

Package: climate (via r-universe)

August 28, 2024

Title Interface to Download Meteorological (and Hydrological) Datasets

Version 1.2.0

Description Automatize downloading of meteorological and hydrological data from publicly available repositories: OGIMET (<http://ogimet.com/index.phtml.en>), University of Wyoming - atmospheric vertical profiling data (<http://weather.uwyo.edu/upperair/>), Polish Institute of Meteorology and Water Management - National Research Institute (<https://danepubliczne.imgw.pl>), and National Oceanic & Atmospheric Administration (NOAA). This package also allows for searching geographical coordinates for each observation and calculate distances to the nearest stations.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.0

Depends R (>= 4.0.0)

Imports XML, httr, curl, data.table, stringi

Suggests dplyr, knitr, maps, testthat, tidyr, rmarkdown

URL <https://github.com/bczernecki/climate>

BugReports <https://github.com/bczernecki/climate/issues>

VignetteBuilder knitr

Repository <https://bczernecki.r-universe.dev>

RemoteUrl <https://github.com/bczernecki/climate>

RemoteRef HEAD

RemoteSha 227557f4e66d6e9591650485fd9b8c6d069c9268

Contents

co2_demo	2
hydro_imgw	3
hydro_imgw_annual	4
hydro_imgw_daily	5
hydro_imgw_monthly	6
imgw_hydro_abbrev	7
imgw_hydro_stations	7
imgw_meteo_abbrev	8
imgw_meteo_stations	9
meteo_imgw	9
meteo_imgw_daily	11
meteo_imgw_datastore	12
meteo_imgw_hourly	13
meteo_imgw_monthly	14
meteo_noaa_co2	15
meteo_noaa_hourly	16
meteo_ogimet	17
nearest_stations_imgw	19
nearest_stations_noaa	20
nearest_stations_ogimet	21
profile_demo	22
sounding_wyoming	23
spheroid_dist	25
stations_meteo_imgw_telemetry	25
stations_ogimet	26
test_url	27
Index	28

co2_demo	<i>Exemplary CO2 dataset from Mauna Loa Observatory (NOAA dataset)</i>
----------	------------------------------------------------------------------------

Description

The object contains pre-downloaded CO2 dataset from Mauna Loa observatory The snapshot was taken 2020/05/05.

Usage

```
co2_demo
```

Format

An object of class `data.frame` with 745 rows and 7 columns.

Examples

```
data(co2_demo)
head(co2_demo)
```

hydro_imgw	<i>Hydrological data from IMGW</i>
------------	------------------------------------

Description

Downloading daily, and monthly hydrological data from the measurement stations available in the danepubliczne.imgw.pl collection

Usage

```
hydro_imgw(
  interval,
  year,
  coords = FALSE,
  value = "H",
  station = NULL,
  col_names = "short",
  ...
)
```

Arguments

interval	temporal resolution of the data ("daily", "monthly", or "semiannual_and_annual")
year	vector of years (e.g., 1966:2000)
coords	add coordinates of the stations (logical value TRUE or FALSE)
value	type of data (can be: state - "H" (default), flow - "Q", or temperature - "T")
station	vector of hydrological stations danepubliczne.imgw.pl; can be given as station name with CAPITAL LETTERS (character) It accepts either names (characters in CAPITAL LETTERS) or stations' IDs (numeric)
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
...	other parameters that may be passed to the 'shortening' function that shortens column names

Value

A data.frame with columns describing the hydrological parameters (e.g. flow, water level) where each row represent a measurement, depending on the interval, at a given hour, month or year. If coords = TRUE additional two columns with geographic coordinates are added.

Examples

```
x = hydro_imgw("monthly", year = 1999)
head(x)
```

hydro_imgw_annual	<i>Semi-annual and annual hydrological data</i>
-------------------	-------------------------------------------------

Description

Downloading hydrological data for the semi-annual and annual period available in the danepubliczne.imgw.pl collection

Usage

```
hydro_imgw_annual(
  year,
  coords = FALSE,
  value = "H",
  station = NULL,
  col_names = "short",
  allow_failure = TRUE,
  ...
)
```

Arguments

year	vector of years (e.g., 1966:2000)
coords	add coordinates of the stations (logical value TRUE or FALSE)
value	type of data (can be: state - "H" (default), flow - "Q", or temperature - "T")
station	name or ID of hydrological station(s). It accepts names (characters in CAPITAL LETTERS) or stations' IDs (numeric)
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE
...	other parameters that may be passed to the 'shortening' function that shortens column names

Examples

```
hydro_yearly = hydro_imgw_annual(year = 2000, value = "H", station = "ANNOPOL")
```

hydro_imgw_daily	<i>Daily hydrological data</i>
------------------	--------------------------------

Description

Downloading daily hydrological data from the danepubliczne.imgw.pl collection

Usage

```
hydro_imgw_daily(  
  year,  
  coords = FALSE,  
  station = NULL,  
  col_names = "short",  
  allow_failure = TRUE,  
  ...  
)
```

Arguments

year	vector of years (e.g., 1966:2000)
coords	add coordinates of the stations (logical value TRUE or FALSE)
station	name or ID of hydrological station(s). It accepts names (characters in CAPITAL LETTERS) or stations' IDs (numeric)
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE
...	other parameters that may be passed to the 'shortening' function that shortens column names

Examples

```
daily = hydro_imgw_daily(year = 2000)
```

hydro_imgw_monthly *Monthly hydrological data*

Description

Downloading monthly hydrological data from the danepubliczne.imgw.pl collection

Usage

```
hydro_imgw_monthly(  
  year,  
  coords = FALSE,  
  station = NULL,  
  col_names = "short",  
  allow_failure = TRUE,  
  ...  
)
```

Arguments

year	vector of years (e.g., 1966:2000)
coords	add coordinates of the stations (logical value TRUE or FALSE)
station	name or ID of hydrological station(s). It accepts names (characters in CAPITAL LETTERS) or stations' IDs (numeric)
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE
...	other parameters that may be passed to the 'shortening' function that shortens column names

Examples

```
monthly = hydro_imgw_monthly(year = 2000)
```

imgw_hydro_abbrev	<i>Definitions of hydrological parameters used for shortening column names from the danepubliczne.imgw.pl collection</i>
-------------------	--------------------------------------------------------------------------------------------------------------------------

Description

The object contains 3 columns that are currently used for improving readability of the downloaded dataset: `fullname`, `abbr_eng`, and `fullname_eng`

Usage

```
imgw_hydro_abbrev
```

Format

The data contains a `data.frame` with ca. 20 elements described in three ways:

fullname original column names as downloaded from the repository

abbr_eng shorten column names with abbreviations derived from the most popular scheme used for meteorological parameters

fullname_eng detailed description of downloaded meteorological variables

The object is created mostly to be used altogether with the `hydro_shortening_imgw()` function

Examples

```
data(imgw_hydro_abbrev)
head(imgw_hydro_abbrev)
```

imgw_hydro_stations	<i>Location of the hydrological stations from the danepubliczne.imgw.pl collection</i>
---------------------	----------------------------------------------------------------------------------------

Description

The object contains weather stations coordinates, ID numbers, and elevations

Usage

```
imgw_hydro_stations
```

Format

The data contains a data.frame with 1304 obs. of 3 variables:

id Station ID

X Longitude

Y Latitude

The object is in the geographic coordinates using WGS84 (EPSG:4326).

Examples

```
data(imgw_hydro_stations)
head(imgw_hydro_stations)
```

imgw_meteo_abbrev	<i>Definitions of meteorological parameters used for shortening column names for the meteorological data from the danepubliczne.imgw.pl collection</i>
-------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

Description

The object contains 3 columns that are currently used for improving readability of the downloaded dataset: `fullname`, `abbr_eng`, and `fullname_eng`

Usage

```
imgw_meteo_abbrev
```

Format

The data contains a data.frame with ca. 250 elements described in three ways:

fullname original column names as downloaded from the repository

abbr_eng shorten column names with abbreviations derived from the most popular scheme used for meteorological parameters

fullname_eng detailed description of downloaded meteorological variables

The object is created mostly to be used altogether with the `meteo_shortening_imgw` function

Examples

```
data(imgw_meteo_abbrev)
head(imgw_meteo_abbrev)
```

imgw_meteo_stations	<i>Location of the meteorological stations from the danepubliczne.imgw.pl collection</i>
---------------------	------------------------------------------------------------------------------------------

Description

The object contains weather stations coordinates, ID numbers, and elevations

Usage

```
imgw_meteo_stations
```

Format

The data contains a data.frame with 1998 obs. of 3 variables:

id Station ID

X Longitude

Y Latitude

station Station name

id2 IMGW-PIB ID for station rank

The object is in the geographic coordinates using WGS84 (EPSG:4326).

Examples

```
data(imgw_meteo_stations)
head(imgw_meteo_stations)
```

meteo_imgw	<i>Meteorological data from IMGW</i>
------------	--------------------------------------

Description

Downloading hourly, daily, and monthly meteorological data from the SYNOP / CLIMATE / PRE-CIP stations available in the danepubliczne.imgw.pl collection

Usage

```
meteo_imgw(
  interval,
  rank = "synop",
  year,
  status = FALSE,
  coords = FALSE,
  station = NULL,
  col_names = "short",
  ...
)
```

Arguments

interval	temporal resolution of the data ("hourly", "daily", "monthly")
rank	rank of the stations: "synop" (default), "climate" or "precip"
year	vector of years (e.g., 1966:2000)
status	leave the columns with measurement and observation statuses (default status = FALSE - i.e. the status columns are deleted)
coords	add coordinates of the station (logical value TRUE or FALSE)
station	vector of hydrological stations danepubliczne.imgw.pl can be name of station CAPITAL LETTERS(character). It accepts names (characters in CAPITAL LETTERS) or stations' IDs (numeric)
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
...	other parameters that may be passed to the 'shortening' function that shortens column names

Value

A data.frame with columns describing the meteorological parameters (e.g. temperature, wind speed, precipitation) where each row represent a measurement, depending on the interval, at a given hour, month or year. If coords = TRUE additional two columns with geographic coordinates are added.

Examples

```
x = meteo_imgw("monthly", year = 2018, coords = TRUE)
head(x)
```

meteo_imgw_daily *Daily IMGW meteorological data*

Description

Downloading daily (meteorological) data from the SYNOP / CLIMATE / PRECIP stations available in the danepubliczne.imgw.pl collection

Usage

```
meteo_imgw_daily(
  rank = "synop",
  year,
  status = FALSE,
  coords = FALSE,
  station = NULL,
  col_names = "short",
  allow_failure = TRUE,
  ...
)
```

Arguments

rank	rank of the stations: "synop" (default), "climate", or "precip"
year	vector of years (e.g., 1966:2000)
status	leave the columns with measurement and observation statuses (default status = FALSE - i.e. the status columns are deleted)
coords	add coordinates of the station (logical value TRUE or FALSE)
station	name of meteorological station(s). It accepts names (characters in CAPITAL LETTERS); Stations' IDs (numeric) are no longer valid
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE
...	other parameters that may be passed to the 'shortening' function that shortens column names

Examples

```
daily = meteo_imgw_daily(rank = "climate", year = 2000)
```

meteo_imgw_datastore *IMGW meteorological data from the IMGW datastore repository*

Description

Downloading hourly (meteorological) data from the telemetric stations available in the danepubliczne.imgw.pl/datastore collection since 2008. Most parameters are collected with 10 minutes interval and thus it is recommended to download only the mandatory years, parameters or stations. For example, 1 year of data with all available parameters requires processing around 4GB of uncompressed data.

Usage

```
meteo_imgw_datastore(
  year,
  parameters = NULL,
  stations = NULL,
  coords = TRUE,
  allow_failure = TRUE
)
```

Arguments

year	numeric vector of years to be downloaded (e.g., 2022:2023)
parameters	<ul style="list-style-type: none"> character vector describing which parameters to be downloaded. Default NULL means to download all available. <ol style="list-style-type: none"> "wd" - wind direction (degrees) "t2m" - temperature at 2 metres above ground level (degree Celsius) "t0m" - ground temperature (degree Celsius) "rr_24h" - precipitation totals for last 24 hours (mm) "rr_1h" - precipitation totals for last 1 hour (mm) "rr_10min" - precipitation totals for last 10 minutes (mm) "ws" - wind speed (m/s) "ws_max" - maximum wind speed for last 10 minutes (m/s) "gust" - wind gust (if present) (m/s) "rh" - relative humidity (%) "water_in_snow" - water equivalent of melted snow cover (mm)
stations	<ul style="list-style-type: none"> character vector with station names as visible in the <code>meteo_imgw_telemetry_stations()</code>. Default NULL means to download data for all available stations.
coords	<ul style="list-style-type: none"> logical - whether to append the dataset with station full name, longitude, latitude and altitude. Default: TRUE
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE

Details

Data from the IMGW automated (telemetry) systems are non validated by experts and may contain invalid values.

Examples

```
# download only air temperature for selected 2 stations in 2022-2023:
imgw_telemetry = meteo_imgw_datastore(year = 2022:2023,
                                       parameters = "t2m",
                                       stations = c("HALA GAŚSIENICOWA",
                                                    "DOLINA PIĘCIU STAWÓW"),
                                       coords = TRUE)
```

meteo_imgw_hourly	<i>Hourly IMGW meteorological data</i>
-------------------	----------------------------------------

Description

Downloading hourly (meteorological) data from the SYNOP / CLIMATE / PRECIP stations available in the danepubliczne.imgw.pl collection

Usage

```
meteo_imgw_hourly(
  rank = "synop",
  year,
  status = FALSE,
  coords = FALSE,
  station = NULL,
  col_names = "short",
  allow_failure = TRUE,
  ...
)
```

Arguments

rank	rank of the stations: "synop" (default), "climate", or "precip"
year	vector of years (e.g., 1966:2000)
status	leave the columns with measurement and observation statuses (default status = FALSE - i.e. the status columns are deleted)
coords	add coordinates of the station (logical value TRUE or FALSE)
station	name or ID of meteorological station(s). It accepts names (characters in CAPITAL LETTERS) or stations' IDs (numeric)

col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE
...	other parameters that may be passed to the 'shortening' function that shortens column names

Examples

```
hourly = meteo_imgw_hourly(rank = "climate", year = 1984)
head(hourly)
```

meteo_imgw_monthly	<i>Monthly IMGW meteorological data</i>
--------------------	-----------------------------------------

Description

Downloading monthly (meteorological) data from the SYNOP / CLIMATE / PRECIP stations available in the danepubliczne.imgw.pl collection

Usage

```
meteo_imgw_monthly(
  rank = "synop",
  year,
  status = FALSE,
  coords = FALSE,
  station = NULL,
  col_names = "short",
  allow_failure = TRUE,
  ...
)
```

Arguments

rank	rank of the stations: "synop" (default), "climate", or "precip"
year	vector of years (e.g., 1966:2000)
status	leave the columns with measurement and observation statuses (default status = FALSE - i.e. the status columns are deleted)
coords	add coordinates of the station (logical value TRUE or FALSE)
station	name or ID of meteorological station(s). It accepts names (characters in CAPITAL LETTERS) or stations' IDs (numeric)
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset

allow_failure logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE

... other parameters that may be passed to the 'shortening' function that shortens column names

Examples

```
monthly = meteo_imgw_monthly(rank = "climate", year = 1969)
head(monthly)

# a descriptive (long) column names:
monthly2 = meteo_imgw_monthly(rank = "synop", year = 2018,
                              col_names = "full")
head(monthly2)

# please note that station names may change over time
# and thus 2 names are required in some cases:
# df = meteo_imgw_monthly(rank = 'synop', year = 1991:2000,
#                          coords = TRUE, station = c("POZNAŃ", "POZNAŃ-ŁAWICA"))
```

meteo_noaa_co2

CO2 Mauna Loa (NOAA) dataset

Description

Carbon Dioxide (CO₂) monthly measurements from Mauna Loa observatory. The source file is available at: ftp://aftp.cmdl.noaa.gov/products/trends/co2/co2_mm_mlo.txt with all further details.

Usage

```
meteo_noaa_co2()
```

Details

Data from March 1958 through April 1974 have been obtained by C. David Keeling of the Scripps Institution of Oceanography (SIO) and were obtained from the Scripps website (scrippsco2.ucsd.edu).

The "average" column contains the monthly mean CO₂ mole fraction determined from daily averages. The mole fraction of CO₂, expressed as parts per million (ppm) is the number of molecules of CO₂ in every one million molecules of dried air (water vapor removed). If there are missing days concentrated either early or late in the month, the monthly mean is corrected to the middle of the month using the average seasonal cycle. Missing months are denoted by -99.99. The "interpolated" column includes average values from the preceding column and interpolated values where data are missing. Interpolated values are computed in two steps. First, we compute for each month the average seasonal cycle in a 7-year window around each monthly value. In this way the seasonal cycle is allowed to change slowly over time. We then determine the "trend" value for each month by removing the seasonal cycle; this result is shown in the "trend" column. Trend values

are linearly interpolated for missing months. The interpolated monthly mean is then the sum of the average seasonal cycle value and the trend value for the missing month. NOTE: In general, the data presented for the last year are subject to change, depending on recalibration of the reference gas mixtures used, and other quality control procedures. Occasionally, earlier years may also be changed for the same reasons. Usually these changes are minor. CO₂ expressed as a mole fraction in dry air, micromol/mol, abbreviated as ppm

Examples

```
#co2 = meteo_noaa_co2()
#head(co2)
#plot(co2$yy_d, co2$co2_avg, type='l')
```

meteo_noaa_hourly	<i>Hourly NOAA Integrated Surface Hourly (ISH) meteorological data</i>
-------------------	------------------------------------------------------------------------

Description

Downloading hourly (meteorological) data from the SYNOP stations available in the NOAA ISD collection. Some stations in the dataset are dated back even up to 1900. By default only records that follow FM-12 (SYNOP) convention are processed. Further details available at: <https://www1.ncdc.noaa.gov/pub/data/noaa/re>

Usage

```
meteo_noaa_hourly(
  station = NULL,
  year = 2019,
  fm12 = TRUE,
  allow_failure = TRUE
)
```

Arguments

station	ID of meteorological station(s) (characters). Find your station's ID at: https://www1.ncdc.noaa.gov/pub/d/history.txt
year	vector of years (e.g., 1966:2000)
fm12	use only FM-12 (SYNOP) records (TRUE by default)
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE

Examples

```
# London-Heathrow, United Kingdom
noaa = meteo_noaa_hourly(station = "037720-99999", year = 1949)
```

meteo_ogimet

Scrapping meteorological (Synop) data from the Ogimet webpage

Description

Downloading hourly or daily (meteorological) data from the Synop stations available at <https://www.ogimet.com/>

Usage

```
meteo_ogimet(
  interval,
  date = c(Sys.Date() - 30, Sys.Date()),
  coords = FALSE,
  station,
  precip_split = TRUE,
  allow_failure = TRUE
)
```

Arguments

interval	'daily' or 'hourly' dataset to retrieve - given as character
date	start and finish date (e.g., date = c("2018-05-01", "2018-07-01")) - character or Date class object. If not provided last 30 days are used.
coords	add geographical coordinates of the station (logical value TRUE or FALSE)
station	WMO ID of meteorological station(s). Character or numeric vector
precip_split	whether to split precipitation fields into 6/12/24h
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE numeric fields (logical value TRUE (default) or FALSE); valid only for hourly time step

Value

A data.frame of measured values with columns describing the meteorological parameters (e.g. air temperature, wind speed, cloudines). Depending on the interval, at a given hour or day. Different parameters are returned for daily and hourly datasets.

1. station_ID - WMO station identifier
2. Lon - longitude
3. Lat - latitude
4. Date - date (and time) of observations
5. TC - air temperature at 2 metres above ground level. Values given in Celsius degrees
6. TdC - dew point temperature at 2 metres above ground level. Values given in Celsius degrees
7. TmaxC - maximum air temperature at 2 metres above ground level. Values given in Celsius degrees

8. TminC - minimum air temperature at 2 metres above ground level. Values given in Celsius degrees
9. ddd - wind direction
10. ffkmh - wind speed in km/h
11. Gustkmh - wind gust in km/h
12. P0hpa - air pressure at elevation of the station in hPa
13. PseahPa - sea level pressure in hPa
14. PTnd - pressure tendency in hPa
15. Nt - total cloud cover
16. Nh - cloud cover by high-level cloud fraction
17. HKm - height of cloud base
18. InsoD1 - insolation in hours
19. Viskm - visibility in kilometres
20. Snowcm - depth of snow cover in centimetres
21. pr6 - precipitation totals in 6 hours
22. pr12 - precipitation totals in 12 hours
23. pr24 - precipitation totals in 24 hours
24. TemperatureCAvg - average air temperature at 2 metres above ground level. Values given in Celsius degrees
25. TemperatureCMax - maximum air temperature at 2 metres above ground level. Values given in Celsius degrees
26. TemperatureCMin - minimum air temperature at 2 metres above ground level. Values given in Celsius degrees
27. TdAvgC - average dew point temperature at 2 metres above ground level. Values given in Celsius degrees
28. HrAvg - average relative humidity. Values given in %
29. WindkmhDir - wind direction
30. WindkmhInt - wind speed in km/h
31. WindkmhGust - wind gust in km/h
32. PresslevHp - Sea level pressure in hPa
33. Precmm - precipitation totals in mm
34. TotCLOct - total cloudiness in octants
35. lowCLOct - cloudiness by low level clouds in octants
36. SunD1h - sunshine duration in hours
37. PreselevHp - atmospheric pressure measured at altitude of station in hPa
38. SnowDepcm - depth of snow cover in centimetres

Examples

```
# downloading daily data for New York - La Guardia (last 30 days by default)
new_york = meteo_ogimet(interval = "daily",
                        station = 72503,
                        coords = TRUE)
```

nearest_stations_imgw *List of nearby meteorological or hydrological IMGW-PIB stations in Poland*

Description

Returns a data frame of meteorological or hydrological stations with their coordinates in particular year. The returned object is valid only for a given year and type of stations (e.g. "synop", "climate" or "precip"). If add_map = TRUE additional map of downloaded data is added.

Usage

```
nearest_stations_imgw(
  type = "meteo",
  rank = "synop",
  year = 2018,
  add_map = TRUE,
  point = NULL,
  no_of_stations = 50,
  allow_failure = TRUE,
  ...
)
```

Arguments

type	data name; "meteo" (default), "hydro"
rank	rank of the stations: "synop" (default), "climate", or "precip"; Only valid if type = "meteo"
year	select year for searching nearest station
add_map	logical - whether to draw a map for a returned data frame (requires maps/mapdata packages)
point	a vector of two coordinates (longitude, latitude) for a point we want to find nearest stations to (e.g. c(15, 53)); If not provided calculated as a mean longitude and latitude for the entire dataset
no_of_stations	how many nearest stations will be returned from the given geographical coordinates. 50 used by default
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE

... extra arguments to be provided to the `graphics::plot()` function (only if `add_map = TRUE`)

Value

A data.frame with a list of nearest stations. Each row represents metadata for station which collected measurements in a given year. Particular columns contain stations metadata (e.g. station ID, geographical coordinates, official name, distance in kilometers from a given coordinates).

Examples

```
df = nearest_stations_imgw(type = "meteo",
  rank = "synop",
  year = 2018,
  point = c(17, 52),
  add_map = TRUE,
  no_of_stations = 4)
```

nearest_stations_noaa *List of nearby SYNOP stations for a defined geographical location*

Description

Returns a data frame of meteorological stations with their coordinates and distance from a given location based on the noaa website. The returned list is valid only for a given day.

Usage

```
nearest_stations_noaa(
  country,
  date = Sys.Date(),
  add_map = TRUE,
  point = NULL,
  no_of_stations = 10,
  allow_failure = TRUE
)
```

Arguments

country	country name (e.g., "SRI LANKA"). Single entries allowed only.
date	optionally, a day when measurements were done in all available locations; current Sys.Date used by default
add_map	logical - whether to draw a map for a returned data frame (requires maps/mapdata packages)

point	a vector of two coordinates (longitude, latitude) for a point we want to find nearest stations to (e.g. c(80, 6)). If not provided the query will be based on a mean longitude and latitude among available dataset.
no_of_stations	how many nearest stations will be returned from the given geographical coordinates; default 30
allow_failure	logical - whether to allow or stop on failure. By default set to TRUE. For debugging purposes change to FALSE

Value

A data.frame with number of nearest station according to given point columns describing stations parameters (e.g. ID station, distance from point, geographic coordinates, etc.) where each row represent a measurement, each station which has a measurements on selected date. If add_map = TRUE additional map of downloaded data is added.

Examples

```
nearest_stations_noaa(country = "SRI LANKA",
  point = c(80, 6),
  add_map = TRUE,
  no_of_stations = 10)

uk_stations = nearest_stations_noaa(country = "UNITED KINGDOM", no_of_stations = 100)
```

nearest_stations_ogimet

List of nearby synop stations for a defined geographical location

Description

Returns a data frame of meteorological stations with their coordinates and distance from a given location based on the ogimet webpage. The returned list is valid only for a given day.

Usage

```
nearest_stations_ogimet(
  country = "United Kingdom",
  date = Sys.Date(),
  add_map = FALSE,
  point = c(2, 50),
  no_of_stations = 10,
  allow_failure = TRUE,
  ...
)
```

Arguments

country	country name; for more than two words they need to be separated with a plus character (e.g., "United+Kingdom"). It is possible to provide more than one country combined into a vector
date	optionally, a day when measurements were done in all available locations; current Sys.Date used by default
add_map	logical - whether to draw a map for a returned data frame (requires maps/mapdata packages)
point	a vector of two coordinates (longitude, latitude) for a point we want to find nearest stations to (e.g. c(0, 0))
no_of_stations	how many nearest stations will be returned from the given geographical coordinates
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE
...	extra arguments to be provided to the <code>graphics::plot()</code> function (only if <code>add_map = TRUE</code>)

Value

A data.frame with number of nearest station according to given point columns describing stations parameters (e.g. ID station, distance from point in km, geographic coordinates, etc.). Each row represent a measurement, each station which has a measurements on selected date. If `add_map = TRUE` additional map of downloaded data is added.

Examples

```
nearest_stations_ogimet(country = "Uniced Kingdom",
                        point = c(-2, 50),
                        add_map = TRUE,
                        no_of_stations = 0,
                        allow_failure = TRUE,
                        main = "Meteo stations in UK")
```

 profile_demo

Exemplary sounding profile from University of Wyoming dataset

Description

The object contains pre-downloaded atmospheric (sounding) profile for Łeba, PL rawinsonde station. The measurement was taken 2000/03/23 at 00 UTC.

Usage

```
profile_demo
```

Format

The data contains list of two data.frames as derived using `sounding_wyoming()` function

Examples

```
data(profile_demo)
head(profile_demo)
```

sounding_wyoming	<i>Sounding data</i>
------------------	----------------------

Description

Downloading the measurements of the vertical profile of atmosphere (also known as sounding data). Data can be retrieved using TEMP and BUFR sounding formatting.

Usage

```
sounding_wyoming(
  wmo_id,
  yy,
  mm,
  dd,
  hh,
  min = 0,
  bufr = FALSE,
  allow_failure = TRUE
)
```

Arguments

wmo_id	international WMO station code (World Meteorological Organization ID); For Polish stations: Łeba - 12120, Legionowo - 12374, Wrocław- 12425
yy	year - single number
mm	month - single number denoting month
dd	day - single number denoting day
hh	hour - single number denoting initial hour of sounding; for most stations this measurement is done twice a day (i.e. at 12 and 00 UTC), sporadically 4 times a day
min	minute - single number denoting initial minute of sounding; applies only to BUFR soundings.
bufr	<ul style="list-style-type: none"> • BUFR or TEMP sounding to be decoded. By default TEMP is used. For BUFR soundings use <code>bufr = TRUE</code>
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE

Value

Returns two lists with values described at: weather.uwyo.edu ; The first list contains:

1. PRES - Pressure (hPa)
2. HGHT - Height (metres)
3. TEMP - Temperature (C)
4. DWPT - Dew point (C)
5. RELH - Relative humidity (%)
6. MIXR - Mixing ratio (g/kg)
7. DRCT - Wind direction (deg)
8. SKNT - Wind speed (knots)
9. THTA = (K)
10. THTE = (K)
11. THTV = (K)

The second list contains metadata and calculated thermodynamic / atmospheric instability indices (for TEMP soundings only)

A list of 2 data.frames where first data frame represents parameters of upper parts o with columns describing the meteorological parameters (e.g. temperature, air pressure) where each row represent a measurement, depending on the height. Second data.frame presents a description of the conditions under which the sounding was carried out.

Source

<http://weather.uwyo.edu/upperair/sounding.html>

Examples

```
#####
# download data for Station 45004 starting 1120Z 11 Jul 2021; Kowloon, HONG KONG, CHINA
# using TEMP and BUFR sounding formats
#####
TEMP = sounding_wyoming(wmo_id = 45004, yy = 2021, mm = 07, dd = 17,
                        hh = 12, min = 00)

#head(TEMP[[1]])

BUFR = sounding_wyoming(wmo_id = 45004, yy = 2021, mm = 07, dd = 17,
                        hh = 12, min = 00, bufr = TRUE)

#head(BUFR[[1]])

#####
### example with a random date to download sounding from LEBA, PL station: ###
#####

profile = sounding_wyoming(wmo_id = 12120,
                           yy = sample(2000:2019,1),
```



```
mm = sample(1:12,1),
dd = sample(1:20,1),
hh = 0)
# head(profile)
# plot(profile[[1]]$HGHT, profile[[1]]$PRES, type = 'l')
```

spheroid_dist

Distance between two points on a spheroid

Description

Calculate the distance between two points on the surface of a spheroid using Vincenty's formula. This function can be used when GIS libraries for calculating distance are not available.

Usage

```
spheroid_dist(p1, p2)
```

Arguments

p1 coordinates of the first point in decimal degrees (LON, LAT)
p2 coordinates of the second point in decimal degrees (LON, LAT)

Value

distance between two locations in kilometers

Examples

```
p1 = c(18.633333, 54.366667) # longitude and latitude for Gdansk
p2 = c(17.016667, 54.466667) # longitude and latitude for Slupsk
spheroid_dist(p1, p2)
```

stations_meteo_imgw_telemetry

IMGW telemetry stations

Description

Retrieving current metadata for stations used in the telemetric systems of the IMGW-PIB datastore (danepubliczne.imgw.pl/datastore)

Usage

```
stations_meteo_imgw_telemetry()
```

Value

data table with metadata for over 500 stations. Metadata contains: station ID, station name, river, latitude, longitude, altitude

Examples

```
telemetry_stations = stations_meteo_imgw_telemetry()
```

stations_ogimet	<i>Scrapping a list of meteorological (Synop) stations for a defined country from the Ogimet webpage</i>
-----------------	----------------------------------------------------------------------------------------------------------

Description

Returns a list of meteorological stations with their coordinates from the Ogimet webpage. The returned list is valid only for a given day

Usage

```
stations_ogimet(
  country = "United+Kingdom",
  date = Sys.Date(),
  add_map = FALSE,
  allow_failure = TRUE
)
```

Arguments

country	country name; for more than two words they need to be separated with a plus character (e.g. "United+Kingdom")
date	a day when measurements were done in all available locations
add_map	logical - whether to draw a map with downloaded metadata (requires maps/mapdata packages)
allow_failure	logical - whether to proceed or stop on failure. By default set to TRUE (i.e. don't stop on error). For debugging purposes change to FALSE

Value

A data.frame with columns describing the synoptic stations in selected countries where each row represent a station. If add_map = TRUE additional map of downloaded data is visualized.

Examples

```
stations_ogimet(country = "Australia", add_map = TRUE)
```

test_url

Download file in a graceful way

Description

Function for downloading & testing url/internet connection according to CRAN policy Example solution strongly based on <https://community.rstudio.com/t/internet-resources-should-fail-gracefully/49199/12> as suggested by kvasilopoulos

Usage

```
test_url(link, output, quiet = FALSE)
```

Arguments

link	character vector with URL to check
output	character vector for output file name
quiet	logical vector (TRUE or FALSE) to be passed to curl_download function. FALSE by default

Examples

```
link = "https://www1.ncdc.noaa.gov/pub/data/noaa/2019/123300-99999-2019.gz"  
output = tempfile()  
test_url(link = link, output = output)
```

Index

* abbreviations

imgw_hydro_abbrev, [7](#)
imgw_meteo_abbrev, [8](#)

* datasets

co2_demo, [2](#)
imgw_hydro_abbrev, [7](#)
imgw_hydro_stations, [7](#)
imgw_meteo_abbrev, [8](#)
imgw_meteo_stations, [9](#)
profile_demo, [22](#)

* hydro

imgw_hydro_abbrev, [7](#)

* meteo

co2_demo, [2](#)
imgw_hydro_stations, [7](#)
imgw_meteo_abbrev, [8](#)
imgw_meteo_stations, [9](#)
profile_demo, [22](#)

* shortening

imgw_hydro_abbrev, [7](#)
imgw_meteo_abbrev, [8](#)

co2_demo, [2](#)

graphics::plot(), [20](#), [22](#)

hydro_imgw, [3](#)

hydro_imgw_annual, [4](#)

hydro_imgw_daily, [5](#)

hydro_imgw_monthly, [6](#)

imgw_hydro_abbrev, [7](#)

imgw_hydro_stations, [7](#)

imgw_meteo_abbrev, [8](#)

imgw_meteo_stations, [9](#)

meteo_imgw, [9](#)

meteo_imgw_daily, [11](#)

meteo_imgw_datastore, [12](#)

meteo_imgw_hourly, [13](#)

meteo_imgw_monthly, [14](#)

meteo_noaa_co2, [15](#)

meteo_noaa_hourly, [16](#)

meteo_ogimet, [17](#)

nearest_stations_imgw, [19](#)

nearest_stations_noaa, [20](#)

nearest_stations_ogimet, [21](#)

profile_demo, [22](#)

sounding_wyoming, [23](#)

spheroid_dist, [25](#)

stations_meteo_imgw_telemetry, [25](#)

stations_ogimet, [26](#)

test_url, [27](#)