Package ‘hydroscoper’

May 14, 2021

Type Package
Title Interface to the Greek National Data Bank for Hydrometeorological Information
Version 1.4.1
Maintainer Konstantinos Vantas <kon.vantas@gmail.com>
Description R interface to the Greek National Data Bank for Hydrological and Meteorological Information. It covers Hydroscope's data sources and provides functions to transliterate, translate and download them into tidy dataframes.
BugReports https://github.com/ropensci/hydroscoper/issues
Depends R (>= 3.4)
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports stringi (>= 1.5), stringr (>= 1.4), tibble(>= 3.1), pingr (>= 2.0), readr (>= 1.4), jsonlite (>= 1.7)
Suggests ggplot2 (>= 3.3), knitr (>= 1.31), rmarkdown (>= 2.7), testthat (>= 3.0)
VignetteBuilder knitr
NeedsCompilation no
Author Konstantinos Vantas [aut, cre]
  <https://orcid.org/0000-0001-6387-8791>), Sharla Gelfand [ctb, rev] (Sharla Gelfand reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/185), Tim Trice [rev] (Tim Trice reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/185)
Repository CRAN
Date/Publication 2021-05-14 16:30:03 UTC
find_stations

R topics documented:

find_stations ........................................... 2
get_data ...................................................... 3
get_tables ..................................................... 4
greece_borders ............................................. 6
hydroscoaper ................................................. 7
hydroscoaper_defunct ....................................... 8
hydro_coords .................................................. 9
hydro_translate ............................................. 9
stations ...................................................... 12
timeseries ...................................................... 13

Description

find_stations returns a tibble with the nearest stations’ distances using a given point’s longitude and latitude values. This function uses the Haversine formula for distance calculation in km.

Usage

find_stations(longitude = 24, latitude = 38)

Arguments

longitudef a numeric value in degrees
latitude a numeric value in degrees

Value

If the given longitude is in [24, 38] and the latitude is in [34, 42] (i.e. are valid values for Greece) returns an ordered tibble with the station_id, name, subdomain and distance values in km. The station’s data that are used come from the ‘stations’ dataset. Otherwise returns an error message.

Examples

# find the five nearest stations to a point near Thessaloniki,
# (lon, lat) = (22.97, 40.60)
head(find_stations(22.97, 40.60), 5)
**get_data**

*Get time series values in a tibble*

**Description**

`get_data` returns a tibble from a Hydroscope's time-series text file.

**Usage**

```r
get_data(subdomain = c("kyy", "ypaat", "emy", "deh"), time_id)
```

**Arguments**

- `subdomain` One of the subdomains of hydroscope.gr
- `time_id` A time series ID

**Value**

If `subdomain` is one of:

- kyy, Ministry of Environment and Energy
- ypaat, Ministry of Rural Development and Food
- deh, Greek Public Power Corporation
- emy, National Meteorological Service

and `time_id` exists in that `subdomain`, returns a tibble with the time series values. Otherwise returns an error message.

The dataframe columns are:

- **date** The time series Dates (POSIXct)
- **value** The time series values (numeric)
- **comment** Comments about the values (character)

**Note**

Data are not available freely in the sub-domains: "deh" (Greek Public Power Corporation) and "emy" (National Meteorological Service).

**Author(s)**

Konstantinos Vantas, <kon.vantas@gmail.com>

**References**

Stations' data are retrieved from the Hydroscope's databases:

- Ministry of Environment, Energy and Climate Change.
- Ministry of Rural Development and Food.
### Examples

```r
# Not run:
# get time series 912 from the Greek Ministry of Environment and Energy
time_series <- get_data("kyy", 912)
```

```r
# End(Not run)
```

---

### Description

A family of functions that return a tibble from a specific database from Hydroscope using the Enhydris API. `get_database` returns a named list of Tibbles using all the family's functions.

### Usage

- `get_stations(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_timeseries(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_instruments(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_water_basins(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_water_divisions(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_political_divisions(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_variables(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_units_of_measurement(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_time_steps(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
- `get_owners(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)`
get_tables

get_instruments_type(
  subdomain = c("kyy", "ypaat", "emy", "deh"),
  translit = TRUE
)

get_station_type(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)

get_database(subdomain = c("kyy", "ypaat", "emy", "deh"), translit = TRUE)

Arguments

subdomain One of the subdomains of Hydroscope in the vector c("kyy","ypaat","emy","deh").
translit Automatically transliterate Greek to Latin.

Value

If subdomain is one of:

- kyy, Ministry of Environment and Energy.
- ypaat, Ministry of Rural Development and Food.
- deh, Greek Public Power Corporation.
- emy, National Meteorological Service.

returns a tibble or a named list with tibbles from the corresponding database. Otherwise returns an error message.

Note

Objects’ IDs are not unique among the different Hydroscope databases. For example, time series’ IDs from http://kyy.hydroscope.gr have same values with time series’ from http://ypaat.hydroscope.gr.
The coordinates of the stations are based on the European Terrestrial Reference System 1989 (ETRS89).

Author(s)

Konstantinos Vantas, <kon.vantas@gmail.com>

References

The data are retrieved from the Hydroscope’s site databases:

- Ministry of Environment, Energy and Climate Change.
- Ministry of Rural Development and Food.
- National Meteorological Service.
- Greek Public Power Corporation.
Examples

```r
## Not run:

# data will be downloaded from Ministry of Environment and Energy (kyy):
subdomain <- "kyy"

# stations
kyy_stations <- get_stations(subdomain)

# time series
kyy_ts <- get_timeseries(subdomain)

# instruments
kyy_inst <- get_instruments(subdomain)

# water basins
kyy_wbas <- get_water_basins(subdomain)

# water divisions
kyy_wdiv <- get_water_divisions(subdomain)

# political divisions
kyy_pol <- get_political_divisions(subdomain)

# variables
kyy_vars <- get_variables(subdomain)

# units of measurement
kyy_units <- get_units_of_measurement(subdomain)

# time steps
kyy_time_steps <- get_time_steps(subdomain)

# owners
kyy_owners <- get_owners(subdomain)

# instruments type
kyy_instr_type <- get_instruments_type(subdomain)

# stations' type
kyy_st_type <- get_station_type(subdomain)

# use all the get_ functions above to create a named list with tibbles
kyy_db <- get_database(subdomain)

## End(Not run)
```

---

**greece_borders**

*Greek borders*
**Description**

The borders of Greece are taken from Geodata.gov.gr. The variables are created using the function `tidy` from the broom package. This data frame was created for use with the `geom_polygon` from `ggplot2` package.

**Usage**

`greece_borders`

**Format**

A tibble with 18,474 rows and 8 variables:

- `long` Longitude in decimal degrees, ETRS89
- `lat` Latitude in decimal degrees, ETRS89
- `order` order, integer
- `hole` hole, boolean
- `piece` piece, integer
- `group` group, numeric

**Source**

Konstantinos Vantas

---

**hydroscoper**

**hydroscoper: Interface to Hydroscope**

**Description**

`hydroscoper` provides an R interface to the Greek National Data Bank for Hydrological and Meteorological Information [http://www.hydroscope.gr](http://www.hydroscope.gr).

`hydroscoper` covers Hydroscope’s data sources using the Enhydris API and provides functions to:

1. Transform the available tables and data sets into tibbles.
2. Transliterate the Greek Unicode names to Latin.
3. Translate various Greek terms to English.

**Enhydris API**

The Enhydris database is implemented in PostgreSQL. Details can be found [here](http://www.hydroscope.gr).
Data Sources

The data are retrieved from the Hydroscope’s databases:

- Ministry of Environment, Energy and Climate Change.
- Ministry of Rural Development and Food.
- National Meteorological Service.
- Greek Public Power Corporation.

Author(s)

Maintainer: Konstantinos Vantas <kon.vantas@gmail.com> [ORCID]

Other contributors:

- Sharla Gelfand (Sharla Gelfand reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/185) [contributor, reviewer]
- Tim Trice (Tim Trice reviewed the package for rOpenSci, see https://github.com/ropensci/onboarding/issues/185) [reviewer]

See Also

Useful links:

- https://github.com/ropensci/hydroscoper
- https://docs.ropensci.org/hydroscoper/
- Report bugs at https://github.com/ropensci/hydroscoper/issues

---

hydroscoper_defunct  Defunct functions in hydroscoper

Description

These functions are no longer available in hydroscoper.

Usage

get_coords(...)

Arguments

...  Defunct function’s parameters

Details

Defunct functions:

- **get_coords**: This function is defunct. Please use hydro_coords to convert Hydroscope’s points raw format to a tidy data frame.
hydro_coords

Convert coordinates from Hydroscope’s points to a tibble

Description

hydro_coords returns a tibble with the stations’ longitude and latitude using as input the variable point from get_stations function.

Usage

hydro_coords(x)

Arguments

x

a string vector with the points retrieved from Hydroscope

Value

a tibble with the longitude and latitude values.

Author(s)

Konstantinos Vantas, <kon.vantas@gmail.com>

Examples

## Not run:
# get stations from the Greek Ministry of Environment and Energy
kyy_stations <- get_stations("kyy")

# create a tibble with stations’ coords
coords <- hydro_coords(kyy_stations$point)

## End(Not run)

hydro_translate

Translate Greek names and terms to English

Description

hydro_translate translates various Hydroscope’s names and terms to English.

Usage

hydro_translate(x, value = c("owner", "variable", "timestep", "division"))
Arguments

- **x** a string vector
- **value** One of the predefined values in c("owner", "variable", "timestep", "division")

Value

If value is one of:

- **owner**, organizations’ names.
- **variable**, hydrometeorological term.
- **timestep**, timestep term.
- **division**, Water Division.

returns a character vector with translations of various hydrometeorological terms or organizations’ names from Greek (with latin characters) to English.

The organizations’ names in **owner** are:

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>min_envir_energy</td>
<td>Ministry of Environment and Energy</td>
</tr>
<tr>
<td>min_agricult</td>
<td>Ministry of Rural Development and Food</td>
</tr>
<tr>
<td>natio_meteo_service</td>
<td>National Meteorological Service</td>
</tr>
<tr>
<td>natio_observ_athens</td>
<td>National Observatory of Athens</td>
</tr>
<tr>
<td>public_power_corp</td>
<td>Public Power Corporation</td>
</tr>
<tr>
<td>natio_argic_resear</td>
<td>National Agricultural Research Foundation</td>
</tr>
<tr>
<td>greek prefectures</td>
<td>Greek Prefectures</td>
</tr>
<tr>
<td>crete_eng_faculty</td>
<td>Technical University of Crete</td>
</tr>
<tr>
<td>crete_natural_museum</td>
<td>Natural History Museum of Crete</td>
</tr>
</tbody>
</table>

The Greek Water Divisions codes in **division** are:

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GR01</td>
<td>Dytike Peloponnesos</td>
</tr>
<tr>
<td>GR02</td>
<td>Boreia Peloponnesos</td>
</tr>
<tr>
<td>GR03</td>
<td>Anatolike Peloponnesos</td>
</tr>
<tr>
<td>GR04</td>
<td>Dytike Sterea Ellada</td>
</tr>
<tr>
<td>GR05</td>
<td>Epeiros</td>
</tr>
<tr>
<td>GR06</td>
<td>Attike</td>
</tr>
<tr>
<td>GR07</td>
<td>Anatolike Sterea Ellada</td>
</tr>
<tr>
<td>GR08</td>
<td>Thessalia</td>
</tr>
<tr>
<td>GR09</td>
<td>Dytike Makedonia</td>
</tr>
<tr>
<td>GR10</td>
<td>Kentrike Makedonia</td>
</tr>
<tr>
<td>GR11</td>
<td>Anatolike Makedonia</td>
</tr>
<tr>
<td>GR12</td>
<td>Thrake</td>
</tr>
<tr>
<td>GR13</td>
<td>Krete</td>
</tr>
</tbody>
</table>
| GR14 | Nesoi Aigaiou
Note

The dictionary used for the Greek to English translation is:

<table>
<thead>
<tr>
<th>Transliterated term</th>
<th>English term</th>
</tr>
</thead>
<tbody>
<tr>
<td>agnosto</td>
<td>unknown</td>
</tr>
<tr>
<td>anemos</td>
<td>wind</td>
</tr>
<tr>
<td>dieuthynse</td>
<td>direction</td>
</tr>
<tr>
<td>parelthon</td>
<td>past</td>
</tr>
<tr>
<td>tachyteta</td>
<td>speed</td>
</tr>
<tr>
<td>mese</td>
<td>average</td>
</tr>
<tr>
<td>brochoptose</td>
<td>precipitation</td>
</tr>
<tr>
<td>diarkeia</td>
<td>duration</td>
</tr>
<tr>
<td>exatmisse</td>
<td>evaporation</td>
</tr>
<tr>
<td>exatmisodiapnoe</td>
<td>evapotranspiration</td>
</tr>
<tr>
<td>thermokrasia</td>
<td>temperature</td>
</tr>
<tr>
<td>edaphous</td>
<td>ground</td>
</tr>
<tr>
<td>bathos</td>
<td>depth</td>
</tr>
<tr>
<td>elachiste</td>
<td>min</td>
</tr>
<tr>
<td>megiste</td>
<td>max</td>
</tr>
<tr>
<td>piiese</td>
<td>pressure</td>
</tr>
<tr>
<td>semeiake</td>
<td>point</td>
</tr>
<tr>
<td>chioni</td>
<td>snow</td>
</tr>
<tr>
<td>ypsometro</td>
<td>elevation</td>
</tr>
<tr>
<td>stathme</td>
<td>level</td>
</tr>
<tr>
<td>plemmyra</td>
<td>flood</td>
</tr>
<tr>
<td>paroche</td>
<td>flow</td>
</tr>
<tr>
<td>broche</td>
<td>precipitation</td>
</tr>
<tr>
<td>katastase</td>
<td>condition</td>
</tr>
<tr>
<td>ektrimemene</td>
<td>estimation</td>
</tr>
<tr>
<td>athroistiko</td>
<td>cumulative</td>
</tr>
<tr>
<td>stereo</td>
<td>sediment</td>
</tr>
<tr>
<td>ygrasia</td>
<td>humidity</td>
</tr>
<tr>
<td>ygro</td>
<td>wet</td>
</tr>
<tr>
<td>apolyte</td>
<td>absolute</td>
</tr>
<tr>
<td>schetike</td>
<td>relative</td>
</tr>
<tr>
<td>asbestio</td>
<td>calcium</td>
</tr>
<tr>
<td>wetu</td>
<td>precipitation</td>
</tr>
<tr>
<td>chionobrochometro</td>
<td>snow_rain_gauge</td>
</tr>
<tr>
<td>xero</td>
<td>dry</td>
</tr>
<tr>
<td>ydrometrese</td>
<td>flow_gauge</td>
</tr>
<tr>
<td>thalasses</td>
<td>sea</td>
</tr>
<tr>
<td>semeio_drosou</td>
<td>dew_point</td>
</tr>
<tr>
<td>oratoteta</td>
<td>visibility</td>
</tr>
<tr>
<td>steria</td>
<td>land</td>
</tr>
<tr>
<td>thalassa</td>
<td>sea</td>
</tr>
<tr>
<td>barometro</td>
<td>barometer</td>
</tr>
<tr>
<td>tase_ydratmon</td>
<td>vapour_pressure</td>
</tr>
<tr>
<td>psychrometro</td>
<td>psychrometer</td>
</tr>
</tbody>
</table>
isodynamo_ypsos  water_equivalent
agogimoteta  conductance
aktinobolia  radiation
anthraka  carbon
dioxeidio  dioxide
ypoloipo  residual
argilio  aluminum
argilos  clay
arseniko  arsenic
pyrihiou  silicon
aera  air
nephokalypse  cloud_cover
nephose  clouds
axiosemeiota  remarkably
nephe  clouds
kairos  weather
diafora  difference
atmosfairiki  atmospheric
stathera  constant
parousa  present
parelthousa  past
kalymeno  cover
el.  min
meg.  max
skleroteta  hardness
eliophaneia  sunshine
eisro_e_se_tamieuteres  inflow_reservoir

Examples

## Not run:

```r
# get data from the Ministry of Environment and Energy
kyy_owners <- get_owners("kyy")
kyy_vars <- get_variables("kyy")
owners_names <- hydro_translate(kyy_owners$name, "owner")
vars <- hydro_translate(kyy_vars$descr, "variable")
```

## End(Not run)

Description

Stations’ data from the Greek National Data Bank for Hydrological and Meteorological Information. This dataset is a comprehensive look-up table with geographical and ownership information of the available stations in all Hydroscope’s databases.
Usage

stations

Format

A tibble with 2,322 rows and 9 variables:

- **station_id**  The station’s ID from the domain’s database
- **name**  The station’s name
- **water_basin**  The station’s Water Basin
- **water_division**  The station’s Water Division
- **owner**  The station’s owner
- **longitude**  The station’s longitude in decimal degrees, ETRS89
- **latitude**  The station’s latitude in decimal degrees, ETRS89
- **altitude**  The station’s altitude, meters above sea level
- **subdomain**  The corresponding Hydroscope’s database

Description

Time series’ data from the Greek National Data Bank for Hydrological and Meteorological Information. This dataset is a comprehensive look-up table of all of the available measurements for a given station in a given Hydroscope’s database, with units of measurement and times of those measurements.

Usage

timeseries

Format

A tibble with 10,804 rows and 9 variables:

- **time_id**  The time series ID
- **station_id**  The corresponding station’s ID
- **variable**  The time series variable type
- **timestep**  The timestep of time series
- **units**  The units of the time series
- **start_date**  The starting date of time series values
- **end_date**  The ending date of time series values
- **subdomain**  The corresponding Hydroscope’s database
Index

* datasets
  * greece_borders, 6
  * stations, 12
  * timeseries, 13
  * _PACKAGE (hydroscoper), 7

find_stations, 2

get_coords, 8
get_coords (hydroscoper_defunct), 8
get_data, 3
get_database (get_tables), 4
get_instruments (get_tables), 4
get_instruments_type (get_tables), 4
get_owners (get_tables), 4
get_political_divisions (get_tables), 4
get_station_type (get_tables), 4
get_stations (get_tables), 4
get_tables, 4
get_time_steps (get_tables), 4
get_timeseries (get_tables), 4
get_units_of_measurement (get_tables), 4
get_variables (get_tables), 4
get_water_basins (get_tables), 4
get_water_divisions (get_tables), 4
greece_borders, 6

hydro_coords, 8, 9
hydro_translate, 9
hydroscoper, 7
hydroscoper-package (hydroscoper), 7
hydroscoper_defunct, 8

stations, 12

timeseries, 13