Package 'rnrfa'

April 15, 2019

Title UK National River Flow Archive Data from R

Version 2.0

Maintainer Claudia Vitolo <cvitolodev@gmail.com>

URL http://cvitolo.github.io/rnrfa/

BugReports https://github.com/cvitolo/rnrfa/issues

Description Utility functions to retrieve data from the UK National River Flow Archive (<http://nrfa.ceh.ac.uk/>). The package contains R wrappers to the UK NRFA data temporary-API. There are functions to retrieve stations falling in a bounding box, to generate a map and extracting time series and general information.

Depends R (>= 3.0.2)

Imports rgdal, dplyr, curl, jsonlite, lubridate, graphics, stats, httr, xts, ggmap, ggplot2, sp, parallel, tibble

Suggests testthat, knitr, covr, lintr, rmarkdown

LazyData true

Encoding UTF-8

License GPL-3

Repository CRAN

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

Author Claudia Vitolo [aut, cre] (<https://orcid.org/0000-0002-4252-1176>),
Matthew Fry [ctb] (Matthew supervised the unofficial API integration.),
Wouter Buytaert [ctb] (This package is part of Claudia Vitolo's PhD work and Wouter is the supervisor.),
Michael Spencer [ctb] (Michael updated the function osg_parse to work with grid references of different lengths.),
Tobias Gauster [ctb] (Tobias improved the function osg_parse introducing vectorisation)

Date/Publication 2019-04-15 10:52:42 UTC

R topics documented:

`rnrfa-package` *UK National River Flow Archive data from R*

Description

rnrfa: UK National River Flow Archive Data from R.

Details

Utility functions to retrieve data from the UK National River Flow Archive (http://nrfa.ceh.ac.uk/). The package contains R wrappers to the UK NRFA data temporary-API. There are functions to retrieve stations falling in a bounding box, to generate a map and extracting time series and general information.

`catalogue` *List of stations from UK NRFA*

Description

This function pulls the list of stations (and related metadata), falling within a given bounding box, from the CEH National River Flow Archive website.

Usage

```
catalogue(bbox = NULL, column_name = NULL, column_value = NULL,
  min_rec = NULL, all = TRUE)
```

Arguments

bbox	this is a geographical bounding box (e.g. list(lon_min = -3.82, lon_max = -3.63, lat_min = 52.43, lat_max = 52.52))
column_name	name of column to filter
column_value	string to search in column_name
min_rec	minimum number of recording years
all	if TRUE it returns all the available metadata. If FALSE, it returns only the following columns: id, name, river, hydrometricArea, operator, haName, catchmentArea, altitude, lat, lon.

Details

coordinates of bounding box are required in WGS84 (EPSG: 4326). If BB coordinates are missing, the function returns the list corresponding to the maximum extent of the network.

Value

tibble table containing the list of stations and related metadata

Author(s)

Claudia Vitolo

Examples

```
## Not run:
  # Retrieve all the stations in the network
  x <- catalogue()

  # Define a bounding box:
  bbox <- list(lon_min=-3.82, lon_max=-3.63, lat_min=52.43, lat_max=52.52)
  # Get stations within the bounding box
  x <- catalogue(bbox)

  # Get stations based on minimum catchment area
  x <- catalogue(column_name = "catchment-area", column_value = 2000)

  # Get stations based on minimum number of recording years
  x <- catalogue(min_rec=30)

## End(Not run)
```

cmr *This function retrieves Catchment Mean Rainfall (cmr).*

Description

Given the station ID number(s), this function retrieves data (time series in zoo format with accompanying metadata) from the WaterML2 service on the NRFA database. Catchment Mean Rainfall is measured in mm/month.

Usage

```
cmr(id, metadata = FALSE, cl = NULL, verbose = FALSE)
```

Arguments

id	station ID number(s), each number should be in the range [3002,236051].
metadata	Logical, FALSE by default. If metadata = TRUE means that the result for a single station is a list with two elements: data (the time series) and meta (metadata).
cl	(optional) This is a cluster object, created by the parallel package. This is set to NULL by default, which sends sequential calls to the server.
verbose	(FALSE by default). If set to TRUE prints GET request on the console.

Value

list composed of as many objects as in the list of station ID numbers. Each object can be accessed using their names or index (e.g. x[[1]], x[[2]], and so forth). Each object contains a zoo time series.

Author(s)

Claudia Vitolo

Examples

```
## Not run:
  cmr(18019)
  cmr(c(54022,54090,54091))

## End(Not run)
```

convert_flow *Convert flow from cumecs to mm/d*

Description

This function converts flow time series from cumecs (m3/s) to mm/d by dividing the flow by the catchment area and converting it to mm/day.

Usage

```
convert_flow(flow_cumecs, catchment_area)
```

Arguments

flow_cumecs	This is the flow time series in cumecs (m3/s)
catchment_area	This is the catchment are in Km2.

Value

Flow time series in mm/d

Examples

```
## Not run:
  convert_flow(30, 2)

## End(Not run)
```

gdf *This function retrieves Gauged Daily Flow (gdf).*

Description

Given the station ID number(s), this function retrieves data (time series in zoo format with accompanying metadata) from the WaterML2 service on the NRFA database. Gauged Daily Flow is measured in mm/day.

Usage

```
gdf(id, metadata = FALSE, cl = NULL, verbose = FALSE)
```

Arguments

id	station ID number(s), each number should be in the range [3002,236051].
metadata	Logical, FALSE by default. If metadata = TRUE means that the result for a single station is a list with two elements: data (the time series) and meta (metadata).
cl	(optional) This is a cluster object, created by the parallel package. This is set to NULL by default, which sends sequential calls to the server.
verbose	(FALSE by default). If set to TRUE prints GET request on the console.

Value

list composed of as many objects as in the list of station ID numbers. Each object can be accessed using their names or index (e.g. x[[1]], x[[2]], and so forth). Each object contains a zoo time series.

Author(s)

Claudia Vitolo

Examples

```
## Not run:
  gdf(18019)
  gdf(c(54022,54090,54091))

## End(Not run)
```

---

get_ts *This function retrieves time series data.*

Description

Given the station identification number(s), this function retrieves data (time series in zoo format with accompanying metadata) from the WaterML2 service on the NRFA database. The time series can be of two types: cmr (catchment mean rainfall, monthly) or gdf (gauged daily flows, daily).

Usage

```
get_ts(id, type, metadata = FALSE, cl = NULL, verbose = FALSE)
```

Arguments

id	station identification number(s), each number should be in the range [3002,236051].
type	The following data-types are available: • gdf = Gauged daily flows • gmf = Gauged monthly flows

- ndf = Naturalised daily flows
- nmf = Naturalised monthly flows
- cdr = Catchment daily rainfall
- cdr-d = Catchment daily rainfall distance to rain gauge
- cmr = Catchment monthly rainfall
- pot-stage = Peaks over threshold stage
- pot-flow = Peaks over threshold flow
- gauging-stage = Gauging stage
- gauging-flow = Gauging flow
- amax-stage = Annual maxima stage
- amax-flow = Annual maxima flow

`metadata` Logical, FALSE by default. If metadata = TRUE means that the result for a single station is a list with two elements: data (the time series) and meta (metadata).

`cl` (optional) This is a cluster object, created by the parallel package. This is set to NULL by default, which sends sequential calls to the server.

`verbose` (FALSE by default). If set to TRUE prints GET request on the console.

Value

list composed of as many objects as in the list of station identification numbers. Each object can be accessed using their names or index (e.g. x[[1]], x[[2]], and so forth). Each object contains a zoo time series.

Author(s)

Claudia Vitolo

Examples

```
## Not run:
  get_ts(18019, type = "cmr")
  get_ts(c(54022,54090,54091), type = "cmr")
  get_ts(18019, type = "gdf")
  get_ts(c(54022,54090,54091), type = "gdf")
  plot(get_ts(id = 23001, type = "ndf"))
  plot(get_ts(id = 23001, type = "nmf"))

## End(Not run)
```

osg_parse *Converts OS Grid Reference to BNG/WGS coordinates.*

Description

This function converts an Ordnance Survey (OS) grid reference to easting/northing or latitude/longitude coordinates.

Usage

```
osg_parse(grid_refs, coord_system = c("BNG", "WGS84"))
```

Arguments

grid_refs	This is a string (or a character vector) that contains the OS grid Reference.
coord_system	By default, this is "BNG" which stands for British National Grids. The other option is to set coord_system = "WGS84", which returns latitude/longitude coordinates (more info can be found here https://www.epsg-registry.org/).

Value

vector made of two elements: the easting and northing (by default) or latitude and longitude coordinates.

Author(s)

Claudia Vitolo

Examples

```
## Not run:
  # single entry
  osg_parse(grid_refs = "TQ722213")

  # multiple entries
  osg_parse(grid_refs = c("SN831869","SN829838"))

## End(Not run)
```

plot_rain_flow *Plot rainfall and flow for a given station*

Description

This function retrieves rainfall and flow time series for a given catchment, divides the flow by the catchment area and converts it to mm/day to that it can be comparable with the rainfall (mm/month). Finally it generates a plots combining rainfall and flow information.

Usage

```
plot_rain_flow(id = NULL, rain = NULL, flow = NULL, area = NULL,
  title = "")
```

Arguments

id	Station identification number
rain	Rainfall time series, measured in mm/month
flow	Flow time series, measured in m3/s
area	Catchment area in Km2
title	(optional) Plot title

Value

Plot rainfall and flow for a given station

Examples

```
## Not run:
  plot_rain_flow(id = 54090)

## End(Not run)
```

plot_trend *Plot trend*

Description

This function plots a previously calculated trend.

Usage

```
plot_trend(df, column_name)
```

Arguments

df	Data frame containing at least 4 column: lat (latitude), lon (longitude), slope and an additional user-defined column column_name.
column_name	name of the column to use for grouping the results.

Value

Two plots, side-by-side, the first showing the distribution of the trend over a map, based on the slope of the linear model that describes the trend. The second plot shows a boxplot of the slope grouped based on the column Region. Region and slope can be user-defined.

Examples

```
## Not run:
  plot_trend(df, Region)

## End(Not run)
```

---

seasonal_averages *Calculate seasonal averages*

Description

This calculates the seasonal averages from a time series.

Usage

```
seasonal_averages(timeseries, season = "Spring", startseason = NULL,
  endseason = NULL, parallel = FALSE)
```

Arguments

timeseries	Time series (xts class).
season	Name of the season (Autumn, Winter, Spring, Summer)
startseason	String encoding the start of the season (e.g. for spring in the northen hemisphere this is "03-21")
endseason	String encoding the end of the season (e.g. for spring in the northen emisphere this is "06-20")
parallel	Logical, FALSE by default. If parallel = TRUE means that the function can be used in parallel computations.

Value

A vector containing the seasonal average and significance level (p-value) for each time series.

Examples

```
## Not run:
  seasonal_averages(timeseries = cmr(18019), season = "Spring")
  seasonal_averages(list(cmr(18019), cmr(18019)), season = "Spring")

## End(Not run)
```

`station_ids` *List of stations identification numbers from UK NRFA*

Description

This function pulls the list of station identification numbers.

Usage

```
station_ids()
```

Value

vector integer identification numbers (one for each station)

Author(s)

Claudia Vitolo

Examples

```
## Not run:
  # Retrieve all the stations ids
  x <- station_ids()

## End(Not run)
```

Index