

Package ‘xtractomatic’

March 12, 2018

Version 3.4.2

Title Accessing Environmental Data from ERD's ERDDAP Server

Description Contains three functions that access environmental data from ERD's ERDDAP service <<http://coastwatch.pfeg.noaa.gov/erddap>>. The `xtracto()` function extracts data along a trajectory for a given "radius" around the point. The `xtracto_3D()` function extracts data in a box. The `xtractogon()` function extracts data in a polygon. There are also two helper functions to obtain information about available data, and plotting functions to plot the results.

Depends R (>= 3.3.0)

License CC0

LazyData true

URL <https://github.com/rmendels/xtractomatic>

BugReports <http://www.github.com/rmendels/xtractomatic/issues>

Imports dplyr, httr, ncd4, readr, sp, stats, utils

Suggests DT, ggplot2, knitr, lubridate, mapdata, reshape2, rmarkdown, webshot

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Roy Mendelssohn [aut, cre],
Cindy Bessey [ctb],
Dave Foley [ctb]

Maintainer Roy Mendelssohn <roy.mendelssohn@noaa.gov>

Repository CRAN

Date/Publication 2018-03-12 21:18:48 UTC

R topics documented:

colors	2
getInfo	2
Marlintag38606	4
mbnms	4
searchData	5
xtracto	6
xtractogon	7
xtractomatic	8
xtracto_3D	10

Index	11
--------------	-----------

colors	<i>cmocean colors The cmocean color palette by Kristen Thyng as of late 2017</i>
--------	--

Description

str(colors) List of 18 \$ algae \$ amp \$ balance \$ curl \$ deep \$ delta \$ dense \$ gray \$ haline \$ ice \$ matter \$ oxy \$ phase \$ solar \$ speed \$ tempo \$ thermal \$ turbid

Usage

colors

Format

An object of class list of length 18.

getInfo	<i>Extract dataset information for a given dtype name or number</i>
---------	---

Description

getInfo displays the dataset information for a given dtype or dtype name

Usage

getInfo(dtype)

Arguments

dtype - character string or integer (1-138) for given dataset

Value

prints out dataset information

Details

`getInfo` gives the dataset information for the given dataset. This includes: `dtype`, `datasetname`, `longname`, `varname`. It will return the following information about any dataset that matches the string:

- `dtype`
- `datasetname`
- `longname`
- `varname`
- `hasAlt`
- `latSouth`
- `lon360`
- `minLongitude`
- `maxLongitude`
- `longitudeSpacing`
- `minLatitude`
- `maxLatitude`
- `latitudeSpacing`
- `minAltitude`
- `maxAltitude`
- `minTime`
- `maxTime`
- `timeSpacing`
- `infoUrl`

Examples

```
getInfo('atsstamday')
```

 Marlntag38606

Marlin Tag Data

Description

Telemetry data of a blue marlin tagged in the Pacific Ocean in 2003

Usage

Marlntag38606

Format

A data frame with 152 obs. of 7 variables:

date time of observation given as YYYY-MM-DD

lon longitude of observation

lat latitude of observation

lowLon low error on longitude

highLon high error on longitude

lowLat low error on latitude

highLat high error on latitude

Source

Dr. Mike Musyl, Pelagic Research Group LLC

 mbnms

MBNMS Boundaries

Description

A dataset containing the latitudes and longitudes of a polygon that define boundaries of the Monterey Bay National Marine Sanctuary.

Usage

mbnms

Format

A data frame with 6666 obs. of 2 variables:

Longitude Longitudes of a boundary polygon

Latitude Latitudes of a boundary polygon

Source

http://sanctuaries.noaa.gov/library/imast_gis.html

searchData

Extract dataset information based on a list of character strings

Description

searchData finds the all datasets that contain the supplied string in the given dataset field

Usage

```
searchData(searchList = list(list("varname", "chl")))
```

Arguments

searchList - A list of lists each list will contain the field to search and the search string

Value

dataframe with any matching information

Details

searchData will search for the given string in any of the fields dtypename,datasetname,longname,varname. Over the list of searches provided, the search sequentially refines the search result based on the next list in the list of lists.

It will return the following information about any dataset that matches the string:

- dtypename
- datasetname
- longname
- varname
- hasAlt
- latSouth
- lon360
- minLongitude
- maxLongitude
- longitudeSpacing
- minLatitude
- maxLatitude
- latitudeSpacing
- minAltitude

- maxAltitude
- minTime
- maxTime
- timeSpacing
- infoUrl

Examples

```
list1 <- 'varname:x_wind'
list2 <- 'datasetname:8day'
mylist <- c(list1, list2)
searchResult <- searchData(mylist)
```

xtracto

Extract environmental data along a trajectory using ERDDAP.

Description

xtracto uses the ERD ERDDAP data web service to extract environmental data along a longitude, latitude and time trajectory

Usage

```
xtracto(dtype, xpos, ypos, tpos = NA, xlen = 0, ylen = 0,
        verbose = FALSE)
```

Arguments

- | | |
|---------|---|
| dtype | - number or string identifying the ERDDAP parameter to extract |
| xpos | - a real array with the longitudes of the trajectory (in decimal degrees East, either 0-360 or -180 to 180) |
| ypos | - a real array with the latitudes of the trajectory (in decimal degrees N; -90 to 90) |
| tpos | - character array with the times of the trajectory in "YYYY-MM-DD". Default is NA for no time. |
| xlen | - optional real array defining the longitude box around the given point (xlen/2 around the point). Default 0. |
| ylen | - optional real array defining the latitude box around the given point (ylen/2 around the point). Default 0. |
| verbose | - optional logical for verbose download out, default FALSE |

Value

A dataframe containing:

- column 1 = mean of data within search radius
- column 2 = standard deviation of data within search radius
- column 3 = number of points found within search radius
- column 4 = time of returned value
- column 5 = min longitude of call (decimal degrees)
- column 6 = max longitude of call (decimal degrees)
- column 7 = min latitude of call (decimal degrees)
- column 8 = max latitude of call (decimal degrees)
- column 9 = requested time in tag
- column 10 = median of data within search radius
- column 11 = median absolute deviation of data within search radius

Examples

```
xpos <- c(230, 235)
ypos <- c(40, 45)
tpos <- c('2006-01-15', '2006-01-20')
xlen <- 0.025
ylen <- 0.025
extract <- xtracto('erdMbsstd8day', xpos, ypos, tpos = tpos, xlen = xlen, ylen = ylen)
```

xtractogon

Extract environmental data in a polygon using ERDDAP.

Description

xtractogon uses the ERD ERDDAP data web service to extract environmental data inside a polygon defined by vectors of latitudes and longitudes

Usage

```
xtractogon(dtype, xpos, ypos, tpos = NA, verbose = FALSE)
```

Arguments

dtype	- number or string identifying the ERDDAP parameter to extract
xpos	- array giving longitudes (in decimal degrees East, either 0-360 or -180 to 180) of polygon
ypos	- array giving latitudes (in decimal degrees N; -90 to 90) of polygon
tpos	- 2-element array giving min and max time (specify both minimum and maximum dates). For the last available time, use "last". Default NA.
verbose	- logical for verbose download out, default FALSE

Value

structure with data and dimensions

- `extract$data` - the masked data array dimensions (lon,lat,time)
- `extract$varname` - the name of the parameter extracted
- `extract$datasetname` - ERDDAP dataset name
- `extract$longitude` - the longitudes on some scale as request
- `extract$latitude` - the latitudes always going south to north
- `extract$time` - the times of the extracts

Details

`xtractogon` extracts the data from the smallest bounding box that contains the polygon, and then uses the function "point.in.polygon" from the "sp" package to mask out the areas outside of the polygon.

Examples

```
tpos <- c("2014-09-01", "2014-10-01")
xpos <- mbnms$Longitude
ypos <- mbnms$Latitude
sanctchl <- xtractogon('erdVH3chlamday', xpos, ypos, tpos = tpos )
```

`xtractomatic`

xtractomatic: Routines to simplify data extraction using ERD's ERDDAP web service.

Description

The `xtractomatic` package is a set of routines to simplify accessing data using ERD's ERDDAP data web service. The package contains three main functions and two helper functions.

Main Functions

- `xtracto` - Extracts an environmental variable along a track defined by its longitude, latitude and time.
- `xtracto_3D` - Extracts an environmental variable in a 3D (longitude,latitude, time) bounding box
- `xtractogon` - Extracts an environmental variable in a spatial polygon through time.

Plotting Functions

- `plotTrack` - Plots the results of a call to `'xtracto()'`
- `plotBBBox` - Plots the results of a call to `'xtracto_3D()'` and `'xtractogon()'`

Helper Functions

- `searchData` - Searches to see if given string is contained in the `datasetname`, `varname`, or `dtypename`. See `?searchData`.
- `getInfo` - Returns information about a given environmental variable. See `?getInfo`.

@section Details: When the `xtractomatic` package is loaded (`"library(xtractomatic)"`) a data structure called `erddapStruct` is automatically loaded into memory, and is explicitly used in `searchData` and `getInfo`, as well implicitly in the other functions. Not all ERDDAP variables are accessed in the routines, and this structure defines information about the datasets, including:

- `dtypename`
- `datasetname`
- `longname`
- `varname`
- `hasAlt`
- `latSouth`
- `lon360`
- `minLongitude`
- `maxLongitude`
- `longitudeSpacing`
- `minLatitude`
- `maxLatitude`
- `latitudeSpacing`
- `minAltitude`
- `maxAltitude`
- `minTime`
- `maxTime`
- `timeSpacing`
- `infoUrl`

Besides the terse help documents, more detail in using the functions are given in the included vignette "Usingxtractomatic". The datasets used in the vignette are included in the "data" directory.

xtracto_3D	<i>Extract environmental data in a 3-D bounding box using ERDDAP.</i>
------------	---

Description

xtracto_3D uses the ERD ERDDAP data web service to extract environmental data in a given longitude, latitude and time bounding box

Usage

```
xtracto_3D(dtype, xpos, ypos, tpos = NA, verbose = FALSE)
```

Arguments

dtype	-	number or string identifying the ERDDAP parameter to extract
xpos	-	2-element array giving min and max longitude (in decimal degrees East, either 0-360 or -180 to 180)
ypos	-	2-element array giving min and max latitude (in decimal degrees N; -90 to 90)
tpos	-	2-element array giving min and max time (specify both minimum and maximum dates). For the last available time, use "last". Default NA.
verbose	-	logical for verbose download output, default FALSE

Value

structure with data and dimensions:

- extract\$data - the data array dimensions (lon,lat,time)
- extract\$varname - the name of the parameter extracted
- extract\$datasetname - ERDDAP dataset name
- extract\$longitude - the longitudes on some scale as request
- extract\$latitude - the latitudes always going south to north
- extract\$time - the times of the extracts

Examples

```
xpos <- c(-130., -125.)
ypos <- c(30., 35.)
tpos <- c('2015-01-16', '2015-02-16')
extract <- xtracto_3D('mhsstdmday', xpos, ypos, tpos = tpos)
```

```
xpos <- c(230, 231)
ypos <- c(40, 41)
tpos <- c('2006-05-05', '2006-05-06')
extract <- xtracto_3D('erdMBSstd8day', xpos, ypos, tpos = tpos, verbose=TRUE)
```

Index

*Topic **datasets**

colors, [2](#)

Marlintag38606, [4](#)

mbnms, [4](#)

colors, [2](#)

getInfo, [2](#)

Marlintag38606, [4](#)

mbnms, [4](#)

searchData, [5](#)

xtracto, [6](#)

xtracto_3D, [10](#)

xtractogon, [7](#)

xtractomatic, [8](#)

xtractomatic-package (xtractomatic), [8](#)