

Package ‘climate’

July 11, 2026

Title Interface to Download Meteorological (and Hydrological) Datasets

Version 1.4.0

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

Depends R (>= 4.1.0)

Imports archive, curl, data.table, httr, R6, stringi

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

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

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

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

NeedsCompilation no

Author Bartosz Czernecki [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6496-1386>>),
Arkadiusz Głogowski [aut] (ORCID:
<<https://orcid.org/0000-0002-7587-8892>>),
Jakub Nowosad [aut] (ORCID: <<https://orcid.org/0000-0002-1057-3721>>),
IMGW-PIB [ctb] (source of the data)

Maintainer Bartosz Czernecki <nwp@amu.edu.pl>

Repository CRAN

Date/Publication 2026-07-10 23:10:15 UTC

Contents

co2_demo	3
compute_relative_humidity	3
find_all_station_names	4
hydro_imgw	5
hydro_imgw_daily	6
hydro_imgw_datastore	6
hydro_imgw_monthly	8
hydro_shortening_imgw	9
imgw_hydro_abbrev	9
imgw_hydro_stations	10
imgw_meteo_abbrev	11
imgw_meteo_stations	11
meteo_imgw	12
meteo_imgw_daily	13
meteo_imgw_datastore	15
meteo_imgw_hourly	16
meteo_imgw_monthly	17
meteo_noaa_co2	18
meteo_noaa_hourly	19
meteo_ogimet	20
meteo_ogimet_synop	22
meteo_shortening_imgw	25
nearest_stations_imgw	26
nearest_stations_noaa	27
nearest_stations_ogimet	28
ogimet_daily	29
profile_demo	30
sounding_wyoming	31
spheroid_dist	33
stations_hydro_imgw_telemetry	33
stations_meteo_imgw_telemetry	34
stations_ogimet	34
synop_parser	35
test_url	36

Index	38
--------------	-----------

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.

Value

data.frame with historical CO2 concentrations `data(co2_demo)` `head(co2_demo)`

compute_relative_humidity	<i>Compute relative humidity from air temperature and dew-point temperature</i>
---------------------------	---

Description

Uses the August-Roche-Magnus approximation to derive relative humidity from the 2-metre air temperature and dew-point temperature.

Usage

```
compute_relative_humidity(t2m, dpt2m)
```

Arguments

t2m	Numeric vector. Air temperature (2 m) in degrees Celsius.
dpt2m	Numeric vector. Dew-point temperature (2 m) in degrees Celsius. Must be the same length as t2m.

Details

The August-Roche-Magnus approximation is:

$$RH = 100 \times \frac{\exp\left(\frac{17.625 T_d}{243.04 + T_d}\right)}{\exp\left(\frac{17.625 T}{243.04 + T}\right)}$$

where T is the air temperature and T_d is the dew-point temperature, both in degrees Celsius. The coefficients (17.625 and 243.04) follow Alduchov & Eskridge (1996).

Value

Numeric vector of relative humidity values in percent (0-100). Returns NA where either input is NA. Values are not clamped, so rounding errors may produce results marginally outside 0-100.

References

Alduchov, O. A., & Eskridge, R. E. (1996). Improved Magnus form approximation of saturation vapor pressure. *Journal of Applied Meteorology*, 35(4), 601–609.

Examples

```
compute_relative_humidity(t2m = 20, dpt2m = 10) # ~52 %
compute_relative_humidity(t2m = 0, dpt2m = 0)    # 100 %
compute_relative_humidity(t2m = c(20, 15, NA), dpt2m = c(10, 12, 8))
```

```
find_all_station_names
```

Find all variants of station' names

Description

For IMGW-PIB stations different naming were used historically. For example, POZNAN and “POZNAN-ŁAWICA, thus both names should be used when searching for the station. This function finds all variants of station' names status information and expand the created object

Usage

```
find_all_station_names(station_name)
```

Arguments

station_name character vector of station names

Value

character vector of station names with all variants of station's names

Examples

```
{
  find_all_station_names(c("WARSZAWA", "POZNĄN"))
}
```

hydro_imgw

Hydrological data from IMGW

Description

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

Usage

```
hydro_imgw(interval, year, value = "H", station = NULL, ...)
```

Arguments

interval	temporal resolution of the data ("daily" or "monthly")
year	vector of years (e.g., 1966:2000)
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)
...	other parameters that may be passed e.g. 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_daily	<i>Daily hydrological data</i>
------------------	--------------------------------

Description

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

Usage

```
hydro_imgw_daily(year, station = NULL, allow_failure = TRUE, ...)
```

Arguments

year	vector of years (e.g., 1966:2000)
station	name or ID of hydrological station(s). It accepts names (characters in CAPITAL LETTERS) or stations' IDs (numeric)
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

Value

data.frame with historical hydrological data for the daily time interval

Examples

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

hydro_imgw_datastore	<i>IMGW hydrological data from the IMGW datastore repository</i>
----------------------	--

Description

Downloading hourly and sub-hourly (hydrological) 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 2-4GB of uncompressed data.

hydro_imgw_monthly	<i>Monthly hydrological data</i>
--------------------	----------------------------------

Description

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

Usage

```
hydro_imgw_monthly(  
  year,  
  coords = FALSE,  
  station = NULL,  
  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)
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

Value

data.frame with historical hydrological data for the monthly summaries

Examples

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

hydro_shortening_imgw *Shortening column names for hydrological variables*

Description

Shortening column names of hydrological parameters to improve the readability of downloaded dataset from the danepubliczne.imgw.pl collection and removing duplicated column names

Usage

```
hydro_shortening_imgw(data, col_names = "short", remove_duplicates = TRUE)
```

Arguments

data	downloaded dataset with original column names
col_names	three types of column names possible: "short" - default, values with shorten names, "full" - full English description, "polish" - original names in the dataset
remove_duplicates	whether to remove duplicated column names (default TRUE - i.e., columns with duplicated names are deleted)

Value

data.frame with shorten names of hydrological parameters

Examples

```
monthly = data = hydro_imgw("monthly", year = 1969, col_names = "polish")

if (is.data.frame(monthly)) {
  abbr = hydro_shortening_imgw(data = monthly,
    col_names = "full",
    remove_duplicates = TRUE)
  head(abbr)
}
```

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)
```

<code>imgw_hydro_stations</code>	<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

Meteorological data from the IMGW-PIB official repository

Description

Downloading hourly, daily, and monthly meteorological data from the SYNOP / CLIMATE / PRECIP stations, or sub-hourly (10-minute) telemetry data from the automated network, all available in the `danepubliczne.imgw.pl` collection.

Usage

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

Arguments

<code>interval</code>	temporal resolution of the data: "hourly", "daily", or "monthly". Not used when <code>rank = "telemetry"</code> (telemetry data are always at 10-minute intervals). Defaults to <code>NULL</code> , which is only valid together with <code>rank = "telemetry"</code> .
<code>rank</code>	rank of the stations: "synop" (default), "climate", "precip", or "telemetry". Use "telemetry" for the automated IMGW datastore network (data available since 2008).

year	vector of years (e.g., 1966:2000). For rank = "telemetry" all years must be ≥ 2008 .
status	leave the columns with measurement and observation statuses (default FALSE — status columns are deleted). Not used when rank = "telemetry".
coords	add coordinates of the station (logical value TRUE or FALSE). Default FALSE.
station	name of meteorological station(s). For ranks "synop", "climate", "precip": station name(s) in CAPITAL LETTERS. Please note that station names may change over time — sometimes two names are required, e.g. c("POZNAN", "POZNAN-LAWICA"). For rank = "telemetry": station name(s) as listed by stations_meteo_imgw_telemetry(). NULL (default) downloads all available stations.
col_names	column name style: "short" (default), "full" (English descriptions), or "polish" (original dataset names). Not used when rank = "telemetry".
parameters	character vector of parameter codes to download. Only used when rank = "telemetry". NULL (default) downloads all available parameters. Accepted values: "wd", "t2m", "t0m", "rr_24h", "rr_1h", "rr_10min", "ws", "ws_max", "gust", "rh", "water_in_snow".
...	other parameters passed to the column-shortening function. Not used when rank = "telemetry".

Value

A data.frame with meteorological parameters where each row is a measurement. For ranks "synop", "climate", "precip": measurements at a given hour, day, or month, depending on interval. If coords = TRUE two additional coordinate columns are appended. For rank = "telemetry": a data.table with 10-minute interval observations (not expert-validated). If coords = TRUE columns name, lon, lat, and alt are appended.

Examples

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

# Telemetry (10-minute) data from automated stations (available since 2008):
tel = meteo_imgw(rank = "telemetry", year = 2022,
                 parameters = "t2m",
                 station = "HALA GASIENICOWA")
head(tel)
```

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 vector of names (characters in CAPITAL LETTERS); Important: Some stations may have changed names over time in the IMGW-PIB database and thus providing both names is needed (e.g. station = c("POZNAŃ", "POZNAŃ-ŁAWICA", "WARSZAWA", "WARSZAWA-OKĘCIE")). 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

Value

data.frame with a daily meteorological measurements

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	- character vector describing which parameters to be downloaded. Default NULL means to download all available. "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	character vector with station names as visible in the stations_meteo_imgw_telemetry(). Default NULL means to download data for all available stations.
coords	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.

Value

data.table with a raw meteorological measurements in 10-min intervals. Please note that this dataset is not validated by experts and may contain invalid values.

Examples

```
imgw_telemetry = meteo_imgw_datastore(year = 2022:2023,
                                       parameters = "t2m",
                                       stations = c("HALA GAŚIENICOWA",
                                                    "DOLINA 5 STAWÓW"),
                                       coords = TRUE)
```

meteo_imgw_hourly

Hourly IMGW meteorological data

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 of meteorological station(s) (character vector)
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

Value

meteorological data for the hourly time interval

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 of meteorological station(s). It accepts names (characters in CAPITAL LETTERS). Stations' IDs (numeric) are no longer supported. Please note that station names may change over time and thus sometimes 2 names are required in some cases, e.g. c("POZNAN", "POZNAN-LAWICA").
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

Value

meteorological data with monthly summaries

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)
```

meteo_noaa_co2	<i>CO2 Mauna Loa (NOAA) dataset</i>
----------------	-------------------------------------

Description

Carbon Dioxide (CO2) monthly measurements from Mauna Loa observatory. The source file is available at: ftp://ftp.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

Value

Data frame with historical CO₂ concentrations

Examples

```
co2 = meteo_noaa_co2()
head(co2)
```

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://www.ncei.noaa.gov/pub/data/noaa/readme>

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://www.ncei.noaa.gov/pub/data/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

Value

data.frame with historical meteorological data in hourly intervals

Examples

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

meteo_ogimet	<i>Download meteorological (Synop) data from the Ogimet service</i>
--------------	---

Description

Unified entry point for downloading hourly or daily meteorological data from **Ogimet**. Two back-ends are supported:

Usage

```
meteo_ogimet(
  interval = "hourly",
  date = c(Sys.Date() - 30, Sys.Date()),
  station = NULL,
  country_name = NULL,
  source = NULL,
  ...
)
```

Arguments

interval	"hourly" (default) or "daily" — time resolution to retrieve.
date	Length-2 character or Date vector giving the start and end of the requested period, e.g. c("2018-05-01", "2018-07-01"). Defaults to the last 30 days.
station	WMO ID(s) of the station(s) to download. Character or numeric vector. Not required when country_name is provided (SYNOP path only).

country_name	Optional character string. When provided, the SYNOP path downloads all Ogimet stations for the named country in a single request (e.g. "Poland", "Germany"), and station is ignored. Valid only with source = "synop" (or the default hourly path).
source	Character. Backend to use: "synop" (raw SYNOP decoding) or "html" (HTML scraping). When NULL (default) the backend is chosen automatically: "synop" for interval = "hourly", "html" for interval = "daily".
...	Optional named arguments:
	allow_failure Logical. When TRUE (default) network or parsing errors are caught and a message is emitted; when FALSE errors propagate.
	simplified Logical. Applies to source = "synop" only. When TRUE (default) a compact 20-column data.frame is returned (see synop output below). When FALSE the full synop_parser() output is returned with 30+ columns.
	precip_split Logical. Split the precipitation field into separate pr6, pr12, and pr24 columns. Valid only for interval = "hourly" with source = "html"; a warning is emitted otherwise. Default TRUE.
	return_list Logical. Applies to source = "synop" only. When TRUE a named list with elements data (compact 20-column data.frame) and full (30+ column parser output) is returned instead of a data.frame. A warning is emitted when used with source = "html". Default FALSE.

Details

- "synop" (default for hourly): Downloads raw SYNOP messages from the Ogimet getsynop endpoint and decodes them with [synop_parser\(\)](#). Supports station mode (one or more WMO IDs) and/or country mode (country_name). A default output columns are described in the **synop output** section below, but can be enhanced optionally with simplified = FALSE or return_list = TRUE to include more of decoded SYNOP fields.
- "html" (default for daily): Scrapes pre-formatted summary tables from the Ogimet using HTML parsing. Supports station mode only (one or more WMO IDs). Output columns are described in the **html output** section below.

Value

synop output (source = "synop", simplified = TRUE or return_list = TRUE \$data): A data.frame with one row per decoded SYNOP observation and approx. 20 columns: date (POSIXct UTC), station, t2m, dpt2m, rel_hum, tmax, tmin, wd, ws, gust, press, slp, press_tend, precip, Nt, Nh, N_base, insol, visibility, snow.

synop output (source = "synop", simplified = FALSE): A data.frame with 30+ columns from [synop_parser\(\)](#), prefixed by station_id and Date.

html output (source = "html", interval = "hourly"): A data.frame with columns: station_ID, optionally Lon/Lat, Date, TC, TdC, TmaxC, TminC, ddd, ffknh, Gustkmh, P0hPa, PseahPa, PTnd, Nt, Nh, HKm, InsoD1, Viskm, Snowcm, and (when precip_split = TRUE) pr6, pr12, pr24.

html output (source = "html", interval = "daily"): A data.frame with columns: station_ID, optionally Lon/Lat, Date, TemperatureCAvg, TemperatureCMax, TemperatureCMin, TdAvgC, HrAvg,

WindkmhDir, WindkmhInt, WindkmhGust, PresslevHp, PreselevHp, Precmm, SunD1h, SnowDepcm, TotClOct, lowClOct, VisKm.

Returns NULL invisibly on failure when `allow_failure = TRUE`.

Examples

```
# Hourly SYNOP data for Poznan-Lawica (default source = "synop")
pozn_h = meteo_ogimet(interval = "hourly",
                      station = 12330,
                      date = c("2009-12-01", "2009-12-04"))

# Daily HTML summaries for New York - La Guardia (default source = "html")
new_york = meteo_ogimet(interval = "daily", station = 72503)

# Hourly with full parser output as a list
pozn_list = meteo_ogimet(interval = "hourly",
                        station = 12330,
                        date = c("2009-12-01", "2009-12-04"),
                        return_list = TRUE)
head(pozn_list$data) # simplified
head(pozn_list$full) # all parser columns

# Country mode: all Polish stations for one day
germany = meteo_ogimet(interval = "hourly",
                      country_name = "Germany",
                      date = c("2009-12-15", "2009-12-15"))

# Force SYNOP backend for daily data
pozn_d = meteo_ogimet(interval = "daily",
                    station = 12330,
                    date = c("2009-12-01", "2009-12-04"),
                    source = "synop")

# Force HTML backend for hourly data
pozn_h2 = meteo_ogimet(interval = "hourly",
                    station = 12330,
                    date = c("2019-06-01", "2019-06-08"),
                    source = "html")
```

meteo_ogimet_synop	<i>Download and decode raw SYNOP messages from the Ogimet getsynop service</i>
--------------------	--

Description

Downloads raw SYNOP messages from the Ogimet getsynop endpoint and decodes them into a tidy `data.frame` using the `synop_parser()` function. Two retrieval modes are supported:

Usage

```
meteo_ogimet_synop(
  station = NULL,
  date = c(Sys.Date() - 30, Sys.Date()),
  country = NULL,
  country_name = NULL,
  simplified = TRUE,
  allow_failure = TRUE
)
```

Arguments

station	Numeric or character vector of WMO station IDs. Optional when country_name is provided; required otherwise.
date	Character or Date vector of length 2 giving the start and end of the requested period, e.g. c("2009-12-01", "2009-12-04"). Defaults to the last 30 days.
country	Optional; passed to synop_parser() for country-specific precipitation indicator decoding (e.g. "RU"). Single string or NULL (default). This is distinct from country_name.
country_name	Optional character string naming the country whose stations should be downloaded, as recognised by Ogimet (e.g. "Poland", "Germany", "France"). When provided, the state=Ogimet parameter is used and station is ignored. The full date range is fetched in a single request.
simplified	Logical. When TRUE (default) returns a compact data.frame with 20 standardised columns (see Value below). When FALSE the full parser output is returned.
allow_failure	Logical. When TRUE (default) network errors are caught and a message is emitted; when FALSE errors propagate to the caller.

Details

- **Station mode** (station provided): fetches messages for one or more WMO station IDs. URL form: `http://www.ogimet.com/cgi-bin/getsynop?block=<id>&begin=<YYYYMMDDhhmm>&end=<YYYYMMDDhhmm>`
- **Country mode** (country_name provided): fetches messages for all Ogimet stations in a country in a single request. URL form: `http://www.ogimet.com/cgi-bin/getsynop?begin=<YYYYMMDDhhmm>&end=<YYYYMMDDhhmm>`

When both station and country_name are supplied, country_name takes precedence and a warning is issued.

Each line of the response is a comma-separated record: station_id,year,month,day,hour,minute,<SYNOP message>. The SYNOP message is decoded via [synop_parser\(\)](#) with as_data_frame = TRUE.

Value

By default (simplified = TRUE), a compact data.frame with one row per decoded SYNOP observation. Columns:

- date — Observation date-time (POSIXct, UTC).
- station — WMO station identifier (character).


```
head(poland)

# Simplified view
poznan_simple = meteo_ogimet_synop(station = 12330,
                                   date = c("2009-12-01", "2009-12-04"),
                                   simplified = TRUE)

head(poznan_simple)
```

meteo_shortening_imgw *Shortening column names for meteorological variables*

Description

Shortening column names of meteorological parameters to improve the readability of downloaded dataset from the danepubliczne.imgw.pl collection and removing duplicated column names

Usage

```
meteo_shortening_imgw(data, col_names = "short", remove_duplicates = TRUE)
```

Arguments

data	downloaded dataset with original column names
col_names	three types of column names possible: "short" - default, values with shortened names, "full" - full English description, "polish" - original names in the dataset
remove_duplicates	whether to remove duplicated column names (default TRUE - i.e., columns with duplicated names are deleted)

Value

data.frame with modified names of meteorological parameters

Examples

```
monthly = meteo_imgw("monthly", rank = "climate", year = 1969)

abbr = meteo_shortening_imgw(data = monthly,
  col_names = "full",
  remove_duplicates = TRUE)

head(abbr)
```

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 <code>graphics::plot()</code> 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() - 1,
  add_map = FALSE,
  point = c(2, 50),
  no_of_stations = 10,
  allow_failure = TRUE,
  ...
)
```

Arguments

country	country name; 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; Sys.Date used by default
add_map	logical - whether to draw a map for a returned data frame (requires maps/mapdata packages)

<code>point</code>	a vector of two coordinates (longitude, latitude) for a point we want to find nearest stations to (e.g. <code>c(0, 0)</code>)
<code>no_of_stations</code>	how many nearest stations will be returned from the given geographical coordinates
<code>allow_failure</code>	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
<code>...</code>	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 = "United Kingdom",
                        point = c(-2, 50),
                        add_map = TRUE,
                        no_of_stations = 50,
                        allow_failure = TRUE,
                        main = "Meteo stations in UK")
```

<code>ogimet_daily</code>	<i>Scrapping daily meteorological (Synop) data from the Ogimet web-page</i>
---------------------------	---

Description

Downloading daily (meteorological) data from the Synop stations available in the <https://www.ogimet.com/> repository. The data are processed only if temperature or precipitation fields are present.

Usage

```
ogimet_daily(
  date = c(Sys.Date() - 30, Sys.Date()),
  station = NA,
  hour = 6,
  allow_failure = TRUE,
  ...
)
```

Arguments

date	start and finish of date (e.g., date = c("2018-05-01","2018-07-01")). By default last 30 days.
station	WMO ID of meteorological station(s). Character or numeric vector
hour	time for which the daily raport is generated. Set default as hour = 6 (i.e. 6 UTC)
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 that may be passed from wrapper top-level functions

Value

data.frame with historical meteorological data for the daily summaries

Examples

```
# downloading daily summaries for last 30 days. station: New York - La Guardia
new_york = ogimet_daily(station = 72503)
```

profile_demo	<i>Exemplary sounding profile from University of Wyoming dataset</i>
--------------	--

Description

The object contains pre-downloaded atmospheric (sounding) profile for Leba, 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. By default automatic detection of input format is used.

Usage

```
sounding_wyoming(
    wmo_id,
    yy,
    mm,
    dd,
    hh,
    min = 0,
    source = "AUTODETECT",
    allow_failure = TRUE
)
```

Arguments

wmo_id	international WMO station code (World Meteorological Organization ID); For Polish stations: Leba - 12120, Legionowo - 12374, Wrocław - 12425
yy	year - calendar year
mm	month - calendar month
dd	day - calendar day
hh	hour - hour of sounding; for most stations measurements are performed twice a day (i.e. at 12 and 00 UTC), sporadically 4 times a day
min	minute - minute of sounding; Default 00; Other values applies only to BUFR soundings.
source	optional; AUTODETECT, BUFR (raw; most detailed) or TEMP (FM-35) input format. By default "AUTODETECT" is used which means that the function will try to detect the format automatically. If a station use only a specific format that is known you can specify it explicitly.
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. SPED - Wind speed (m/s)
9. THTA - (K)
10. THTE - (K)
11. THTV - (K)

The second list contains metadata for time of observation and station location

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 AUTODETECT, TEMP and BUFR sounding formats

# autodect input format:
sounding_auto = sounding_wyoming(wmo_id = 45004,
                                yy = 2021, mm = 07, dd = 17, hh = 12)

# temp (fm35) input format:
sounding_temp = sounding_wyoming(wmo_id = 45004,
                                yy = 2021, mm = 07, dd = 17, hh = 12,
                                source = "TEMP")

# bufr input format:
sounding_bufr = sounding_wyoming(wmo_id = 45004,
                                yy = 2021, mm = 07, dd = 17, hh = 12, min = 00,
                                source = "BUFR")
```

spheroid_dist	<i>Distance between two points on a spheroid</i>
---------------	--

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

numerical vector with distance between two locations (in kilometers)

Examples

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

stations_hydro_imgw_telemetry	<i>IMGW hydrological telemetry stations</i>
-------------------------------	---

Description

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

Usage

```
stations_hydro_imgw_telemetry()
```

Value

data table with metadata for over 850 stations. Metadata contains: station ID, station name, river, latitude, longitude, altitude, km_of_river

Examples

```
hydro_telemetry_stations = stations_hydro_imgw_telemetry()
```

```
stations_meteo_imgw_telemetry
```

IMGW meteorological telemetry stations

Description

Retrieving current metadata for meteorological 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, year_est, latitude, longitude, altitude

Examples

```
meteo_telemetry_stations = stations_meteo_imgw_telemetry()
```

```
stations_ogimet
```

Scrapping a list of meteorological (Synop) stations for a defined country from the Ogimet webpage

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; Every word must be written with capital letters (e.g. "United Kingdom")
date	a day when measurements were done in all available locations
add_map	logical - whether to draw a map based on downloaded dataset (requires maps package)
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)
```

synop_parser

Parse SYNOP messages into structured lists or a data frame

Description

This function decodes SYNOP FM-12 meteorological messages which are commonly used for reporting weather observations under the regulations of the World Meteorological Organization (WMO). It parses one or more SYNOP messages and returns their structured representation as generated by the SYNOP R6 decoder.

Usage

```
synop_parser(message, country = NULL, simplify = TRUE, as_data_frame = FALSE)
```

Arguments

message	Character vector with SYNOP messages.
country	Optional; A single character value passed to the precipitation indicator decoder to adjust country-specific behaviour (e.g. "RU").
simplify	Logical. If TRUE (default) and a single message is provided, the function returns the decoded list directly instead of a length-one list. Ignored when as_data_frame = TRUE.
as_data_frame	Logical. If TRUE, return a data.frame with one row per message and commonly-used decoded fields as columns. Missing or unparsed fields are filled with NA. Default is FALSE.

Details

Currently, the decoder contains most of the core logic for parsing the main sections of SYNOP messages that are commonly used in atmospheric sciences. However, it does not yet cover all possible SYNOP groups and fields, and some fields may be missing or incomplete.

Value

When `as_data_frame = FALSE` (default): a list of decoded SYNOP messages, or the decoded list directly when `simplify = TRUE` and a single message is supplied. When `as_data_frame = TRUE`: a `data.frame` with one row per message and the following columns (all numeric/character as appropriate, NA when not present in the message): `station_type`, `station_id`, `region`, `obs_day`, `obs_hour`, `wind_unit`, `wind_estimated`, `visibility`, `cloud_cover`, `wind_direction`, `wind_speed`, `air_temperature`, `dewpoint_temperature`, `station_pressure`, `sea_level_pressure`, `pressure_tendency`, `pressure_change`, `precipitation_amount`, `precipitation_time`, `cloud_base_min`, `cloud_base_max`, `low_cloud_type`, `middle_cloud_type`, `high_cloud_type`, `low_cloud_amount`, `maximum_temperature` (Section 3 daily maximum, °C), `minimum_temperature` (Section 3 daily minimum, °C), `gust` (highest gust speed from Section 3 group 910ff/911ff, in the wind unit of the message), `cloudiness_height` (cloud cover in oktas of the highest cloud layer reported in Section 3, i.e. cirrus/cirrocumulus/cirrostratus; NA when absent), `sunshine_duration` (daily sunshine in hours, from Section 3 group 55SSS), `snow_depth` (total snow depth in cm; 0 for trace amounts, NA for non-continuous cover or unmeasurable depth), `snow_depth_state` (descriptive state of ground with snow/ice per WMO code table 0975, e.g. "Even layer of loose dry snow covering ground completely"), `source` (the original SYNOP message string). Row names are sequential integers.

Examples

```
synop_code = "AAXX 01004 88889 12782 61506 10094 20047 30111 40197 53007 60001 81541"
synop_parser(synop_code)
synop_parser(rep(synop_code, 2), simplify = FALSE)
synop_parser(synop_code, as_data_frame = TRUE)
synop_parser(rep(synop_code, 2), as_data_frame = 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 = TRUE)
```

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

Value

No return value, called for side effects

Examples

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

Index

- * **abbreviations**
 - imgw_hydro_abbrev, [9](#)
 - imgw_meteo_abbrev, [11](#)
- * **datasets**
 - co2_demo, [3](#)
 - imgw_hydro_abbrev, [9](#)
 - imgw_hydro_stations, [10](#)
 - imgw_meteo_abbrev, [11](#)
 - imgw_meteo_stations, [11](#)
 - profile_demo, [30](#)
- * **hydro**
 - imgw_hydro_abbrev, [9](#)
- * **meteo**
 - co2_demo, [3](#)
 - imgw_hydro_stations, [10](#)
 - imgw_meteo_abbrev, [11](#)
 - imgw_meteo_stations, [11](#)
 - profile_demo, [30](#)
- * **shortening**
 - imgw_hydro_abbrev, [9](#)
 - imgw_meteo_abbrev, [11](#)
- co2_demo, [3](#)
- compute_relative_humidity, [3](#)
- compute_relative_humidity(), [24](#)
- find_all_station_names, [4](#)
- graphics::plot(), [26](#), [29](#)
- hydro_imgw, [5](#)
- hydro_imgw_daily, [6](#)
- hydro_imgw_datastore, [6](#)
- hydro_imgw_monthly, [8](#)
- hydro_shortening_imgw, [9](#)
- imgw_hydro_abbrev, [9](#)
- imgw_hydro_stations, [10](#)
- imgw_meteo_abbrev, [11](#)
- imgw_meteo_stations, [11](#)
- meteo_imgw, [12](#)
- meteo_imgw_daily, [13](#)
- meteo_imgw_datastore, [15](#)
- meteo_imgw_hourly, [16](#)
- meteo_imgw_monthly, [17](#)
- meteo_noaa_co2, [18](#)
- meteo_noaa_hourly, [19](#)
- meteo_ogimet, [20](#)
- meteo_ogimet_synop, [22](#)
- meteo_shortening_imgw, [25](#)
- nearest_stations_imgw, [26](#)
- nearest_stations_noaa, [27](#)
- nearest_stations_ogimet, [28](#)
- ogimet_daily, [29](#)
- profile_demo, [30](#)
- sounding_wyoming, [31](#)
- spheroid_dist, [33](#)
- stations_hydro_imgw_telemetry, [33](#)
- stations_meteo_imgw_telemetry, [34](#)
- stations_ogimet, [34](#)
- synop_parser, [35](#)
- synop_parser(), [21–24](#)
- test_url, [36](#)