

Package ‘ags’

July 22, 2025

Title Crosswalk Municipality and District Statistics in Germany

Version 1.0.1

Description

Construct time series for Germany's municipalities (Gemeinden) and districts (Kreise) using a annual crosswalk constructed by the Federal Office for Building and Regional Planning (BBSR).

URL <https://sumtxt.github.io/ags/>

BugReports <https://github.com/sumtxt/ags/issues>

Imports stats ($\geq 4.2.0$), rlang ($\geq 1.0.4$), dplyr ($\geq 1.0.9$), stringr ($\geq 1.4.0$),

Depends R ($\geq 4.2.0$)

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

Suggests knitr, rmarkdown, ggplot2

VignetteBuilder knitr

NeedsCompilation no

Author Moritz Marbach [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-7101-2821>>)

Maintainer Moritz Marbach <m.marbach@ucl.ac.uk>

Repository CRAN

Date/Publication 2023-02-21 11:40:11 UTC

Contents

ags_dist	2
btw_sn	3
code_bundesland	3
format_ags	4
xwalk_ags	5

ags_dist	<i>Defines a distance metric for the AGS</i>
----------	--

Description

Defines a distance metric for the AGS

Usage

```
ags_dist(x, y, landw = 10^6, kreisw = 10^3, gemw = 1, ceiling = 99999999)
```

Arguments

x, y	vectors of AGS values
landw	weight of the Bundesland (Land) integers
kreisw	weight of the Kreis (district) integers
gemw	weight of the Gemeinde (municipality) integers
ceiling	truncate all distances at this value

Details

The distance metric is defined as

$$\text{abs}(x[1:2] - y[1:2]) * \text{landw} + \text{abs}(x[3:5] - y[3:5]) * \text{kreisw} + \text{abs}(x[6:8] - y[6:8]) * \text{gemw},$$

where $z[a:b]$ means all digits between a and b for integer z.

With the default weights, this sum is the absolute difference between x and y.

Value

A numerical vector.

Examples

```
ags_dist(14053, 14059)
```

btw_sn	<i>Number of voters and valid votes in Saxony (1994-2017)</i>
--------	---

Description

The dataset includes the number of voters and valid votes in all federal elections (Bundestagswahlen) across districts in Saxony.

Usage

```
btw_sn
```

Format

A data frame with 155 rows and 4 variables:

district AGS of the district.

year Election year.

voters Number of eligible voters.

valid Number of valid votes.

Source

<https://www.regionalstatistik.de>

code_bundesland	<i>Convert the Name or the AGS of a Bundesland</i>
-----------------	--

Description

Convert the Name or the AGS of a Bundesland

Usage

```
code_bundesland(
  sourcevar,
  origin = "ags",
  destination = "name",
  factor = FALSE
)
```

Arguments

sourcevar	Vector which contains the codes or names to be converted.
origin	The following options are available: • ags: AGS (default). • name: Bundesland name.
destination	The following options are available: • ags: Bundesland AGS (default). • iso: The Bundesland two-character abbreviation. • name: Bundesland name. • name_eng: Bundesland name in English.
factor	If TRUE returns ordered factor.

Details

This function converts a string of Bundesland names into the AGS, the standardized (English) name, or the Bundesland abbreviation.

If origin="AGS", the first two digits will be used to identify a Bundesland. It is therefore important that sourcevar is supplied as a character vector with a leading zeros if applicable.

Value

A character vector.

See Also

[format_ags\(\)](#) for formatting AGS.

Examples

```
library(dplyr)
data(btw_sn)

btw_sn %>%
  mutate(bl=code_bundesland(district, origin="ags",
    destination="name"))
```

format_ags

Formats AGS with a Leading Zero

Description

Formats AGS with a Leading Zero

Usage

```
format_ags(ags, type, verbose = FALSE)
```

Arguments

ags	Input vector that will be coerced into an integer vector. Factor vectors are first coerced to a character vector and then to an integer vector.
type	Type of AGS supplied as ags. Three options are available: • land: Bundesland AGS (Bundeslandschlüssel, 2 digits) • district: District AGS (Kreisschlüssel, 5 digits) • municipality: Municipality AGS (Gemeindeschlüssel, 8 digits) The abbreviations l, d, and m are also accepted.
verbose	If TRUE the function outputs additional information.

Value

A character vector.

Examples

```
format_ags(c(1,14), type="land")
format_ags(c(1002,14612), type="district")
format_ags(c(01002000,14612000), type="municipality")
```

xwalk_ags

Crosswalk Municipality or District Statistics

Description

This function constructs time series of counts for Germany's municipalities (Gemeinden) and districts (Kreise).

Usage

```
xwalk_ags(
  data,
  ags,
  time,
  xwalk,
  variables = NULL,
  strata = NULL,
  weight = NULL,
  fuzzy_time = FALSE,
  verbose = TRUE
)
```

Arguments

<code>data</code>	A data frame or a data frame extension (e.g. a tibble).
<code>ags</code>	Name of the character variable (quoted) with municipality AGS (Gemeinden, 8 digits) or district AGS (Kreise, 5 digits).
<code>time</code>	Name of the variable (quoted) identifying the year (YYYY format). Values will be coerced to integers.
<code>xwalk</code>	Name of the crosswalk. The following crosswalks are available: • <code>xd19</code>, <code>xd20</code> for district-level data between 1990-2019/2020. • <code>xm19</code>, <code>xm20</code> for municipality-level data between 1990-2019/2020.
<code>variables</code>	Either a vector of names (quoted) for variables to interpolate or <code>NULL</code> to disable interpolation and return the data matched with the <code>xwalk</code> .
<code>strata</code>	Vector of variable names (quoted) or <code>NULL</code> . See details.
<code>weight</code>	Name of the interpolation weight or <code>NULL</code> . The following are available: • <code>pop</code>: Population weights. • <code>size</code>: Area weights. • <code>emp</code>: Weights based on the number of employees (1998 onwards).
<code>fuzzy_time</code>	If <code>FALSE</code> the crosswalk and the data are matched exactly by <code>ags</code> and <code>time</code> . If <code>TRUE</code> they are matched exactly by <code>ags</code> and as best as possible on <code>time</code> . See details below.
<code>verbose</code>	If <code>TRUE</code> the function outputs information on the number of matched and unmatched rows.

Details

This function facilitates the use of crosswalks constructed by the BBSR for municipalities and districts in Germany (Milbert 2010). The crosswalks map one year's set of district/municipality identifiers to later year's identifiers and provide weights to perform area or population weighted interpolation.

All data rows with NAs in either the `ags` or `time` variable are excluded. The same applies to all rows with a value in `ags` or `time` that never appears in the crosswalk.

Fuzzy matching uses the absolute difference between the year reported in the data and a crosswalk year. If there is a tie, crosswalk years from before the year reported in the data are preferred.

If area or population weighted interpolation is requested (i.e., when `variables` are supplied), the combination of the variables set in `ags`, `time` and `strata` need to uniquely identify a row in data.

Caution: Data from <https://www.regionalstatistik.de/> *sometimes* includes annual values for merged units (e.g., Städteregion Aachen, 05334) and for their former parts (Kreis Aachen, 05354 and Stadt Aachen, 05313). When such data is crosswalked with `fuzzy_time=TRUE` and interpolated, the final counts will be off by approximately factor 2. The reason is that the final output is the sum of the interpolated counts for the parts and the measured count of the merged unit.

Value

If interpolation is requested, the crosswalked and interpolated data are returned. If interpolation is not requested, the data matched with the crosswalk are returned. The following variables are added:

- row_id row number of data before matching.
- ags[*] the crosswalked AGS.
- year_xw the matched year from the crosswalk.
- [*]_conv the interpolation weight.
- diff the absolute difference between year_xw and time.

References

Milbert, Antonia. 2010. "[Gebietsreformen–politische Entscheidungen und Folgen für die Statistik.](#)" BBSR-Berichte kompakt 6/2010. Bundesinstitut für Bau-, Stadt- und Raumforschung.

Examples

```
data(btw_sn)

btw_sn_ags20 <- xwalk_ags(
  data = btw_sn,
  ags = "district",
  time = "year",
  xwalk = "xd20",
  variables = c("voters", "valid"),
  weight = "pop"
)

head(btw_sn_ags20)
```

Index

* **datasets**

btw_sn, [3](#)

ags_dist, [2](#)

btw_sn, [3](#)

code_bundesland, [3](#)

format_ags, [4](#)

format_ags(), [4](#)

xwalk_ags, [5](#)