Package 'VulnToolkit'

July 21, 2025

Type Package

Title Analysis of Tidal Datasets

Version 1.1.4

Date 2021-07-30

BugReports https://github.com/troyhill/VulnToolKit/issues

Description Contains functions for analysis and summary of tidal datasets. Also provides access to tidal data collected by the National Oceanic and Atmospheric Administration's Center for Operational Oceanographic Products and Services and the Permanent Service for Mean Sea Level. For detailed description and application examples, see Hill, T.D. and S.C. Anisfeld (2021) <doi:10.6084/m9.figshare.14161202.v1> and Hill, T.D. and S.C. Anisfeld (2015) <doi:10.1016/j.ecss.2015.06.004>.

License GPL-3

URL https://github.com/troyhill/VulnToolkit

LazyData true

Imports XML, plyr

Depends R (>= 3.5.0)

Suggests covr, testthat, knitr, rmarkdown

RoxygenNote 7.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Troy Hill [aut, cre] (ORCID: <https://orcid.org/0000-0003-2980-4099>), Shimon Anisfeld [ctb]

Maintainer Troy Hill <Hill.Troy@gmail.com>

Repository CRAN

Date/Publication 2021-08-02 13:10:02 UTC

Contents

`dur.bias` *Calculates bias in flooding duration estimates due to logger deployment time*

Description

Seasonality in water levels imparts bias to flooding duration datasets built from short-term water level data. `dur.bias` calculates and presents these biases in visual and tabular form.

Using data from a nearby, user-defined NOAA station, bias is estimated by comparing the relationship between flooding duration and elevation in datasets covering two time periods: (1) the time periodof water level logger deployment, and (2) the reference time period (some number of years).

Usage

```
dur.bias(
  data,
  dateCol = 1,
  station = 8518750,
  ref.period = c("20120101", "20121231"),
  time = "GMT"
)
```

Arguments

data	water level dataset. must have a time stamp column named 'datetime' with the first ten digits following the format YYYY-MM-DD.Different separators are fine, but the placement and number of digits for year, month, and day are critical. Date range must be continuous, or the "estimated" curve won't be accurate (datasets with gaps won't work well)
dateCol	index number of the column in 'data' containing dates. Must be a date/POSIX datatype.
station	name or number of NOAA station to be used for reference data. Ideally, use the station closest to where your water level data was collected. This is fed to noaa so must be compatible with noaa's station argument.
ref.period	downloads NOAA tide data to set reference flooding-elevation relationship. This is used to evaluate accuracy of flooding duration estimates determined by water level deployment. Use of whole years is recommended. dates must be entered in format YYYYMMDD, in the form (c(start_date, end_date)).
time	time zone to download reference data (GMT is default; the NOAA website does not have all data available in all time zones.)

Details

Seasonality in water levels imparts bias to flooding duration datasets built from short-term water level data. dur.bias calculates and presents these biases in visual and tabular form.

Value

(1) A summary of root mean square errors is printed, (2) a data frame of elevations, flooding duration estimates, and their associated bias estimates is saved as output, and (3) two plots are made, graphically showing the relationship between predicted and reference flooding duration curves.

See Also

noaa,fld.dur

Examples

```
## Not run:
NL_extract <- NL_6min_2013[NL_6min_2013[, 1] < "2013-02-01", ]
dur.bias(data = NL_extract, station = 8518750, ref.period = c("20130101", "20131231"))

## End(Not run)
```

`dur.events` *Calculates the duration of the median (or other percentile) flooding event*

Description

Calculates the duration of the median (or other percentile) flooding event

Usage

```
dur.events(level, elevation, percentile = 0.5, units = "1 observation")
```

Arguments

`level`	a numeric vector of water levels or tide data
`elevation`	elevation(s) of interest (e.g., marsh platform, MHW). A vector of elevations is accepted. Elevation should be in the same vertical datum as `level`
`percentile`	the percentile(s) to find (median flooding event is used by default)
`units`	the time interval between measurements. This argument enables conversion of flooding durations to hours. `units` must be a string of the form `XX units`, where `XX` is an integer and `units` can be `second`, `minute`, `hour`, or their plurals. When time units are specified, all time units are converted to hours in output. When units are not specified, the output reports the number of sequential observations

Details

The duration of individual flooding events is calculated from water level data. `dur.events()` differs from `fld.dur()` in that the former examines individual flooding events, rather than cumulative inundation times.

Value

the duration of the median (or other percentile) flooding event occurring at a given elevation. If `percentile` is of length one, the value(s) returned are the corresponding flooding duration percentiles for the elevation(s) of interest. To ensure clarity when more than one `percentile` is sought, in those cases a dataframe is output to report elevations and flooding durations

Examples

```
data(NL_6min_2013)
MHW <- 0.9117  # New London MHW in 2013: 0.9117 m relative to MLLW

# median flooding duration at three elevations
dur.events(NL_6min_2013[, 2], elevation = c(0.5, MHW, 1.5), units = "6 minutes")

# a dataframe is output when results are more complex
dur.events(NL_6min_2013[, 2], elevation = MHW,
```

```
           percentile = c(0.1, 0.5, 0.9), units = "6 minutes")
dur.events(NL_6min_2013[, 2], elevation = c(0.5, MHW, 1.5),
  percentile = c(0.1, 0.5, 0.9), units = "6 minutes")

# dur.events() differs from fld.dur() in that it examines individual
# flooding events, rather than cumulative inundation

# The median flooding event at MHW lasts 2.9 hrs
dur.events(NL_6min_2013[, 2], elevation = MHW, units = "6 minutes")

# And over an entire year, MHW was flooded 13% of the time
fld.dur(z = MHW, NL_6min_2013[, 2])
```

`fld.depth` *Calculates flooding depth above an elevation of interest*

Description

Flooding depths are calculated from water level or tide data. If water levels are used, the median (or other percentile) flooding depth is calculated based on all observations of flooded conditions (when water depth is equal to or greater than the elevation of interest). If a high/low tide dataset is used, flooding depth percentiles will be just for high tides that flood the selected elevation. The latter case includes only peak high water levels, and so will yield greater flooding depths for the same elevation.

Using data from a nearby, user-defined NOAA station, bias is estimated by comparing the relationship between flooding duration and elevation in datasets covering two time periods: (1) the time periodof water level logger deployment, and (2) the reference time period (some number of years).

Usage

```
fld.depth(level, elevation, percentile = 0.5)
```

Arguments

`level`	a numeric vector of water levels or tide data
`elevation`	elevation(s) of interest (marsh platform, MHW). A vector of elevations is accepted. Elevation should be in the same vertical datum as `level`
`percentile`	the percentile(s) to calculate (median flooding depth is calculated by default). A vector of percentiles is accepted.

Value

if 'percentile' argument is of length one, the value(s) returned are the corresponding flooding percentiles for the elevation(s) of interest. To ensure clarity when more than one 'percentiles' are sought, in those cases a dataframe is output to report elevations and flooding depths.

See Also

noaa, fld.dur

Examples

```
data(NL_6min_2013)
MHW <- 0.9117  # New London MHW in 2013: 0.9117 m relative to MLLW
fld.depth(level = NL_6min_2013[, 2], elevation = MHW,
         percentile = 0.5)

fld.depth(level = NL_6min_2013[, 2], elevation = MHW,
         percentile = c(0.25, 0.5, 0.75))

fld.depth(level = NL_6min_2013[, 2], elevation = c(0, MHW, 1.5),
         percentile = 0.75)

fld.depth(level = NL_6min_2013[, 2], elevation = c(0, MHW, 1.5),
         percentile = c(0.25, 0.5, 0.75))
```

---

fld.dur *Flooding duration*

Description

Calculates the percent of time an elevation is submerged

Usage

```
fld.dur(z, level)
```

Arguments

z	elevation of interest
level	a numeric vector of water levels. Time interval between measurements must be uniform.

Details

Calculates the percent of time an elevation is submerged

Value

value the decimal fraction of measurements in level that fall above z

See Also

fld.frq

Examples

```
data(NL_6min_2013)
a <- fld.dur(0.9117, NL_6min_2013[,2]); a # flooding duration at MHW
a * length(NL_6min_2013[,2]) / 10 # convert to hours per year

b <- fld.dur(0, NL_6min_2013[,2]); b # flooding duration at MLLW
b * length(NL_6min_2013[,2]) / 10 # hours per year

elev.dur <- data.frame(elev = seq(from = -0.5, to = 1.25, by = 0.005))
elev.dur$dur <- fld.dur(elev.dur$elev, NL_6min_2013[,2]) * length(NL_6min_2013[,2]) / 10

plot(elev.dur$dur ~ elev.dur$elev, pch = 19,
ylab = "flooding duration (hours per year)", xlab = "elevation (m; MLLW)")
```

---

fld.frq *Flooding frequency*

Description

Calculates the frequency of high tides flooding an elevation of interest.

Usage

```
fld.frq(z, ht, units = "percent")
```

Arguments

z	elevation of interest
ht	a numeric vector of high tide levels
units	units for output. Default is `percent`; the proportion of tides flooding elevation `x`. Alternatively, `tides` will return the number of flooding tides in the dataset

Details

Calculates the frequency of high tides flooding an elevation of interest.

Value

value the number or percent of high tides in `vec` that fall above `z`

See Also

`fld.dur`

Examples

```
# get a dataset of high/low tides
HT.NL <- HL(level = NL_6min_2013[, 2], time = NL_6min_2013[, 1], tides = "H")
a <- fld.frq(z = 0.9117, ht = HT.NL[, 1], units = "tides") # number of flooding tides at MHW
a / length(HT.NL[, 1]) # flooding tides as a percentage of all tides in time period

fld.frq(0.9117, HT.NL[,1], units = "percent") # check

b <- as.numeric((NL_6min_2013[nrow(NL_6min_2013), 1] -
NL_6min_2013[1, 1])) / 365.242 # fraction of year covered by dataset

elev.frq <- data.frame(elev = seq(from = 0, to = 1.5, by = 0.005))

# error thrown if units = "tides" and length(x) > length(ht)
fld.frq(elev.frq$elev, HT.NL[, 1], units = "tides")

# a work-around
elev.frq$frq <- fld.frq(elev.frq$elev, HT.NL[, 1], units = "percent") *
length(HT.NL[, 1]) / b

plot(elev.frq$frq ~ elev.frq$elev, pch = 19,
ylab = "flooding frequency (tides per year)", xlab = "elevation (m; MLLW)")
```

`form.no` *Calculate tidal form number*

Description

Uses harmonic constituent data from the NOAA CO-OPS website to calculate tidal form numbers as the ratio of the sum of K1 and O1 diurnal harmonic constituent amplitudes to the sum of the M2 and S2 semidiurnal amplitudes. Requires an internet connection.

Usage

```
form.no(station)
```

Arguments

`station`	station ID number or vector of IDs, available on CO-OPS website (https://www.tidesandcurrents.noaa.gov/stations.html?type=Water+Levels).

Value

a dataframe of station number(s) and corresponding tidal form number(s).

Examples

```
## Not run:
a <- form.no()
stn.list <- c("8467150", "8461490", "9454240")
b <- form.no(stn.list)
b

## End(Not run)
```

---

harcon *Scrapes harmonic constituent data from NOAA CO-OPS website*

Description

Scrapes harmonic constituent data from NOAA CO-OPS website. Requires an internet connection.

Usage

```
harcon(station)
```

Arguments

station	station ID number or vector of IDs, available on CO-OPS website (https://www.tidesandcurrents.noaa.gov/stations.html?type=Water+Levels).

Value

a dataframe of harmonic constituents and their associated phases, amplitudes, and speeds.

Examples

```
## Not run:
bport.cons <- harcon(8467150) # Bridgeport, CT
bport.cons

## End(Not run)
```

HL *Extracts high and low tides from a record of water levels*

Description

Extracts high and low tides from a record of water levels. For reliable results, the time interval between samples must be constant throughout the dataset (e.g., 1 sample per hour, 1 sample per 6 minutes). Datasets with irregularly-spaced observations should be expanded to the smallest time interval to avoid issues.

Usage

```
HL(
  level,
  time,
  period = 13,
  phantom = TRUE,
  tides = "all",
  semidiurnal = TRUE,
  verbose = FALSE
)
```

Arguments

level	a numeric vector of water levels
time	a vector (numeric or POSIX*) indicating the time of water level measurements. Units must be minutes.
period	a single numeric or integer estimate of tidal period (full tidal cycle). Units must be hours.
phantom	a protective measure taken to prevent the inclusion of an artificial ('phantom') high or low tide at the end of the dataset, which occurs because the final observation is always a local maxima/minima for the period following the previous low/high tide. If the water level measurements end precisely at a low or high tide, this can be changed to FALSE.
tides	is used to optionally subset the output to include only high or low tides. This argument can be 'all' (default), 'H', or 'L'
semidiurnal	logical. If TRUE, higher- and lower- high/low tides are reported in a separate column called 'tide2'. Tides are grouped by day and assigned higher- or lower-status. When a single high or low tide exists in a day, tides on adjacent days are used to make an assignment.
verbose	logical. If TRUE, messages are returned to console identifying the presence of NAs in the input data

Value

a dataframe of tide levels, associated time stamps, and tide type ('H' or 'L'). If there are NAs present in the water level or time datasets, a message reports this information in the console but the NAs are not removed or otherwise acted upon. The column indicating semidiurnal tides ('tide2') may be NA if tides are identical or in cases where there is a single tide in the first day of the dataset.

See Also

HL.plot

Examples

```
HL.NL <- VulnToolkit::HL(level = NL_6min_2013[,2], time = NL_6min_2013[,1])
head(HL.NL)

## Not run:
### Important note: Problems arise when using irregularly-spaced data.
### The best solution is for the user to ensure their data are evenly-spaced.
### An example:
###
### pull data at 6-minute intervals
dat.hr   <- noaa(begindate = "20200101", enddate = "20200630", station = "8467150",
                 interval = "hourly")

### pull data at 1-hr intervals
dat.6min <- noaa(begindate = "20200701", enddate = "20201231", station = "8467150",
                 interval = "6 minute")

dat.combined <- rbind(dat.hr, dat.6min)
### if time interval is inconsistent the HL output can be unreliable
HL.plot(level = dat.combined[, 2], time = dat.combined[, 1])

### expand 1-hr dataset using the 6-minute time interval
timeSeq <- data.frame(time_GMT = seq.POSIXt(from = min(dat.hr$time_GMT),
                                            to = max(dat.hr$time_GMT),
                                            by = "6 min"))
### create consistently-spaced dataset
dat.hr2       <- plyr::join_all(list(timeSeq, dat.hr))
dat.combined2 <- rbind(dat.hr2, dat.6min)
HL.plot(level = dat.combined2[, 2], time = dat.combined2[, 1])

## End(Not run)
```

HL.plot *Plots water level data and high/low tides extracted by* HL()

Description

Plots water level data and high/low tides extracted by `HL()`. Purpose is for quick and easy visual assessment of `HL()` output.

Usage

```
HL.plot(
  level,
  time,
  period = 13,
  phantom = TRUE,
  tides = "all",
  semidiurnal = TRUE,
  verbose = FALSE
)
```

Arguments

`level`	a numeric vector of water levels
`time`	a vector (numeric or POSIX*) indicating the time of water level measurements. Units must be minutes.
`period`	a single numeric or integer estimate of tidal period (full tidal cycle). Units must be hours.
`phantom`	a protective measure taken to prevent the inclusion of an artificial high or low tide at the end of the dataset. If the water level measurements end precisely at a low or high tide, this can be changed to FALSE.
`tides`	is used to optionally subset the output to include only high or low tides. This argument can be 'all' (default), 'H', or 'L'
`semidiurnal`	logical. If TRUE, higher- and lower- high/low tides are reported in a separate column called 'tide2'
`verbose`	logical. If TRUE, messages are returned to console identifying the presence of NAs in the input data

Value

a plot of water levels, with red and blue dots superimposed on high and low tides.

See Also

HL

Examples

```
HL.plot(level = NL_6min_2013[,2], time = NL_6min_2013[,1])
HL.plot(level = NL_6min_2013[1:1000,2], time = NL_6min_2013[1:1000,1])

### HL.plot can be annotated using base graphics
abline(h = -0.3, lty = 2)
```

```
text(x = as.POSIXct("2013-01-01 00:00", format = "%F %R"), y = -0.25,
     "elevation of interest", pos = 4)
mtext("plot title", side = 3)

### or extended with ggplot in various ways.
## Not run:
library(ggplot2) # install with install.packages("ggplot2")
ggHL <- function(level, time, plotVariable = "tide2", period = 13,
        phantom = TRUE, tides = "all", semidiurnal = TRUE) {
  hl <- VulnToolkit::HL(level = level, time = time, period = period,
                        phantom = phantom, tides = tides, semidiurnal = semidiurnal)
  wll.2 <- data.frame(1:length(level), level, time)
  plot_return <- ggplot(data = wll.2, aes(x = time, y = level)) +
    geom_line(col = "darkgray") + xlab("") +
    geom_point(data = hl, aes(x = time, y = level, col = get(plotVariable))) +
    theme_classic() + theme(legend.title = element_blank())
  plot_return
}
newPlot <- ggHL(level = NL_6min_2013[1:1000,2], time = NL_6min_2013[1:1000,1])
newPlot

## End(Not run)
```

NL_6min_2013 *New London water levels, 2013*

Description

Water levels from NOAA-COOPS station number 8461490 in New London, CT. Data recorded at 6-minute intervals, in meters relative to MHW, and in the GMT time zone.

Usage

```
data(NL_6min_2013)
```

Format

A dataframe with 87591 rows and 3 variables.

Source

downloaded from National Atmospheric and Oceanic Administration's Center for Operational Oceanographic Products and Services using `VulnToolkit::noaa()`

noaa *Downloads NOAA CO-OPS tide data*

Description

Scrapes water level data (and other measurements) from NOAA CO-OPS website. NOAA's site limits the time period for data downloads, but these constraints are avoided by 'noaa()'. Requires internet connection and curl (check availablility using 'Sys.which("curl")').

Usage

```
noaa(begindate = "begindate", enddate = "enddate", station = "8467150",
met = FALSE, units = "meters", datum = "MHW", interval = "HL", time = "GMT",
continuous = TRUE)
```

Arguments

begindate	first day of data to download. Format must be YYYYMMDD. If left unspecified, the first complete day of data will be used.
enddate	final day of data to download. Format must be YYYYMMDD. If left unspecified, the last complete day of data will be used.
station	station number, found on the NOAA Tides and Currents website (https://www.tidesandcurrents.noaa.gov/s Station numbers can be numeric or a character string (necessary if first character is a zero). Default station is Bridgeport, CT.
met	whether meteorological data should be returned. This value can be 'TRUE' or 'FALSE'; if 'TRUE', all ancillary parameters are returned. At present, this only works with 6-minute and hourly data
units	can be 'feet' or 'meters'. Default is 'meters'
datum	vertical reference datum, set to 'MHW' by default. Can be 'station', 'NAVD', 'MLLW', 'MLW', 'MSL', 'MTL', 'MHW', 'MHHW', or 'IGLD' (some datums are not available at some sites)
interval	sets measurement interval; can be 'HL' (default), '6 minute', 'hourly', or 'monthly'.
time	can be 'LST', 'GMT', or 'LST/LDT'. Not all time zones are available for all data. GMT appears to have wider availability than LST, so it is the default.
continuous	determines whether a continuous time series is produced, with lengthy gaps in data filled in with NAs. By default, this is FALSE. This option only applies to data at evenly spaced intervals (i.e., 6 minute or hourly)

Details

Download water level and other data from NOAA CO-OPS website.

Value

dataset a dataframe with water levels, associated time stamps, a station ID column, and tide type (if interval is set to HL). The NOAA CO-OPS website has many odd data availabilty problems. Some data are not available in all time intervals or time zones.

Examples

```
## Not run:
# Example requires an internet connection
bport2013 <- noaa(begindate = 20130101, enddate = 20131231,
  station = "8467150", interval = "6 minute")

test2.1 <- noaa(begindate = "20100101", enddate = "20120101", interval = "hourly")
test2.2 <- noaa(begindate = "20100101", enddate = "20120101", interval = "hourly",
               continuous = TRUE)
nrow(test2.1) # includes data on NOAA site (incomplete record)
nrow(test2.2) # fills gaps with NAs
test2.3 <- noaa(begindate = "20100101", enddate = "20120101", interval = "hourly",
               met = TRUE)

## End(Not run)
```

---

noaa.datums *Scrapes elevation datums from NOAA CO-OPS website*

Description

Scrapes elevation datums from NOAA CO-OPS website

Usage

```
noaa.datums(station = 8467150)
```

Arguments

station	station ID number, available on the CO-OPS website.

Details

Scrapes elevation datums from NOAA CO-OPS website. Requires an internet connection.

Value

a dataframe of vertical datum names and their elevations in meters relative to the station datum for the 1983-2001 epoch. Also contains a column of times associated with relevant datums (record maximum and minimums, lowest and highest astronomical tides).

See Also

noaa

Examples

```
## Not run:
# examples require internet connection
bport.datums <- noaa.datums() # Bridgeport, CT
battery.datums <- noaa.datums(station = 8518750) # Battery, NYC

## End(Not run)
```

`noaa.parameters` *Reports data available for a NOAA station*

Description

Reports data available for a NOAA station

Usage

```
noaa.parameters(stn = 8467150)
```

Arguments

`stn`	NOAA station number (note that station names are not yet acceptable inputs). Default station is Bridgeport, CT.

Details

`noaa.parameters` reports the parameters (meteorological and tidal) available for a specified NOAA station.

Value

a dataframe with parameters and associated start and end dates. Where a parameter's availability is not continuous, multiple rows are reported. This function can be used to, for example, select meteorological parameters to include in calls to `noaa`

See Also

`noaa`

Examples

```
## Not run:
# examples require internet connection
noaa.parameters()
LA.stns <- noaa.stations(state = "LA")
noaa.parameters(LA.stns$number[1])

## End(Not run)
```

number.tides *Numbers tidal cycles, flood tides, and ebb tides*

Description

Numbers tidal cycles, flood tides, and ebb tides in a set of water level data.

Usage

```
number.tides(data, datetime, hl)
```

Arguments

data	dataframe to modify (containing water levels, time stamps)
datetime	date/time column from full dataset (used as 'time' argument in call to HL)
hl	output from HL

Details

Numbers tidal cycles, flood tides, and ebb tides in a set of water level data.

Value

the dataframe noted in data, with additional columns assigning a number to each tidal cycle, ebb tide, and flood tide.

See Also

HL

Examples

```
# build high-low dataset
HL.NL <- HL(level = NL_6min_2013[,2], time = NL_6min_2013[,1])
# number tides in original
nos <- number.tides(data = NL_6min_2013, datetime = NL_6min_2013[,1], HL.NL)
head(nos)
```

psmsl *Download sea level data from Permanent Service for Mean Sea Level*

Description

psmsl imports monthly/annual tide data from stations around the world. Dataset compiled and stewarded by the Permanent Service for Mean Sea Level (http://www.psmsl.org).

Usage

```
psmsl(station = 12, type = "RLR", interval = "annual")
```

Arguments

station	name or ID, or a vector of station names/IDs. Elements can be a character string (must match actual station name identically), or numeric station ID (no quotes: i.e., 12 rather than "12"). Use psmsl.stations to find stations, or check http://www.psmsl.org. If multiple stations are included, their data is combined using rbind, making a long (rather than wide) dataset. Default station is the Battery, in New York City.
type	data quality class; can be 'metric' (default; data has been rigorously standardized) or 'RLR'. See http://www.psmsl.org for documentation.
interval	time interval over which mean sea level is calculated. Can be 'monthly' or 'annual'.

Value

A data frame data.frame containing the requested Permanent Service for Mean Sea Level data.

See Also

psmsl.stations can be used to find stations

Examples

```
## Not run:
battery  <- psmsl()
stations <- psmsl(station = c(1372, 12), interval = "monthly")

## End(Not run)
```

psmsl.stations *Generates a list of active and historic PSMSL stations*

Description

Generates a list of active and historic tide stations hosted by the Permanent Service for Mean Sea Level (http://www.psmsl.org)

Usage

```
psmsl.stations(type = "RLR", country = "all", sort.by = "country")
```

Arguments

type	data quality class; can be 'metric' or 'RLR' - see http://www.psmsl.org for documentation
country	if desired, the full list of stations can be filtered by up to three alphabetical characters. Specific country codes can be entered ("USA"), or abbreviated codes in case the user isn't sure of the country code ("U"; "US"). Upper case and lower case codes are both acceptable. Default is 'all' stations.
sort.by	the criterion for sorting the final dataframe. By default, output is sorted alphabetically by country code. Any column name can be used for sorting: 'name', 'id', 'lat', 'long', 'country', 'date', 'coastline', or stn

Details

A data.frame containing all Permanent Service for Mean Sea Level stations meeting country code criterion. Data frame is sorted by the column specified in argument sort.by

Value

A data.frame containing all Permanent Service for Mean Sea Level stations meeting country code criterion. Data frame is sorted by the column specified in argument sort.by

See Also

psmsl

Examples

```
## Not run:
# examples require internet connection
stn.df <- psmsl.stations()
stn.df2 <- psmsl.stations(country = "USA", sort.by = "date")
stn.df3 <- psmsl.stations(type = "metric")

## End(Not run)
```

`wave.dur` *Find the proportion of time when water surface is near a selected elevation*

Description

Find the proportion of time when water surface is near a selected elevation

Usage

```
wave.dur(level, elevation, buffer = 0.050)
```

Arguments

`level`	a numeric vector of water levels
`elevation`	elevation(s) of interest (e.g., marsh platform, MHW). A vector of elevations is accepted. Elevation should be in the same vertical datum as `level` and have units of meters
`buffer`	the vertical range used to calculate exposure. Default is 0.050 meters (5 cm)

Value

a vector of values representing the fraction of time in the entire dataset where the water surface is near the elevation of interest. Numerical range is from 0-1. NAs are omitted.

Examples

```
data(NL_6min_2013)

### an elevation of interest: 0.9117 m
MHW <- 0.9117

wave.dur(NL_6min_2013[, 2], MHW)

## Not run:
### function also accepts and returns vectors
elevs           <- data.frame(elevs = c(-1000:1500) / 10^3)
elevs$wave.dur  <- wave.dur(NL_6min_2013[, 2], elevs$elevs)
plot(elevs ~ wave.dur, data = elevs)

## End(Not run)
```

Index