

Package ‘fude’

March 1, 2026

Type Package

Title Utilities for Fude Polygon

Version 0.4.0

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 (>= 4.1.0)

Imports dplyr, DT, glue, leaflet, magrittr, purrr, readxl, sf, shiny,
units

Suggests testthat (>= 3.0.0)

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation no

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

Maintainer Takeshi Nishimura <takenishi@gmail.com>

Repository CRAN

Date/Publication 2026-03-01 13:40:02 UTC

Contents

bind_fude	2
cite_fude	3
city_code_table	3
combine_fude	4
extract_boundary	5
extract_fude	6
get_boundary	6
kcity_code_table	8
lg_code_table	8
ls_fude	9
pref_code_table	10
rcom_code_table	10
read_fude	11
read_ikasudb	12
rename_fude	12
shiny_fude	13
Index	14

bind_fude	<i>Bind multiple Fude Polygon data</i>
-----------	--

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, quiet = TRUE)
d2 <- read_fude(path, 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.

`city_code_table`*City code/name correspondence table*

Description

A dataset containing codes/names of cities in Japan.

Usage

```
city_code_table
```

Format

A data frame with 1,905 rows and 9 variables:

key Unique city 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
local_government_cd Local government codes
census_year Year of the census from which the data is derived

combine_fude	<i>Combine the Fude Polygon data with the agricultural community boundary data</i>
--------------	--

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 = "", rcom = "", year = NULL)
```

Arguments

data	Fude Polygon data as returned by read_fude() .
boundary	Agricultural community boundary data as returned by get_boundary() .
city	A character vector of local government names or 6-digit local government codes to extract.
kcity	A regular expression. One or more former municipality names (in Japanese) to extract.
rcom	A regular expression. One or more agricultural community names (in Japanese) to extract.
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)
```

extract_boundary	<i>Extract specified agricultural community boundary data</i>
------------------	---

Description

`extract_boundary()` extracts specified subsets of agricultural community boundary data returned by [get_boundary\(\)](#).

Usage

```
extract_boundary(boundary, city = "", kcity = "", rcom = "", layer = FALSE)
```

Arguments

boundary	Agricultural community boundary data as returned by get_boundary() .
city	A character vector of local government names or 6-digit local government codes to extract.
kcity	A regular expression. One or more former municipality names (in Japanese) to extract.
rcom	A regular expression. One or more agricultural community names (in Japanese) to extract.
layer	Logical. If TRUE, the returned object includes not only agricultural community boundaries but also prefecture and municipality boundaries.

Value

An `sf::sf()` object.

See Also

[read_fude\(\)](#).

extract_fude	<i>Extract specified Fude Polygon data</i>
--------------	--

Description

extract_fude() extracts specified subsets of Fude Polygon data returned by [read_fude\(\)](#).

Usage

```
extract_fude(data, year = NULL, city = NULL, kcity = "", rcom = "")
```

Arguments

data	Fude Polygon data as returned by read_fude() .
year	A numeric vector of years to extract.
city	A character vector of local government names or 6-digit local government codes to extract.
kcity	A regular expression. One or more former municipality names (in Japanese) to extract.
rcom	A regular expression. One or more agricultural community names (in Japanese) to extract.

Value

An `sf::sf()` object.

See Also

[read_fude\(\)](#).

get_boundary	<i>Get the agricultural community boundary data</i>
--------------	---

Description

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

Usage

```
get_boundary(  
  data,  
  boundary_data_year = 2020,  
  rcom_year = 2020,  
  boundary_type = 1,  
  path = NULL,  
  suffix = FALSE,  
  to_wgs84 = TRUE,  
  encoding = "CP932",  
  quiet = FALSE  
)
```

Arguments

data	Either Fude Polygon data as returned by <code>read_fude()</code> , or a two-digit prefecture code.
boundary_data_year	Year when the agricultural community boundary data were created.
rcom_year	Year of the agricultural community boundary data.
boundary_type	The type of boundary data: 1 = agricultural community, 2 = former municipality, 3 = municipality.
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.
suffix	Logical. If FALSE, suffixes such as "-SHI" and "-KU" in local government names are removed.
to_wgs84	Logical. If TRUE, transform coordinates to WGS 84 (EPSG:4326).
encoding	Character encoding of the source files (e.g., "CP932").
quiet	Logical. If TRUE, suppress messages about reading progress.

Value

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

Examples

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

kcity_code_table	<i>Kcity code/name correspondence table</i>
------------------	---

Description

A dataset containing codes/names of kcities in Japan.

Usage

kcity_code_table

Format

A data frame with 12,110 rows and 10 variables:

key Unique kcity 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 city names in Kanji

local_government_cd Local government codes

census_year Year of the census from which the data is derived

lg_code_table	<i>Local government code/name correspondence table</i>
---------------	--

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

Inspect 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 Fude Polygon data as returned by [read_fude\(\)](#).

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

rcom_code_table *Community code/name correspondence table*

Description

A dataset containing codes/names of communities in Japan.

Usage

rcom_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 Year of the census from which the data is derived

read_fude	<i>Read a Fude Polygon ZIP file</i>
-----------	-------------------------------------

Description

read_fude() reads MAFF Fude Polygon data from a ZIP file and returns the layers as a list of `sf::sf()` objects. The ZIP may contain one or more spatial data files such as **GeoJSON** (.json or .geojson) and **FlatGeobuf** (.fgb). The function also works with ZIP files you created, as long as the original filenames are unchanged.

Usage

```
read_fude(
  path = NULL,
  pref = NULL,
  year = 2025,
  rcom_year = 2020,
  supplementary = FALSE,
  to_wgs84 = TRUE,
  quiet = FALSE
)
```

Arguments

path	Path to a ZIP file containing one or more supported spatial files (.geojson, .json, and .fgb).
pref	Prefecture name or a two-digit prefecture code.
year	Year when the Fude Polygon data were created.
rcom_year	Year of the agricultural community boundary data.
supplementary	Logical. If TRUE, add supplementary information for each polygon.
to_wgs84	Logical. If TRUE, transform coordinates to WGS 84 (EPSG:4326).
quiet	Logical. If TRUE, suppress messages about reading progress.

Value

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

Examples

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

read_ikasudb	<i>Read a shuraku Excel file</i>
--------------	----------------------------------

Description

read_ikasudb() reads a shuraku Excel file provided by MAFF.

Usage

```
read_ikasudb(boundary, path, na = c("-", ". . ."), zero = TRUE)
```

Arguments

boundary	Agricultural community boundary data as returned by get_boundary() .
path	Path to an .xlsx file.
na	Character vector of strings to interpret as missing values. Defaults to c("-", "\u2026").
zero	Logical. If TRUE, treat masked values ("x" and "X") as zero.

Value

An `sf::sf()` object.

rename_fude	<i>Rename the Fude Polygon data</i>
-------------	-------------------------------------

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	Fude Polygon data as returned by read_fude() .
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. <ul style="list-style-type: none"> "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, 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, height = 1000, rcom = FALSE)
```

Arguments

<code>data</code>	A list or data frame containing Fude Polygon data.
<code>height</code>	Height of the map.
<code>rcom</code>	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

* datasets

- city_code_table, 3
- kcity_code_table, 8
- lg_code_table, 8
- pref_code_table, 10
- rcom_code_table, 10

bind_fude, 2

cite_fude, 3

city_code_table, 3

combine_fude, 4

combine_fude(), 2

extract_boundary, 5

extract_fude, 6

get_boundary, 6

get_boundary(), 4, 5, 12

kcity_code_table, 8

lg_code_table, 8

ls_fude, 9

pref_code_table, 10

rcom_code_table, 10

read_fude, 11

read_fude(), 2, 4–7, 9, 12, 13

read_ikasudb, 12

rename_fude, 12

sf::sf(), 2, 4–7, 11–13

shiny_fude, 13