Package ‘rfInterface’

February 20, 2019

Type Package

Title Functions for Using UK Riverflow and Rainfall API

Date 2018-12-17

Version 0.1.0

Author Adam Griffin <adagri@ceh.ac.uk>

Maintainer Adam Griffin <adagri@ceh.ac.uk>

Description rfInterface provides functions to easily access EA and NRFA river flow data and SEPA rainfall data, with utility functions to convert station identifiers and location grid references. Preview of data at https://gateway-staging.ceh.ac.uk/hydrology-ukscape.

Encoding UTF-8

LazyData true

License GPL-3

Imports jsonlite, lubridate, graphics, grDevices, pastecs

RoxygenNote 6.1.1

R topics documented:

autoAxis *Add automatically scaled horizontal date axis.*

Description

Using a provided time series, this functions adds a horizontal datetime axis to an existing plot in the same fashion as the `axis` function. Ticks and labels are automatically determined. Additional graphical `par` arguments can be passed to customise apperance.

Usage

```
autoAxis(series, ...)
```

Arguments

`series`	vector of datetime objects (readable by `as.Date`)
`...`	additional `par` arguments to be passed to `axis`. Can include any argument passed to axis except `side`, `at`, `labels` and `tick`.

Value

Adds axis to existing plot in current device.

Examples

```
## Not run:
    ts <- seq(as.Date("01-01-1900"),as.Date("01-01-1902"),by="day")
    autoAxis(series = ts, cex.axis=0.6, las=1)

## End(Not run)
```

idConverter *Convert between different station identifiers.*

Description

Given a station identifier, and its source (either NRFA or EA/SEPA), this function provides the identifier designated by the other organisation. idConverter assumes that any given station is uniquely monitored by EA or SEPA, not both.

Usage

```
idConverter(num, source = c("NRFA", "EA", "SEPA"))
```

Arguments

`num`	vector of numbers or strings containing identifier(s) of station (all should come from same source)
`source`	string indicating source of provided identifier(s), not target.

Value

the identifier(s) from the other data source not selected in 'source'. If not found, returns NA.

Examples

```
## Not run:
idConverter(39071, "NRFA")  # NRFA to EA/SEPA
idConverter("0130TH", "EA")  # EA to NRFA

## End(Not run)
```

idToRef *Obtain station reference for river flow API.*

Description

For EA/SEPA/COSMOS stations for which data is provided, this provides the reference string for the API, given the identifier.

Usage

```
idToRef(id)
```

Arguments

id	EA/SEPA/COSMOS station id.

Value

ref string to request flow from API

Examples

```
## Not run:
idToRef("SS50F007")

## End(Not run)
```

`import_metadata` *Import metadata from river flow API.*

Description

Using the river flow/rainfall API, station information can be extracted for single or multiple sites. All data must be of the same type from the same organisation.

Usage

```
import_metadata(ids, dat, org = c("NRFA", "EA", "SEPA", "COSMOS"))
```

Arguments

`ids`	identifier for stations (not EA refs)
`dat`	string indicating datatype, as written in metadata.
`org`	organisation from whom the data is obtained.

Value

a list containing:

- id - measuring authority station identifier
- ref - API reference string
- name - station name
- organisation
- station aliases under different organisations
- datatype - list of descriptors of data
- startDate - character string of first record
- dataUrl - string of URL to obtain data from API directly.

If not found, returns NA for each such station.

Examples

```
## Not run:
import_metadata(ids=c("SX67F051", "SS50F007"), org="EA", dat="gdf")
import_metadata(ids="SX67F051", org="EA", dat="gdf")

## End(Not run)
```

import_ts — *Import time series from river flow API.*

Description

Using the river flow/rainfall API, time series can be extracted, either selecting single dates, periods of record, or entire records for single or multiple sites. Metadata can also be returned for stations in the dataset. All data must be of the same type from the same organisation over the same period.

Usage

```
import_ts(ids, dat, org = c("NRFA", "EA", "SEPA", "COSMOS"),
  startDate = NULL, endDate = NULL, metadata = FALSE,
  datetime = TRUE)
```

Arguments

`ids`	identifier for stations (not EA refs)
`dat`	string indicating datatype, as written in metadata.
`org`	organisation from whom the data is obtained.
`startDate`	string of the form `YYYY-MM-DD` to indicate start of period desired, or single date. Whole record given if no startDate provided.
`endDate`	string of the form `YYYY-MM-DD` to indicate end of period desired. If no startDate provided, this is ignored.
`metadata`	if `TRUE`, returns metadata for each station selected.
`datetime`	if `TRUE`, converts datetime column into POSIXlt.

Value

a dataframe containing the dates and magnitudes of the selected data. If multiple stations selected, the dataframes are contained in a named list. If metadata is true, each station will consist of a list containing `detail` and `data`. If not found, returns NA for each such station.

Examples

```
## Not run:
import_ts(ids=c("SX67F051", "SS50F007"), org="EA", dat="gdf",
    startDate="2017-01-01", endDate="2017-02-01")
import_ts(ids="SX67F051", org="EA", dat="gdf", metadata=T)

## End(Not run)
```

POTextract *Extract Peak-over-Threshold series from timeseries (NRFA criteria)*

Description

Using a given time series, a POT series is returned using the NRFA criteria:

- Peaks must be seperated by a time interval at least 3 times as long as the estimated time to rise at the location of interest.
- Between peaks, the trough must be no more than two-thirds the size of either peak.
- All peaks must exceed a given threshold: current standard is to choose a threshold which is expected to permit, on average, five events per year.

This is based on the work of Ilaria Prosdocimi and the packages pastecs and ilaProsUtils.

Usage

```
POTextract(series, datetime = NULL, threshold = 0, timeOfRise = 0)
```

Arguments

series	Either a vector of flow magnitudes, or a dataframe with two #' columns, magnitude and datetime.
datetime	If series is not a dataframe, this is a vector of datetimes #' corresponding to the events in series vector. If not supplied, and no datetime supplied, ordinal values are assumed (1,2,...,N).
threshold	positive number describing minimum size of peak event. If not supplied, takes the 90th percentile of the data.
timeOfRise	positive number describing time of rise at the location of the measurements.

Value

a list containing:

- a dataframe of the peak events, including datetime and magnitude
- a boolean vector the length of the original series indicating which measurements are peaks.

Examples

```
## Not run:
    xt <- (arima.sim(list(order=c(3,0,0), ar=c(0.5,0.25,0.125)), n=100) + 2)^2
    dt <- as.Date(1:100, origin="2000-01-01")
    POTextract(xt, dt, threshold = 20, timeOfRise = 3)

## End(Not run)
```

`rainFlowPlot` *Plot rainfall and flow time series on the same plot.*

Description

Given an identifier for a rain measurement and/or a flow gauging station, this function will plot the combined plot of rainfall and river flow, or a single plot of the desired time series.

Usage

```
rainFlowPlot(rainID = NULL, flowID = NULL, rainOrg = NULL,
  flowOrg = NULL, rainDat = "cdr", flowDat = "gdf",
  startDate = NULL, endDate = NULL, filepath = NULL)
```

Arguments

`rainID`	identifier for station recording rainfall
`flowID`	identifier for station recording flow
`rainOrg`	organisation responsible for the station in `rainID`
`flowOrg`	organisation responsible for the station in `flowID`
`rainDat`	rainfall measurement designator as described in metadata
`flowDat`	flow measurement designator as described in metadata
`startDate`	string of the form `YYYY-MM-DD` to indicate start of period desired, or single date. Whole record given if no startDate provided.
`endDate`	string of the form `YYYY-MM-DD` to indicate end of period desired. If no startDate provided, this is ignored.
`filepath`	if provided, this gives the path to save a .png image. ".'png" must be included at the end of the filepath.

Value

plots figure to open device unless `filepath` provided, in which case a .png file is saved instead.

Examples

```
## Not run:
rainFlowPlot(rainID = "234176", flowID = "9005",
             rainOrg = "SEPA", flowOrg = "NRFA",
             startDate = "2017-07-01", endDate = "2018-07-01")

## End(Not run)
```

`rfInterface` *Functions for Using UK Riverflow and Rainfall API*

Description

rfInterface: Functions for Using UK Riverflow and Rainfall API

Details

rfInterface provides functions to easily access EA and NRFA river flow data and SEPA rainfall data, with utility functions to convert station identifiers and location grid references. Preview of data at `https://gateway-staging.ceh.ac.uk/hydrology-ukscape`.

The Environment Agency (EA) data is currently restricted to Devon/Cornwall riverflow data starting from 2017-01-01. EA stations have a station identifier (either numeric or alphanumeric, approximately 8 characters), and a reference string which the API uses (alphanumeric with dashes, approximately 36 characters).

The Scottish Environment Protection Agency (SEPA) data is currently restricted to rainfall in Scotland from 2018-02-13. Rainfall is presented as daily catchment total rainfall for given catchments corresponding to riverflow gauging stations. SEPA stations have a 6-7 digit station number, which is also its API reference string.

The National River Flow Archive (NRFA) provides gauged daily river flow across the UK. Currently river flow data is restricted to the period 2017-01-01 to 2018-09-01. NRFA stations have a 6-7 digit station number, which is also its API reference string.

`stationList` *Gives a list of stations from the relevant organisation.*

Description

Returns a dataframe of stations from a selected organisation. Columns include NRFA station number, latitude, longitude, id, JSON reference, and corresponding identifiers under other organisations.

Usage

```
stationList(org = c("EA", "NRFA", "SEPA", "COSMOS"))
```

Arguments

`org`	organisation to obtain list of stations from.

Value

dataframe containing station names, ids and other important information.

Examples

```
## Not run:
stationList("NRFA")

## End(Not run)
```

`ts_reformat` *Reformats a time series to have datetime objects.*

Description

Converts a data.frame with strings for datetimes into one with POSIXlt date objects.

Usage

```
ts_reformat(ts)
```

Arguments

`ts`	time series data.frame object of two columns: datetime (string of datetimes in form `YYYY-MM-DDTHH:MM:SSZ`) and data (numeric).

Value

data.frame with replaced datetime column containing equivalent POSIXlt objects.

Index