'VulnToolkit'

July 15, 2014

---

fld.dur	*Flooding duration*

Description

Calculates the percent of time an elevation is submerged

Usage

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

Arguments

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

Value

value	the decimal fraction of measurements in /codevec that fall above /codez

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
```

---

fld.frq   *Flooding frequency*

Description

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

Usage

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

Arguments

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

Value

value	the number or percent of high tides in /codevec that fall above /codez

Examples

```
# get a dataset of high/low tides
data(NL_6min_2013)
HT.NL <- HL(level = NL_6min_2013[, 2], time = NL_6min_2013[, 1], tides = "H")
a <- fld.frq(0.9117, HT.NL[, 1]) # 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 cove

elev.frq <- data.frame(elev = seq(from = 0, to = 1.5, by = 0.005))
fld.frq(elev.frq$elev, HT.NL[, 1]) # error thrown if units = "tides" and length(x) > length(ht)
elev.frq$frq <- fld.frq(elev.frq$elev, HT.NL[, 1], units = "percent") * length(HT.NL[, 1]) / b # a work-aroun

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

---

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 internet connection.

Usage

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

Arguments

station   station ID number(s), available on CO-OPS site (http://co-ops.nos.noaa.gov/stations.html?type=Wat
               A list

Value

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

Author(s)

Troy D. Hill; Hill.Troy@Gmail.com

References

http://wetlandsandr.wordpress.com/

Examples

```
# Example is commented out because it requires an internet connection
# a <- form.no()
# stn.list <- c("8467150", "8461490", "9454240")
# b <- form.no(stn.list)
# b
```

---

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

Description

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

Usage

```
    harcon(station)
```

Arguments

station   station ID number, available on CO-OPS site (http://co-ops.nos.noaa.gov/stations.html?type=Water+L

Value

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

Examples

```
# Example is commented out because it requires an internet connection
# bport.cons <- harcon(8467150) # Bridgeport, CT
# bport.cons
```

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")
```

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'

Value

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

Examples

```
data(NL_6min_2013)
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   *Extracts high and low tides from a record of water levels*

Description

Extracts high and low tides from a record of water levels

Usage

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

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'

Value

`dataset`	a dataframe of tide levels, associated time stamps, and tide type ('H' or 'L').

Examples

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

`NL_6min_2013`	*New London water levels, 2013*

Description

Water levels from NOAA-COOPS station #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 http://co-ops.nos.noaa.gov/inventory.html?id=8461490 using `VulnToolkit::noaa()`

References

Center for Operational Oceanographic Products and Services

---

noaa   *Scrapes water level data from NOAA-COOPS website*

Description

Scrapes water level data from NOAA-COOPS website. Requires internet connection.

Usage

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

Arguments

begindate, enddate

           set desired date range and must be in YYYYMMDD format. If one or both dates are left unspecified, the first and/or last complete day of available data will be used.

station   station name or ID number, available on CO-OPS site (http://co-ops.nos.noaa.gov/stations.html?type= Default is Bridgeport, CT station.

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', or 'hourly'

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.

Value

dataset   a dataframe of water levels, associated time stamps, a station ID column, and tide type (if interval is set to 'HL')

Examples

```
# Example is commented out because it requires an internet connection
# bport2013 <- noaa(begindate = 20130101, enddate = 20131231,
#      station = "Bridgeport, CT", interval = "6 minute")
```

---

noaa.datums   *Scrapes elevation datums from NOAA-COOPS website*

Description

Scrapes elevation datums from NOAA-COOPS website. Requires internet connection.

Usage

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

Arguments

station
station name or ID number, available on CO-OPS site (http://co-ops.nos.noaa.gov/stations.html?type=
Default is Bridgeport, CT station.

Value

dataset
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).

Author(s)

Troy D. Hill; Hill.Troy@Gmail.com

References

http://wetlandsandr.wordpress.com/

Examples

```
# Example is commented out because it requires an internet connection
# bport.datums <- noaa.datums() # Bridgeport, CT
# battery.datums <- noaa.datums(station = 8518750) # Battery, NYC
```

---

noaa.stations   *Prints active NOAA stations*

Description

noaa.stations returns active NOAA CO-OPS tide stations.

Usage

```
noaa.stations(state = "all")
```

Arguments

state
state limits printed records to a state or states of interest. Default is to show 'all'
stations.

Details

This code returns a dataframe showing all active stations in the NOAA CO-OPS network. This can
be used to identify stations for use in noaa() (see example below). The argument state offers an
easy way of subsetting the data frame to the state(s) of interest.

Value

A dataframe with station names, states/territories, and station numbers. If a 'state' argument is
provided that is not found on the NOAA site, an error message is displayed. If a vector of states is
provided, some of which are invalid, a dataframe will be produced for the valid criteria and invalid
criteria will be noted.

See Also

NOAA CO-OPS site: http://co-ops.nos.noaa.gov/stations.html?type=Water+Levels Historic stations which can also be accessed by `noaa()`: http://co-ops.nos.noaa.gov/stations.html

Examples

```
# examples require internet connection
# a <- noaa.stations() # all active stations
# b <- noaa.stations(state = "MA")
# c <- noaa.stations(state = c("RI", "CT"))
# d <- noaa.stations(state = c("OR", "RI", "MA", "Germany", "Pluto")) # two erroneous entries
# highlow <- noaa(station = d[1,1], begindate = 20130101)
```

---

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()`

Value

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

Examples

```
# build high-low dataset
data(NL_6min_2013)
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   *Downloads tide data hosted by the Permanent Service for Mean Sea*
                               Level (www.psmsl.org)

Description

This function scrapes data from www.psmsl.org according to user-defined criteria.

Usage

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

Arguments

station
: station name or number, or a vector of station names/numbers. 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 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' or 'RLR'. See 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.

References

http://wetlandsandr.wordpress.com/

See Also

`psmsl.stations`

Examples

```
# examples require internet connection
# battery <- psmsl()
# stations <- psmsl(station = c(1372, 12), interval = "monthly")
```

psmsl.stations	*Display PSMSL tide stations*

Description

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

Usage

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

Arguments

type
: data quality class; can be 'metric' or 'RLR' - see 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', 'GLOSS_ID', 'country', 'date', 'coastline', or 'number'

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'

References

http://wetlandsandr.wordpress.com/

See Also

psmsl

Examples

```
# examples require internet connection
# stn.df <- psmsl.stations()
# stn.df2 <- psmsl.stations(country = "USA", sort.by = "date")
```

---

vuln.kit *Calculates selected hydrologic parameters and vulnerability metrics*

Description

This function takes a set of water level data as an input, and calculates a set of flooding parameters.

Usage

```
vuln.kit(level, datetime, platform, units = "meters", frq.dur.inc = 0.005,
                 TV.inc = 0.1)
```

Arguments

level	a numeric vector of water levels
datetime	a POSIX* vector of time stamps that correspond to the measurements in 'level'
platform	elevation of the marsh platform (or another vertical position of interest). Should be in the units specified by `units` and relative to the same vertical datum as `level`.
units	length units used. 'meters' is default; 'feet' is alternative. If units are "feet," data is converted internally and output in meters.
frq.dur.inc	elevation interval used to calculate flooding frequency, duration, D90, and Ax. Defaults to 0.005 m. Units must correspond to `units` argument
TV.inc	Elevation interval used to calculate vulnerability metrics (DV, D90V). Defaults to 0.1 m. Units must correspond to `units` argument.

Value

output	a list containing two items:
dataset	a dataframe of elevations (relative to elevation set in `platform` argument), flooding frequencies (flooding events per year), flooding durations (hr yr-1), duration of 90th percentile flooding event (D90; hr), and mean flooding depth (A; m)
metrics	a dataframe containing the flooding frequency, flooding duration, D90, mean flooding depth, duration vulnerability, and D90 vulnerability, calculated at the vertical elevation set by `platform` argument

Examples

```
data(NL_6min_2013)
NL2013 <- vuln.kit(level = NL_6min_2013[,2], datetime = NL_6min_2013[,1],
                 platform = 0.9117) # MHW in 2013: 0.9117 m relative to MLLW
NL2013$metrics
```

Index