

Package ‘hddtools’

August 9, 2018

Title Hydrological Data Discovery Tools

Version 0.8.2

Maintainer Claudia Vitolo <cvitolodev@gmail.com>

URL <http://ropensci.github.io/hddtools>,
<https://github.com/ropensci/hddtools>

BugReports <https://github.com/ropensci/hddtools/issues>

Description

Facilitates discovery and handling of hydrological data, access to catalogues and databases.

Depends R (>= 3.2.1), rgdal

Imports zoo, sp, RCurl, XML, rnrf, Hmisc, raster, stringr, gdata,
tibble

Suggests testthat, leaflet, rmarkdown, knitr, pkgdown

VignetteBuilder knitr

License GPL-3

Repository CRAN

RoxygenNote 6.1.0

Encoding UTF-8

NeedsCompilation no

Author Claudia Vitolo [aut, cre] (<<https://orcid.org/0000-0002-4252-1176>>),
Wouter Buytaert [ctb] (This package was developed during Claudia's PhD
at Imperial College London and Wouter was the PhD supervisor.),
Erin Le Dell [ctb] (Erin reviewed the package for rOpenSci, see
<https://github.com/ropensci/onboarding/issues/73>),
Michael Sumner [ctb] (Michael reviewed the package for rOpenSci, see
<https://github.com/ropensci/onboarding/issues/73>)

Date/Publication 2018-08-09 21:40:03 UTC

R topics documented:

bboxSpatialPolygon	2
catalogueData60UK	3
catalogueGRDC	4
catalogueMOPEX	6
catalogueSEPA	7
Data60UKcatalogue	8
getContent	9
GRDCcatalogue	9
grdcLTMMMD	11
hddtools	11
KGClimateClass	12
MOPEXcatalogue	13
SEPAcatalogue	14
tsData60UK	14
tsGRDC	15
tsMOPEX	18
tsSEPA	19
Index	20

bboxSpatialPolygon	<i>Convert a bounding box to a SpatialPolygons object Bounding box is first created (in lat/lon) then projected if specified</i>
--------------------	--

Description

Convert a bounding box to a SpatialPolygons object Bounding box is first created (in lat/lon) then projected if specified

Usage

```
bboxSpatialPolygon(boundingbox, proj4stringFrom = NULL,
  proj4stringTo = NULL)
```

Arguments

boundingbox	Bounding box: a 2x2 numerical matrix of lat/lon coordinates
proj4stringFrom	Projection string for the current boundingbox coordinates (defaults to lat/lon, WGS84)
proj4stringTo	Projection string, or NULL to not project

Value

A SpatialPolygons object of the bounding box

References

<http://gis.stackexchange.com/questions/46954/clip-spatial-object-to-bounding-box-in-r>

Examples

```
## Not run:
  boundingbox <- raster::extent(-180, +180, -50, +50)
  bbSP <- bboxSpatialPolygon(boundingbox = boundingbox)

## End(Not run)
```

catalogueData60UK *Data source: Data60UK catalogue*

Description

This function interfaces the Data60UK database catalogue (available from <http://nrfaapps.ceh.ac.uk/datauk60/data.html>) containing 61 datasets. Dataset catalogue is available from <http://nrfaapps.ceh.ac.uk/datauk60/data.html>.

Usage

```
catalogueData60UK(areaBox = NULL, columnName = NULL,
  columnValue = NULL, useCachedData = TRUE)
```

Arguments

areaBox	bounding box, a list made of 4 elements: minimum longitude (lonMin), minimum latitude (latMin), maximum longitude (lonMax), maximum latitude (latMax)
columnName	name of the column to filter
columnValue	value to look for in the column named columnName
useCachedData	logical, set to TRUE to use cached data, set to FALSE to retrieve data from online source. This is TRUE by default.

Value

This function returns a data frame made of 61 rows (gauging stations) and 6 columns: "id" (hydro-metric reference number), "River", "Location", "gridReference", "Latitude", "Longitude".

Author(s)

Claudia Vitolo

Examples

```
## Not run:
# Retrieve the whole catalogue
Data60UK_catalogue_all <- catalogueData60UK()

# Filter the catalogue based on a bounding box
areaBox <- raster::extent(-4, -2, +52, +53)
Data60UK_catalogue_bbox <- catalogueData60UK(areaBox)

# Filter the catalogue based on an ID
Data60UK_catalogue_ID <- catalogueData60UK(columnName = "stationID",
                                           columnValue = "62001")

## End(Not run)
```

catalogueGRDC

Data source: Global Runoff Data Centre catalogue

Description

This function interfaces the Global Runoff Data Centre database which provides river discharge data for about 9000 sites over 157 countries.

Usage

```
catalogueGRDC(areaBox = NULL, columnName = NULL, columnValue = NULL,
              useCachedData = TRUE)
```

Arguments

areaBox	bounding box, a list made of 4 elements: minimum longitude (lonMin), minimum latitude (latMin), maximum longitude (lonMax), maximum latitude (latMax)
columnName	name of the column to filter
columnValue	value to look for in the column named columnName
useCachedData	logical, set to TRUE to use cached data, set to FALSE to retrieve data from online source. This is TRUE by default.

Value

This function returns a data frame made of 9481 rows (gauging stations, as per October 2017) and 26 columns:

- grdc_no: GRDC station number
- wmo_reg: WMO region
- sub_reg: WMO subregion


```
# Get only catchments within river Thames
GRDC_catalogue_river <- catalogueGRDC(columnName = "river",
                                     columnValue = "Thames")

## End(Not run)
```

catalogueMOPEX *Data source: MOPEX catalogue*

Description

This function interfaces the MOPEX database catalogue (available from ftp://hydrology.nws.noaa.gov/pub/gcip/mopex/US_Data/) containing 438 daily datasets.

Usage

```
catalogueMOPEX(areaBox = NULL, columnName = NULL, columnValue = NULL,
               useCachedData = TRUE)
```

Arguments

areaBox	bounding box, a list made of 4 elements: minimum longitude (lonMin), minimum latitude (latMin), maximum longitude (lonMax), maximum latitude (latMax)
columnName	name of the column to filter
columnValue	value to look for in the column named columnName
useCachedData	logical, set to TRUE to use cached data, set to FALSE to retrieve data from online source. This is TRUE by default.

Value

This function returns a data frame made of 431 rows (gauging stations) and 12 columns containing stations metadata.

Author(s)

Claudia Vitolo

Examples

```
## Not run:
# Retrieve the MOPEX catalogue
MOPEX_catalogue_all <- catalogueMOPEX()

# Define a bounding box
areaBox <- raster::extent(-95, -92, 37, 41)
# Filter the catalogue based on bounding box
```

```
MOPEX_catalogue_bbox <- catalogueMOPEX(areaBox = areaBox)

# Get only catchments within NC
MOPEX_catalogue_state <- catalogueMOPEX(columnName = "state",
                                         columnValue = "NC")

## End(Not run)
```

catalogueSEPA

Data source: SEPA catalogue

Description

This function provides an unofficial SEPA database catalogue of river level data (available from <http://pennine.ddns.me.uk/riverlevels/ConciseList.html>) containing info for 1752 stations. Some are NRFA stations.

Usage

```
catalogueSEPA(columnName = NULL, columnValue = NULL,
              useCachedData = TRUE)
```

Arguments

columnName	name of the column to filter
columnValue	value to look for in the column named columnName
useCachedData	logical, set to TRUE to use cached data, set to FALSE to retrieve data from online source. This is TRUE by default.

Value

This function returns a data frame made of a maximum of 830 rows (stations) and 8 columns: "id-NRFA", "aspxpage", "stationId", "River", "Location", "GridRef", "Operator" and "CatchmentArea(km2)". Column idNRFA shows the National River Flow Archive station id. Column "aspxpage" returns the Environment Agency gauges id. The column "stationId" is the id number used by SEPA. Use the stationId to retrieve the time series of water levels.

Author(s)

Claudia Vitolo

Examples

```
## Not run:
# Retrieve the whole catalogue
SEPA_catalogue_all <- catalogueSEPA()

# Get only catchments with area above 5000 Km2
SEPA_catalogue_area <- catalogueSEPA(columnName = "CatchmentAreaKm2",
                                     columnValue = ">= 5000")

# Get only catchments within river Avon
SEPA_catalogue_river <- catalogueSEPA(columnName = "River",
                                       columnValue = "Avon")

## End(Not run)
```

Data60UKcatalogue

Data set: The Data60UK Catalogue

Description

The Data60UK catalogue

Usage

```
data("Data60UKcatalogue")
```

Format

A data frame with 61 stations (rows) and 6 metadata fields (columns).

stationID Station id number.

River String describing the river's name.

Location String describing the location.

gridReference British National Grid Reference.

Latitude

Longitude

Source

<http://nrfaapps.ceh.ac.uk/datauk60/data.html>

getContent	<i>Extracts links from ftp page</i>
------------	-------------------------------------

Description

This function extracts all the links in a ftp page

Usage

```
getContent(dirs)
```

Arguments

dirs is the url from which links should be extracted

Value

vector containing all the links in the page

Author(s)

Claudia Vitolo

Examples

```
## Not run:  
# Retrieve mopex daily catalogue  
url <- "ftp://hydrology.nws.noaa.gov/pub/gcip/mopex/US_Data/Us_438_Daily/"  
getContent(dirs = url)  
  
## End(Not run)
```

GRDCcatalogue	<i>Data set: The GRDC Catalogue</i>
---------------	-------------------------------------

Description

The GRDC catalogue

Usage

```
data("GRDCcatalogue")
```

Format

A data frame with 9252 stations (rows) and 46 metadata fields.

grdc_no : GRDC station number

wmo_reg : WMO region

sub_reg : WMO subregion

nat_id : national station ID

river : river name

station : station name

country : country code (ISO 3166)

lat : latitude, decimal degree

long : longitude, decimal degree

area : catchment size, km²

altitude : height of gauge zero, m above sea level

ds_stat_no : GRDC station number of next downstream GRDC station

d_start : daily data available from year

d_end : daily data available until year

d_yrs : length of time series, daily data

d_miss : percentage of missing values (daily data)

m_start : monthly data available from

m_end : monthly data available until

m_yrs : length of time series, monthly data

m_miss : percentage of missing values (monthly data)

t_start : earliest data available

t_end : latest data available

t_yrs : maximum length of time series, daily and monthly data

lta_discharge : mean annual streamflow, m³/s

r_volume_yr : mean annual volume, km³

r_height_yr : mean annual runoff depth, mm

Source

http://www.bafg.de/GRDC/EN/02_srvcs/21_tmsrs/211_ctlgs/catalogues_node.html

`grdcLTMMMD`*Data set: The grdcLTMMMD look-up table*

Description

The grdcLTMMMD look-up table

Usage

```
data("grdcLTMMMD")
```

Format

A data frame with 6 rows and 4 columns.

WMO_Region an integer between 1 and 6

Coverage

Number_of_stations

Archive url to spreadsheet

Source

<http://www.bafg.de/GRDC>

`hddtools`*hddtools: Hydrological Data Discovery Tools*

Description

Many governmental bodies and institutions are currently committed to publish open data as the result of a trend of increasing transparency, based on which a wide variety of information produced at public expense is now becoming open and freely available to improve public involvement in the process of decision and policy making. Discovery, access and retrieval of information is, however, not always a simple task. Especially when access to data APIs is not allowed, downloading a meta-data catalogue, selecting the information needed, requesting datasets, de-compression, conversion, manual filtering and parsing can become rather tedious. The R package hddtools is an open source project, designed to make all the above operations more efficient by means of reusable functions.

The package facilitate access to various online data sources such as:

- **KGClimateClass** (<http://koeppen-geiger.vu-wien.ac.at/>): The Koppen Climate Classification map is used for classifying the world's climates based on the annual and monthly averages of temperature and precipitation
- **GRDC** (http://www.bafg.de/GRDC/EN/Home/homepage_node.html): The Global Runoff Data Centre (GRDC) provides datasets for all the major rivers in the world

- **Data60UK** (<http://tdwg.catchment.org/datasets.html>): The Data60UK initiative collated datasets of areal precipitation and streamflow discharge across 61 gauging sites in England and Wales (UK).
- **MOPEX** (<http://tdwg.catchment.org/datasets.html>): This dataset contains historical hydrometeorological data and river basin characteristics for hundreds of river basins in the US.
- **SEPA** (<http://apps.sepa.org.uk/waterlevels/>): The Scottish Environment Protection Agency (SEPA) provides river level data for hundreds of gauging stations in the UK.

This package complements R's growing functionality in environmental web technologies by bridging the gap between data providers and data consumers. It is designed to be an initial building block of scientific workflows for linking data and models in a seamless fashion.

References

Vitolo C, Buytaert W, 2014, HDDTOOLS: an R package serving Hydrological Data Discovery Tools, AGU Fall Meeting, 15-19 December 2014, San Francisco, USA.

KGClimateClass	<i>Function to identify the updated Koppen-Greiger climate zone (on a 0.1 x 0.1 degrees resolution map).</i>
----------------	--

Description

Given a bounding box, the function identifies the overlapping climate zones.

Usage

```
KGClimateClass(areaBox = NULL, updatedBy = "Peel", verbose = FALSE)
```

Arguments

areaBox	bounding box, a list made of 4 elements: minimum longitude (lonMin), minimum latitude (latMin), maximum longitude (lonMax), maximum latitude (latMax)
updatedBy	this can either be "Kottek" or "Peel"
verbose	if TRUE more info are printed on the screen

Value

List of overlapping climate zones.

Author(s)

Claudia Vitolo

References

Kottek et al. (2006): <http://koeppen-geiger.vu-wien.ac.at/>. Peel et al. (2007): <http://people.eng.unimelb.edu.au/mpeel/koppen.html>.

Examples

```
## Not run:  
# Define a bounding box  
areaBox <- raster::extent(-3.82, -3.63, 52.41, 52.52)  
# Get climate classes  
KGClimateClass(areaBox = areaBox)  
  
## End(Not run)
```

MOPEXcatalogue

Data set: The MOPEX Catalogue

Description

The MOPEX catalogue

Usage

```
data("MOPEXcatalogue")
```

Format

A data frame with 431 stations (rows) and 12 metadata fields (columns).

stationID Station id number.
longitude
latitude
elevation
V5
datestart
dateend
V8
V9
state
V11
basin String describing the river's name.

Source

ftp://hydrology.nws.noaa.gov/pub/gcip/mopex/US_Data/

 SEPAcatalogue

Data set: The SEPA Catalogue

Description

The SEPA catalogue

Usage

```
data("SEPAcatalogue")
```

Format

A data frame with 830 observations on the following 8 variables.

idNRFA National River Flow Archive id number.

aspxpage Environment Agency gauges id.

stationId SEPA station id.

River String describing the river's name.

Location String describing the location.

GridRef British National Grid Reference.

Operator The operator's name.

CatchmentAreaKm2 Area of the catchment.

Source

<http://pennine.ddns.me.uk/riverlevels/ConciseList.html>

 tsData60UK

Interface for the Data60UK database of Daily Time Series

Description

This function extract the dataset containing daily rainfall and streamflow discharge at one of the Data60UK locations.

Usage

```
tsData60UK(stationID, plotOption = FALSE, twindow = NULL)
```

Arguments

stationID hydrometric reference number (string)

plotOption boolean to define whether to plot the results. By default this is set to FALSE.

twindow is a vector of dates and times for which the data should be retrieved

Value

The function returns a data frame containing 2 time series (as zoo objects): "P" (precipitation) and "Q" (discharge).

Author(s)

Claudia Vitolo

Examples

```
## Not run:
stationID <- catalogueData60UK()$stationID[1]
Morwick <- tsData60UK(stationID = stationID)
Morwick <- tsData60UK(stationID = stationID, plotOption = TRUE)

## End(Not run)
```

 tsGRDC

Interface for the Global Runoff Data Centre database of Monthly Time Series

Description

This function interfaces the Global Runoff Data Centre monthly mean daily discharges database.

Usage

```
tsGRDC(stationID, plotOption = FALSE)
```

Arguments

stationID	7 string that identifies a station, GRDC station number is called "grdc no" in the catalogue.
plotOption	boolean to define whether to plot the results. By default this is set to TRUE.

Details

Please note that not all the GRDC stations listed in the catalogue have monthly data available.

Value

The function returns a list of 6 tables:

- **LTVD** This is a table containing seasonal variability of river discharge based on original daily data. It is made of 8 columns:
 - LTV_LMM_Monthly__MM: calendar month

- LTV_LDM_Day__Value: lowest daily discharge value in each calendar month in the given time series, calculated from lowest values of each calendar month in consecutive calendar years.
- LTV_LDM_Day__YYYY_MM_DD: Date of occurrence of lowest daily discharge
- LTV_MDM_Day__MM: calendar month
- LTV_MDM_Day__Value: mean of daily discharge values in each calendar month in the given time series, calculated from monthly means of each calendar month in consecutive calendar years.
- LTV_HDM_Day__MM: calendar month
- LTV_HDM_Day__Value: highest daily discharge value in each calendar month in the given time series, calculated from highest values of each calendar month in consecutive calendar years.
- LTV_HDM_Day__YYYY_MM_DD: Date of occurrence of highest daily discharge
- **LTVM**This is a table containing seasonal variability of river discharge based on monthly data. It is made of 8 columns:
 - LTV_LMM_Monthly__MM: calendar month
 - LTV_LMM_Monthly__Value: lowest monthly discharge value in each calendar month in the given time series, calculated from lowest values of each calendar month in consecutive calendar years.
 - LTV_LMM_Monthly__YYYY_MM_DD: Date of occurrence of lowest monthly discharge
 - LTV_MMM_Month__MM: calendar month
 - LTV_MMM_Month__Value: mean of monthly discharge values in each calendar month in the given time series, calculated from values of each calendar month in consecutive calendar years..
 - LTV_HMM_Month__MM: calendar month
 - LTV_HMM_Month__Value: highest monthly discharge value in each calendar month in the given time series, calculated from highest values of each calendar month in consecutive calendar years.
 - LTV_HMM_Month__YYYY_MM_DD: Date of occurrence of highest monthly discharge
- **PVD**This is a table containing ... It is made of 7 columns:
 - LQ_Day__Value: lowest daily discharge value in the given time series, calculated from lowest values of consecutive calendar years.
 - LQ_Day__YYYY_MM_DD: Date of occurrence of lowest daily discharge
 - MLQ_Day__Value: mean of lowest daily discharge values in the given time series, calculated from lowest values of consecutive calendar years.
 - MQ_Day__Value: mean of daily discharge values in the given time series, calculated from yearly means of consecutive calendar years.
 - MHQ_Day__Value: mean of highest daily discharge values in the given time series, calculated from highest values of consecutive calendar years.
 - HQ_Day__Value: highest daily discharge value in the given time series, calculated from highest values of consecutive calendar years.
 - HQ_Day__YYYY_MM_DD: Date of occurrence of highest daily discharge
- **PVM**This is a table containing ... It is made of 5 columns:

- LQ_Month__Value: lowest monthly discharge value in the given time series, calculated from lowest yearly values of consecutive calendar years.
- LQ_Month__YYYY_MM: month of first occurrence of lowest monthly discharge
- MQ_Month__Value: mean of monthly discharge values in the given time series, calculated from yearly means of consecutive calendar years.
- HQ_Month__Value: highest monthly discharge value in the given time series, calculated from highest yearly values of consecutive calendar years.
- HQ_Month__YYYY_MM: month of first occurrence of highest monthly discharge
- **YVD**This is a table containing ... It is made of 12 columns:
 - Year_Min_Day__YYYY: calendar year
 - Year_Min_Day__Value: Lowest daily discharge value in the given calendar year, calculated from 12 lowest monthly values in the year in question.
 - Year_Min_Day__YYYY_MM_DD: date of first occurrence
 - Year_Mean_Min_Day__YYYY: calendar year
 - Year_Mean_Min_Day__Value: mean of lowest daily discharge values in the given calendar year, calculated from 12 lowest monthly values in the year in question.
 - Year_Mean_Day__YYYY: calendar year
 - Year_Mean_Day__Value: Mean of daily discharge values in the given calendar year, calculated from 12 monthly means in the year in question.
 - Year_Mean_Max_Day__YYYY: calendar year
 - Year_Mean_Max_Day__Value: mean of highest daily discharge values in the given calendar year, calculated from 12 highest monthly values in the year in question.
 - Year_Max_Day__YYYY: calendar year
 - Year_Max_Day__Value: highest daily discharge value in the given calendar year, calculated from 12 highest monthly values in the year in question.
 - Year_Max_Day__YYYY_MM_DD: date of first occurrence
- **YVM**This is a table containing ... It is made of 8 columns:
 - Year_Min_Month__YYYY: calendar year
 - Year_Min_Month__Value: lowest monthly discharge value in the given calendar year, calculated from 12 monthly values in the year in question.
 - Year_Min_Month__YYYY_MM: month of first occurrence
 - Year_Mean_Month__Value: mean of monthly discharge values in the given calendar year, calculated from 12 monthly values in the year in question.
 - Year_Max_Month__YYYY: calendar year
 - Year_Max_Month__Value: highest monthly discharge value in the given calendar year, calculated from 12 monthly values in the year in question.
 - Year_Max_Month__YYYY_MM: month of first occurrence

Author(s)

Claudia Vitolo

Examples

```
## Not run:
Adaitu <- tsGRDC(stationID = "1577602")
Adaitu <- tsGRDC(stationID = catalogueGRDC()$grdc_no[1000],
                 plotOption = TRUE)

## End(Not run)
```

tsMOPEX

Interface for the MOPEX database of Daily Time Series

Description

This function extract the dataset containing daily rainfall and streamflow discharge at one of the MOPEX locations.

Usage

```
tsMOPEX(stationID, plotOption = FALSE, timeExtent = NULL)
```

Arguments

stationID	hydrometric reference number (string)
plotOption	boolean to define whether to plot the results. By default this is set to TRUE.
timeExtent	is a vector of dates and times for which the data should be retrieved

Value

The function returns a data frame containing 2 time series (as zoo objects): "P" (precipitation) and "Q" (discharge).

Author(s)

Claudia Vitolo

Examples

```
## Not run:
stationID <- catalogueMOPEX()$stationID[1]
BroadRiver <- tsMOPEX(stationID = stationID)
BroadRiver <- tsMOPEX(stationID = stationID, plotOption = TRUE)

## End(Not run)
```

`tsSEPA`*Interface for the MOPEX database of Daily Time Series*

Description

This function extract the dataset containing daily rainfall and streamflow discharge at one of the MOPEX locations.

Usage

```
tsSEPA(stationID, plotOption = FALSE, timeExtent = NULL)
```

Arguments

<code>stationID</code>	hydrometric reference number (string)
<code>plotOption</code>	boolean to define whether to plot the results. By default this is set to TRUE.
<code>timeExtent</code>	is a vector of dates and times for which the data should be retrieved

Value

The function returns river level data in a zoo object.

Author(s)

Claudia Vitolo

Examples

```
## Not run:  
sampleID <- catalogueSEPA()$stationId[1]  
sampleTS <- tsSEPA(stationID = sampleID, plotOption = TRUE)  
  
## End(Not run)
```

Index

*Topic **datasets**

Data60UKcatalogue, [8](#)

GRDCcatalogue, [9](#)

grdcLTMM, [11](#)

MOPEXcatalogue, [13](#)

SEPAcatalogue, [14](#)

bboxSpatialPolygon, [2](#)

catalogueData60UK, [3](#)

catalogueGRDC, [4](#)

catalogueMOPEX, [6](#)

catalogueSEPA, [7](#)

Data60UKcatalogue, [8](#)

getContent, [9](#)

GRDCcatalogue, [9](#)

grdcLTMM, [11](#)

hddtools, [11](#)

hddtools-package (hddtools), [11](#)

KGClimateClass, [12](#)

MOPEXcatalogue, [13](#)

SEPAcatalogue, [14](#)

tsData60UK, [14](#)

tsGRDC, [15](#)

tsMOPEX, [18](#)

tsSEPA, [19](#)