

# Package ‘SchoolDataIT’

May 14, 2025

**Type** Package

**Title** Retrieve, Harmonise and Map Open Data Regarding the Italian School System

**Version** 0.2.6

**Author** Leonardo Cefalo [aut, cre] (ORCID:

<<https://orcid.org/0009-0001-7776-723X>>),

Alessio Pollice [ctb, ths] (ORCID:

<<https://orcid.org/0000-0002-2818-9373>>),

Paolo Maranzano [ctb] (ORCID: <<https://orcid.org/0000-0002-9228-2759>>)

**Maintainer** Leonardo Cefalo <[leonardo.cefalo@uniba.it](mailto:leonardo.cefalo@uniba.it)>

**Description** Compiles and displays the available data sets regarding the Italian school system, with a focus on the infrastructural aspects.

Input datasets are downloaded from the web, with the aim of updating everything to real time.

The functions are divided in four main modules, namely

'Get', to scrape raw data from the web

'Util', various utilities needed to process raw data

'Group', to aggregate data at the municipality or province level

'Map', to visualize the output datasets.

**License** GPL (>= 3)

**Encoding** UTF-8

**URL** <https://github.com/lcef97/SchoolDataIT>

**LazyData** true

**RoxygenNote** 7.3.2

**Imports** curl, dplyr, ggplot2, grDevices, httr, leafpop, magrittr, mapview, readr, rlang, rvest, sf, stringr, tidyr, utils, xml2

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), tidyverse

**Config/testthat/edition** 3

**Depends** R (>= 2.10)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-05-14 08:40:05 UTC

## Contents

example_AdmUnNames20220630 . . . . .	2
example_InnerAreas . . . . .	3
example_input_DB23_MIUR . . . . .	4
example_input_nstud23 . . . . .	6
example_input_Registry23 . . . . .	7
example_Invalsi23_prov . . . . .	8
example_Prov22_shp . . . . .	9
example_School2mun23 . . . . .	10
Get_AdmUnNames . . . . .	11
Get_BroadBand . . . . .	12
Get_DB_MIUR . . . . .	13
Get_InnerAreas . . . . .	15
Get_Invalsi_IS . . . . .	16
Get_nstud . . . . .	18
Get_nteachers_prov . . . . .	19
Get_Registry . . . . .	21
Get_School2mun . . . . .	22
Get_Shapefile . . . . .	23
Group_DB_MIUR . . . . .	25
Group_nstud . . . . .	27
Group_teachers4stud . . . . .	30
Map_DB . . . . .	31
Map_Invalsi . . . . .	34
Map_School_Buildings . . . . .	36
Set_DB . . . . .	38
Util_Check_nstud_availability . . . . .	44
Util_DB_MIUR_num . . . . .	47
Util_Invalsi_filter . . . . .	48
Util_nstud_wide . . . . .	50
<b>Index</b>	<b>53</b>

---

example\_AdmUnNames20220630

*Subset of the administrative codes of municipalities*

---

## Description

This table includes the administrative codes of the municipalities from four regions: Molise, Campania, Apulia and Basilicata, as of June 30th 2022; some strings in field Municipality\_description including accents have been forced to ASCII. The whole dataset can be retrieved with the command `Get_AdmUnNames(Date = "2022-06-30")`

## Usage

example\_AdmUnNames20220630

**Format**

## 'example\_AdmUnNames20220630' A data frame with 1,074 rows and 5 columns:

- Province\_code Numeric; the NUTS-3 administrative code
- Province\_initials Character; abbreviated NUTS-3 denomination.
- Municipality\_code Character; the ISTAT LAU (municipality) ID.
- Municipality\_description Character; the municipality name.
- Cadastral\_code Character; a LAU - level ID code, different from the official ISTAT municipality code. It is used in the school registry (see [example\\_input\\_Registry23](#))

**Source**

<<https://www.istat.it/it/archivio/6789>>

**See Also**

[Get\\_AdmUnNames](#)

---

example_InnerAreas	<i>Subset of the school registry in school year 2022/23</i>
--------------------	---

---

**Description**

This dataframe includes the classification of municipalities , from four regions: Molise, Campania, Apulia and Basilicata. Only the first 10 columns are included; some strings in field Municipality\_description including accents have been forced to ASCII. The whole dataset can be retrieved with the command `Get_InnerAreas()`. For the definition of ISTAT inner areas class, see [Get\\_InnerAreas](#)

**Usage**

```
example_InnerAreas
```

**Format**

## 'example\_InnerAreas' A data frame with 1074 rows and 10 columns:

- Municipality\_code Character; the ISTAT LAU (municipality) ID.
- Municipality\_code\_numeric Numeric; the ISTAT LAU (municipality) ID in numeric format.
- Cadastral\_code Character; a LAU - level ID code, different from the official ISTAT municipality code.
- Region\_code Numeric; the region (NUTS-2 administrative level) ID
- Region\_description Character; the region (NUTS-2 administrative level) name.
- Province\_code Numeric; the NUTS-3 administrative code.
- Province\_initials Character; abbreviated NUTS-3 denomination.

- Province\_description Character; the province (NUTS-3 administrative level) denomination.
- Municipality\_description Character; the municipality name.
- Inner\_area\_code\_2014\_2020 Character; the ISTAT inner areas classification between 2014 and 2020.
- Inner\_area\_description\_2014\_2020 Character; the description of the classes identified in the previous column
- Inner\_area\_code\_2021\_2027 Character; the ISTAT inner areas classification between 2021 and 2027.
- Inner\_area\_description\_2021\_2027 Character; the description of the classes identified in the previous column
- Destination\_municipality\_code Character; For non-central municipalities (classes C, D, E, F), the ID of the closest pole municipality according to the 2021-2027 classification
- Destination\_municipality\_code Character; The denomination of the municipalities in the previous column
- Destination\_pole\_code Character; An internal ID convention for the destination poles; it includes a letter (the class of the destination pole, either A or B); a number of two digits (the region code of the destination pole) and the progressive number of poles within a region.

### Source

<<https://www.istat.it/it/archivio/273176>>

### See Also

[Get\\_InnerAreas](#)

---

example\_input\_DB23\_MIUR

*Subset of the school buildings database in school year 2022/23*

---

### Description

This dataframe includes the schools directly identifiable as primary, middle or high school, from four regions: Molise, Campania, Apulia and Basilicata. Only the first 35 columns are included. Some strings including accents in fields `Other_disturbances_proximity`, `Other_specific_criticalities` and `Other` have been forced to ASCII. The whole dataset can be retrieved with the command `Get_DB_MIUR(2023)`

### Usage

example\_input\_DB23\_MIUR

**Format**

## 'example\_input\_DB23\_MIUR' A data frame with 7479 rows and 35 columns:

- Year Numeric; the school year.
- School\_code Character; the school ID.
- Order Character; the school order, either primary, middle or high school.
- Reference\_institute\_code Character; the ID of the reference institute.
- Building\_code Character; the building ID; the first 6 digits usually identify the municipality.
- Municipality\_code Character; the ISTAT LAU (municipality) ID.
- Municipality\_description Character; the municipality name.
- Province\_initials Character; abbreviated NUTS-3 denomination.
- Postal\_code Character; the ZIP code; slightly finer than municipality boundaries. for big municipalities.
- Context\_without\_disturbances Character; whether the school belongs to an environment devoid of disturbances; otherwise, the types of disturbances are listed in columns 11 - 18.
- Dumps\_proximity Character; whether the school is close to dumps (disturbance element).
- Pollutant\_industries\_proximity Character; whether the school is close to pollutant industries (disturbance element).
- Pollutant\_waters\_proximity Character; whether the school is close to pollutant or stagnant streams or ponds (disturbance element).
- Air\_pollution\_sourcer\_proximity Character; whether the school is close to sources of air pollution (disturbance element).
- Acoustic\_pollution\_sourcer\_proximity Character; whether the school is close to sources of acoustic pollution (disturbance element).
- Electromagnetic\_radiation\_sources\_proximity Character; whether the school is close to sources of electromagnetic radiation (disturbance element).
- Graveyards\_proximity Character; whether the school is close to a graveyard (disturbance element).
- Other\_disturbances\_proximity Character; other disturbance elements to which the school is close, other than those already listed.
- School\_area\_specific\_criticalities Character; whether any specific criticality element occurs inside the school area; specified in columns 20 - 27.
- Layby absence Character; whether the access to the area pertaining to the school building lacks a lay-by or pitch (school area criticality element).
- Unfenced area Character; whether the school building area lacks fences or enclosures (school area criticality element).
- Large\_traffic Character; whether the school area is close to large traffic streams (school area criticality element).
- Railway\_traffic Character; whether the school area is close to railway traffic streams (school area criticality element).
- Abandoned\_industries Character; whether the school area is located in pre-existences of abandoned industries (school area criticality element).

- `Decayed_urban_area` Character; whether the school belongs or is close to a decayed area (school area criticality element).
- `Risky_industries_proximity` Character; whether the school is close to perilous industrial areas (school area criticality element).
- `Other_specific_criticalities` Character; specific criticality elements regarding the school area, other than those already listed.
- `School_bus` Character; whether the school is reached by school-bus service.
- `Urban_public_transport` Character; whether the school is served by a urban public transport station in the range of 250 meters.
- `Interurban_public_transport` Character; whether the school is served by a inter-urban public transport station in the range of 500 meters.
- `Railway_transport` Character; whether the school ranges 500 meters or less from a train station.
- `Private_transport` Character; whether the school can be reached by private transport.
- `Disabled_people_transport` Character; whether the school is provided with disabled people specific transport.
- `Bicycle_lane` Character; whether the building is in proximity of a bicycle/bike lane.
- `Other` Character; whether the building can be reached in any other specific way.

### Source

[Homepage](#); more in detail, the dataset blocks are downloaded respectively from: [cols 10-18](#); [cols 20-27](#); [cols 28-35](#)

### See Also

[Get\\_DB\\_MIUR](#)

---

example\_input\_nstud23 *Subset of the students and classes counts in school year 2022/23*

---

### Description

This dataframe includes students and classes counts for the schools from four regions: Molise, Campania, Apulia and Basilicata. The whole dataset can be retrieved with the command `Get_nstud(2023, filename = "ALUCORSOINDCLASTA")`

### Usage

```
example_input_nstud23
```

**Format**

## 'example\_input\_nstud23' A data frame with 21208 rows and 7 columns:

- Year Numeric; the school year.
- School\_code Character; the school ID.
- Order Character; the school order, either primary, middle or high school.
- Grade Numeric; the school grade.
- Classes Numeric; the count of classes of a given grade in each school
- Male\_students Numeric; the count of male students in all classes of a given educational grade in each school
- Female\_students Numeric; the count of female students in all classes of a given educational grade in each school

**Source**

[Specific link](#)

**See Also**

[Get\\_nstud](#)

---

example\_input\_Registry23

*Subset of the school registry in school year 2022/23*

---

**Description**

This dataframe includes the schools directly identifiable as primary, middle or high school, from four regions: Molise, Campania, Apulia and Basilicata. Only the first 10 columns are included. The whole dataset can be retrieved with the command `Get_Registry(2023)`

**Usage**

`example_input_Registry23`

**Format**

## 'example\_input\_Registry23' A data frame with 5929 rows and 10 columns:

- Year Numeric; the school year.
- Area Character; the macro-area of the municipality, i.e. North, Center or South.
- Region\_description Character; the region (NUTS-2 administrative level) name.
- Province\_description Character; the province (NUTS-3 administrative level) name.
- Reference\_institute\_code Character; the ID of the reference institute.

- School\_code Character; the school ID.
- Cadastral\_code Character; a LAU - level ID code, different from the official LAU municipality code. The Italian Ministry of Education does provide this code in the place of the LAU code for both the Schools registry and the early school buildings DBs.
- Municipality\_description Character; the municipality name.
- School\_address Character; the school physical address.
- Postal\_code Character; the ZIP code, slightly finer than municipality boundaries for big municipalities.

### Source

[Source link](#)

### See Also

[Get\\_Registry](#)

---

example\_Invalsi23\_prov

*Subset of the Invalsi scores in school year 2022/23*

---

### Description

This dataframe includes the Invalsi scores of the schools from four regions: Molise, Campania, Apulia and Basilicata, for the school year 2022/23. The whole dataset can be retrieved with the command `Get_Invalsi_IS(level = "NUTS-3")`

### Usage

example\_Invalsi23\_prov

### Format

## 'example\_Invalsi23\_prov' A data frame with 240 rows and 11 columns:

- Year Character; the school year.
- Grade Numeric; the school grade; only includes the school grades subjected to the Invalsi survey. Either 2, 5, 8, 10 or 13.
- Subject Character; the school subject in which the test is taken; either Italian, Mathematics, English reading or English listening.
- Province\_code Numeric; the NUTS-3 administrative code.
- Province\_initials Character; abbreviated NUTS-3 denomination.
- Province\_description Character; the province (NUTS-3 administrative level) denomination.

- Average\_percentage\_score Numeric; the province-level percentage of sufficient tests, only for primary schools; ranges 0-100.
- Std\_dev\_percentage\_score Numeric; the standard deviation of the percentage of sufficient tests, only for primary schools.
- WLE\_average\_score Numeric; the province-level average WLE (Weighted Likelihood Estimator) score.
- Std\_dev\_WLE\_score Numeric; the standard deviation of WLE scores.
- Students\_coverage Numeric; the percentage of students for which the Invalsi tests are reported.

### Source

<[https://serviziostatistico.invalsi.it/en/archivio-dati/?\\_sft\\_invalsi\\_ss\\_data\\_collective=open-data](https://serviziostatistico.invalsi.it/en/archivio-dati/?_sft_invalsi_ss_data_collective=open-data)>

### See Also

[Get\\_Invalsi\\_IS](#)

---

example_Prov22_shp	<i>Subset of Italian provinces shapefile</i>
--------------------	--

---

### Description

This is the shapefile for the provinces belonging to four regions: Molise, Campania, Apulia and Basilicata, as of January 1st 2022. These are the latest administrative units boundaries relevant at the beginning of the school year 2022/23. The whole shapefile can be retrieved with the command `Get_Shapefile(Year = 2022, level = "NUTS-3")`

### Usage

```
example_Prov22_shp
```

### Format

## 'example\_Prov22\_shp' A Spatial polygon data frame with 13 rows/polygons and 15 columns:

- COD\_RIP Numeric; the code for the macroarea (1 for Northwest, 2 for Northeast, 3 for Center, 4 for South and 5 for Isles)
- COD\_REG Numeric; the region (NUTS-2 administrative level) ID
- COD\_PROV Numeric; the NUTS-3 administrative code
- COD\_CM Numeric; the administrative code for Metropolitan Cities (which are always at the NUTS-3 level), obtained as 200 + NUTS-3 code, if the unit is a Metropolitan city; 0 otherwise.
- COD\_UTS Numeric; the administrative code for Metropolitan cities if the unit is a Metropolitan City; the province code otherwise.

- DEN\_PROV Character; the province (NUTS-3 administrative level) name, if the unit is not a Metropolitan City; blank otherwise.
- DEN\_CM Character; the Metropolitan City (NUTS-3 administrative level) name, if the unit is a Metropolitan City; blank otherwise.
- DEN\_UTS Character; the province or Metropolitan City (NUTS-3 administrative level) name.
- SIGLA Character; abbreviated NUTS-3 denomination.
- TIPO\_UTS Character; the NUTS-3 type of the unit; either "Provincia" (Province) or "Citta metropolitana" (Metropolitan City)
- Shape\_Leng Numeric; the polygon perimeter.
- Shape\_Area Numeric; the polygon area.
- geometry the polygon geometry.

### Source

<<https://www.istat.it/it/archivio/222527>>

### See Also

[Get\\_Shapefile](#)

---

example_School2mun23	<i>Association of the municipality code to a subset of public schools 2022/23</i>
----------------------	---

---

### Description

This list maps the IDs of the schools from four regions (Molise, Campania, Apulia and Basilicata) to the corresponding LAU codes. The whole dataset can be retrieved with the command `Get_School2mun(2023)`

### Usage

```
example_School2mun23
```

### Format

## 'example\_School2mun23' A list of four elements

- Registry\_from\_buildings A data frame of 5527 rows and 5 columns, including the schools listed in the buildings registry.
- Registry\_from\_registry A data frame of 5929 rows and 5 columns, including the schools listed in the schools registry.
- Any A data frame of 5954 rows and 5 columns, including schools listed in any of the registries
- Both A data frame of 5510 rows and 5 columns, including schools listed in both registries

For each element, rows correspond to school IDs; the columns are:

- `School_code` Character; the school ID.
- `Province_code` Numeric; the NUTS-3 administrative code.
- `Province_initials` Character; abbreviated NUTS-3 denomination.
- `Municipality_code` Character; the ISTAT LAU (municipality) ID.
- `Municipality_description` Character; the municipality name.

### Source

Buildings registry (2021 onwards); Buindings registry(until 2019); Schools registry

### See Also

[Get\\_School2mun](#)

---

Get_AdmUnNames	<i>Download the names and codes of Italian LAU and NUTS-3 administrative units</i>
----------------	--

---

### Description

This function downloads a file provided by the Italian National Institute of Statistics including all the codes of administrative units in Italy. As of today, it is the easiest way to map directly cadastral codes to municipality codes.

### Usage

```
Get_AdmUnNames(Date = Sys.Date(), autoAbort = FALSE)
```

### Arguments

<code>Date</code>	Character. The date at which administrative unit codes are sought for. Important: must be in the format: "yyyy-mm-dd". Current date by default.
<code>autoAbort</code>	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

### Value

An object of class `tbl_df`, `tbl` and `data.frame`, including: NUTS-3 code, NUTS-3 abbreviation, LAU code, LAU name (description) and cadastral code. All variables are characters except for the NUTS-3 code.

### Source

<<https://situas.istat.it/web/#/territorio>>

## Examples

```
Get_AdmUnNames("2025-01-01", autoAbort = TRUE)
```

---

Get_BroadBand	<i>Download the data regarding the broad band connection activation in Italian schools</i>
---------------	--

---

## Description

Retrieves the data regarding the activation date of the broad band connection in schools. It also indicates whether the connection was activated or not at a certain date.

## Usage

```
Get_BroadBand(
  Date = Sys.Date(),
  verbose = TRUE,
  show_col_types = FALSE,
  autoAbort = FALSE
)
```

## Arguments

Date	Object of class Date. The date at which it is required to determine if the broad band connection has been activated or not. By default it is the current date.
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
show_col_types	Logical. If TRUE, if the verbose argument is also TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

## Details

Ultra - Broadband is defined as everlasting internet connection with a maximum speed of 1 gigabit per second, with a minimum guaranteed speed of 100 megabits/second both on the uploading and downloading operations, until the peering point is reached, as declared on the data provider's website: <<https://bandaultralarga.italia.it/scuole-voucher/progetto-scuole/>>. In the example the broadband availability at the beginning of school year 2022/23 (1st september 2022) is shown.

**Value**

An object of class `tbl_df`, `tbl` and `data.frame`. The variables `BB_Activation_date` and `BB_Activation_staus` indicate the activation date and activation status of the broadband connection at the selected date.

**Source**

Broadband dashboard: <<https://bandaultralarga.italia.it/scuole-voucher/dashboard-scuole/>>

**Examples**

```
Broadband_220901 <- Get_BroadBand(Date = as.Date("2022-09-01"), autoAbort = TRUE)

Broadband_220901

Broadband_220901[, c(9,6,13,14)]
```

---

Get\_DB\_MIUR

---

*Download the database of Italian public schools buildings*


---

**Description**

This function downloads the School Buildings Open Database provided by the Italian Ministry of Education, University and Research.

It is one of the main sources of information regarding the infrastructure system of public schools in Italy. For a given year, all available data are downloaded (except for the structural units section, which has a different level of detail) and gathered into a unique dataframe.

**Usage**

```
Get_DB_MIUR(
  Year = 2023,
  verbose = TRUE,
  input_Registry = NULL,
  input_AdmUnNames = NULL,
  show_col_types = FALSE,
  certifications = FALSE,
  autoAbort = FALSE
)
```

**Arguments**

Year	Numeric or character value. Reference school year (last available is 2023). Available in the formats: 2023, "2022/2023", 202223, 20222023. 2022 by default (other databases are not currently available for 2023).
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
input_Registry	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The school registry corresponding to the year in scope, obtained as output of the function <a href="#">Get_Registry</a> . If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default.
input_AdmUnNames	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The ISTAT file including all the codes and all the names of the administrative units for the year in scope, obtained as output of the function <a href="#">Get_AdmUnNames</a> . Only necessary for school years 2015/16, 2017/18 and 2018/19. If NULL and required, it will be downloaded automatically but not saved in the global environment. NULL by default.
show_col_types	Logical. If TRUE, if the verbose argument is also TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
certifications	Logical. From year 2021/22 onwards, whether to include some safety certifications in the database. Given the particular level of definition of this file, it requires extra computational time (other than the downloading time). FALSE by default.
autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

**Details**

This function downloads the raw data; missing observations are not edited; all variables are characters. Since certifications are defined at the level of structural units of the single buildings, here the fields read as the percentage of structural units in a building having a given certificate. To edit the output of this function and convert the relevant variables to numeric or Boolean, please [Util\\_DB\\_MIUR\\_num](#). Schools different from primary, middle or high schools are classified as "NR". In the example, the data for school year 2022/23 are retrieved.

**Value**

An object of class `tbl_df`, `tbl` and `data.frame`.

**Source**

[Homepage](#)

**Examples**

```
input_DB23_MIUR <- Get_DB_MIUR(2023, autoAbort = TRUE)
```

```
input_DB23_MIUR[-c(1,4,6,9)]
```

---

Get\_InnerAreas

Download the classification of peripheral municipalities

---

### Description

Retrieves the classification of Italian municipalities into six categories; classes D, E, and F are the so-called internal/inner areas; classes A, B and C are the central areas.

### Usage

```
Get_InnerAreas(verbose = TRUE, autoAbort = FALSE)
```

### Arguments

verbose	Logical. Whether to keep track of computational time. TRUE by default.
autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

### Details

Classes are defined according to these criteria; see the methodological note (in Italian) for more detail:

- A - Standalone pole municipalities, the highest degree of centrality; they are characterised by a thorough and self-sufficient combined endowment of school, health and transport infrastructure, i.e. there are at least a lyceum and a technical high school; a railway station of medium dimensions and a hospital provided with an emergency ward.
- B - Intermunicipality poles; the endowment of such infrastructures is complete if a small set of contiguous municipalities is considered

The remaining classes are defined in terms of the national distribution of the road distances from a municipality to the closest pole:

- C - Belt municipalities, travel time below the median ( $< 27'42''$ ).
- D - Intermediate municipalities, travel time between the median and the third quartile ( $27'42'' - 40'54''$ ).
- E - Peripheral municipalities, travel time between the third quartile and 97.5th percentile ( $40'54'' - 1h\ 6' 54''$ ).
- F - Ultra-peripheral municipalities, travel time over the 97.5th percentile ( $> 1h\ 6' 54''$ ).

For more information regarding the dataset, it is possible to check the ISTAT methodological note (in Italian) available at <https://www.istat.it/it/files//2022/07/FOCUS-AREE-INTERNE-2021.pdf>

**Value**

An object of class `tbl_df`, `tbl` and `data.frame`.

**Source**

<<https://www.istat.it/notizia/la-geografia-delle-aree-interne-nel-2020-vasti-territori-tra-potenzialita-e-debolezze/>>

**Examples**

```
InnerAreas <- Get_InnerAreas(autoAbort = TRUE)
InnerAreas[, c(1,9,13)]
```

---

Get\_Invalsi\_IS

*Download the Invalsi census survey data*

---

**Description**

Downloads the full database of the Invalsi scores, detailed either at the municipality or province level.

**Usage**

```
Get_Invalsi_IS(
  level = "LAU",
  verbose = TRUE,
  show_col_types = FALSE,
  multiple_out = TRUE,
  autoAbort = FALSE,
  category = FALSE
)
```

**Arguments**

<code>level</code>	Character. The level of aggregation of Invalsi census data. Either "NUTS-3", "Province", "LAU", "Municipality". "LAU" by default.
<code>verbose</code>	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
<code>show_col_types</code>	Logical. If TRUE, if the verbose argument is also TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
<code>multiple_out</code>	Logical. Whether keeping multiple dataframes as outputs (thus overriding the level argument) or not. TRUE by default.

autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.
category	Logical. Whether to focus on a specific category of students participating to the census survey. Warning: experimental. FALSE by default.

## Details

Numeric variables provided are:

- Average\_percentage\_score Average direct score (percentage of sufficient tests)
- Std\_dev\_percentage\_score Standard deviation of the direct score
- WLE\_average\_score Average WLE score. The WLE score is calculated through the Rasch's psychometric model and is suitable for middle and high schools in that it is cleaned from the effect of cheating (which would affect both the average score and the score variability). By construction it has a mean around 200 points.
- Std\_dev\_WLE\_score Standard deviation of the WLE score. By construction it ranges around 40 points at the school level.
- Students\_coverage Students coverage percentage

Additional numeric variables, not always available for all observational units, are:

- Mean and SD of ESCS indicator
- First-Fifth\_Level: Distribution of the proficiency level of students
- Targets\_percentage: Percentage of students reaching targets

Numeric codes 888 and 999 denote not applicable and not available fields respectively.

If multiple\_out == TRUE, provides the following datasets:

- Municipality\_data: LAU-level data
- Province\_data: NUTS-3-level data
- Region\_data: NUTS-2-level data
- LLS\_data: data at the level of local labour systems (Sistemi Locali del Lavoro; see [ISTAT webpage](#) for details)
- Inner\_Areas\_2021\_data aggregated data for inner areas according to the 2020 taxonomy
- Inner\_Areas\_2014\_data aggregated data for inner areas according to the former 2014 taxonomy
- Macroarea\_data data aggregated for North-West, North-East, Center, South and Islands

## Value

Unless multiple\_out == TRUE, an object of class tbl\_df, tbl and data.frame. Otherwise, a list including objects of the aforementioned classes

## Source

<[https://serviziostatistico.invalsi.it/en/archivio-dati/?\\_sft\\_invalsi\\_ss\\_data\\_collective=open-data](https://serviziostatistico.invalsi.it/en/archivio-dati/?_sft_invalsi_ss_data_collective=open-data)>

Examples

```
Get_Invalsi_IS(level = "NUTS-3", autoAbort = TRUE, verbose = FALSE)
```

---

Get_nstud	<i>Download students' number data</i>
-----------	---------------------------------------

---

Description

This functions downloads the data regarding the number of students, from the open website of the Italian Ministry of Education, University and Research

Usage

```
Get_nstud(  
  Year = 2023,  
  filename = c("ALUCORSOETASTA", "ALUCORSOINDCLASTA"),  
  verbose = TRUE,  
  show_col_types = FALSE,  
  autoAbort = FALSE  
)
```

Arguments

Year	Numeric or character. Reference school year (last available is 2023). Available in the formats: 2022, "2021/2022", 202122, 20212022. 2023 by default
filename	<p>Character. A string included in the name of the file to download. By default it is c("ALUCORSOETASTA", "ALUCORSOINDCLASTA"), which are the file names used so far for the number of students by age and the number of students in public schools by age and class.</p> <p>Other file names are the following. The output is not currently supported by the remainder of the functions involving the number of students.</p> <p>"ALUITASTRACITSTA" for the number of Italian and foreign students in public schools</p> <p>"ALUSECGRADOINDSTA" for the number of students of public schools by high school address</p> <p>"ALUTEMPOSCUOLASTA" for the number of students of public schools by school running time</p> <p>"ALUCORSOETAPAR", "ALUCORSOINDCLAPAR", "ALUITASTRACITPAR", "ALUSECGRADOINDPAR", "ALUTEMPOSCUOLAPAR" for the data of the previous file but referring to private schools.</p>
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.

- show\_col\_types Logical. If TRUE, if the verbose argument is also TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
- autoAbort Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

### Value

By default, a list of two `tbl_df`, `tbl` and `data.frame` objects:

- `$ALUCORSOETASTA`: The number of students by school, school grade and age. It provides a higher number of school than the other element
- `$ALUCORSOINDCLASTA`: The number of students and classes by school and school grade. This is a long-format dataframe.

### Source

[Homepage](#)

### Examples

```
Get_nstud(2023, filename = "ALUCORSOINDCLASTA", autoAbort = TRUE)
```

---

Get_nteachers_prov	<i>Download the number of teachers in Italian schools by province</i>
--------------------	---

---

### Description

This functions downloads the number of teachers by province from the open website of the Italian Ministry of Education, University and Research.

### Usage

```
Get_nteachers_prov(
  Year = 2023,
  verbose = TRUE,
  show_col_types = FALSE,
  filename = c("DOCTIT", "DOCSUP"),
  autoAbort = FALSE
)
```

**Arguments**

Year	Numeric or character value. Reference school year for the school registry data (last available is 2023). Available in the formats: 2022, "2021/2022", 202122, 20212022. 2023 by default
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
show_col_types	Logical. If TRUE, if the 'verbose' argument is also TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
filename	Character. Which data to retrieve among the province counts of teachers/school personnel. By default it is c("DOCTIT", "DOCSUP"), which are the file names used so far for the number of tenured and temporary teachers respectively. Other file names are the following: "ATATIT" for the number of tenured non-teaching personnel "ATASUP" for the number of temporary non-teaching personnel
autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

**Details**

Please notice that by default, the function returns the count of the number of tenured and temporary teachers. If either the count of non-teaching personnel or the count of a single category of teaching personnel is needed, please adapt the `filename` argument accordingly.

**Value**

An object of class `tbl_df`, `tbl` and `data.frame`.

**Source**

[Homepage](#)

**Examples**

```
nteachers23 <- Get_nteachers_prov(2023, filename = "DOCTIT", autoAbort = TRUE)
nteachers23[, c(3,4,5)]
```

---

Get_Registry	<i>Download the registry of Italian public schools from the school registry section</i>
--------------	---

---

## Description

This function returns two main pieces of information regarding Italian schools, namely:

- The denomination of the region, province and municipality to which the school belongs.
- The mechanographical code to the reference institute of each school.

It is possible to access schools in all the national territory, including the autonomous provinces of Aosta, Trento and Bozen.

## Usage

```
Get_Registry(
  Year = 2023,
  filename = c("SCUANAGRAFESTAT", "SCUANAAUTSTAT"),
  show_col_types = FALSE,
  autoAbort = FALSE
)
```

## Arguments

Year	Numeric or character. Reference school year (last available is 2024). Available in the formats: 2023, "2022/2023", 202223, 20222023. 2023 by default.
filename	Character. A string included in the name of the file to download, identifying the schools included. By default it is c("SCUANAGRAFESTAT", "SCUANAAUTSTAT"), i.e. the file names used for public school registries, respectively across all the national territory except for the autonomous provinces of Aosta, Trento or Bozen, and only in the three. If instead the registry of the private schools is needed, please insert "SCUANAGRAFEPAR" and/or "SCUANAAUTPAR". For the registry of private schools, either in all the national territory except for the aforementioned provinces, and for these provinces, please use "SCUANAGRAFEPAR" and "SCUANAAUTPAR" respectively. Please notice that data regarding private schools are not available for most functions in this package.
show_col_types	Logical. If TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

## Details

Schools different from primary, middle or high schools are classified as "NR".

**Value**

An object of class tbl\_df, tbl and data.frame.

**Source**

[Homepage](#)

**Examples**

```
Get_Registry(2024, filename = "SCUANAGRAFESTAT", autoAbort = TRUE)
```

---

Get_School2mun	<i>Associate a Municipality (LAU) code to each school</i>
----------------	---

---

**Description**

This function associates the relevant municipality codes to all the schools listed in the two main registries provided by the Italian Ministry of Education, University and Research, namely:

- The registry of school buildings, here referred to as Registry\_from\_buildings ([Get\\_DB\\_MIUR](#))
- The official schools registry, here referred to as Registry\_from\_registry (see [Get\\_Registry](#))

**Usage**

```
Get_School2mun(  
  Year = 2023,  
  show_col_types = FALSE,  
  verbose = TRUE,  
  input_AdmUnNames = NULL,  
  input_Registry = NULL,  
  autoAbort = FALSE  
)
```

**Arguments**

Year	Numeric or character value (last available is 2023). Available in the formats: 2023, "2022/2023", 202223, 20222023. 2023 by default.
show_col_types	Logical. If TRUE, if the verbose argument is also TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.

input_AdmUnNames	Object of class tbl_df, tbl and data.frame, obtained as output of the function <a href="#">Get_AdmUnNames</a> The ISTAT file including all the administrative units codes for the year in scope. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default.
input_Registry	Object of class tbl_df, tbl and data.frame, obtained as output of the function <a href="#">Get_Registry</a> The school registry corresponding to the year in scope. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default
autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

Value

- An object of class list, including 4 elements:
- \$Registry\_from\_buildings: Object of class tbl\_df, tbl and data.frame: the schools listed in the buildings registry
  - \$Registry\_from\_registry: Object of class tbl\_df, tbl and data.frame: the schools listed in the schools registry
  - \$Any: Object of class tbl\_df, tbl and data.frame: schools listed anywhere
  - \$Both: Object of class tbl\_df, tbl and data.frame: schools listed in both the sections

Source

Buildings registry (2021 onwards); Buindings registry(until 2019); Schools registry

Examples

```
Get_School2mun(Year = 2023, autoAbort = TRUE)
```

---

Get_Shapefile	<i>Download the shapefiles of Italian NUTS-3 and LAU administrative units</i>
---------------	---

---

Description

Downloads either the boundaries or the centroids of the relevant administrative units, either provinces or municipalities, from the ISTAT website. Geometries are expressed in meters.

**Usage**

```
Get_Shapefile(
  Year,
  level = "LAU",
  lightShp = TRUE,
  autoAbort = FALSE,
  centroids = FALSE
)
```

**Arguments**

Year	Numeric. Reference year for the administrative units.
level	Character. Either "LAU"/"Municipality", "NUTS-3"/"Province", "NUTS-2"/"Region", . "LAU" by default
lightShp	Logical. If TRUE, the function downloads a generalised, i.e. less detailed, and lighter version of the shapefiles. TRUE by default.
autoAbort	Logical. Whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.
centroids	Logical. Whether to switch from polygon geometry to point geometry. In the latter case, the point is located at the centroid of the relevant area. FALSE by default.

**Value**

A spatial data frame of class `data.frame` and `sf`.

**Source**

<<https://www.istat.it/it/archivio/222527>>

**Examples**

```
library(magrittr)
```

```
Prov23_shp <- Get_Shapefile(2023, lightShp = TRUE, level = "NUTS-3", autoAbort = TRUE)
ggplot2::ggplot() + ggplot2::geom_sf(data = Prov23_shp) +
  ggplot2::ggtitle("Italian provinces in 2023/01/01")
```

---

Group_DB_MIUR	<i>Aggregate the database of Italian public schools buildings at the municipality and province level</i>
---------------	--

---

## Description

This function transforms the output of the [Util\\_DB\\_MIUR\\_num](#) function (which is detailed at the level of single school buildings) at the municipality/LAU and province/NUTS-3 level. It also allows the user to classify the grade of centrality of municipalities through the variable `Inner_area`.

## Usage

```
Group_DB_MIUR(
  data = NULL,
  Year = 2023,
  count_units = TRUE,
  countname = "nbbuildings",
  count_missing = TRUE,
  verbose = TRUE,
  track_deleted = TRUE,
  InnerAreas = TRUE,
  ord_InnerAreas = FALSE,
  input_InnerAreas = NULL,
  autoAbort = FALSE,
  ...
)
```

## Arguments

<code>data</code>	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The database of school buildings, preferably already converted to numeric, obtained via <a href="#">Util_DB_MIUR_num</a>
<code>Year</code>	Numeric or Character. The reference school year, if either <code>data</code> or <code>input_InnerAreas</code> must be retrieved. Available in the formats: 2023, "2022/2023", 202223, 20222023. Important: use the same <code>Year</code> argument used to retrieve the input school buildings data if they are provided as input. 2023 by default
<code>count_units</code>	Logical. Whether the rows to aggregate at each level must be counted or not. True by default.
<code>countname</code>	character. The name of the variable indicating the number of schools included in each municipality of province, if the argument 'count' is TRUE. "nbbuildings" by default.
<code>count_missing</code>	Logical. Whether the function should return two dataframes including the percentage of NAs in the data object at the territorial level. TRUE by default
<code>verbose</code>	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
<code>track_deleted</code>	Logical. If TRUE, the function returns the IDs of schools not included. TRUE by default.

InnerAreas	Logical. Whether an indicator of the percentage of schools belonging to peripheral (Inner) areas must be included or not.
ord_InnerAreas	Logical. Whether the Inner areas classification should be treated as an ordinal variable rather than as a binary one (see <a href="#">Get_InnerAreas</a> for the classification). Please notice that the function creates a column for each class, and if this database must be used in a statistical model, one of the 6 resulting columns must be dropped. False by default.
input_InnerAreas	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The classification of peripheral municipalities, needed only if <code>InnerAreas == TRUE</code> , obtained as output of the <a href="#">Get_InnerAreas</a> function. If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default
autoAbort	Logical. In case any data must be retrieved, whether to automatically abort the operation and return <code>NULL</code> in case of missing internet connection or server response errors. <code>FALSE</code> by default.
...	Additional arguments to the function <a href="#">Util_DB_MIUR_num</a> in case no data are provided or data.

## Details

Numerical variables are summarised by the mean; Boolean variables are summarised by the mean as well, thus they become frequency indicators. Qualitative values, if included, are summarised by the mode. Summary measures do not include NAs. The output dataframes are also detailed at the school order level (i.e. Primary, Middle, High school, or different orders). This means that rows are unique combinations of territorial unities and school order.

## Value

An object of class `list` including:

- `$Municipality_data`: object of class `tbl_df`, `tbl` and `data.frame`, the output dataframe detailed at the municipality level; all variables besides the first 5 (which identify the record) are numeric
- `$Province_data`: object of class `'tbl_df'`, `'tbl'` and `'data.frame'`, the output dataframe detailed at the province level; all variables besides the first 3 (which identify the record) are numeric
- `$Municipality_missing` (Only if `count_missing == TRUE`); object of class `tbl_df`, `tbl` and `data.frame`, the percentage of NAs in each variable at the municipality level.
- `$Province_missing`: (Only if `count_missing == TRUE`); object of class `'tbl_df'`, `'tbl'` and `'data.frame'`, the percentage of NAs in each variable at the province level.
- `$deleted`: character vector. The schools removed from the original dataframe for data quality reasons. This object is returned only if `track_deleted == TRUE`

## Examples

```
library(magrittr)
DB23_MIUR <- example_input_DB23_MIUR %>% Util_DB_MIUR_num(verbose = FALSE) %>%
```

```
Group_DB_MIUR(InnerAreas = FALSE)
```

```
DB23_MIUR$Municipality_data[, -c(1,2,4)]
summary(DB23_MIUR$Municipality_data)
```

```
DB23_MIUR$Province_data[, -c(1,3)]
summary(DB23_MIUR$Province_data)
```

---

Group_nstud	<i>Aggregate the students number data by class at the municipality and province level</i>
-------------	---

---

## Description

This function creates two dataframes with the number of students, classes and students by class, aggregated at the province and municipality level

## Usage

```
Group_nstud(
  data = NULL,
  Year = 2023,
  check = TRUE,
  verbose = TRUE,
  check_registry = "Any",
  InnerAreas = TRUE,
  ord_InnerAreas = FALSE,
  check_ggplot = FALSE,
  missing_to_1 = FALSE,
  input_Registry = NULL,
  input_InnerAreas = NULL,
  input_Prov_shp = NULL,
  input_School2mun = NULL,
  input_AdmUnNames = NULL,
  autoAbort = FALSE,
  ...
)
```

**Arguments**

data	Either an object of class <code>list</code> , obtained as output of the <a href="#">Get_nstud</a> function, or an object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the <a href="#">Util_nstud_wide</a> function, if <code>NULL</code> , the function will download it automatically but it will not be saved in the global environment. <code>NULL</code> by default.
Year	Numeric or character value. The reference school year, if either of the <code>input_</code> arguments must be retrieved. Available in the formats: <code>2022</code> , <code>"2022/2023"</code> , <code>"202223"</code> , <code>"20222023"</code> . <code>2023</code> by default
check	Logical. If <code>TRUE</code> , the function runs the test of the students number availability across all school included in the school registries (see <a href="#">Util_Check_nstud_availability</a> ). <code>TRUE</code> by default
verbose	Logical. If <code>TRUE</code> , the user keeps track of the main underlying operations. <code>TRUE</code> by default.
check_registry	Character. If <code>check == TRUE</code> , the school registries included in the <code>input_School2mun</code> object (see <a href="#">Get_School2mun</a> ) whose availability has to be checked. Either <code>"Registry_from_buildings"</code> (buildings section), <code>"Registry_from_registry"</code> (registry section), <code>"Any"</code> or <code>"Both"</code> . <code>"Any"</code> by default.
InnerAreas	Logical. If <code>check == TRUE</code> , Whether it must be checked if municipalities belong to Inner areas or not. <code>TRUE</code> by default.
ord_InnerAreas	Logical. If <code>check == TRUE</code> and <code>InnerAreas == TRUE</code> , whether the Inner areas classification should be treated as an ordinal variable rather than as a categorical one (see <a href="#">Get_InnerAreas</a> for the classification). <code>FALSE</code> by default.
check_ggplot	Logical. If <code>check == TRUE</code> , whether to display or not a static map of the availability of the students number by province; see also <a href="#">Util_Check_nstud_availability</a> . <code>TRUE</code> by default.
missing_to_1	Logical. Only needed if data is not provided in wide format. Whether the number of classes should be imputed to 1 when it is missing; see <a href="#">Util_nstud_wide</a> . <code>FALSE</code> by default.
input_Registry	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the function <a href="#">Get_Registry</a> . If <code>check == TRUE</code> , the school registry (the properly said one, from the registry section). If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default
input_InnerAreas	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The classification of peripheral municipalities, obtained as output of the <a href="#">Get_InnerAreas</a> function. Needed only if <code>check == TRUE</code> and <code>InnerAreas == TRUE</code> . If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default
input_Prov_shp	Object of class <code>sf</code> , <code>tbl_df</code> , <code>tbl</code> , <code>data.frame</code> . The relevant shapefile of Italian municipalities, if both the <code>check</code> and <code>check_ggplot</code> options are chosen. If <code>NULL</code> it is downloaded automatically but not saved in the global environment. <code>NULL</code> by default.
input_School2mun	Object of class <code>list</code> with elements of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the function <a href="#">Get_School2mun</a> . The mapping from school

codes to municipality (and province) codes. Needed only if 'check == TRUE'. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default.

input_AdmUnNames	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the function <a href="#">Get_AdmUnNames</a> The ISTAT file including all the codes and the names of the administrative units for the year in scope. Only needed if <code>check == TRUE</code> and the argument <code>input_School2mun</code> is NULL. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default.
autoAbort	Logical. In case any data must be retrieved, whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.
...	Additional arguments to the function <a href="#">Util_nstud_wide</a> if data is not provided.

## Details

Numerical variables are summarised by the mean; Boolean variables are summarised by the mean as well, thus they become frequency indicators. Qualitative values, if included, are summarised by the mode. Summary measures do not include NAs.

## Value

An object of class `list` including:

- `$Municipality_data`: object of class `tbl_df`, `tbl` and `data.frame`, the output dataframe detailed at the municipality level
- `$Province_data`: object of class `'tbl_df'`, `'tbl'` and `'data.frame'`, the output dataframe detailed at the province level

## Examples

```
Year <- 2023

nstud23_aggr <- Group_nstud(data = example_input_nstud23, Year = Year,
                           input_Registry = example_input_Registry23,
                           InnerAreas = FALSE,
                           input_School2mun = example_School2mun23)

summary(nstud23_aggr$Municipality_data[,c(46,47,48)])

summary(nstud23_aggr$Province_data[,c(44,45,46)])
```

---

Group_teachers4stud	<i>Arrange the number of teachers per students in public Italian schools at the province level</i>
---------------------	--

---

## Description

This function provides the average number of teachers per students in Italian public schools at the province level.

## Usage

```
Group_teachers4stud(
  Year = 2023,
  input_nteachers = NULL,
  nteachers_filename = c("DOCTIT", "DOCSUP"),
  verbose = TRUE,
  input_nstud_raw = NULL,
  input_nstud_aggr = NULL,
  autoAbort = FALSE,
  ...
)
```

## Arguments

Year	Numeric or character value. Reference school year for the school registry data (last available is 2022). Available in the formats: 2022, "2021/2022", 202122, 20212022. 2023 by default
input_nteachers	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The number of teachers by province, obtained as output of the function <a href="#">Get_nteachers_prov</a> . If <code>NULL</code> , the function will download it automatically but it will not be saved in the global environment. <code>NULL</code> by default.
nteachers_filename	Character. If <code>input_nteachers</code> is not provided, which data to retrieve regarding the number of teachers/personnel; see <a href="#">Get_nteachers_prov</a> <code>c("DOCTIT", "DOCSUP")</code> by default, i.e. tenured teachers and temporary teachers.
verbose	Logical. If <code>TRUE</code> , the user keeps track of the main underlying operations. <code>TRUE</code> by default.
input_nstud_raw	Object of class 'list', including two objects of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the <a href="#">Get_nstud</a> function with the default filename parameter. Not necessary if the argument <code>input_nstud_aggr</code> is provided. If <code>NULL</code> , the function will download it automatically but it will not be saved in the global environment. <code>NULL</code> by default.

<code>input_nstud_aggr</code>	Object of class <code>list</code> , including two objects of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the function <code>Group_nstud</code> . If <code>NULL</code> , the function will compute it manually but it will not be saved in the global environment. <code>NULL</code> by default.
<code>autoAbort</code>	Logical. In case any data must be retrieved, whether to automatically abort the operation and return <code>NULL</code> in case of missing internet connection or server response errors. <code>FALSE</code> by default.
<code>...</code>	Arguments to <code>Group_nstud</code> if argument <code>input_nstud_aggr</code> is not provided

**Value**

An object of class `tbl_df`, `tbl` and `data.frame`

**Examples**

```
input_nstud23 <- Get_nstud(2023, filename = "ALUCORSOINDCLASTA", autoAbort = TRUE)
Registry23 <- Get_Registry(2023, autoAbort = TRUE)
School2mun23 <- Get_School2mun(2023, input_Registry = Registry23, autoAbort = TRUE)

nstud23.aggr <- Group_nstud(Year = 2023, data = input_nstud23,
  input_Registry = Registry23, input_School2mun = School2mun23,
  autoAbort = TRUE)

input_nteachers23 <- Get_nteachers_prov(2023, autoAbort = TRUE)

teachers4stud <- Group_teachers4stud(Year = 2023,
  input_nteachers = input_nteachers23,
  input_nstud_aggr = nstud23.aggr, autoAbort = TRUE)

teachers4stud[, -c(1, 2, 10, 11)]

summary(teachers4stud)
```

---

Map\_DB

---

*Map school data*


---

**Description**

This function displays a map of the data arranged through the function `Set_DB`. It supports two kinds of map:

- Interactive map (default option), which allows the user to visualize all the data in scope through the interactive popup, and

- Static map (ggplot), which can be easily exported in .pdf objects.

The user must select a variable to display. It is possible to insert either a readily-downloaded database obtained through the function [Set\\_DB](#) or the basic inputs to plug in that function, other than an input shapefile. Relevant arguments not provided by the user will be download automatically, but not saved into the global environment. However we suggest to plug in at least some inputs, as otherwise the running time may be long. This function generalises the functionalities of the more data-specific functions [Map\\_School\\_Buildings](#) and [Map\\_Invalsi](#).

## Usage

```
Map_DB(
  data = NULL,
  Year = 2023,
  field,
  level = "LAU",
  plot = "mapview",
  popup_height = 200,
  col_rev = FALSE,
  pal = "viridis",
  input_shp = NULL,
  region_code = c(1:20),
  main_pos = "top",
  main = "",
  order = NULL,
  autoAbort = FALSE,
  only_observed = FALSE,
  ...
)
```

## Arguments

data	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the <a href="#">Set_DB</a> function. If <code>NULL</code> , it will be arranged automatically but not saved into the global environment. <code>NULL</code> by default.
Year	Numeric or Character. The reference school year, needed if either data or <code>input_shp</code> are not provided. Available in the formats: 2023, "2022/2023", 202223, 20222023. 2023 by default.
field	Character. The variable to display in the map.
level	Character. The administrative level of detailed at which the target variable must be displayed. Either "LAU"/"Municipality" or "NUTS-3"/"Province". If the "data" argument is plugged in, please select the same level. "LAU" by default.
plot	Character. The type of map to display; either "mapview" for interactive maps, or "ggplot" for static maps. "mapview" by default.
popup_height	Numeric. The height of the popup table in terms of pixels if the "mapview" mode is chosen. 200 by default.
col_rev	Logical. Whether the scale of the colour palette should be reverted or not. <code>FALSE</code> by default.

pal	Character. The palette to use if the "mapview" mode is chose. "viridis" by default.
input_shp	Object of class sf, tbl_df, tbl and data.frame. The relevant shapefiles of Italian administrative boundaries, at the selected level of detail (LAU or NUTS-3). If NULL, it is downloaded automatically but not saved in the global environment. NULL by default.
region_code	Numeric. The NUTS-2 codes of the units that must be displayed. If the level is set to "LAU", choosing a limited number of regions is recommended. By default, c(1, 3, 5:20), i.e. all Italian regions except the provinces of Aosta, Trento and Bozen which have data availability issues.
main_pos	Character. Where the header should be placed if the ggplot mode is chosen. The header is located on the top if "top" is given as input, and above the legend scale otherwise. "top" by default.
main	Character. The title to display in the "ggplot" rendering options.
order	Character. The educational level. Either "Primary" (primary school), "Middle" (middle school), or "High" (high school). If the data include the Invalsi census survey, please select a level consistent with the chosen educational grade. "Media" by default.
autoAbort	Logical. In case any data must be retrieved, whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.
only_observed	Logical. Whether to remove unobserved areas from the plot. FALSE by default.
...	Additional arguments for the input database, if not provided; see <a href="#">Set_DB</a>

### Value

If plot == "mapview", an object of class mapview. Otherwise, if plot == "ggplot", an object of class gg and ggplot.

### Examples

```
DB23 <- Set_DB(Year = 2023, level = "NUTS-3",
  Invalsi_grade = c(10,13), NA_autoRM = TRUE,
  input_Invalsi_IS = example_Invalsi23_prov, input_nstud = example_input_nstud23,
  input_InnerAreas = example_InnerAreas,
  input_School2mun = example_School2mun23,
  input_AdmUnNames = example_AdmUnNames20220630,
  nteachers = FALSE, BroadBand = FALSE, SchoolBuildings = FALSE)
```

```
Map_DB(DB23, field = "Students_per_class_13", input_shp = example_Prov22_shp, level = "NUTS-3",
  col_rev = TRUE, plot = "ggplot")
```

```
Map_DB(DB23, field = "Inner_area", input_shp = example_Prov22_shp, order = "High",
```

```

level = "NUTS-3", col_rev = TRUE, plot = "ggplot")

Map_DB(DB23, field = "M_Mathematics_10", input_shp = example_Prov22_shp, level = "NUTS-3",
plot = "ggplot")

```

---

Map\_Invalsi

*Display a map of Invalsi scores*


---

## Description

This function displays either a static or interactive map of the Invalsi scores, either at the municipality or province level. It supports two kinds of map:

- Interactive map (default option), which allows the user to visualize all the data in scope through the interactive popup, and
- Static map (ggplot), which can be easily exported in .pdf objects.

## Usage

```

Map_Invalsi(
  data = NULL,
  Year = 2023,
  subj_toplot = "ITA",
  grade = 8,
  level = "LAU",
  main = "",
  main_pos = "top",
  region_code = c(1:20),
  plot = "mapview",
  pal = "viridis",
  WLE = FALSE,
  col_rev = FALSE,
  popup_height = 200,
  only_observed = FALSE,
  verbose = TRUE,
  input_shp = NULL,
  autoAbort = FALSE
)

```

## Arguments

data	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The raw Invalsi survey data that has to be filtered, obtained as output of the <a href="#">Get_Invalsi_IS</a> function. If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default
------	--

Year	Numeric or character value. Reference school year for the data (last available is 2022/23). Available in the formats: 2022, "2021/2022", 202122, 20212022. 2022 by default
subj_toplot	Character. The school subject to display in the map, The school subject to include, one among: "Englis_listening"/"ELI", "English_reading"/"ERE", "Italian"/"ITA" and "Mathematics"/"MAT". "ITA" (Italian) by default.
grade	Numeric. The school grade to chose. Either 2 (2nd year of primary school), 5 (last year of primary school), 8 (last year of middle shcool), 10 (2nd year of high school) or 13 (last year of school). 8 by default
level	Character. The level of aggregation of Invalsi census data. Either "NUTS-3", "Province", "LAU", "Municipality". If an input dataframe is provided, please select the same level of aggregation. "LAU" by default
main	Character. A customary title to the map. If NULL, the title will mention: subject, year and school grade. Empty by default.
main_pos	Character. Where the header should be placed if the ggplot mode is chosen. The header is located on the top if "top" is given as input, and above the legend scale otherwise. "top" by default.
region_code	Numeric. The NUTS-2 codes of the units that must be displayed. If the level is set to "LAU", choosing a limited number of regions is recommended. By default, c(1, 3, 5:20), i.e. all Italian regions except the provinces of Aosta, Trento and Bozen which have data availability issues.
plot	Character. The type of map to display; either "mapview" for interactive maps, or "ggplot" for static maps. "mapview" by default.
pal	Character. The palette to use if the "mapview" mode is chose. "viridis" by default.
WLE	Logical. Whether the variable to chose should be the average WLE score rather that the percentage of sufficient tests, if both are available. FALSE by default
col_rev	Logical. Whether the scale of the colour palette should be reverted or not, if the mapview mode is chosen. FALSE by default
popup_height	Numeric. The height of the popup table in terms of pixels if the "mapview" mode is chosen. 200 by default.
only_observed	Logical. Whether to remove unobserved areas from the plot. FALSE by default.
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
input_shp	Object of class sf, tbl_df, tbl, data.frame. The relevant shapefiles of Italian administrative boudaries, at the selected level of detail (LAU or NUTS-3). If NULL, it is downloaded automatically but not saved in the global environment. NULL by default.
autoAbort	Logical. In case any data must be retrieved, whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

### Value

If plot == "mapview", an object of class mapview. Otherwise, if plot == "ggplot", an object of class gg and ggplot.

## Examples

```
Map_Invalsi(subj = "Italian", grade = 13, level = "NUTS-3", Year = 2023, WLE = FALSE,
  data = example_Invalsi23_prov, input_shp = example_Prov22_shp, plot = "ggplot")
```

```
Map_Invalsi(subj = "Italian", grade = 5, level = "NUTS-3", Year = 2023, WLE = TRUE,
  data = example_Invalsi23_prov, input_shp = example_Prov22_shp, plot = "ggplot")
```

---

Map\_School\_Buildings    *Display data fom the school buildings database*

---

## Description

This function displays a map of the data downloaded trough the [Get\\_DB\\_MIUR](#) function. It supports two kinds of map:

- Interactive map (default option), which allows the user to visualize all the data in scope through the interactive popup, and
- Static map (ggplot), which can be easily exported in .pdf objects.

## Usage

```
Map_School_Buildings(
  data = NULL,
  field,
  order = NULL,
  level = "LAU",
  region_code = c(1:20),
  plot = "mapview",
  pal = "viridis",
  col_rev = FALSE,
  popup_height = 200,
  main_pos = "top",
  main = "",
  only_observed = FALSE,
  verbose = TRUE,
  input_shp = NULL,
  autoAbort = FALSE,
  ...
)
```

**Arguments**

<code>data</code>	Object of class <code>list</code> or <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . Input data obtained as output of the function <a href="#">Group_DB_MIUR</a> . If <code>NULL</code> , it will be downloaded automatically but not saved in the global environment. <code>NULL</code> by default.
<code>field</code>	Character. The variable to display in the map.
<code>order</code>	Character. The school order. Either "Primary", "Middle", or "High" (high school). If <code>NULL</code> , an average of the three school orders will be displayed for the target variable. <code>NULL</code> by default.
<code>level</code>	Character. The administrative level of detailed at which the target variable must be displayed. Either "LAU"/"Municipality" or "NUTS-3"/"Province". "LAU" by default.
<code>region_code</code>	Numeric. The NUTS-2 codes of the units that must be displayed. If the level is set to "LAU", choosing a limited number of regions is recommended. By default, <code>c(1:20)</code> , i.e. all Italian regions.
<code>plot</code>	Character. The type of map to display; either "mapview" for interactive maps, or "ggplot" for static maps. "mapview" by default.
<code>pal</code>	Character. The palette to use if the "mapview" mode is chose. "viridis" by default.
<code>col_rev</code>	Logical. Whether the scale of the colour palette should be reverted or not, if the "mapview" mode is chosen. <code>FALSE</code> by default
<code>popup_height</code>	Numeric. The height of the popup table in terms of pixels if the "mapview" mode is chosen. 200 by default.
<code>main_pos</code>	Character. Where the header should be placed if the ggplot mode is chosen. The header is located on the top if "top" is given as input, and above the legend scale otherwise. "top" by default.
<code>main</code>	Character. The customary title to display in the "ggplot" rendering options
<code>only_observed</code>	Logical. Whether to remove unobserved areas from the plot. <code>FALSE</code> by default.
<code>verbose</code>	Logical. If <code>TRUE</code> , the user keeps track of the main underlying operations. <code>TRUE</code> by default.s
<code>input_shp</code>	Object of class <code>sf</code> , <code>tbl_df</code> , <code>tbl</code> , <code>data.frame</code> . The relevant shapefiles of Italian administrative boudaries, at the selected level of detail (LAU or NUTS-3). If <code>NULL</code> it is downloaded automatically but not saved in the global environment. <code>NULL</code> by default.
<code>autoAbort</code>	Logical. In case any data must be retrieved, whether to automatically abort the operation and return <code>NULL</code> in case of missing internet connection or server response errors. <code>FALSE</code> by default.
<code>...</code>	If data is not provided, the arguments to <a href="#">Group_DB_MIUR</a> .

**Value**

If `plot == "mapview"`, an object of class `mapview`. Otherwise, if `plot == "ggplot"`, an object of class `gg` and `ggplot`.

## Examples

```
library(magrittr)

DB23_MIUR <- example_input_DB23_MIUR %>%
  Util_DB_MIUR_num(track.deleted = FALSE) %>%
  Group_DB_MIUR(InnerAreas = FALSE, count_missing = FALSE)

DB23_MIUR %>% Map_School_Buildings(field = "School_bus",
  order = "Primary", level = "NUTS-3", plot = "ggplot",
  input_shp = example_Prov22_shp)

DB23_MIUR %>% Map_School_Buildings(field = "Railway_transport",
  order = "High", level = "NUTS-3", plot = "ggplot",
  input_shp = example_Prov22_shp)

DB23_MIUR %>% Map_School_Buildings(field = "Context_without_disturbances",
  order = "Middle", level = "NUTS-3", plot = "ggplot",
  input_shp = example_Prov22_shp, col_rev = TRUE)
```

---

Set\_DB

---

*Build up a comprehensive database regarding the school system*


---

## Description

This function generates a unique dataframe of the school system data including a customary choice of available datasets. This function allows the user to aggregate the desired datasets, when available, among these:

- Invalsi census survey
- School buildings
- Number of students and school classes
- Number of teachers
- Broadband connection availability

To save as much time as possible it is possible to plug in ready-made input data; otherwise they will be downloaded automatically but not saved in the global environment. When a new dataset is joined to the existing ones, it is possible that some observations in this datasets are missing. In this case, by default, the choice of keeping as much observational units as possible, or to remove units with missing variables is left to the user.

**Usage**

```

Set_DB(
  Year = 2023,
  level = "LAU",
  conservative = TRUE,
  Invalsi = TRUE,
  SchoolBuildings = TRUE,
  nstud = TRUE,
  nteachers = TRUE,
  BroadBand = TRUE,
  verbose = TRUE,
  show_col_types = FALSE,
  Invalsi_subj = c("ELI", "ERE", "ITA", "MAT"),
  Invalsi_grade = c(2, 5, 8, 10, 13),
  Invalsi_WLE = FALSE,
  SchoolBuildings_certifications = FALSE,
  SchoolBuildings_include_numerics = TRUE,
  SchoolBuildings_include_qualitatives = FALSE,
  SchoolBuildings_row_cutout = FALSE,
  SchoolBuildings_col_cut_thresh = 20000,
  SchoolBuildings_flag_outliers = TRUE,
  SchoolBuildings_count_missing = FALSE,
  nstud_imputation_thresh = 19,
  nstud_missing_to_1 = FALSE,
  UB_nstud_byclass = 99,
  LB_nstud_byclass = 1,
  UB_nstud_byclass_grade = NULL,
  LB_nstud_byclass_grade = NULL,
  nstud_filter_by_grade = FALSE,
  InnerAreas = TRUE,
  ord_InnerAreas = FALSE,
  nstud_check = TRUE,
  nstud_check_registry = "Any",
  BroadBand_impute_missing = TRUE,
  Date = as.Date(paste0(substr(year.patternA(Year), 1, 4), "-09-01")),
  NA_autoRM = NULL,
  input_Invalsi_IS = NULL,
  input_Registry = NULL,
  input_SchoolBuildings = NULL,
  input_nstud = NULL,
  input_School2mun = NULL,
  input_AdmUnNames = NULL,
  input_InnerAreas = NULL,
  input_teachers4student = NULL,
  input_nteachers = NULL,
  input_BroadBand = NULL,
  autoAbort = FALSE
)

```

**Arguments**

Year	Numeric or Character. The relevant school year. Available in the formats: 2023, "2022/2023", 202223, 20222023. Important: if input datasets are plugged in, please select the same Year argument used to download the input data. 2023 by default.
level	Character. The administrative level of detail at which data must be aggregated. Either "LAU"/"Municipality" or "NUTS-3"/"Province". "LAU" by default.
conservative	Logical. If FALSE, only the schools included in all the datasets are taken as input. TRUE by default.
Invalsi	Logical. Whether the Invalsi census data must be included (see <a href="#">Get_Invalsi_IS</a> ). TRUE by default.
SchoolBuildings	Logical. Whether the school buildings dataset must be included (see <a href="#">link{Get_DB_MIUR, Util_DB_MIUR_num}</a> ). TRUE by default.
nstud	Logical. Whether the students number per class must be included (see <a href="#">Get_nstud</a> ). TRUE by default.
nteachers	Logical. Whether the number of teachers by province must be included (see <a href="#">link{Get_nteachers_prov}</a> ). TRUE by default.
BroadBand	Logical. Whether the broadband availability in schools must be included (see <a href="#">Get_BroadBand</a> ). TRUE by default
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
show_col_types	Logical. If TRUE, if the verbose argument is also TRUE, the columns of the raw dataset are shown during the download. FALSE by default.
Invalsi_subj	Character. If Invalsi == TRUE, the school subject(s) to include, among "Englis_listening"/"ELI", "English_reading"/"ERE", "Italian"/"Ita" and "Mathematics"/"MAT". All four by default.
Invalsi_grade	Numeric. If Invalsi == TRUE, the educational grade to choose. Either 2 (2nd year of primary school), 5 (last year of primary school), 8 (last year of middle shcool), 10 (2nd year of high school) or 13 (last year of school). All by default.
Invalsi_WLE	Logical. Whether to express Invalsi scores as averagev WLE score rather that the percentage of sufficient tests, if both are Invalsi_grade is either or 2 5. FALSE by default
SchoolBuildings_certifications	Logical. If the school buldings database has to be downloaded, whether to include safety certifications. Only relevant from schol year 2020/21 onwards (see <a href="#">Get_DB_MIUR</a> ). FALSE by default
SchoolBuildings_include_numerics	Logical. Whether to include strictly numeric variables alongside with Boolean ones in the school buildings database (see <a href="#">Util_DB_MIUR_num</a> ). TRUE by default.
SchoolBuildings_include_qualitatives	Logical. Whether to include qualitative variables alongside with Boolean ones in the school buildings database (see <a href="#">Util_DB_MIUR_num</a> ). FALSE by default.

- SchoolBuildings\_row\_cutout  
Logical. Whether to filter out rows including missing fields in the school buildings database (see [Util\\_DB\\_MIUR\\_num](#)). FALSE by default.
- SchoolBuildings\_col\_cut\_thresh  
Numeric. The threshold of missing values allowed for each variable in the school buildings database (see [Util\\_DB\\_MIUR\\_num](#)). If a variable has a higher number of missing observations, then it is cut out. 20.000 by default. Warning: if the option SchoolBuildings\_row\_cutout is active, please select a lower threshold (e.g. 1000)
- SchoolBuildings\_flag\_outliers  
Logical. Whether to assign NA to outliers in numeric variables; see [Util\\_DB\\_MIUR\\_num](#) for more details. TRUE by default.
- SchoolBuildings\_count\_missing  
Logical. Whether the function should return the percentage of NAs in the input school buildings database (see also [Group\\_DB\\_MIUR](#)). FALSE by default.
- nstud\_imputation\_thresh  
Numeric. If `nstud_missing_to_1 == TRUE`, the minimum threshold below which the number of classes is imputed to 1 if missing; see also [Util\\_nstud\\_wide](#). 19 by default.
- nstud\_missing\_to\_1  
Numeric. If `nstud == TRUE`, whether the number of classes should be imputed to 1 when it is missing and the number of students is below a threshold (argument `nstud_imputation_thresh`, see [Util\\_nstud\\_wide](#)). FALSE by default.
- UB\_nstud\_byclass  
Numeric. Either a unique value for all school orders, or a vector of three order-specific values in the order: primary, middle, high. If focus is on class size, the upper limit of the acceptable school-level (if `filter_by_grade == FALSE`) or grade-level (otherwise) average of the number of students by class. If a whole school or any grade in a school respectively has a higher number of students by class, the record is considered an outlier and filtered out. 99 by default, i.e. no restriction is made. Please notice that boundaries are included in the acceptance interval.
- LB\_nstud\_byclass  
Numeric. Either a unique value for all school orders, or a vector of three order-specific values in the order: primary, middle, wide. If focus is on class size, the lower limit of the acceptable school-level (if `filter_by_grade == FALSE`) or grade\_level (otherwise) average of the number of students by class. If a whole school or any grade in a school respectively has a smaller number of students by class, the record is considered an outlier and filtered out. 1 by default. Please notice that boundaries are included in the acceptance interval.
- UB\_nstud\_byclass\_grade  
Numeric. IF `filter_by_grade == TRUE`, the upper limit of the acceptable grade-level average class size. If NULL it is set equal to `UB_nstud_byclass`. NULL by default.
- LB\_nstud\_byclass\_grade  
Numeric. IF `filter_by_grade == TRUE`, the lower limit of the acceptable grade-level average class size. If NULL it is set equal to `LB_nstud_byclass`. NULL by default.

nstud_filter_by_grade	Logical. If focus is on class size, whether to remove all school grades with average class size outside of the acceptance boundaries. FALSE by default.
InnerAreas	Logical. Whether the percentage of schools belonging to inner/internal areas must be included (see <a href="#">Get_InnerAreas</a> ). TRUE by default.
ord_InnerAreas	Logical. If check == TRUE and InnerAreas == TRUE, whether the Inner areas classification should be treated as an ordinal variable rather than as a categorical one (see <a href="#">Get_InnerAreas</a> for the classification). FALSE by default.
nstud_check	Logical. If nstud == TRUE, whether to check the students number availability across all school included in the school registries (see <a href="#">Util_Check_nstud_availability</a> ). TRUE by default.
nstud_check_registry	Character. If nstud == TRUE and nstud_check == TRUE, the school registries whose availability has to be checked. Either "Registry_from_buildings" (buildings registry), "Registry_from_registry" (proper registry), "Any" or "Both". "Any" by default.
BroadBand_impute_missing	Whether the schools not included in the Broadband dataset must be considered in the total of schools (i.e. the denominator to the Broadband availability indicator). TRUE by default.
Date	Character or Date. The threshold date to broadband activation to consider it activated for a school, i.e. the date before which the works of broadband activation must be finished in order to consider a school as provided with the broadband. By default, September 1st at the beginning of the school year.
NA_autoRM	Logical. Either TRUE, FALSE or NULL. If TRUE, the values missing in a single dataset are automatically deleted from the final DB. If FALSE, the missing observations are kept automatically. If NULL, the choice is left to the user by an interactive menu. NULL by default.
input_Invalsi_IS	Object of class tbl_df, tbl and data.frame. If INVALSI == TRUE, the raw Invalsi survey data, obtained as output of the <a href="#">Get_Invalsi_IS</a> function. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default
input_Registry	Object of class tbl_df, tbl and data.frame. The school registry corresponding to the year in scope, obtained as output of the function <a href="#">Get_Registry</a> . If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default
input_SchoolBuildings	Object of class tbl_df, tbl and data.frame. If SchoolBuildings == TRUE, the raw school buildings dataset obtained as output of the function <a href="#">Get_DB_MIUR</a> . If NULL, it will be downloaded automatically but not saved in the global environment. NULL by default.
input_nstud	Object of class list, including two objects of class tbl_df, tbl and data.frame. If nstud == TRUE, the students and classes counts, obtained as output of the function <a href="#">Get_nstud</a> with default filename parameter. If NULL, the function will download it automatically but it will not be saved in the global environment. NULL by default.

<code>input_School2mun</code>	Object of class <code>list</code> with elements of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . If <code>nstud == TRUE</code> , the mapping from school codes to municipality (and province) codes. Needed only if <code>check == TRUE</code> , obtained as output of the function <a href="#">Get_School2mun</a> . If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default.
<code>input_AdmUnNames</code>	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the function <a href="#">Get_AdmUnNames</a> . If necessary, the ISTAT file including all the codes and the names of the administrative units for the year in scope. Required either if <code>nstud == TRUE &amp; nstud_check == TRUE</code> , or if <code>SchoolBuildings == TRUE</code> , <code>input_DB_MIUR</code> is not provided, and the school year is one of 2015/16, 2017/18 or 1018/19. If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default.
<code>input_InnerAreas</code>	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . If <code>InnerAreas == TRUE</code> , the classification of peripheral municipalities, obtained as output of the function <a href="#">Get_InnerAreas</a> . If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default.
<code>input_teachers4student</code>	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . If <code>nteachers == TRUE</code> and <code>nstud == TRUE</code> , the number of teachers for students by province. Please notice that this object cannot be considered a substitute for the number of students by class since it provides no information on the number of schools in single educational grades but only at the school order level. Obtained as output of the function <a href="#">Group_teachers4stud</a> . If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default.
<code>input_nteachers</code>	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . If <code>nteachers == TRUE</code> , the number of teachers by province, obtained as output of the function <a href="#">Get_nteachers_prov</a> . If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default.
<code>input_BroadBand</code>	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . If <code>BroadBand == TRUE</code> , the raw Broadband connection dataset obtained as output of the function <a href="#">Get_BroadBand</a> . If <code>NULL</code> , it will be downloaded automatically but not saved in the global environment. <code>NULL</code> by default.
<code>autoAbort</code>	Logical. In case any data must be retrieved, whether to automatically abort the operation and return <code>NULL</code> in case of missing internet connection or server response errors. <code>FALSE</code> by default.

**Value**

An object of class `tbl_df`, `tbl` and `data.frame`

**See Also**

[Util\\_DB\\_MIUR\\_num](#), [Group\\_DB\\_MIUR](#), [Group\\_nstud](#), [Util\\_Check\\_nstud\\_availability](#), [Get\\_School2mun](#) for similar arguments.

## Examples

```
DB23_prov <- Set_DB(Year = 2023, level = "NUTS-3", Invalsi_grade = c(5, 8, 13),
  Invalsi_subj = "Italian", nteachers = FALSE, BroadBand = FALSE,
  SchoolBuildings_count_missing = FALSE, NA_autoRM = TRUE,
  input_SchoolBuildings = example_input_DB23_MIUR[, -c(11:18, 10:27)],
  input_Invalsi_IS = example_Invalsi23_prov,
  input_nstud = example_input_nstud23,
  input_InnerAreas = example_InnerAreas,
  input_School2mun = example_School2mun23,
  input_AdmUnNames = example_AdmUnNames20220630)
```

```
DB23_prov
```

```
summary(DB23_prov[, -c(22:62)])
```

---

```
Util_Check_nstud_availability
```

*Check how many schools in the school registries are included in the students count dataframe*

---

## Description

This function checks for which schools listed in the two registries (the buildings registry and the properly said schools registry) the count of students is available. The first registry is referred to as Registry\_from\_buildings and the second one as Registry\_from\_registry.

## Usage

```
Util_Check_nstud_availability(
  data,
  Year,
  cutout = c("IC", "IS", "NR"),
  verbose = TRUE,
  ggplot = TRUE,
  toplot_registry = "Any",
  InnerAreas = TRUE,
  ord_InnerAreas = FALSE,
  input_Registry = NULL,
  input_InnerAreas = NULL,
  input_Prov_shp = NULL,
```

```

    input_AdmUnNames = NULL,
    input_School2mun = NULL,
    autoAbort = FALSE
  )

```

## Arguments

data	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the <a href="#">Util_nstud_wide</a> function
Year	Numeric or character value. Reference school year. Available in the formats: 2023, "2022/2023", 202223, 20222023.
cutout	Character. The types of schools not to be taken into account (because not relevant or because they are out of scope in the students number section). By default <code>c("IC", "IS", "NR")</code> , i.e. the check does not regard comprehensive institutes, superior institutes, and all the schools that cannot be classified either as primary, middle or high schools.
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
ggplot	Logical. If TRUE, the function displays a static map of the availability of the students number by province (but it does not save the ggplot object into the global environment). TRUE by default.
toplot_registry	Character. If the ggplot option is chosen, the students number availability of which registry must be plotted; either "Registry_from_buildings", "Registry_from_registry", "Any" or "Both". "Any" by default.
InnerAreas	Logical. Whether it must be checked if municipalities belong to inner areas or not. TRUE by default.
ord_InnerAreas	Logical. Whether the inner areas classification should be treated as an ordinal variable rather than as a categorical one (see <a href="#">Get_InnerAreas</a> for the classification). FALSE by default.
input_Registry	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the function <a href="#">Get_Registry</a> The school registry from the registry section. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default
input_InnerAreas	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The classification of peripheral municipalities, obtained as output of the <a href="#">Get_InnerAreas</a> function. Needed only if the InnerAreas option is chosen. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default
input_Prov_shp	Object of class <code>sf</code> , <code>tbl_df</code> , <code>tbl</code> , <code>data.frame</code> . The relevant shapefile of Italian municipalities, if the ggplot option is chosen. If NULL it is downloaded automatically but not saved in the global environment. NULL by default.
input_AdmUnNames	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> , obtained as output of the function <a href="#">Get_AdmUnNames</a> The ISTAT file including all the codes and the names of

the administrative units for the year in scope. Only needed if the argument `input_School2mun` is NULL and has to be computed. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default.

`input_School2mun`

Object of class `list` with elements of class `tbl_df`, `tbl` and `data.frame`, obtained as output of the function `Get_School2mun`. The mapping from school codes to municipality (and province) codes. If NULL, it will be downloaded automatically, but not saved in the global environment. NULL by default.

`autoAbort`

Logical. In case any data must be retrieved, whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

### Value

An object of class `list` including two elements:

- `$Municipality_data`
- `$Province_data`

Both the elements are objects of class `list` including four elements:

- `$Registry_from_buildings`: object of class of class `tbl_df`, `tbl` and `data.frame`: the availability of the number of students in the schools listed in the buildings section.
- `$Registry_from_registry`: object of class of class `tbl_df`, `tbl` and `data.frame`: the availability of the number of students in the schools listed in the registry section.
- `$Any`: object of class of class `tbl_df`, `tbl` and `data.frame`: the availability of the number of students in the schools listed anywhere.
- `$Both`: object of class of class `tbl_df`, `tbl` and `data.frame`: the availability of the number of students in the schools listed in both sections.

### Source

**Buildings Registry; Schools Registry**

### Examples

```
nstud23 <- Util_nstud_wide(example_input_nstud23, verbose = FALSE)

Util_Check_nstud_availability(nstud23, Year = 2023,
  input_Registry = example_input_Registry23, InnerAreas = FALSE,
  input_School2mun = example_School2mun23, input_Prov_shp = example_Prov22_shp)
```

---

Util_DB_MIUR_num	<i>Convert the raw school buildings data to numeric or Boolean variables</i>
------------------	--

---

## Description

This function transforms the output variables of the [Get\\_DB\\_MIUR](#) into Boolean or Numeric. Additionally, it removes the columns with an excessive number of missing observations (20.000 by default), and if required it may also delete the rows including missing fields. In this case, it is possible to keep track of the deleted rows.

## Usage

```
Util_DB_MIUR_num(
  data = NULL,
  include_numerics = TRUE,
  include_qualitatives = FALSE,
  row_cutout = FALSE,
  track_deleted = TRUE,
  verbose = TRUE,
  col_cut_thresh = 20000,
  flag_outliers = TRUE,
  autoAbort = FALSE,
  ...
)
```

## Arguments

<code>data</code>	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . Input data obtained through the function <a href="#">Get_DB_MIUR</a> . If <code>NULL</code> it will be downloaded automatically with the appropriate arguments, but not saved in the global environment. <code>NULL</code> by default.
<code>include_numerics</code>	Logical. Whether to include strictly numeric variables alongside with Boolean ones. <code>TRUE</code> by default.
<code>include_qualitatives</code>	Logical. Whether to include qualitative variables alongside with Boolean ones. <code>FALSE</code> by default.
<code>row_cutout</code>	Logical. Whether to filter out rows including missing fields. <code>FALSE</code> by default.
<code>track_deleted</code>	Logical. If <code>TRUE</code> , the function returns the names of the school not included in the output dataframe. <code>TRUE</code> by default.
<code>verbose</code>	Logical. If <code>TRUE</code> , the user keeps track of the main underlying operations. <code>TRUE</code> by default.
<code>col_cut_thresh</code>	Numeric. The threshold of missing values allowed for each variable. If a variable as a higher number of missing observations, then it is cut out. 20.000 by default. Warning: if the option <code>row_cutout</code> is active, please select a lower threshold (e.g. 1000)

flag_outliers	Logical. Whether to assign NA to outliers in numeric variables. TRUE by default.
autoAbort	Logical. In case any data must be retrieved, whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.
...	Additional arguments to the function Get_DB_MIUR if data is not provided.

## Details

The outliers to be set to NA if flag\_outliers is active are defined as follows: School area or free area surface of less than 50 squared meters, building volume of less than 150 cubic meters, 0 floors in the building.

## Value

If track\_deleted == TRUE, An object of class list including two objects:

- \$data: object of class tbl\_df, tbl and data.frame, the output dataframe.
- \$deleted: object of class tbl\_df, tbl and data.frame. The school IDs of the deleted units.

If track\_deleted == FALSE, the output is only the first element of the list.

## Examples

```
library(magrittr)

DB23_MIUR_num <- example_input_DB23_MIUR %>% Util_DB_MIUR_num(track_deleted = FALSE)

DB23_MIUR_num[, -c(1,4,6,8,9,10)]
summary(DB23_MIUR_num)
```

---

Util_Invalsi_filter	<i>Filter the Invalsi data by subject, school grade and year.</i>
---------------------	---

---

## Description

This function filters the database of Invalsi scores (see [Get\\_Invalsi\\_IS](#)) by school year, education grade and subject and returns a dataframe in wide format. Each row corresponds to one territorial unit (either municipality or province); the numerical variables are three (the mean score, the score's standard deviation and the students coverage percentage) for each selected subject.

**Usage**

```
Util_Invalsi_filter(
  data = NULL,
  subj = c("ELI", "ERE", "ITA", "MAT"),
  grade = 8,
  level = "LAU",
  WLE = FALSE,
  Year = 2023,
  verbose = TRUE,
  autoAbort = FALSE
)
```

**Arguments**

data	Object of class <code>tbl_df</code> , <code>tbl</code> and <code>data.frame</code> . The raw Invalsi survey data that has to be filtered, obtained as output of the <a href="#">Get_Invalsi_IS</a> function. If <code>NULL</code> , it will be downloaded automatically, but not saved in the global environment. <code>NULL</code> by default
subj	Character. The school subject(s) to include, among "Englis_listening"/"ELI", "English_reading"/"ERE", "Italian"/"ITA" and "Mathematics"/"MAT". All four by default.
grade	Numeric. The school grade to chose. Either 2 (2nd year of primary school), 5 (last year of primary school), 8 (last year of middle shcool), 10 (2nd year of high school) or 13 (last year of school). 8 by default
level	Character. The level of aggregation of Invalsi census data. Either "NUTS-3", "Province", "LAU", "Municipality". If an input dataframe is provided, please select the same level of aggregation. "LAU" by default
WLE	Logical. Whether the variable to choose should be the average WLE score rather than the percentage of sufficient tests, if both are available. <code>FALSE</code> by default
Year	Numeric or character value. Reference school year for the data (last available is 2022/23). Available in the formats: 2022, "2021/2022", 202122, 20212022. 2023 by default
verbose	Logical. If <code>TRUE</code> , the function informs about the time needed. <code>TRUE</code> by default.
autoAbort	Logical. In case any data must be retrieved, whether to automatically abort the operation and return <code>NULL</code> in case of missing internet connection or server response errors. <code>FALSE</code> by default.

**Value**

An object of class `tbl_df`, `tbl` and `data.frame`. For all subjects and school grades, the variables indicate:

- M The mean score, either WLE or percentage of sufficient tests
- S The standard deviation of the score
- C The students coverage percentage (expressed in the scale 1 - 100)

## Examples

```
Util_Invalsi_filter(subj = c("Italian", "Mathematics"), grade = 5, level = "NUTS-3", Year = 2023,
                    WLE = FALSE, data = example_Invalsi23_prov)

Util_Invalsi_filter(subj = c("Italian", "Mathematics"), grade = 5, level = "NUTS-3", Year = 2023,
                    WLE = TRUE, data = example_Invalsi23_prov)

Invalsi23_high <- Util_Invalsi_filter(subj = "Italian", grade = c(10,13), level = "NUTS-3",
                                     Year = 2023, data = example_Invalsi23_prov)

summary(Invalsi23_high)
```

---

Util_nstud_wide	<i>Clean the raw dataframe of the number of students and arrange it in a wide format</i>
-----------------	--

---

## Description

This function rearranges the output of the [Get\\_nstud](#) function in such a way to represent the counts of students and, if required, either the number of students by class and number of classes, or the counts of students per school timetable (running time) in a unique observation per school. If the focus is on class size, this function firstly cleans the data from the outliers in terms of average number of students by class at the school level and imputates the number of classes to 1 when missing.

## Usage

```
Util_nstud_wide(
  data = NULL,
  missing_to_1 = FALSE,
  nstud_imputation_thresh = 19,
  UB_nstud_byclass = 99,
  LB_nstud_byclass = 1,
  filter_by_grade = FALSE,
  UB_nstud_byclass_grade = NULL,
  LB_nstud_byclass_grade = NULL,
  verbose = TRUE,
  autoAbort = FALSE,
  ...
)
```

**Arguments**

data	Object of class list, including two objects of class tbl_df, tbl and data.frame, obtained as output of the <a href="#">Get_nstud</a> function with the default filename parameter. If NULL, the function will download it automatically but it will not be saved in the global environment. NULL by default.
missing_to_1	Logical. If focus is on class size, whether the number of classes should be imputed to 1 when it is missing and the number of students is below a threshold (argument <code>nstud_imputation_thresh</code> ). TRUE by default.
nstud_imputation_thresh	Numeric. If focus is on class size, the minimum threshold below which the number of classes is imputed to 1 if missing, if <code>missing_to_1 == TRUE</code> . E.g. if the threshold is 19, for all the schools in which there are 19 or less students in a given grade but the number of classes for that grade is missing, the number of classes is imputed to 1. 19 by default.
UB_nstud_byclass	Numeric. Either a unique value for all school orders, or a vector of three order-specific values in the order: primary, middle, high. If focus is on class size, the upper limit of the acceptable school-level (if <code>filter_by_grade == FALSE</code> ) or grade-level (otherwise) average of the number of students by class. If a whole school or any grade in a school respectively has a higher number of students by class, the record is considered an outlier and filtered out. 99 by default, i.e. no restriction is made. Please notice that boundaries are included in the acceptance interval.
LB_nstud_byclass	Numeric. Either a unique value for all school orders, or a vector of three order-specific values in the order: primary, middle, wide. If focus is on class size, the lower limit of the acceptable school-level (if <code>filter_by_grade == FALSE</code> ) or grade_level (otherwise) average of the number of students by class. If a whole school or any grade in a school respectively has a smaller number of students by class, the record is considered an outlier and filtered out. 1 by default. Please notice that boundaries are included in the acceptance interval.
filter_by_grade	Logical. If focus is on class size, whether to remove all school grades with average class size outside of the acceptance boundaries. FALSE by default.
UB_nstud_byclass_grade	Numeric. IF <code>filter_by_grade == TRUE</code> , the upper limit of the acceptable grade-level average class size. If NULL it is set equal to <code>UB_nstud_byclass</code> . NULL by default.
LB_nstud_byclass_grade	Numeric. IF <code>filter_by_grade == TRUE</code> , the lower limit of the acceptable grade-level average class size. If NULL it is set equal to <code>LB_nstud_byclass</code> . NULL by default.
verbose	Logical. If TRUE, the user keeps track of the main underlying operations. TRUE by default.
autoAbort	Logical. In case any data must be retrieved, whether to automatically abort the operation and return NULL in case of missing internet connection or server response errors. FALSE by default.

... Arguments to [Get\\_nstud](#), needed if data is not provided.

### Details

In the example, we compare the dataframe obtained with the default settings and the one imposed setting narrow inclusion criteria

### Value

An object of class `tbl_df`, `tbl` and `data.frame`

### Examples

```
nstud.default <- Util_nstud_wide(example_input_nstud23)

nstud.narrow <- Util_nstud_wide(example_input_nstud23,
  UB_nstud_byclass = 35, LB_nstud_byclass = 5 )

nrow(nstud.default)
nrow(nstud.narrow)

nstud.default

summary(nstud.default)
```

# Index

## \* datasets

- example\_AdmUnNames20220630, [2](#)
- example\_InnerAreas, [3](#)
- example\_input\_DB23\_MIUR, [4](#)
- example\_input\_nstud23, [6](#)
- example\_input\_Registry23, [7](#)
- example\_Invalsi23\_prov, [8](#)
- example\_Prov22\_shp, [9](#)
- example\_School2mun23, [10](#)

- example\_AdmUnNames20220630, [2](#)
- example\_InnerAreas, [3](#)
- example\_input\_DB23\_MIUR, [4](#)
- example\_input\_nstud23, [6](#)
- example\_input\_Registry23, [3](#), [7](#)
- example\_Invalsi23\_prov, [8](#)
- example\_Prov22\_shp, [9](#)
- example\_School2mun23, [10](#)

- Get\_AdmUnNames, [3](#), [11](#), [14](#), [23](#), [29](#), [43](#), [45](#)
- Get\_BroadBand, [12](#), [40](#), [43](#)
- Get\_DB\_MIUR, [6](#), [13](#), [22](#), [36](#), [40](#), [42](#), [47](#)
- Get\_InnerAreas, [3](#), [4](#), [15](#), [26](#), [28](#), [42](#), [43](#), [45](#)
- Get\_Invalsi\_IS, [9](#), [16](#), [34](#), [40](#), [42](#), [48](#), [49](#)
- Get\_nstud, [7](#), [18](#), [28](#), [30](#), [40](#), [42](#), [50–52](#)
- Get\_nteachers\_prov, [19](#), [30](#), [43](#)
- Get\_Registry, [8](#), [14](#), [21](#), [22](#), [23](#), [28](#), [42](#), [45](#)
- Get\_School2mun, [11](#), [22](#), [28](#), [43](#), [46](#)
- Get\_Shapefile, [10](#), [23](#)
- Group\_DB\_MIUR, [25](#), [37](#), [41](#), [43](#)
- Group\_nstud, [27](#), [31](#), [43](#)
- Group\_teachers4stud, [30](#), [43](#)

- Map\_DB, [31](#)
- Map\_Invalsi, [32](#), [34](#)
- Map\_School\_Buildings, [32](#), [36](#)

- Set\_DB, [31–33](#), [38](#)

- Util\_Check\_nstud\_availability, [28](#), [42](#), [43](#), [44](#)

- Util\_DB\_MIUR\_num, [14](#), [25](#), [26](#), [40](#), [41](#), [43](#), [47](#)
- Util\_Invalsi\_filter, [48](#)
- Util\_nstud\_wide, [28](#), [29](#), [41](#), [45](#), [50](#)