Package ‘ARPALData’

October 12, 2022

Type Package

Title Retrieving, Managing and Analysing Air Quality and Weather Data for Lombardy (Italy) using ARPA Lombardia Open Database

Version 1.2.3

Description Contains functions for retrieving, managing and analysing air quality and weather data from Regione Lombardia open database (<https://www.dati.lombardia.it/>). Data are collected by ARPA Lombardia (Lombardia Environmental Protection Agency), Italy, through its ground monitoring network. See the webpage <https://www.arpalombardia.it/Pages/ARPA_Home_Page.aspx> for further information on ARPA Lombardia's activities and history. Data quality (e.g. missing values, exported values, graphical mapping) has been checked involving members of the ARPA Lombardia's office for air quality control. The package makes available observations since 1989 (for weather) and 1996 (for air quality) and are updated with daily frequency by the regional agency.

License GPL (>= 2)

Imports RSocrata, tidyverse, tidyselect, utils, dplyr, lubridate, magrittr, rlang, readr, stringi, stringr, tm, tidyrr, purrr, eurostat, sf, ggplot2, tibble, data.table, aweek, parallel, doParallel, moments, mondate

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Paolo Maranzano [aut, cre, cph]
(https://orcid.org/0000-0002-9228-2759), Andrea Algieri [aut, cph]

Maintainer Paolo Maranzano <pmaranzano.ricercastatistica@gmail.com>

Repository CRAN

Date/Publication 2022-03-07 20:20:04 UTC

R topics documented:

ARPALData .................................................. 2
ARPALdf_Summary ....................................... 2
**ARPALdf_Summary**

ARPALdf_Summary_map .................................................. 4
get_ARPA_Lombardia_AQ_data ........................................... 5
get_ARPA_Lombardia_AQ_municipal_data ................................. 6
get_ARPA_Lombardia_AQ_municipal_registry ............................. 8
get_ARPA_Lombardia_AQ_registry ........................................ 9
get_ARPA_Lombardia_W_data ............................................. 9
get_ARPA_Lombardia_W_registry .......................................... 11
get_ARPA_Lombardia_zoning ............................................. 12
get_Lombardia_geospatial ................................................ 13
is_ARPALdf ................................................................. 13
is_ARPALdf_AQ .............................................................. 14
is_ARPALdf_AQ_mun ......................................................... 15
is_ARPALdf_W ............................................................... 15
map_Lombardia_stations .................................................. 16
registry_KNN_dist ........................................................ 17
Time_aggregate ............................................................. 18

**Description**

Contains functions for downloading and managing air quality and weather data from Regione Lombardia open database. Data are collected by ARPA Lombardia (Lombardia Environmental Protection Agency), Italy.

**Author(s)**

Paolo Maranzano <pmaranzano.ricercastatistica@gmail.com>

**ARPALdf_Summary**

*Summary statistics for a data frame of class 'ARPALdf'*

**Description**

`ARPALdf_Summary` returns many descriptive statistics summarizing the data contained in a data frame of class `ARPALdf`. Statistics are calculated at overall level (full sample), by station ID and by year. For each variable are reported the basic positioning indices (min, max, mean, median, quantile) and variability indices (range, standard deviation). Other reported statistics are the linear correlation (Pearson) by station and some graphical representation of the distribution (kernel density plot, histogram, Hampel filter and boxplot). In addition, the function returns useful data-quality information: gap length (i.e. number of missing observations for each variable by station and by year).
ARPALdf_Summary

Usage

ARPALdf_Summary(
  Data,
  by_IDStat = 1,
  by_Year = 1,
  gap_length = 1,
  correlation = 1,
  histogram = 0,
  density = 0,
  outlier = 0,
  verbose = T
)

Arguments

Data: Dataset of class 'ARPALdf' containing the data to be summarised.
by_IDStat: Logic value (0 or 1). Use 1 to compute summary statistics by Station ID. Default is 1.
by_Year: Logic value (0 or 1). Use 1 to compute summary statistics by year. Default is 1.
gap_length: Logic value (0 or 1). Use 1 to compute summary statistics for the gap length of each variable. Default is 1.
correlation: Logic value (0 or 1). Use 1 to compute linear correlation of available variables. Default is 1.
histogram: Logic value (0 or 1). Use 1 to plot the histogram of each variable. Default is 0.
density: Logic value (0 or 1). Use 1 to plot the kernel density plot of each variable. Default is 0.
outlier: Logic value (0 or 1). Use 1 to analyse extreme values of each variable (boxplot and Hampel filter). Default is 0.
verbose: Logic value (T or F). Toggle warnings and messages. If 'verbose=T' (default) the function prints on the screen some messages describing the progress of the tasks. If 'verbose=F' any message about the progression is suppressed.

Value

A list of data.frames containing summary descriptive statistics for a data frame of class 'ARPALdf'. Summary statistics are computed for the overall sample (Descr), by Station ID (Descr_by_IDStat) and by year (Descr_by_Year). Available statistics are: number of NAs, number of negative values, minimum, mean, maximum and standard deviation.

Examples

## Download daily air quality data from all the stations for year 2020
d <- get_ARPA_Lombardia_AQ_data(ID_station = NULL, Year = 2020, Frequency = "daily")
## Summarising observed data
sum_stats <- ARPALdf_Summary(Data = d)
ARPALdf_Summary_map

Generate a map of summary statistics for a given ARPALdf data.frame

Description

`ARPALdf_Summary_map` represents on a map (polygon of Lombardy) the data contained in a data frame of class `ARPALdf` containing the values or the descriptive statistics by station. Data can be either a ARPALdf of observed data (from `get_ARPA_Lombardia_xxx` commands) and an ARPALdf obtained as summary descriptive statistic (from `ARPALdf_Summary` command).

Usage

```
ARPALdf_Summary_map(
  Data,  
  Title_main,  
  Title_legend = "Variable",  
  Variable,  
  prov_line_type = 1,  
  prov_line_size = 1,  
  col_scale = c("#00FF00", "#FFFF00", "#FF0000"),  
  val_midpoint = NULL,  
  xlab = "Longitude",  
  ylab = "Latitude"
)
```

Arguments

- **Data**: Dataset of class 'ARPALdf' containing the values or the descriptive statistics to plot on the map. Data can be either a ARPALdf of observed data (from `get_ARPA_Lombardia_xxx` commands) and an ARPALdf obtained as summary descriptive statistic (from 'ARPALdf_Summary' command).
- **Title_main**: Title of the plot.
- **Title_legend**: Title for the legend
- **Variable**: Summary variable to represent
- **prov_line_type**: Linetype for Lombardy provinces. Default is 1.
- **prov_line_size**: Size of the line for Lombardy provinces. Default is 1.
- **col_scale**: Vector indicating the minimum, the middle and the average point colors. Default is c("green","yellow","red").
- **val_midpoint**: Numeric. Value associated to the middle-point scale color. Default is NULL (midpoint is set equal to the average of the variable to represent).
- **xlab**: x-axis label. Default is 'Longitude'.
- **ylab**: y-axis label. Default is 'Latitude'.
get_ARPA_Lombardia_AQ_data

Value

A map of selected stations across the Lombardy region

Examples

```r
## Download daily air quality data from all the stations for year 2020
d <- get_ARPA_Lombardia_AQ_data(ID_station = NULL, Year = 2020, Frequency = "daily")
## Summarising observed data
s <- ARPALdf_Summary(Data = d)
## Mapping of the average NO2 in 2020 at several stations
ARPALdf_Summary_map(Data = s$Descr_by_IDStat$Mean_by_stat,
                      Title_main = "Mean NO2 by station in 2020", Variable = "NO2")
```

get_ARPA_Lombardia_AQ_data

Download air quality data from ARPA Lombardia website

Description

`get_ARPA_Lombardia_AQ_data` returns observed air quality measurements collected by ARPA Lombardia ground detection system for Lombardy region in Northern Italy. Available airborne pollutant concentrations are: NO2, NOx, PM10, PM2.5, Ozone, Arsenic, Benzene, Benzo-a-pirene, Ammonia, Sulfur Dioxide, Black Carbon, CO, Nickel, Cadmium and Lead. Data are available from 1996 and are updated up to the current date. For more information about the municipal data visit the section ‘Monitoraggio aria’ at the webpage: https://www.dat.ambiente.it/stories/s/auv9-c2sj

Usage

```r
get_ARPA_Lombardia_AQ_data(
  ID_station = NULL,
  Year = 2020,
  Frequency = "hourly",
  Var_vec = NULL,
  Fns_vec = NULL,
  by_sensor = F,
  parallel = T,
  verbose = T
)
```

Arguments

- `ID_station`: Numeric value. ID of the station to consider. Using `ID_station = NULL`, all the available stations are selected. Default is `ID_station = NULL`. 
Year Numeric vector. Year(s) of interest. Default is Year = 2020. Specifying more than one year the code works in parallel computing (half of the available cores) using parLapply() function.

Frequency Temporal aggregation frequency. It can be "hourly", "daily", "weekly", "monthly" or "yearly". Default is Frequency = "hourly".

Var_vec Character vector of variables to aggregate. If NULL (default) all the variables are averaged.

Fns_vec Character vector of aggregation function to apply to the selected variables. Available functions are mean, median, min, max, sum, qPP (PP-th percentile), sd, var, vc (variability coefficient), skew (skewness) and kurt (kurtosis).

by_sensor Logic value (T or F). If 'by_sensor=T', the function returns the observed concentrations by sensor code, while if 'by_sensor=F' (default) it returns the observed concentrations by station.

parallel Logic value (T or F). If 'parallel=T' (default), data downloading is performed using parallel computing (socketing), while if 'parallel=F' (default) the download is performed serially. 'parallel=T' works only when 'Year' is a vector with multiple values, i.e. for a single year the serial computing is performed.

verbose Logic value (T or F). Toggle warnings and messages. If 'verbose=T' (default) the function prints on the screen some messages describing the progress of the tasks. If 'verbose=F' any message about the progression is suppressed.

Value
A data frame of class 'data.frame' and 'ARPALdf'. The object is fully compatible with Tidyverse.

Examples

```r
## Download hourly air quality data for 2020 at station 501.
get_ARPA_Lombardia_AQ_data(ID_station=501,Year=2020,Frequency="hourly",by_sensor = 0)
## Download monthly data for NOx and NO2 observed during 2020 for all the stations active on the network. For NOx is computed the 25th percentile, while for NO2 is computed the maximum concentration observed.
get_ARPA_Lombardia_AQ_data(ID_station=NULL,Year=2020,Frequency="monthly", Var_vec=c("NOx","NO2"),Fns_vec=c("q25","max"),by_sensor = 0)
## Download hourly air quality data for 2020 at station 501. Data are reported by sensor.
get_ARPA_Lombardia_AQ_data(ID_station=501,Year=2020,by_sensor = TRUE)
```

---

get_ARPA_Lombardia_AQ_municipal_data

Download air quality data at municipal from ARPA Lombardia website
get_ARPA_Lombardia_AQ_municipal_data

Description

‘get_ARPA_Lombardia_AQ_municipal_data’ returns the air quality levels at municipal level estimated by ARPA Lombardia using a physico-chemical model which simulates air quality based on weather and geo-physical variables. For each municipality of Lombardy, ARPA estimates the average (NO2_mean) and maximum daily (NO2_max_day) level of NO2, the daily maximum (Ozone_max_day) and the 8-hours moving window maximum (Ozone_max_8h) of Ozone and the average levels of PM10 (PM10_mean) and PM2.5 (PM2.5_mean). Data are available from 2017 and are updated up to the current date. For more information about the municipal data visit the section 'Stime comunali dell'aria' at the webpage: https://www.datilombardia.it/stories/s/auv9-c2sj

Usage

going_ARPA_Lombardia_AQ_municipal_data(
    ID_station = NULL,
    Year = 2020,
    Frequency = "daily",
    Var_vec = NULL,
    Fns_vec = NULL,
    by_sensor = F,
    parallel = T,
    verbose = T
)

Arguments

ID_station Numeric value. ID of the station to consider. Using ID_station = NULL, all the available stations are selected. Default is ID_station = NULL.

Year Numeric vector. Year(s) of interest. Default is Year = 2020. Specifying more than one year the code works in parallel computing (half of the available cores) using parLapply() function.

Frequency Temporal aggregation frequency. It can be "daily", "weekly", "monthly" or "yearly". Default is Frequency = "daily".

Var_vec Character vector of variables to aggregate. If NULL (default) all the variables are averaged.

Fns_vec Character vector of aggregation function to apply to the selected variables. Available functions are mean, median, min, max, sum, qPP (PP-th percentile), sd, var, vc (variability coefficient), skew (skewness) and kurt (kurtosis).

by_sensor Boolean. If 'by_sensor=1', the function returns the observed concentrations by sensor code, while if 'by_sensor=0' (default) it returns the observed concentrations by station.

parallel Logic value (T or F). If 'parallel=T' (default), data downloading is performed using parallel computing (socketting), while if 'parallel=F' (default) the download is performed serially. 'parallel=T' works only when 'Year' is a vector with multiple values, i.e. for a single year the serial computing is performed.

verbose Logic value (T or F). Toggle warnings and messages. If 'verbose=T' (default) the function prints on the screen some messages describing the progress of the tasks. If 'verbose=F' any message about the progression is suppressed.
get_ARPA_Lombardia_AQ_municipal_registry

Details

More detailed description.

Value

A data frame of class 'data.frame' and 'ARPALdf'. The object is fully compatible with Tidyverse. The column 'NameStation' identifies the name of each municipality. The column 'IDStation' is an ID code (assigned from ARPA) uniquely identifying each municipality.

Examples

```r
## Download daily concentrations at municipal levels observed in 2020
## for all the cities in Lombardy
get_ARPA_Lombardia_AQ_municipal_data(ID_station=NULL, Year=2020,
                                      Frequency="daily")

## Download monthly concentrations of NO2 (average and maximum) observed in 2020
## at city number 100451.
get_ARPA_Lombardia_AQ_municipal_data(ID_station=100451, Year=2020,
                                      Frequency="monthly", Var_vec=c("NO2_mean","NO2_max"),
                                      Fns_vec=c("mean","max"))

## Download daily concentrations observed in 2020 at city number 100451.
## Data are reported by sensor.
get_ARPA_Lombardia_AQ_municipal_data(ID_station=100451, Year=2020, by_sensor = 1)
```
get_ARPA_Lombardia_AQ_registry

Examples

get_ARPA_Lombardia_AQ_municipal_registry()

get_ARPA_Lombardia_AQ_registry

Download metadata (registry) on air quality monitoring stations from ARPA Lombardia website

Description

'get_ARPA_Lombardia_AQ_registry' returns the registry (list) of all the air quality sensors and stations belonging to the ARPA Lombardia network. The information reported are: ID of each sensor and station, geo-location (coordinates in degrees), altitude (mt), starting date and ending date. The column 'NameStation' identifies the name of each station, while 'IDStation' is an ID code (assigned from ARPA) uniquely identifying each station. For more information about the municipal data visit the section 'Monitoraggio aria' at the webpage: https://www.dati.lombardia.it/stories/s/auv9-c2sj

Usage

get_ARPA_Lombardia_AQ_registry()

Value

A data frame of class 'data.frame' and 'ARPALdf'. The object is fully compatible with Tidyverse.

Examples

get_ARPA_Lombardia_AQ_registry()

get_ARPA_Lombardia_W_data

Download weather/meteorological data from ARPA Lombardia website

Description

'get_ARPA_Lombardia_W_data' returns observed air weather measurements collected by ARPA Lombardia ground detection system for Lombardy region in Northern Italy. Available meteorological variables are: temperature (Celsius degrees), rainfall (cm), wind speed (m/s), wind direction (degrees), relative humidity ( Data are available from 1989 and are updated up to the current date. For more information about the municipal data visit the section 'Monitoraggio aria' at the webpage: https://www.dati.lombardia.it/stories/s/auv9-c2sj
get_ARPA_Lombardia_W_data

### Usage

```r
get_ARPA_Lombardia_W_data(
  ID_station = NULL,
  Year = 2020,
  Frequency = "10mins",
  Var_vec = NULL,
  Fns_vec = NULL,
  by_sensor = F,
  parallel = T,
  verbose = T
)
```

### Arguments

- **ID_station**: Numeric value. ID of the station to consider. Using `ID_station = NULL`, all the available stations are selected. Default is `ID_station = NULL`.
- **Year**: Numeric vector. Year(s) of interest. Default is `Year = 2020`. Specifying more than one year the code works in parallel computing (half of the available cores) using `parLapply()` function.
- **Frequency**: Temporal aggregation frequency. It can be "10mins", "hourly", "daily", "weekly", "monthly". Default is `Frequency = "10mins"`.
- **Var_vec**: Character vector of variables to aggregate. If `NULL` (default) all the variables are averaged, except for 'Temperature' and 'Snow_height', which are cumulated.
- **Fns_vec**: Character vector of aggregation function to apply to the selected variables. Available functions are `mean`, `median`, `min`, `max`, `qPP` (PP-th percentile), `sd`, `var`, `vc` (variability coefficient), `skew` (skewness) and `kurt` (kurtosis). Attention: for Wind Speed and Wind Speed Gust only mean, min and max are available; for Wind Direction and Wind Direction Gust only mean is available.
- **by_sensor**: Logic value (T or F). If `by_sensor=T`, the function returns the observed concentrations by sensor code, while if `by_sensor=F` (default) it returns the observed concentrations by station.
- **parallel**: Logic value (T or F). If `parallel=T` (default), data downloading is performed using parallel computing (socketing), while if `parallel=F` (default) the download is performed serially. `parallel=T` works only when `Year` is a vector with multiple values, i.e. for a single year the serial computing is performed.
- **verbose**: Logic value (T or F). Toggle warnings and messages. If `verbose=T` (default) the function prints on the screen some messages describing the progress of the tasks. If `verbose=F` any message about the progression is suppressed.

### Details

More detailed description.

### Value

A data frame of class 'data.frame' and 'ARPALdf'. The object is fully compatible with Tidyverse.
get_ARPA_Lombardia_W_registry

Examples

```r
## Download all the weather measurements at station 100 during 2020. Data have 10 minutes frequency.
get_ARPA_Lombardia_W_data(ID_station = 100, Year = 2020, Frequency = "10mins")
## Download all the weather measurements at station 100 during 2020. Data have 10 minutes frequency.
## Data are reported by sensor.
get_ARPA_Lombardia_W_data(ID_station = 100, Year = 2020, by_sensor = TRUE)
```

---

**Description**

`get_ARPA_Lombardia_W_registry` returns the registry (list) of all the weather sensors and stations belonging to the ARPA Lombardia network. The information reported are: ID of each sensor and station, geo-location (coordinates in degrees), altitude (mt), starting date and ending date. The column 'NameStation' identifies the name of each station, while 'IDStation' is an ID code (assigned from ARPA) uniquely identifying each station. For more information about the municipal data visit the section 'Meteo' at the webpages: https://www.dati.lombardia.it/stories/s/auv9-c2sj and https://www.dati.lombardia.it/Ambiente/Stazioni-Meteorologiche/nf78-nj6b

**Usage**

```r
get_ARPA_Lombardia_W_registry()
```

**Value**

A data frame of class 'data.frame' and 'ARPALdf'. The object is fully compatible with Tidyverse.

**Examples**

```r
get_ARPA_Lombardia_W_registry()
```
get_ARPA_Lombardia_zoning

Download ARPA Lombardia zoning geometries

Description

'get_ARPA_Lombardia_zoning' returns the geometries (polygonal shape file) and a map of the ARPA zoning of Lombardy. The zoning reflects the main orographic characteristics of the territory. Lombardy region is classified into seven type of areas: large urbanised areas, urbanized areas in rural contexts, rural areas, mountainous areas and valley bottom. For more information about the municipal data visit the section 'Zonizzazione ARPA Lombardia' at the webpage: https://www.arpalombardia.it/Pages/Aria/Rete-di-rilevamento/Zonizzazione.aspx

Usage

get_ARPA_Lombardia_zoning(
  plot_map = 1,
  title = "ARPA Lombardia zoning",
  line_type = 1,
  line_size = 1,
  xlab = "Longitude",
  ylab = "Latitude"
)

Arguments

plot_map Logic value (0 or 1). If plot_map = 1, the ARPA Lombardia zoning is represented on a map, if plot_map = 0 only the geometry (polygon shapefile) is stored in the output. Default is plot_map = 1.
title Title of the plot. Default is 'ARPA Lombardia zoning'.
line_type Linetype for the zones’ borders. Default is 1.
line_size Size of the line for the zones. Default is 1.
xlab x-axis label. Default is 'Longitude'.
ylab y-axis label. Default is 'Latitude'.

Value

The function returns an object of class 'sf' containing the polygon borders of the seven zones used by ARPA Lombardia to classify the regional territory. If plot_map = 1, it also returns a map of the zoning.

Examples

zones <- get_ARPA_Lombardia_zoning(plot_map = 1)
DESCRIPTION

'get_Lombardia_geospatial' returns the polygonal (shape file) object containing the geometries of Lombardy. Shapefile are available at different NUTS levels: 'LAU' for the shapefile of municipalities of Lombardy, 'NUTS3' for the shapefile of provinces of Lombardy and 'NUTS2' for the shapefile of Lombardy.

USAGE

get_Lombardia_geospatial(NUTS_level = "LAU")

ARGUMENTS

NUTS_level NUTS level required: use "NUTS2" for regional geometries, "NUTS3" for provincial geometries, or "LAU" for municipal geometries. Default NUTS_level = "LAU".

VALUE

A data frame of class 'data.frame', "sf" and 'ARPALdf'.

EXAMPLES

# shape <- get_Lombardia_geospatial(NUTS_level = "LAU")

DESCRIPTION

'is_ARPALdf' checks if the input object belongs to the class 'ARPALdf'

USAGE

is_ARPALdf(Data)

ARGUMENTS

Data Object to check if the class of a dataframe is 'ARPALdf', i.e. ARPAL dataframe.
Value

The function returns 'True' if the object is of class 'ARPALdf' and it returns 'False' if the object isn't of class 'ARPALdf'.

Examples

d <- get_ARPA_Lombardia_AQ_registry()
is_ARPALdf(d)

---

is_ARPALdf_AQ  
Check if a given object belongs to class 'ARPALdf_AQ'

Description

'is_ARPALdf_AQ' checks if the input object belongs to the class 'ARPALdf_AQ'.

Usage

is_ARPALdf_AQ(Data)

Arguments

Data  
Object to check if the class of a dataframe is 'ARPALdf_AQ', i.e. ARPAL dataframe for air quality data.

Value

The function returns 'True' if the object is of class 'ARPALdf_AQ' and it returns 'False' if the object isn't of class 'ARPALdf_AQ'.

Examples

d <- get_ARPA_Lombardia_AQ_registry()
is_ARPALdf_AQ(d)
is_ARPALdf_AQ_mun

Check if a given object belongs to class 'ARPALdf_AQ_mun'

Description

'is_ARPALdf_AQ_mun' checks if the input object belongs to the class 'ARPALdf_AQ_mun'

Usage

is_ARPALdf_AQ_mun(Data)

Arguments

Data Object to check if the class of a dataframe is 'ARPALdf_AQ_mun', i.e. ARPAL dataframe for air quality data at municipal level (See 'get_ARPA_Lombardia_AQ_municipal_data'. command).

Value

The function returns 'True' if the object is of class 'ARPALdf_AQ_mun' and it returns 'False' if the object isn't of class 'ARPALdf_AQ_mun'

Examples

```r
D <- get_ARPA_Lombardia_AQ_registry()
is_ARPALdf_AQ_mun(D)
```

is_ARPALdf_W

Check if a given object belongs to class 'ARPALdf_W'

Description

'is_ARPALdf_W' checks if the input object belongs to the class 'ARPALdf_W'

Usage

is_ARPALdf_W(Data)

Arguments

Data Object to check if the class of a dataframe is 'ARPALdf_W', i.e. ARPAL dataframe for weather data.
Value

The function returns ‘True’ if the object is of class ‘ARPALdf_W’ and it returns ‘False’ if the object isn’t of class ‘ARPALdf_W’

Examples

```r
d <- get_ARPA_Lombardia_W_registry()
is_ARPALdf_W(d)
```

map_Lombardia_stations

Generate a map of the selected stations

Description

‘get_ARPA_Lombardia_AQ_data’ represents on a map (polygon of Lombardy) the location of the stations contained in a data frame of class ‘ARPALdf’. Data can be either a ARPALdf of observed data (from ‘get_ARPA_Lombardia_xxx’ commands) and an ARPALdf obtained as registry (from ‘get_ARPA_Lombardia_xxx_registry’ command).

Usage

```r
map_Lombardia_stations(
  data,
  title = "Map of ARPA stations in Lombardy",
  prov_line_type = 1,
  prov_line_size = 1,
  col_points = "blue",
  xlab = "Longitude",
  ylab = "Latitude"
)
```

Arguments

data                  Dataset of class ‘ARPALdf’ containing the stations to plot on the map. It can be either a ARPALdf of observed data (from ‘get_ARPA_Lombardia_xxx’ commands) and an ARPALdf obtained as registry (from ‘get_ARPA_Lombardia_xxx_registry’ command).
title                 Title of the plot. Default is ‘Map of ARPA stations in Lombardy’
prov_line_type        Linetype for Lombardy provinces. Default is 1.
prov_line_size        Size of the line for Lombardy provinces. Default is 1.
col_points            Color of the points. Default is ‘blue’.
xlab                  x-axis label. Default is ‘Longitude’.
ylab                  y-axis label. Default is ‘Latitude’.
registry_KNN_dist

Value

A map of selected stations across the Lombardy region

Examples

```r
## Download daily concentrations observed at all the stations in 2020.
d <- get_ARPA_Lombardia_AQ_data(ID_station = NULL, Year = 2020, Frequency = "daily")
## Map the stations included in 'd'
map_Lombardia_stations(data = d, title = "Air quality stations in Lombardy")
```

<table>
<thead>
<tr>
<th>registry_KNN_dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifies the K-nearest-neighbours (stations) to all the monitoring sites included in a given ARPALdf registry data.frame. The neighbours are identified computing the Euclidean distance among the sites’ coordinates.</td>
</tr>
</tbody>
</table>

Description

# For each element included in reg_X, it identifies the k-nearest neighbours locations (among those included in reg_Y) according to an Euclidean distance metric. reg_X and reg_Y must be two 'ARPALdf' objects obtained using get_ARPA_Lombardia_xxx_registry'.

Usage

```r
registry_KNN_dist(reg_X, reg_Y, k = 1)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reg_X</td>
<td>Dataset of class 'ARPALdf' containing the stations list obtained as registry (from 'get_ARPA_Lombardia_xxx_registry' command). The object must contain the following columns: 'IDStation','NameStation','Longitude' and 'Latitude'.</td>
</tr>
<tr>
<td>reg_Y</td>
<td>Dataset of class 'ARPALdf' containing the stations list obtained as registry (from 'get_ARPA_Lombardia_xxx_registry' command). The object must contain the following columns: 'IDStation','NameStation','Longitude' and 'Latitude'.</td>
</tr>
<tr>
<td>k</td>
<td>Integer value. Represents the number of neighbours the user wants to identify.</td>
</tr>
</tbody>
</table>

Value

A data.frame object having the same length of reg_X. For each row (stations in reg_X) it contains the name and the IDStation code for the k-nearest neighbours.
Examples

```r
library(tidyverse)
regAQ <- get_ARPA_Lombardia_AQ_registry()
regAQ <- regAQ %>% filter(Pollutant %in% c("PM10","Ammonia"))
regW <- get_ARPA_Lombardia_W_registry()
registry_KNN_dist(regAQ,regW,k=2)
```

Time_aggregate

### Aggregate ARPALdf data to hourly, daily, weekly, monthly and yearly aggregations

#### Description

`Time_aggregate` returns an ARPALdf object aggregating observations by hourly, daily, weekly, monthly and yearly periods. The function can be applied only to ARPALdf objects. User can indicate specific variables to aggregate and an aggregation function among mean, median, sum (cumulated), min, max and quantile for each variable. It is possible to specify different aggregation functions on the same variable by repeating the name of the variable in Var_vec and specifying the functions in Fns_vec.

#### Usage

```r
Time_aggregate(Dataset, Frequency, Var_vec = NULL, Fns_vec = NULL, verbose = T)
```

#### Arguments

- **Dataset**: ARPALdf dataframe to aggregate.
- **Frequency**: Temporal aggregation frequency. It can be "hourly", "daily", "weekly", "monthly" or "yearly".
- **Var_vec**: Vector of variables to aggregate. If NULL (default) all the variables are averaged, expect for 'Temperature' and 'Snow_height' which are summed.
- **Fns_vec**: Vector of aggregation function to apply to the selected variables. Available functions are mean, median, min, max, sum, qPP (PP-th percentile), sd, var, vc (variability coefficient), skew (skewness) and kurt (kurtosis). Attention: for Wind Speed and Wind Speed Gust only mean, min and max are available; for Wind Direction and Wind Direction Gust only mean is available.
- **verbose**: Logic value (T or F). Toggle warnings and messages. If 'verbose=T' (default) the function prints on the screen some messages describing the progress of the tasks. If 'verbose=F' any message about the progression is suppressed.

#### Value

A data frame
Examples

```r
## Download hourly observed concentrations from all the stations in the network during 2019.
data <- get_ARPA_Lombardia_AQ_data(ID_station=NULL, Year=2020, Frequency="hourly")
## Aggregate all the data to daily frequency
Time_aggregate(Dataset=data, Frequency="daily", Var_vec=NULL, Fns_vec=NULL)
## Aggregate NO2 to weekly maximum concentrations and NOx to weekly minimum concentrations.
Time_aggregate(Dataset=data, Frequency="weekly", Var_vec=c("NO2", "NOx"), Fns_vec=c("max", "min"))
```
Index

ARPALData, 2
ARPALdf_Summary, 2
ARPALdf_Summary_map, 4

get_ARPA_Lombardia_AQ_data, 5
get_ARPA_Lombardia_AQ_municipal_data, 6
get_ARPA_Lombardia_AQ_municipal_registry, 8
get_ARPA_Lombardia_AQ_registry, 9
get_ARPA_Lombardia_W_data, 9
get_ARPA_Lombardia_W_registry, 11
get_ARPA_Lombardia_zoning, 12
get_Lombardia_geospatial, 13

is_ARPALdf, 13
is_ARPALdf_AQ, 14
is_ARPALdf_AQ_mun, 15
is_ARPALdf_W, 15

map_Lombardia_stations, 16

registry_KNN_dist, 17

Time_aggregate, 18