Package 'EEAaq'

August 23, 2023

Title Handle Air Quality Data from the European Environment Agency Data Portal

Version 0.0.3

Description This software downloads and manages air quality data at the European level from the European Environmental Agency (EEA) dataflows (<https://www.eea.europa.eu/data-and-maps/data/aqereporting-9>).

The package allows dynamically mapping the stations, summarising and time aggregating the measurements and building

spatial interpolation maps.

See the webpage <https://www.eea.europa.eu/en> for further information on EEA's activities and history.

Imports

sf,readr,gstat,grDevices,leaflet,tictoc,raster,mondate,aweek,htmlwidgets,dplyr,ggplot2,lubridate,stringr,tibble,tidyr,ggpubr

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 2.10)

LazyData true

License GPL (>= 3)

NeedsCompilation no

Author Agostino Tassan Mazzocco [aut, cre, cph],
Paolo Maranzano [aut, cph] (<https://orcid.org/0000-0002-9228-2759>),
Riccardo Borgoni [aut, cph] (<https://orcid.org/0000-0002-2520-3512>)

Maintainer Agostino Tassan Mazzocco <agostino.tassan@gmail.com>

Repository CRAN

Date/Publication 2023-08-23 15:50:25 UTC

R topics documented:

`code_extr` *Extract numeric NUTS code from the chracter*

Description

This function take as argument the chracter NUTS or LAU code ("NUTS0", "NUTS1", "NUTS2", "NUTS3", "LAU") and mutate it in a numerical hierarchical coding.

Usage

```
code_extr(level)
```

Arguments

`level`	Character value of the NUTS level

Value

integer corresponding to the relative NUTS level or LAU.

`EEAaq_export` *Export and save an* `EEAaq_df` *class object*

Description

`EEAaq_export` saves an `EEAaq_df` class object as a *.csv* or a *.txt* file, and exports the associated shapefile as well.

Usage

```
EEAaq_export(data, filepath, format, shape = FALSE)
```

Arguments

data	an EEAaq_df class object.
filepath	character string giving the file path
format	character string giving the format of the file. It must be one of 'csv' and 'txt'.
shape	logical value (T or F). If TRUE the shapefile associated to the EEAaq_df object given in input is saved in the same directory specified in filepath. If FALSE (the default), only the data frame containing the data is saved.

Value

No return value, called for side effects.

Examples

```
#Download a dataset with the function EEAaq_get_data, which generate an EEAaq_df object.
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU", pollutant = "PM10",
  from = 2021, to = 2021, verbose = TRUE)
temp <- tempdir()
filepath <- paste0(temp, "/data.csv")
EEAaq_export(data = data, filepath = filepath, format = "csv", shape = TRUE)
```

EEAaq_get_data *Download air quality data at european level from the EEA download service*

Description

This function imports air quality datasets at european level, based on the zone, time and pollutant specifications. This function generates an EEAaq_df object, or an EEAaq_df_sfc.

Usage

```
EEAaq_get_data(
  zone_name = NULL,
  NUTS_level = NULL,
  pollutant = NULL,
  from = NULL,
  to = NULL,
  ID = FALSE,
  quadrant = NULL,
  polygon = NULL,
  verbose = TRUE
)
```

Arguments

zone_name	character vector specifying the names of the zones to consider. The reference is the NUTS and LAU nomenclature by Eurostat. See *Details*.
NUTS_level	character that specify the level of NUTS or LAU, to which zone_name belongs. Allowed values are 'NUTS0', 'NUTS1', 'NUTS2', 'NUTS3', 'LAU'. For further information see *Details*.
pollutant	the pollutants for which to download data. It may be: • character vector representing the short names of the pollutants to analyse. The reference is the variable Notation in the dataset pollutants provided by this package. • numeric vector representing the codes of the pollutants to analyse. The reference is the variable Code in the dataset pollutants provided by this package.
from	the starting point of the time window to consider. It may be: • integer representing the year (data are downloaded starting from the first day of the year specified) • character containing a specific day of the year in the format yyyy-mm-dd
to	the ending point of the time window to consider. It may be: • integer representing the year (data are downloaded ending at the last day of the year specified) • character containing a specific day of the year in the format yyyy-mm-dd
ID	logic value (T or F). If TRUE, the character specified in the parameter zone_name is the unique identifier code provided by Eurostat. The reference is the NUTS_ID column from the NUTS dataset or the LAU_ID from the LAU dataset. If FALSE (the default), the character specified in the parameter zone_name is the zone name expressed in latin characters. The reference is the NAME_LATN column from the NUTS dataset and the LAU_NAME column from the LAU dataset.
quadrant	a list of bidimensional numeric vectors containing the coordinates in **WGS84** format. If the list has two elements, the function builds a square using the two coordinates as opposite extremes. If the list contains three or more elements, every point is a vertex of a polygon, in particular the convex hull of the specified points.
polygon	A sfc_POLYGON or sfc_MULTIPOLYGON class object (see https://CRAN.R-project.org/package=sf for further information), specifying the area of interest. The polygon can be imported via shapefile or other formats from the user.
verbose	logic value (T or F). If TRUE (the default) information about the function progress are printed. If FALSE no message is printed.

Details

The NUTS classification (Nomenclature of territorial units for statistics) is a hierarchical system for dividing up the economic territory of the EU and the UK. The levels are defined as follows:

- **NUTS 0**: the whole country

- **NUTS 1**: major socio-economic regions
- **NUTS 2**: basic regions for the application of regional policies
- **NUTS 3**: small regions for specific diagnoses

Further information is available at https://ec.europa.eu/eurostat/web/nuts/background. These LAUs (Local Administrative Units) are the building blocks of the NUTS, and comprise the municipalities and communes of the European Union. For further information see https://ec.europa.eu/eurostat/web/nuts/local-administrative-units. Note that a specific name can be associated with both LAU and NUTS levels. For instance "Milano" is either a city or a NUTS 3 area in Italy. To download data referred to the city, specify `zone_name = "Milano"` and `NUTS_level = "LAU"`, while to download data referred to the province, specify `zone_name = "Milano"` and `NUTS_level = "NUTS3"`. One of `zone_name`, `quadrant` and `polygon` should always be specified by the user.

Value

A data frame of class `EEAaq_df`, if `zone_name` is specified, and of class `EEAaq_df_sfc` if whether the parameter `quadrant` or `polygon` is specified.

Examples

```
library(dplyr)
library(utils)
#Download of the PM10 data in Milan (Lombardia, Italy) during 2023.
EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU",
  pollutant = "PM10", from = 2023, to = 2023, ID = FALSE, verbose = TRUE)

#Alternatively, it is possible to obtain the same result using
#the LAU_ID of Milan and the pollutant code:
EEAaq_get_data(zone_name = "015146", NUTS_level = "LAU",
  pollutant = "PM10", from = 2023, to = 2023, ID = TRUE, verbose = TRUE)

#Another way to get the nitrogen dioxide in Milan during 2023,
#is to extract the geometry from the LAU dataset and use it with the polygon parameter:
temp <- tempfile()
download.file("https://github.com/AgostinoTassanMazzocco/EEAaq/raw/main/LAU.rds",
  temp, quiet = TRUE)
milan <- readRDS(temp) %>%
    filter(LAU_NAME == "Milano") %>%
    pull(geometry)
EEAaq_get_data(pollutant = "PM10", from = 2023, to = 2023, polygon = milan)

#Another option is to choose the exact dates, for example,
#in order to import Milan PM10 january 2023 data,
#it is possible to write:
EEAaq_get_data(polygon = milan, pollutant = "PM10", from = "2023-01-01", to = "2023-01-31")

#Spcifying the parameter quadrant, it is possible to choose
#a specific quadrant on the geographic map.
```

```
#For instance, using the points (11.5, 46.5), (10.5, 46),
#will be imported data about the air quality stations
#located inside the rectangle which has, as opposite vertexes,
#the two selected points. In this case
#PM10 and PM2.5 data are downloaded for the whole 2021:
EEAaq_get_data(pollutant = c("PM10", "PM2.5"),
  quadrant = list(c(11.5, 46.5), c(10.5, 46)),
  from = 2021, to = 2021, verbose = TRUE)
```

EEAaq_get_stations *Download EEA measurement station information dataset*

Description

Download the updated dataset from EEA, containing measurement station information. For further information about the variables see stations.

Usage

```
EEAaq_get_stations(byStation = FALSE, complete = TRUE)
```

Arguments

byStation	Logic value (T or F). If TRUE the dataset is organized by station (one row for each measurement station). If FALSE the dataset is organized by sampling point. Each station have multiple sampling points.
complete	Logic value (T or F). If TRUE, the dataset contains all the variables given by the EEA. If FALSE the dataset contains only a few variables, the most importants. For further details about the variables, see stations.

Value

A tibble containing the stations information. Further details available here stations.

Examples

```
EEAaq_get_stations(byStation = FALSE, complete = TRUE)
```

EEAaq_idw_map *Build a spatial interpolation map based on the Inverse Distance Weighting technique.*

Description

EEAaq_idw_map recieves as input a EEAaq_taggr_df or a EEAaq_taggr_df_sfc class object and produces a spatial interpolation map. Depending on the time frequency of the aggregation, multiple maps are generated, one for each timestamp. It may be exported as pdf, jpeg, png, gif and html.

Usage

```
EEAaq_idw_map(
  data = NULL,
  pollutant = NULL,
  aggr_fun,
  bounds_level = NULL,
  distinct = FALSE,
  gradient = TRUE,
  idp = 2,
  nmax = NULL,
  maxdist = NULL,
  dynamic = FALSE,
  fill_NUTS_level = NULL,
  tile = "Esri.WorldGrayCanvas",
  save = NULL,
  filepath = NULL,
  width = 1280,
  height = 720,
  res = 144,
  delay = 1,
  verbose = TRUE
)
```

Arguments

data	an object of class EEAaq_taggr_df or EEAaq_taggr_df_sfc, which is the output of the EEAaq_time_aggregate function.
pollutant	vector containing the pollutant for which to build the map. It must be one of the pollutants contained in data.
aggr_fun	charachter containing the aggregation function to use for computing the interpolation. It must be one of the statistics contained in data.
bounds_level	character containing the NUTS level or LAU for which draw internal boundaries. Admissible values are 'NUTS0', 'NUTS1', 'NUTS2', 'NUTS3', 'LAU'.

distinct	logic value (T or F). If TRUE, each map generated is printed and saved in distinct pages (for instance if data has a monthly frequency in a yearly time window, 12 distinct plots are generated). If FALSE (the default), the maps are printed in a single page.
gradient	logic value (T or F). If TRUE (the default) the maps generated are colored with a continuous color scale. If FALSE, the color scale is discrete.
idp	numeric value that specify the inverse distance weighting power. For further information see idw.
nmax	numeric value; specify the number of nearest observations that should be used for the inverse distance weighting computing, where nearest is defined in terms of the space of the spatial locations. By default, all observations are used. For further information see idw
maxdist	numeric value; only observations within a distance of maxdist from the prediction location are used for the idw computation. By default, all observations are used. If combined with nmax, both criteria apply.
dynamic	logic value (T or F). If TRUE the function creates a Leaflet map widget using **htmlwidgets** (for further information see leaflet). If FALSE (the default) the maps generated are static.
fill_NUTS_level	character containing the NUTS level or LAU for which to aggregate the idw computing, in order to obtain a uniform coloring inside each area at the specified level. (For instance if fill_NUTS_level = "LAU", each municipality is filled by the mean value of the pollutant concentration, computed by the idw, of each pixel inside the respective municipality). Allowed values are 'NUTS0', 'NUTS1', 'NUTS2', 'NUTS3', 'LAU'.
tile	character representing the name of the provider tile. To see the full list of the providers, run leaflet::providers. For further information see addProviderTiles.
save	character representing in which extension to save the map. Allowed values are 'jpeg', 'png', 'pdf' (if dynamic = FALSE), 'gif' (if dynamic = FALSE & distinct = TRUE), 'html' (if dynamic = TRUE).
filepath	a character string giving the file path.
width, height	the width and the height of the plot, expressed in pixels (by default width = 1280, height = 720). This parameters are available only for save 'jpeg', 'png' and 'gif'. For further information see png or jpeg.
res	the nominal resolution in ppi which will be recorded in the bitmap file, if a positive integer (by default res = 144). This parameter is available only for save 'jpeg', 'png'. For further information see png or jpeg.
delay	numeric value specifying the time to show each image in seconds, when save = "gif".
verbose	logic value (T or F). If TRUE (the default) information about the function progress are printed. If FALSE no message is printed.

Details

EEAaq_idw_map create a spatial interpolation map, based on the Inverse Distance Weighting method (Shepard 1968). This method starts from the available georeferenced data and estimates the value

of the variable in the points where it's unknown as a weighted average of the known values, where weights are given by an inverse function of the distance of every point from the fixed stations. The greater the distance of a point from a station, the smaller the weight assigned to the values of the respective station for the computing of that unknown point. Given the sampling plan s_i for $i = 1, ..., n$, which represent the location of the air quality stations, the pollutant concentration value $Y(s_i) = Y_i$ represents the value of the pollutant concentration detected by the site s_i and u is the point for which the value of the concentration in unknown.

$$\hat{Y}(u) = \sum_{i=1}^{n} Y_i \omega_i(u),$$

where

$$\omega_i(u) = \frac{g(d(s_i, u))}{\sum_{i=1}^{n} g(d(s_i, u))}$$

represent the weights assigned to each location s_i and $d(s_i, u)$ is the distance between u and s_i.

Value

cosa restituisce la funzione

Examples

```
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU",
  pollutant = "PM10", from = 2023, to = 2023, ID = FALSE, verbose = TRUE)

#Monthly aggregation
t_aggr <- EEAaq_time_aggregate(data = data, frequency = "monthly",
                               aggr_fun = c("mean", "min", "max"))

#12 maps are generated, one for each month
EEAaq_idw_map(data = t_aggr, pollutant = "PM10",
aggr_fun = "mean", distinct = TRUE,
gradient = TRUE, idp = 2,
dynamic = FALSE)

#Let's try to change the parameters fill_NUTS_level and dynamic
#Now we are going to use a dataset containing PM10 concentrations
#in Milan province (NUTS 3), during 2022
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "NUTS3",
  pollutant = "PM10", from = 2022, to = 2022)
#yearly aggregation
t_aggr <- EEAaq_time_aggregate(data = data, frequency = "yearly",
  aggr_fun = "mean")
#Let's generate one dynamic map, containing the municipalities inside the Milan province
#filled with the mean concentration value for 2022, computed via idw:
EEAaq_idw_map(data = t_aggr, pollutant = "PM10", aggr_fun = "mean",
  distinct = TRUE, gradient = FALSE, dynamic = TRUE, fill_NUTS_level = "LAU")
```

EEAaq_import *Reverse function of* EEAaq_export. *Reads an* EEAaq_df *object*

Description

Given the file containing the data saved with EEAaq_export, and the associated shapefile, EEAaq_read imports the EEAaq_df class object.

Usage

```
EEAaq_import(file_data, file_shape)
```

Arguments

file_data	file path of the 'csv' or 'txt' file containing the air quality data to import.
file_shape	file path of the shapefile associated to the dataset used in file_data

Value

No return value, called for side effects.

Examples

```
#Download a dataset with the function EEAaq_get_data, which generate an EEAaq_df object.
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU", pollutant = "PM10",
  from = 2021, to = 2021, verbose = TRUE)
temp <- tempdir()
filepath <- paste0(temp, "/data.csv")
#Export the dataset and the associated shape
EEAaq_export(data = data, filepath = filepath, format = "csv", shape = TRUE)
#Import the EEAaq_df object saved in the previous code line
EEAaq_import(file_data = filepath, file_shape = paste0(temp, "/data.shp"))
```

EEAaq_map_stations *Create a map representing the stations based on the data given in input or the information specified in the parameters*

Description

Build static or dynamic maps, representing the location of the stations that detects the specified pollutants. It recieves in input an EEAaq_df or an EEAaq_df_sfc class object, or, alternatively, it's possible to specify the required zones and pollutants with the same nomenclature system of the EEAaq_get_data function.

Usage

```
EEAaq_map_stations(
  data = NULL,
  pollutant = NULL,
  zone_name = NULL,
  NUTS_level = NULL,
  ID = FALSE,
  bounds_level = NULL,
  color = TRUE,
  dynamic = FALSE
)
```

Arguments

data	an EEAaq_df or EEAaq_df_sfc class object, which is the output of the EEAaq_get_data function.
pollutant	character vector containing the short names of the pollutants for which locate the stations.
zone_name	character vector specifying the names of the zones to consider. The reference is the NUTS and LAU nomnclature by Eurostat.
NUTS_level	chracter that specify the level of NUTS or LAU, to which the zone_name belongs.
ID	logic value (T or F). If TRUE the character specified in the parameter zone_name is the unique identifier code provided by Eurostat. If FALSE (the default) it indicates that was specified the full name in latin characters.
bounds_level	character containing the NUTS level or LAU for which draw internal boundaries. Admissible values are "NUTS0", "NUTS1", "NUTS2", "NUTS3", "LAU" and it must be of a lower level then the one specified in the parameter NUTS_level.
color	logical value (T or F). If TRUE (the default) the points are colored based on the pollutant they are able to detect. If FALSE the points have the same color.
dynamic	logical value (T or F). If TRUE the map is interactive and dynamic. If FALSE (the default) the map is static.

Value

A map representing the specified area and the points representing the location of the stations able to detect the specified pollutants.

Examples

```
#Using as example PM data in Lombardia (Italy) during the whole 2022,
#it's possible to map the stations in two ways.
#First of all specifying the zone information:
EEAaq_map_stations(pollutant = c("PM10", "PM2.5"), zone_name = "Lombardia",
                   NUTS_level = "NUTS2", ID = FALSE, color = FALSE)
#In this case each point have the same color.
```

```
#Alternatively, it is possible to use the data already downloaded in the parameter data,
#coloring the points based on the pollutants the respective station detects.
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU",
  pollutant = "PM10", from = 2023, to = 2023, ID = FALSE, verbose = TRUE)
EEAaq_map_stations(data = data, color = TRUE)
```

EEAaq_summary *Generate an* EEAaq_df *data summary*

Description

This function, applied to an EEAaq_df or EEAaq_df_sfc class object, produces a list of data frames, containing relevant information about the data, such as descriptive statistics, missing values statistics, gap length and correlation.

Usage

```
EEAaq_summary(data = NULL, verbose = TRUE)
```

Arguments

data	an EEAaq_df or EEAaq_df_sfc class object, which is the output of the EEAaq_get_data function.
verbose	logic value (T or F). If TRUE (the default) messages about the function progress are printed. If FALSE no message is printed.

Value

The function EEAaq_summary computes and return a list of summary statistics of the dataset given in data. In particular the elements of the list are:

- Summary global missing count, missing rate, negative count, minimum, maximum, mean and standard deviation, organized by pollutant.
- Summary_byStat list of data frames, one for each different station, containing the descriptive statistics (missing count, missing rate, negative count, minimum, maximum, mean and standard deviation), organized by station.
- gap_length one data frame for each pollutant, containing the gap length organized by station.
- Corr_Matrix if data contains more than one pollutant, the correlation matrix between pollutans is provided, organised by station.

Examples

```
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU",
  pollutant = "PM10", from = 2023, to = 2023, ID = FALSE, verbose = TRUE)
EEAaq_summary(data)
```

EEAaq_time_aggregate *Time aggregation of an* EEAaq_df *class object.*

Description

EEAaq_time_aggregate compute a time aggregation of an EEAaq_df or EEAaq_df_sfc class object, based on the specified frequency and the aggregation functions aggr_fun.

Usage

```
EEAaq_time_aggregate(
  data = NULL,
  frequency = "monthly",
  aggr_fun = c("mean", "min", "max")
)
```

Arguments

data	an EEAaq_df or EEAaq_df_sfc class object, which is the output of the EEAaq_get_data function.
frequency	vector containing the time frequency for which to aggregate the data object. Admissible values are 'yearly', 'monthly', 'weekly', 'daily' 'hourly'.
aggr_fun	character vector containing one or more agregation functions. Admissible values are 'mean', 'median', 'min', 'max', 'sd', 'var', 'quantile_pp' (where pp is a number in the range [0,1], representing the required percentile).

Value

A EEAaq_taggr_df or a EEAaq_taggr_df_sfc class object, which is a tibble containing the required time aggregation.

Examples

```
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU",
  pollutant = "PM10", from = 2023, to = 2023, ID = FALSE, verbose = TRUE)
EEAaq_time_aggregate(data = data, frequency = "monthly", aggr_fun = c("mean", "min", "max"))
EEAaq_time_aggregate(data = data, frequency = "yearly", aggr_fun = "mean")
```

is_EEAaq_df *Check if a given object is an* EEAaq_df *class object*

Description

Given an object as input, is_EEAaq_df verify that the given object belongs to the EEAaq_df class.

Usage

```
is_EEAaq_df(data)
```

Arguments

data	the object for which verify the if it belongs to the EEAaq_df class.

Value

logical value (T ot F). If TRUE the object given in input is an EEAaq_df object. If FALSE the object doesn't belong to the EEAaq_df class.

Examples

```
#Download a dataset with the function EEAaq_get_data, which generate an EEAaq_df object.
data <- EEAaq_get_data(zone_name = "Milano", NUTS_level = "LAU", pollutant = "PM10",
  from = 2021, to = 2021, verbose = TRUE)
#Check if the imported object belongs to the EEAaq_df class
is_EEAaq_df(data = data)
```

my_summarise *Aggregate data based on a specific statistic*

Description

Given data and the aggregation function desired, this function compute a time aggregation of the data.

Usage

```
my_summarise(data, fun_aggr)
```

Arguments

data	An EEAaq_df or EEAaq_df_sfc class object, which is the output of the EEAaq_get_data function.
fun_aggr	Vector character containing the aggregation function for which to time aggregate.

Value

A tibble with the required aggregation.

`pollutants` *Dataset containing information about the available pollutants*

Description

This dataset belongs to the European Environment Agency institution. Contains short information about the available pollutants.

Usage

```
pollutants
```

Format

A data frame with 650 rows and 4 variables:

Code Unique pollutant code identifier

Notation pollutant's short name

Label pollutant's full name

RecommendedUnit Measurement unit, recommended for the pollutant

Source

http://dd.eionet.europa.eu/vocabulary/aq/pollutant/view

`stations` *Air quality measurement stations information.*

Description

This dataset belongs to the European Environment Agency. Assessment methods meta-data (data set D) describe technical facilities used for the measurement of one pollutant or one of its compounds. It contains information about the measurement stations mapped by the EEA in Europe. This dataset may be out of date, and for this reason the use of the `EEAaq_get_stations` function is suggested. For further information see https://cmshare.eea.europa.eu/s/8LGQLRGX8YEiSg9/download

Usage

```
stations
```

Format

A data frame with 68859 rows, 80 variables and a geometry representing the station's location point:

Country Country or territory name

ISO ISO 3166-1 alpha-2 code, representing the country. It's possible to refer to this variable as the NUTS 0 level

SamplingPointId Inspire identifier (Local Id) of sampling point, given by data provider

AirQualityStationEoICode EoI code of air quality measurement station (used in AirBase), given by data provider

AirQualityStationNatCode National code of air quality measurement station, given by data provider

AirQualityStationName Name of air quality measurement station (as in AirBase), given by data provider

AirPollutant Air polluting substance, level of which is measured and reported to the EEA. See `pollutants` for further information

OperationalActivityBegin Start time of the sampling point

OperationalActivityEnd End time of the sampling point

SamplingPointStatus Categorical variable which assumes two possible values:

- *active*: if the sampling point is still active (`OperationalActivityEnd = NA`)
- *closed*: if the sampling point activity is ceased

Longitude Longitude of air quality measurement station, according to the geographical coordinate system WGS84 (decimal degrees)

Latitude Latitude of air quality measurement station, according to the geographical coordinate system WGS84 (decimal degrees)

Altitude Altitude of air quality measurement station (m.a.s.l.)

NUTS1 Name in Latin characters of the area at the level NUTS 1 where the measuring station is located

NUTS1_ID NUTS system identification code of the area at NUTS 1 level in which the measuring station is located

NUTS2 Name in Latin characters of the area at the level NUTS 2 where the measuring station is located

NUTS2_ID NUTS system identification code of the area at NUTS 2 level in which the measuring station is located

NUTS3 Name in Latin characters of the area at the level NUTS 3 where the measuring station is located

NUTS3_ID NUTS system identification code of the area at NUTS 3 level in which the measuring station is located

LAU Name in Latin characters of the area at the level LAU where the measuring station is located

LAU_ID LAU system identification code of the area at LAU level in which the measuring station is located

AirQualityStationArea Area of Air Quality Measurement Station classification - information whether it is measuring air pollution in urban, suburban, rural (etc.) environment

AirQualityStationType Type of Air Quality Measurement Station - information whether it is measuring background,industrial or traffic related air pollution

B-GNamespace Inspire identifier/namespace of reporting entity, given by data provider

Year Latest year for which the data flow item has been reported

AirQualityNetwork Inspire identifier (Local Id) of air quality network, given by the data provider

AirQualityNetworkName Name of air quality measurement network, given by the data provider

Timezone Time zone in which aggregations and statistics are calculated

AltitudeUnit Unit of measurement of the altitude of the station

SampleId Inspire identifier (Local Id) of sample (Feature of Interest), given by data provider

InletHeight Height of the sampling point inlet

InletHeightUnit Unit of measurement of the height of the sampling point inlet

BuildingDistance The horizontal distance of the inlet to the nearest building

BuildingDistanceUnit Unit of measurement of the distance of the inlet to the nearest building

KerbDistance The horizontal distance of the inlet to the nearest kerb

KerbDistanceUnit Unit of measurement of the distance of the inlet to the nearest kerb

DistanceSource The distance from predominant industrial source or source area

DistanceSourceUnit Unit of measurement of the distance from predominant industrial source or source area

MainEmissionSources The main emission source for the pollutant

HeatingEmissions Amount of emissions from domestic heating for a representative area of approximately $1km^2$

HeatingEmissionsUnit Unit of measurement of the heating emissions

Mobile Mobile station qualifier fixed (0) or mobile (1)

TrafficEmissions Amount of emissions from road traffic for a section of road representative of at least 100 m

TrafficEmissionsUnit Unit of measurement of the traffic emissions

IndustrialEmissions Amount of emissions from industry for a representative area of approximately $1km^2$

IndustrialEmissionsUnit Unit of measurement of the industrial emissions

Municipality The name of the municipality in which the monitoring station is located

DispersionLocal The location of the station in relation to nearby buildings & trees using a controlled vocabulary

DispersionRegional The regional dispersion characteristics or topographic situation on a scale of several kilometres affecting the station from a controlled vocabulary

DistanceJunction Distance of the station from a major junction

DistanceJunctionUnit Unit of measurement of the distance of the station from a major junction

HeavyDutyFraction The fraction of the total traffic volume (assessed as AADT) that is composed of HGVs on the adjacent road

HeightFacades The average height of the building facades adjacent to the station (in meters) at the location of the station

StreetWidth The width of the street (in meters) at the location of the station

TrafficSpeed The average speed of vehicles in km/h on the adjacent road

TrafficVolume The total traffic volume (as an annual average daily traffic) on the adjacent road

ProcessId Inspire identifier (Local Id) of sampling process (procedure), given by data provider

ProcessActivityBegin Start time of the measurement process

ProcessActivityEnd End time of the measurement process

MeasurementType The classification (grouping) of measurement methods into generic types. The types of measurements include: Automatic analyser, Remote sensor, Active sampling and Passive sampling

MeasurementMethod Information on method used for measuring air polluting substances

OtherMeasurementMethod Other Measurement Method

MeasurementEquipment Information on equipment used for measuring air polluting substances

OtherMeasurementEquipment Other Measurement Equipment

SamplingMethod Information on the sampling methods used for Active or passive sampling measurement types (i.e.Passive adsorbent, Low Volume Sampling withautomatic filter change...)

OtherSamplingMethod Other Sampling Method

AnalyticalTechnique Information on analytical technique

OtherAnalyticalTechnique Other Analytical Technique

EquivalenceDemonstrated Specifies the equivalence status of the measuring/sampling process according to Annex VI.B of Dir. 2008/50EC and Annex V of Dir. 2004/107EC

DemonstrationReport Link to the equivalence demonstration report

DetectionLimit The measuring/sampling process including detection limit

DetectionLimitUnit Unit of measurement of the detection limit

Documentation Title of documentation on data quality

QAReport Link to report with Quality Assurance information

Duration The expected sampling duration of the measurement or sampling method

DurationUnit Unit of measurement of the duration

Cadence The time interval between the start of two consecutive measurements or samples

CadenceUnit Unit of measurement of the cadence

SourceDataURL URL of source data reported to the EEA

Imported Date and time of source data import into EEA's databases

Source

https://discomap.eea.europa.eu/App/AQViewer/index.html?fqn=Airquality_Dissem.b2g.Measurements

Index