name, get.data = FALSE)
```

Arguments

name	A character argument giving the name of a pre-made shapefile. Will be partially matched. Use "all" to list all available ones.
get.data	Logical indicating whether spatial data should be returned instead of names of spatial data objects.

Details

Custom shapefiles for [basemap](#) should be defined as lists with (at least) following names (everything should be provided as characters):

- **land** Name of the object containing land polygons. Required.
- **glacier** Name of the object containing glacier polygons. Use NULL if glaciers are not needed.
- **bathy** Name of the object containing land polygons. Use NULL if bathymetry is not needed.

All linked spatial data objects must be in same projection. High-resolution pre-made data are still under development and may not be available. Pre-made shapefiles contain additional elements that are used in the [basemap](#) function, but not required for custom shapefile datasets.

Value

Returns a data frame of provided pre-made shapefiles, if name = "all". Returns a shapefile list containing the information for a particular map otherwise.

Author(s)

Mikko Vihtakari

See Also

Other basemap functions: [basemap\(\)](#), [qmap\(\)](#), [transform_coord\(\)](#)

Examples

```
shapefile_list("all")
shapefile_list("Arctic") # partial matching
```

theme_map	<i>A ggplot2 theme for maps</i>
-----------	---------------------------------

Description

A ggplot2 theme for maps.

Usage

```
theme_map(..., grid.col, grid.size)
```

Arguments

...	additional arguments passed to ggtheme .
grid.col	Character code specifying the color of grid lines. Use NA to remove the grid lines.
grid.size	Numeric value specifying the width of grid lines.

Value

A ggplot2 theme layer.

See Also

Other customize shapefiles: [auto_limits\(\)](#), [reorder_layers\(\)](#)

transform_coord	<i>Transform spatial coordinates to another projection</i>
-----------------	--

Description

Transforms spatial coordinates from original projection (decimal degrees assumed) to another projection.

Usage

```
transform_coord(  
  x = NULL,  
  lon = NULL,  
  lat = NULL,  
  new.names = "auto",  
  rotate = FALSE,  
  proj.in = 4326,  
  proj.out = NULL,  
  verbose = FALSE,  
  bind = FALSE,  
  na = "ignore"  
)
```


Arguments

x	Data frame to be transformed. Can be omitted if numeric vectors are assigned to lon and lat.
lon, lat	Either a name of the longitude and latitude columns in x or a numeric vector containing longitude and latitude coordinates. Use NULL to guess the longitude and/or latitude columns in x.
new.names	Character vector of length 2 specifying the names of transformed longitude and latitude columns, respectively. Alternatively NULL, which returns column names from x or "auto", which uses NULL if bind = FALSE and c("lon.proj", "lat.proj") if bind = TRUE.
rotate	Logical indicating whether the projected maps should be rotated to point towards the pole relative to the mid-longitude limit.
proj.in	The original CRS . If NULL, the projection is taken from x. x must be a spatial object in that case.
proj.out	Character. Either NULL, CRS the coordinates should be transformed to or a name of shapefiles in shapefile_list . If NULL, the output projection will be automatically determined from data. This option requires decimal degrees as input option.
verbose	Logical indicating whether information about the projection should be returned as message. Set to FALSE to make the function silent.
bind	logical. Should only transformed coordinates be returned (FALSE, default) or should x be returned with transformed coordinates (TRUE)?
na	character specifying the NA action for missing coordinates. The "ignore" option ignores the coordinates and returns NAs to transformed coordinates. The "remove" option removes missing values from x returning a message while doing it. Any other character argument will trigger na.fail stopping the function in case of missing coordinates.

Details

If x is specified, the function guesses longitude and latitude columns from x by default.

Value

Returns a data frame with transformed spatial coordinates.

Author(s)

Mikko Vihtakari

See Also

Other basemap functions: [basemap\(\)](#), [qmap\(\)](#), [shapefile_list\(\)](#)

Examples

```
# Coordinates are automatically transformed to the pre-made shapefile
# projections:
x <- data.frame(lon = c(-150, 150), lat = c(60, 90))
transform_coord(x)
transform_coord(x, bind = TRUE)

x <- data.frame(lon = c(-150, 150), lat = c(20, 50))
transform_coord(x, bind = TRUE) # no transformation required.
```

vector_bathymetry	Create a polygon bathymetry from a raster bathymetry file
-------------------	---

Description

Vectorizes bathymetry rasters. Designed to be used for the output of [raster_bathymetry](#) function. Warning: processing may take a long time if the bathymetry raster is large.

Usage

```
vector_bathymetry(
  bathy,
  drop.crumbs = NULL,
  remove.holes = NULL,
  smooth = FALSE
)
```

Arguments

bathy	bathyRaster object from the raster_bathymetry function.
drop.crumbs	Single numeric value specifying a threshold (area in km2) for disconnected polygons which should be removed. Set to NULL to bypass the removal. Uses the drop_crumbs function.
remove.holes	Single numeric value specifying a threshold (area in km2) for holes which should be removed. Set to NULL to bypass the removal. Uses the fill_holes function. Currently VERY slow.
smooth	Logical indicating whether the pixelated contours should be smoothed. Uses the smooth_ksmooth function.

Details

The `drop.crumbs` and `remove.holes` arguments can be used to make the resulting object smaller in file size. The `smooth` argument can be used to remove the pixelated contours, but often increases file size. Note also that using this option will bias the contours with respect to real world.

Value

An [sf](#) object containing the depth polygons. Uses same projection than bathy (see [CRS](#)).

Author(s)

Mikko Vihtakari

See Also

Other create shapefiles: [clip_shapefile\(\)](#), [geonorge_bathymetry\(\)](#), [raster_bathymetry\(\)](#)

Index

- * **basemap functions**
 - basemap, [2](#)
 - qmap, [17](#)
 - shapefile_list, [23](#)
 - transform_coord, [24](#)
- * **create shapefiles**
 - geonorge_bathymetry, [14](#)
 - raster_bathymetry, [20](#)
 - vector_bathymetry, [26](#)
- * **customize shapefiles**
 - reorder_layers, [22](#)
 - theme_map, [24](#)
- * **datasets**
 - fdir_main_areas, [12](#)
 - fdir_sub_areas, [13](#)
 - ices_areas, [16](#)
- * **fishery**
 - fdir_main_areas, [12](#)
 - fdir_sub_areas, [13](#)
 - ices_areas, [16](#)
- * **mapfiles**
 - dd_land, [9](#)
 - dd_rbathy, [10](#)
- * **maps**
 - dd_land, [9](#)
 - dd_rbathy, [10](#)
- * **shapefiles**
 - dd_land, [9](#)
 - dd_rbathy, [10](#)
 - fdir_main_areas, [12](#)
 - fdir_sub_areas, [13](#)
 - ices_areas, [16](#)
 - shapefile_list, [23](#)
- auto_limits, [22, 24](#)
- basemap, [2, 15, 19, 22, 23, 25](#)
- bounding box, [20](#)
- clip_shapefile, [14, 21, 27](#)
- Coordinate reference system, [4, 18](#)
- coordinate reference system, [20](#)
- CRS, [14, 20, 25, 26](#)
- cut, [20](#)
- dd_land, [9, 10](#)
- dd_rbathy, [9, 10](#)
- define_shapefiles, [5](#)
- dist2land, [10](#)
- drop_crums, [26](#)
- fdir_main_areas, [12, 13, 16](#)
- fdir_sub_areas, [12, 13, 16](#)
- fill_holes, [26](#)
- FS, [7](#)
- geom_stars, [4, 18](#)
- geonorge_bathymetry, [14, 21, 27](#)
- get_depth, [15](#)
- ggplot, [7, 19](#)
- ggplot2, [5, 7](#)
- ggtheme, [4, 19, 24](#)
- guess the correct columns, [3](#)
- guess the longitude and/or latitude
 - columns, [11, 15, 25](#)
- guess_coordinate_columns, [18](#)
- ices_areas, [12, 13, 16](#)
- list containing shapefile information,
 - [3, 18](#)
- LS, [4, 7, 19](#)
- option, [5](#)
- qmap, [7, 17, 23, 25](#)
- qplot, [17](#)
- Raster, [10](#)
- raster, [6](#)
- raster_bathymetry, [14, 20, 26, 27](#)

read_stars, [5](#), [6](#), [21](#)
reorder_layers, [22](#), [24](#)

s2, [11](#), [15](#)
scale_alpha, [4](#), [18](#)
sf, [3](#), [5](#), [11](#), [14](#), [15](#), [26](#)
sf::st_bbox, [3](#), [18](#)
shapefile_list, [3](#), [5–7](#), [11](#), [15](#), [18](#), [19](#), [23](#), [25](#)
Simple feature collection, [9](#)
smooth_ksmooth, [26](#)
spatial, [25](#)
spatial polygon, [6](#)
st_crs, [4](#), [18](#), [20](#)
st_distance, [11](#)
st_downsample, [21](#)
st_extract, [15](#)
st_polygon, [20](#)
st_warp, [21](#)
stars, [5](#), [20](#), [21](#)

theme, [4](#), [18](#)
theme_map, [22](#), [24](#)
transform_coord, [7](#), [19](#), [23](#), [24](#)

vector_bathymetry, [14](#), [20](#), [21](#), [26](#)