Package ‘fude’

December 22, 2024

Type Package

Title Utilities for Fude Polygon

Version 0.3.7

Description Provides utilities to facilitate handling of Fude Polygon data downloadable from the Ministry of Agriculture, Forestry and Fisheries website <https://open.fude.maff.go.jp>.

License MIT + file LICENSE

URL https://github.com/takeshinishimura/fude, https://takeshinishimura.github.io/fude/

BugReports https://github.com/takeshinishimura/fude/issues

Encoding UTF-8

LazyData true

Depends R (>= 3.5)

Imports dplyr, DT, forcats, glue, leaflet, magrittr, purrr, rlang, sf, shiny, tidyr, units

Suggests testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/testthat/edition 3

NeedsCompilation no

Author Takeshi Nishimura [aut, cph, cre] (<https://orcid.org/0000-0003-1761-4300>)

Maintainer Takeshi Nishimura <takenishi@gmail.com>

Repository CRAN

Date/Publication 2024-12-22 05:20:02 UTC

Contents

---

bind_fude *Bind multiple Fude Polygon data*

Description

`bind_fude()` binds a list of polygon data. It also binds a list of data combined by `combine_fude()`.

Usage

```
bind_fude(...)
```

Arguments

`...`	Database lists to be combined. They should all have the same named elements.

Value

A list of `sf::sf()` object(s).

See Also

`read_fude()`, `combine_fude()`.

Examples

```
path <- system.file("extdata", "castle.zip", package = "fude")
d1 <- read_fude(path, stringsAsFactors = FALSE, quiet = TRUE)
d2 <- read_fude(path, stringsAsFactors = FALSE, quiet = TRUE)
bind_fude(d1, d2)
```

`cite_fude` *Generate Citation Text for Fude Polygon Data*

Description

Generates citation text in Japanese and English for Fude Polygon Data.

Usage

```
cite_fude(data)
```

Arguments

`data` A list or data frame containing Fude Polygon data.

Value

A list with two elements: `ja` for Japanese citation text and `en` for English citation text.

Examples

```
data <- list(fude = data.frame(issue_year = c(2021, 2020), boundary_edit_year = c(2019, 2020)))
cite_fude(data)
```

`combine_fude` *Combine the Fude Polygon data with the agricultural community boundary data*

Description

`combine_fude()` uses the agricultural community boundary data to reduce the Fude Polygon data to the community units.

Usage

```
combine_fude(data, boundary, city, kcity = "", community = "", year = NULL)
```

Arguments

`data` List of `sf::sf()` objects.

`boundary` List of one or more agricultural community boundary data provided by the MAFF.

`city` A local government name in Japanese to be extracted. In the case of overlapping local government names, this must contain the prefecture name in Japanese and the prefecture code in romaji (e.g., "Fuchu-shi, 13", "fuchu 13", "34 fuchu-shi", "34, FUCHU-CHO"). Alternatively, it could be a 6-digit local government code.

`kcity` String by regular expression. One or more former village name in Japanese to be extracted.

`community` String by regular expression. One or more agricultural community name in Japanese to be extracted.

`year` Year in the column name of the `data`. If there is more than one applicable local government code, it is required.

Value

A list of `sf::sf()` objects.

See Also

`read_fude()`.

Examples

```
path <- system.file("extdata", "castle.zip", package = "fude")
d <- read_fude(path, stringsAsFactors = FALSE)
b <- get_boundary(d)
db <- combine_fude(d, b, "\u677e\u5c71\u5e02", "\u57ce\u6771", year = 2022)
```

`community_code_table` *Community code/name correspondence table*

Description

A dataset containing codes/names of communities in Japan.

Usage

```
community_code_table
```

Format

A data frame with 149,511 rows and 13 variables:

KEY Unique community codes

PREF_NAME Prefecture names in Kanji

PREF_KANA Prefecture names in Hiragana

PREF_ROMAJI Prefecture names in Romaji

CITY_NAME City names in Kanji

CITY_KANA City names in Hiragana

CITY_ROMAJI City names in Romaji

KCITY_NAME Former village names in Kanji

RCOM_NAME Community names in Kanji

RCOM_KANA Community names in Hiragana

RCOM_ROMAJI Community names in Romaji

local_government_cd Local government codes

census_year The year of the census from which the data is derived

`extract_boundary` *Extract specified agricultural community boundary data*

Description

`extract_boundary()` extracts the specified data from the list returned by `get_boundary()`.

Usage

```
extract_boundary(boundary, city = "", kcity = "", community = "", all = FALSE)
```

Arguments

`boundary` List of one or more agricultural community boundary data provided by the MAFF.

`city` A local government name in Japanese to be extracted. In the case of overlapping local government names, this must contain the prefecture name in Japanese and the prefecture code in romaji (e.g., "Fuchu-shi, 13", "fuchu 13", "34 fuchu-shi", "34, FUCHU-CHO"). Alternatively, it could be a 6-digit local government code.

`kcity` String by regular expression. One or more former village name in Japanese to be extracted.

`community` String by regular expression. One or more agricultural community name in Japanese to be extracted.

`all` logical.

Value

A list of `sf::sf()` object(s).

See Also

`read_fude()`.

extract_fude *Extract specified Fude Polygon data*

Description

extract_fude() extracts the specified data from the list returned by read_fude().

Usage

```
extract_fude(
  data,
  year = NULL,
  city = NULL,
  kcity = "",
  community = "",
  list = TRUE
)
```

Arguments

data List of sf::sf() objects.

year Years to be extracted.

city Local government names or codes to be extracted.

kcity String by regular expression. One or more former village name in Japanese to be extracted.

community String by regular expression. One or more agricultural community name in Japanese to be extracted.

list logical. If FALSE, the object to be extracted is no longer a list.

Value

A list of sf::sf() object(s).

See Also

read_fude().

get_boundary *Get the agricultural community boundary data*

Description

get_boundary() downloads and reads one or more agricultural community boundary data provided by the MAFF.

Usage

```
get_boundary(
  data,
  year = 2020,
  census_year = 2020,
  path = NULL,
  to_wgs84 = TRUE,
  quiet = FALSE
)
```

Arguments

data	List of sf::sf() objects or one or more strings representing prefecture codes.
year	The year when the agricultural community boundary data was created.
census_year	The year of the Agricultural and Forestry Census.
path	Path to the ZIP file containing the agricultural community boundary data; use a local ZIP file instead of going looking for a ZIP file. Specify a directory containing one or more ZIP files, not the ZIP file itself.
to_wgs84	logical. Convert JGD2000 to WGS 84.
quiet	logical. Suppress information about the data to be read.

Value

A list of sf::sf() objects.

Examples

```
path <- system.file("extdata", "castle.zip", package = "fude")
d <- read_fude(path)
b <- get_boundary(d)
```

`lg_code_table` *Local government code/name correspondence table*

Description

A dataset containing codes/names of local governments in Japan.

Usage

```
lg_code_table
```

Format

A data frame with 1,992 rows and 6 variables:

lg_code Local government codes

pref_kanji Prefecture names in Kanji

city_kanji Local government names in Kanji

pref_kana Prefecture names in Katakana

city_kana Local government names in Katakana

romaji Local government names in Romaji

`ls_fude` *Itemize the structure of Fude Polygon data*

Description

`ls_fude()` lists the year and the local government names (or codes) in order to understand what is included in the list returned by `read_fude()`.

Usage

```
ls_fude(data)
```

Arguments

`data`	List of `sf::sf()` objects.

Value

A data.frame.

See Also

`read_fude()`.

`pref_code_table` *Prefecture code/name correspondence table*

Description

A dataset containing codes/names of prefectures in Japan.

Usage

```
pref_code_table
```

Format

A data frame with 47 rows and 2 variables:

pref_code Prefecture codes

pref_kanji Prefecture names in Kanji

`read_fude` *Read a Fude Polygon ZIP file*

Description

`read_fude()` reads Fude Polygon data as a list. The data can be downloaded from the MAFF website as a ZIP file, which contains one or more spatial data files, such as **GeoJSON** files (`.json` or `.geojson`) and **FlatGeobuf** files (`.fgb`). The function also works with ZIP files you created, as long as you do not change the filenames of the original files.

Usage

```
read_fude(
  path = NULL,
  pref = NULL,
  year = 2024,
  census_year = 2020,
  stringsAsFactors = TRUE,
  to_wgs84 = TRUE,
  quiet = FALSE,
  supplementary = FALSE
)
```

Arguments

`path`	Path to the ZIP file containing one or more supported spatial data files. Supported formats include `.geojson`, `.json`, and `.fgb`.
`pref`	The year when the Fude Polygon data was created.
`year`	The year when the Fude Polygon data was created.
`census_year`	The year of the Agricultural and Forestry Census.
`stringsAsFactors`	logical. Should character vectors be converted to factors?
`to_wgs84`	logical. Convert JGD2000 to WGS 84.
`quiet`	logical. Suppress information about the data to be read.
`supplementary`	logical. If TRUE, add supplementary information for each polygon.

Value

A list of `sf::sf()` objects.

Examples

```
path <- system.file("extdata", "castle.zip", package = "fude")
d <- read_fude(path, stringsAsFactors = FALSE)
```

`rename_fude` *Rename the Fude Polygon data*

Description

`rename_fude()` renames the 6-digit local government code of the list returned by `read_fude()` to the corresponding Japanese name in order to make the data human-friendly.

Usage

```
rename_fude(data, suffix = TRUE, romaji = NULL, quiet = TRUE)
```

Arguments

`data`	List of `sf::sf()` objects.
`suffix`	logical. If `FALSE`, suffixes such as "SHI" and "KU" in local government names are removed.
`romaji`	If not `NULL`, rename the local government name in romaji instead of Japanese. Romanji format is upper case unless specified. • `"title"`: Title case. • `"lower"`: Lower case. • `"upper"`: Upper case.
`quiet`	logical. Suppress information about the data to be read.

Value

A list of `sf::sf()` objects.

See Also

`read_fude()`.

Examples

```
path <- system.file("extdata", "castle.zip", package = "fude")
d <- read_fude(path, stringsAsFactors = FALSE, quiet = FALSE)
d2 <- rename_fude(d)
d2 <- rename_fude(d, suffix = FALSE)
d2 <- d |> rename_fude(romaji = "upper")
```

`shiny_fude` *Prepare Leaflet Map for Fude Polygon Data*

Description

Prepares a Leaflet map for Fude Polygon data.

Usage

```
shiny_fude(data, community = FALSE)
```

Arguments

`data`	A list or data frame containing Fude Polygon data.
`community`	A logical value indicating whether to overlay community data on the map.

Value

A Leaflet map object with Fude Polygon data with an HTML table.

Index