Package 'shoredate'

January 23, 2023

Type Package

Title Shoreline Dating of Stone Age Sites on the Norwegian Skagerrak Coast

Version 1.0.0

Description Provides tools for shoreline dating Stone Age sites located on the Norwegian Skagerrak coast using methods presented in Roalkvam (2023) <doi:10.1016/j.quascirev.2022.107880>. Shoreline dating is based on the present-day elevation of a site, a reconstruction of past relative sea-level change, and empirically derived estimates of the likely elevation of the sites above sea-level when they were in use. The geographical and temporal coverage of the method thus follows from local geological reconstructions of shoreline displacement and the degree to which settlements in the region have been found to have been located on or close to the shoreline when they were in use. Methods for numerical treatment and visualisation of the dates are provided, along with basic tools for evaluating the location of sites within the region and corresponding variation in local shoreline displacement.

Language en-US

License GPL (>= 3)

URL `https://github.com/isakro/shoredate`

BugReports `https://github.com/isakro/shoredate/issues`

Encoding UTF-8

RoxygenNote 7.2.1

Imports ggplot2, ggrepel, ggridges, ggsn, grid, scales, sf, terra, utils

Suggests covr, elevatr, rgdal, knitr, rmarkdown, testthat (>= 3.0.0), vdiffr

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Isak Roalkvam [aut, cre] (<`https://orcid.org/0000-0001-6974-1374`>)

Maintainer Isak Roalkvam <isakroa@protonmail.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-01-23 09:50:02 UTC

R topics documented:

`check_target_location` *Check if a target site is set to the correct CRS and is located within the spatial limit*

Description

Checks if a target sites is set to the correct coordinate reference system WGS84 UTM32N (EPSG: 32632). Throws an error if this is not the case. Checks if the location intersects the study area for which the method was derived (see Roalkvam 2022). Returns a warning if this is not the case.

Usage

```
check_target_location(target)
```

Arguments

`target` A spatial object of class `sf`, representing a site location.

Value

An error if the site does not intersect the study area or is lacking or is set to the incorrect coordinate reference system.

Examples

```
# Create example point using the required coordinate system
# WGS84 UTM32N (EPSG: 32632)
target_point <- sf::st_sfc(sf::st_point(c(579570, 6582982)), crs = 32632)

# Check the CRS and the location of the site
check_target_location(target_point)
```

create_isobases *Create isobases*

Description

Function to create isobases for interpolating shoreline displacement curves. This is done from the centre points of the supplied displacement curves. Isobases can also be created for multiple directions, which is useful for testing the sensitivity of dates to the direction of the isobases.

Usage

```
create_isobases(isobase_direction)
```

Arguments

`isobase_direction`
A vector holding a single or multiple directions for the isobases.

Value

A simple feature holding the isobases represented as lines.

Examples

```
# Create isobases in a specified direction
isobases <- create_isobases(327)
plot(sf::st_geometry(isobases))

# Create isobases using different directions
isobases <- create_isobases(c(327, 338))

# Plot for visualisation
plot(sf::st_geometry(isobases))
```

`displacement_plot` *Plot shoreline displacement curves*

Description

Function for plotting shoreline displacement curves. Calling to plot without providing interpolated curves will display the four underlying curves.

Usage

```
displacement_plot(
  interpolated_curve = NA,
  displacement_curves = c("Horten", "Porsgrunn", "Tvedestrand", "Arendal"),
  greyscale = FALSE
)
```

Arguments

`interpolated_curve`
List holding one or more interpolated shoreline displacement curves.

`displacement_curves`
Character vector specifying which geologically informed displacement curves that should be plotted. Accepted values are `c("Horten", "Porsgrunn", "Tvedestrand", "Arendal")`. All are included by default.

`greyscale`
Logical value indicating whether the plot should include colours or not. Defaults to FALSE.

Value

A plot displaying the underlying shoreline displacement curves and, if provided, an interpolated curve.

Examples

```
# Empty plot for speed of the example
displacement_plot(displacement_curves = "")
```

interpolate_curve *Interpolate displacement curve using IDW*

Description

Interpolate the trajectory of past shoreline displacement to a single location based on the distance of the location to the shoreline isobases of the geologically derived displacement curves. This is done using inverse distance weighting.

Usage

```
interpolate_curve(
  target,
  isobases = NA,
  power = 2,
  cal_reso = 10,
  verbose = FALSE
)
```

Arguments

`target` A spatial target location to where the new displacement curve is interpolated.

`isobases` 4 spatial lines representing the shoreline isobases of the existing displacement curves. Multiple sets of 4 isobases with different isobase dirctions can be provided (see `create_isobases()`).

`power` A numerical value indicating the inverse distance power for IDW. Defaults to 2.

`cal_reso` Numeric value specifying the resolution to use on the calendar scale. Defaults to 10.

`verbose` Logical value indicating whether progress should be printed to console. Defaults to FALSE.

Value

A list holding an interpolated displacement curve for each isobase direction. Each displacement curve is represented by a data frame with the columns `bce` where negative values indicate years BCE and positive CE, `lowerelev`, representing the lower limit for the elevation of the shoreline for each year. `upperelev`, the upper limit for elevation of the shoreline for each year, and `direction` which indicates the direction of the isobases used when interpolating the curve.

Examples

```
# Create example point using the required coordinate system
# WGS84 / zone UTM32N (EPSG: 32632)
target_point <- sf::st_sfc(sf::st_point(c(579570, 6582982)), crs = 32632)

# Interpolate shoreline displacement curve to the target point location
# setting the resolution on the calendar scale to 1000 years for speed.
```

```
target_curve <- interpolate_curve(target_point, cal_reso = 2000)
```

print.shoreline_date *print.shoreline_date*

Description

Print the dates in a `shoreline_date` object. Each date is printed with site name, elevation and highest density region. If the isobase direction is different or more are provided than the default, the directions and dates associated with these are also printed.

Usage

```
## S3 method for class 'shoreline_date'
print(x, ...)
```

Arguments

`x`	Object of class `shoreline_date`.
`...`	Additional arguments.

Value

Print the site names, elevations, non-default isobase directions and HDRs contained in a `shoreline_date` object to console.

Examples

```
target_point <- sf::st_sfc(sf::st_point(c(538310, 6544255)), crs = 32632)

# Reduce date resolution with cal_reso and elevation_reso for speed.
target_date <- shoreline_date(site = target_point,
                              elevation = 70,
                              elev_reso = 1,
                              cal_reso = 400)
target_date
```

shoredate_hdr *Highest density region of shoreline dates*

Description

Function to find 95% highest density region (HDR) for a provided shoreline date. Negative values denote years BCE while positive values denote CE.

Usage

```
shoredate_hdr(bce, probability, site_name, cal_reso, prob = 0.95)
```

Arguments

`bce`	A vector holding calendar years associated with a date
`probability`	A vector holding the probability corresponding to each year in `date`.
`site_name`	A vector holding the name of the site that has been dated.
`cal_reso`	Resolution on the calendar scale used when dating the site.
`prob`	A numerical value indicating the probability coverage of the HDR. Defaults to 0.95.

Value

A list holding start and end points for segments of the highest density region of a shoreline date, the probability coverage and site name.

Examples

```
# Create point to shoreline date
target_point <- sf::st_sfc(sf::st_point(c(538310, 6544255)), crs = 32632)

# Reduce date resolution with cal_reso and elevation_reso for speed.
target_date <- shoreline_date(sites = target_point,
                              elevation = 80,
                              elev_reso = 1,
                              cal_reso = 400)

# `shoredate_hdr()` is already called under the hood with `shoreline_date()`,
# and is printed when calling the `shoreline_date object`
target_date

# However, `shoredate_hdr()` can be applied separately by pulling the
# necessary data from the date
(shoredate_hdr(target_date[[1]][[1]]$date$bce,
               target_date[[1]][[1]]$date$probability,
               target_date[[1]][[1]]$site_name,
               target_date[[1]][[1]]$cal_reso))
```

`shoredate_plot` *Plot shoreline date*

Description

Function for plotting shoreline dates along with associated metadata.

Usage

```
shoredate_plot(
  shorelinedates,
  elevation_distribution = TRUE,
  displacement_curve = TRUE,
  site_name = FALSE,
  parameters = FALSE,
  isobase_direction = FALSE,
  highest_density_region = TRUE,
  hdr_label = TRUE,
  multiplot = FALSE,
  greyscale = FALSE
)
```

Arguments

`shorelinedates` Object of class `shoreline_date`.

`elevation_distribution` Logical value indicating whether the gamma distribution should be displayed. Default is TRUE.

`displacement_curve` Logical value indicating whether the displacement curve should be displayed. Default is TRUE.

`site_name` Logical value indicating whether the name of the site should be printed. Defaults to FALSE.

`parameters` Logical value indicating whether the parameters of the gamma distribution should be printed. Default is FALSE.

`isobase_direction` Logical value indicating whether the direction of the isobases should be printed. Default is FALSE.

`highest_density_region` Logical value indicating whether the 95% highest density region should be displayed. Defaults to TRUE.

`hdr_label` Logical value indicating whether the numeric values for the highest density regions should be displayed. Default is TRUE.

`multiplot` Logical value indicating whether multiple dates should be plotted individually, or be collapsed into a single plot. The only other graphical option with `multiplot` set to TRUE is `highest_density_region`. Default is FALSE.

`greyscale` Logical value indicating whether the plot should be in greyscale or not. Defaults to FALSE.

Details

`shoredate_plot()` returns a plot displaying the provided shoreline dates. A single plot is created for each date, where a range of settings can be adjusted to display or hide various parameters and results. Setting the parameter `multiplot` to `TRUE` returns a sparser version for multiple dates, where the only option is whether or not to display the highest density region in addition to each date. `multiplot` does not allow for multiple isobase directions. Negative values denote years BCE while positive values denote CE.

Value

Plot(s) displaying shoreline dates and associated metadata.

Examples

```
# Create example point with correct coordinate reference system
target_point <- sf::st_sfc(sf::st_point(c(538310, 6544255)), crs = 32632)

# Reduce date resolution with cal_reso and elevation_reso for speed.
target_date <- shoreline_date(sites = target_point, elevation = 80,
                              elev_reso = 10,
                              cal_reso = 500)

shoredate_plot(target_date)
```

`shoredate_sumplot` *Plot the summed probability distribution of multiple shoreline dates*

Description

Function to plot the sum of the probabilities of multiple shoreline dates as resulting from running `sum_shoredates()`.

Usage

```
shoredate_sumplot(shoredates_sum, sample_size = TRUE)
```

Arguments

`shoredates_sum` Object of class `shoredates_sum`.

`sample_size` Logical indicating whether or not to display the number of summed dates on the plot. Defaults to TRUE.

Value

A line plot showing the provided summed probability distribution.

Examples

```
target_points <- sf::st_sfc(sf::st_point(c(538310, 6544255)),
                            sf::st_point(c(538300, 6544250)))
target_points <- sf::st_as_sf(target_points, crs = 32632)

# Shoreline date, reducing resolution on elevation and calendar scales for
# speed.
target_dates <- shoreline_date(target_points,
                               elevation = c(65, 70),
                               elev_reso = 10,
                               cal_reso = 750)

target_sum <- sum_shoredates(target_dates)

shoredate_sumplot(target_sum)
```

---

shoreline_date *Shoreline date*

Description

A function for shoreline dating Stone Age sites based on their present-day elevation, their likely elevation above sea-level when in use and the trajectory of past shoreline displacement on the Norwegian Skagerrak coast.

Usage

```
shoreline_date(
  sites,
  elevation = NA,
  elev_reso = 0.01,
  cal_reso = 10,
  isobase_direction = 327,
  sum_isobase_directions = FALSE,
  model_parameters = c(0.286, 0.048),
  elev_fun = "mean",
  interpolated_curve = NA,
  hdr_prob = 0.95,
  normalise = TRUE,
  sparse = FALSE,
  verbose = FALSE
)
```

Arguments

`sites`	Vector giving one or more site names, or, if displacement curves are to be interpolated, objects of class `sf` representing the sites to be dated. In the case of a spatial geometry, the first column is taken as site name.

`elevation` Vector of numeric elevation values for each site or a an elevation raster of class `SpatRaster` from the package `terra` from where the elevation values are to be derived.

`elev_reso` Numeric value specifying the resolution with which to step through the gamma distribution representing the distance between site and shoreline. Defaults to 0.01m.

`cal_reso` Numeric value specifying the resolution to use on the calendar scale. Defaults to 10.

`isobase_direction`
A vector of numeric values defining the direction(s) of the isobases. Defaults to 327.

`sum_isobase_directions`
Logical value indicating that if multiple isobase directions are specified in `isobase_direction` the results should be summed for each site using `sum_shoredates`. Defaults to FALSE.

`model_parameters`
Vector of two numeric values specifying the shape and scale of the gamma distribution. Defaults to c(0.286, 0.048), denoting the shape and scale, respectively.

`elev_fun` Statistic to define site elevation if this is to be derived from an elevation raster. Uses `terra::extract()`. Defaults to mean.

`interpolated_curve`
List holding shoreline displacement curve. `interpolate_curve()` will be run if this is not provided.

`hdr_prob` Numeric value specifying the coverage of the highest density region. Defaults to 0.95.

`normalise` Logical value specifying whether the shoreline date should be normalised to sum to unity. Defaults to TRUE.

`sparse` Logical value specifying if only site name and shoreline date should be returned. Defaults to FALSE. Note that sparse dates are only compatible with `sum_shoredates()`.

`verbose` Logical value indicating whether progress should be printed to console. Defaults to FALSE.

Value

A nested list of class `shoreline_date` holding the shoreline date results and associated metadata for each dated site for each isobase direction. The elements of each date is:

- `site_name` name of the site.
- `site_elev` elevation of the site.
- `date` data frame with the columns bce where negative values indicate years BCE and positive CE, as well as `probability`, which gives the probability mass for each year.
- `hdr_start` start values for the HDR ranges.
- `hdr_end` end values for the HDR ranges.
- `hdr_prob` probability level for the HDR.

- `dispcurve` data frame holding the displacement curve used for dating the site. This has the columns `bce`, giving years BCE/CE. `lowerelev`, the lower limit for the elevation of the shoreline for each year. `upperelev`, the upper limit for elevation of the shoreline for each year.
- `dispcurve_direction` direction of the isobases in use.
- `model_parameters` parameters for the gamma distribution. The first value gives the shape and the second value the scale of the distribution.
- `gammdat` data frame holding the gamma distribution. The column `offset` denotes the vertical distance (m) from the shoreline, as specified by the `elev_reso` argument. `px` is the cumulative probability at each step of `offset`, and `probs` is the probability of each step found by subtracting the preceding value from each value of `px`.
- `cal_reso` resolution on the calendar scale.

Examples

```
# Create example point using the required CRS WGS84 UTM32N (EPSG: 32632)
target_point <- sf::st_sfc(sf::st_point(c(538310, 6544255)), crs = 32632)

# Date target point, manually specifying the elevation instead of providing
# an elevation raster and setting the resolution on the calendar scale to
# 200 years and elevation scale to 1 for speed.
shoreline_date(sites = target_point,
               elevation = 80,
               elev_reso = 1,
               cal_reso = 400)
```

`sum_shoredates` *Sum shoreline dates*

Description

Function for finding the summed probability distribution of multiple shoreline dates.

Usage

```
sum_shoredates(
  shoreline_dates,
  cut_off = -2500,
  cut_off_level = 1,
  normalise = TRUE
)
```

Arguments

`shoreline_dates`
Object of class `shoreline_date`.

`cut_off` Calender year specifying where dates should be cut off. Defaults to 2500 BCE.

`cut_off_level` Numerical value between 0 and 1 indicating the probability mass that has to faller after the cut-off for a date to be excluded. Defaults to 1, retaining all dates.

`normalise` Logical value indicating whether the probability sum of the dates should be normalised to sum to unity. Defaults to TRUE.

Value

List of class `shoredate_sum` holding the elements:

- `sum` data frame with the columns `bce` where negative values indicate years BCE and positive CE, as well as `probability`, which gives the probability mass for each year.
- `dates_n` number of dates that make up the sum after applying any specified cut-off. One date per site per isobase direction.

Examples

```
target_points <- sf::st_sfc(sf::st_point(c(538310, 6544255)),
                            sf::st_point(c(538300, 6544250)))
target_points <- sf::st_as_sf(target_points, crs = 32632)

# Shoreline date, reducing resoltuion on elevation and calendar scales for
# speed.
target_dates <- shoreline_date(target_points,
                               elevation = c(65, 70),
                               elev_reso = 10,
                               cal_reso = 500)

sum_shoredates(target_dates)
```

`target_plot` *Target plot*

Description

Function to plot the centroids of one or more sites to be dated, relative to the shoreline isobases of the employed displacement curves. The basemap is a simplified representation of the coastline within the study area. Calling the function without providing a target displays the map with the isobases. This can be combined with `create_isobases()` to visualise isobases with a different direction than the default of 327.

Usage

```
target_plot(targets = NA, isobases = NA, greyscale = FALSE)
```

Arguments

`targets` Objects of class `sf` representing the sites to be dated. The first column beyond geom is taken as site name.

`isobases` Spatial lines as object of class `sf` representing the shoreline isobases. Defaults to isobases with a direction of 327, but create_isobases() can be used to create isobases with other directions that can then be passed to `target_plot()`.

`greyscale` Logical value indicating whether the plot should include colours or not. Defaults to FALSE.

Value

A map displaying the location of the shoreline isobases, and, if provided, the position of target locations represented as centroids.

Examples

```
# Display the background map and default isobases
target_plot()
```

Index