

Package: fude (via r-universe)

October 13, 2026

Type Package

Title Utilities for Fude Polygon

Version 0.4.1.9000

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

LazyDataCompression xz

Depends R (>= 4.2.0)

Imports dplyr, readr, readxl, sf, units

Suggests DT, leaflet, shiny, testthat (>= 3.0.0)

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libx11-dev

Repository <https://takeshinishimura.r-universe.dev>

Date/Publication 2026-10-13 05:02:12 UTC

RemoteUrl <https://github.com/takeshinishimura/fude>

RemoteRef HEAD

RemoteSha 6d09b849c6be32d71b2ee8b245f4b27525318991

Contents

bind_fude	2
cite_fude	3
city_id_table	3
city_year_id	4
combine_fude	5
extract_boundary	6
extract_fude	6
get_boundary	7
kcity_id_table	8
kcity_year_id	9
lg_code_table	10
ls_fude	10
merge_split_zip	11
pref_code_table	12
rcom_id_table	12
rcom_year_id	13
read_fude	13
read_ikasudb	14
rename_fude	15
shiny_fude	16
Index	17

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

Description

bind_fude() combines multiple Fude Polygon data objects by binding elements with the same names across inputs. It can also be used on objects that have already been combined by [combine_fude\(\)](#).

Usage

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

Arguments

... Two or more Fude Polygon data objects to combine. Named elements that appear in multiple inputs are row-bound into a single [sf::sf\(\)](#) object.

Value

A named list of [sf::sf\(\)](#) objects.

See Also

```
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	<i>Generate citation text for Fude Polygon data</i>
-----------	---

Description

cite_fude() generates citation text in Japanese and English from Fude Polygon data and related boundary data.

Usage

```
cite_fude(data)
```

Arguments

data	A Fude Polygon data object, boundary data object, or a data frame/list containing issue_year and/or boundary_data_year.
------	---

Value

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

city_id_table	<i>City ID/name correspondence table</i>
---------------	--

Description

A dataset containing codes/names of cities in Japan.

Usage

```
city_id_table
```

Format

A data frame with 1,970 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
id ID

city_year_id	<i>Community year/city ID correspondence table</i>
--------------	--

Description

A dataset containing year/ID of communities in Japan.

Usage

```
city_year_id
```

Format

A data frame with 5,753 rows and 2 variables:

rcom_year Agricultural community reference year
id Reference ID for city_id_table

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

Description

combine_fude() combines Fude Polygon data with agricultural community boundary data and returns the polygons associated with the specified municipality, former municipality, and/or agricultural community.

Usage

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

Arguments

data	A Fude Polygon data object returned by read_fude() .
boundary	Agricultural community boundary data returned by get_boundary() .
city	A character vector of municipality names or local government codes used to identify target municipalities. If NULL, all municipalities are kept.
kcity	A character vector of regular expression patterns used to match former municipality names in Japanese.
rcom	A character vector of regular expression patterns used to match agricultural community names in Japanese.
year	Numeric scalar or NULL. When multiple Fude Polygon datasets match the specified municipality, year is used to choose the target dataset.

Value

A named list of [sf::sf\(\)](#) objects.

See Also

[read_fude\(\)](#), [get_boundary\(\)](#)

Examples

```
path <- system.file("extdata", "castle.zip", package = "fude")
d <- read_fude(path)
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 subsets of agricultural community boundary data returned by [get_boundary\(\)](#) by municipality, former municipality, and/or agricultural community.

Usage

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

Arguments

boundary	Agricultural community boundary data returned by get_boundary() .
city	A character vector of municipality names or local government codes used to identify target municipalities. If NULL, all municipalities are kept.
kcity	A character vector of regular expression patterns used to match former municipality names in Japanese.
rcom	A character vector of regular expression patterns used to match agricultural community names in Japanese.
layer	Logical. If TRUE, return a list containing extracted agricultural community boundaries together with former municipality, municipality, and prefecture boundary layers.

Value

If layer = FALSE, an [sf::sf\(\)](#) object. If layer = TRUE, a named list of [sf::sf\(\)](#) objects.

See Also

[get_boundary\(\)](#)

extract_fude	<i>Extract a subset of Fude Polygon data</i>
--------------	--

Description

extract_fude() extracts a subset of Fude Polygon data returned by [read_fude\(\)](#) by year, municipality, former municipality, and/or agricultural community.

Usage

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

Arguments

data	A Fude Polygon data object returned by <code>read_fude()</code> . data may be a single data frame or a list of data frames.
year	A numeric vector of issue years to extract. If NULL, all years are kept.
city	A character vector of municipality names or local government codes used to identify target municipalities. If NULL, all municipalities are kept.
kcity	A character vector of regular expression patterns used to match former municipality names in Japanese.
rcom	A character vector of regular expression patterns used to match agricultural community names in Japanese.

Value

An `sf::sf()` object containing the extracted subset.

See Also

`read_fude()`

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

Description

`get_boundary()` downloads and reads one or more MAFF agricultural community boundary datasets and returns them as a named list of `sf::sf()` objects. The target prefectures are determined from data.

Usage

```
get_boundary(  
  data,  
  boundary_data_year = NULL,  
  rcom_year = 2025,  
  boundary_type = 1,  
  path = NULL,  
  suffix = FALSE,  
  crs = NULL,  
  encoding = "CP932",  
  quiet = FALSE  
)
```

Arguments

data	Either a Fude Polygon data object returned by <code>read_fude()</code> , or a prefecture code or Japanese prefecture name.
boundary_data_year	The year of the boundary dataset. If NULL, <code>rcom_year</code> is used.
rcom_year	The agricultural community reference year used in the MAFF file name.
boundary_type	Integer specifying the boundary level to read: 1 for agricultural community, 2 for former municipality, and 3 for municipality.
path	Path to a directory containing boundary ZIP files. If NULL, ZIP files are downloaded automatically.
suffix	Logical. If FALSE, suffixes are removed from romaji municipality names, such as "-shi" and "-ku".
crs	Coordinate reference system to transform the output data to. If NULL, the source CRS is kept.
encoding	Character encoding of the source shapefile attributes, such as "CP932".
quiet	Logical. If TRUE, suppress messages during download and reading.

Value

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

Examples

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

kcity_id_table	<i>Kcity ID/name correspondence table</i>
----------------	---

Description

A dataset containing codes/names of kcities in Japan.

Usage

```
kcity_id_table
```

Format

A data frame with 12,642 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

id ID

kcity_year_id	<i>Community year/Kcity ID correspondence table</i>
---------------	---

Description

A dataset containing year/ID of communities in Japan.

Usage

```
kcity_year_id
```

Format

A data frame with 36,375 rows and 2 variables:

rcom_year Agricultural community reference year

id Reference ID for kcity_id_table

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	<i>List the contents of Fude Polygon data</i>
---------	---

Description

ls_fude() summarizes the contents of a Fude Polygon data object returned by [read_fude\(\)](#). It reports the data name, issue year, local government code, number of records, and corresponding prefecture and municipality names.

Usage

```
ls_fude(data)
```

Arguments

data A Fude Polygon data object returned by [read_fude\(\)](#).

Value

A data frame with one row per combination of data name, issue year, and local government code.

See Also

[read_fude\(\)](#)

merge_split_zip	<i>Merge Split ZIP Files of MAFF Fude Polygon Data</i>
-----------------	--

Description

merge_split_zip() combines split ZIP files of Fude Polygon data provided by MAFF into a single ZIP archive.

Usage

```
merge_split_zip(path, output_dir = getwd(), quiet = FALSE)
```

Arguments

path	Path to the first part (.zip.001) of a split ZIP archive.
output_dir	Directory in which to save the merged ZIP file. Defaults to the current working directory.
quiet	Logical. If TRUE, suppress progress messages.

Details

All parts of the split ZIP archive (e.g., .zip.001, .zip.002, and .zip.003) must be downloaded to the same directory before running this function.

The files are concatenated in numerical order to reconstruct the original ZIP archive. The resulting ZIP file is not extracted. The original split files are preserved.

Value

Invisibly returns a character string containing the path to the merged ZIP file.

Examples

```
## Not run:  
merge_fude_zip("MB0001_2026_2025_38.zip.001")  
  
## End(Not run)
```

pref_code_table	<i>Prefecture code/name correspondence table</i>
-----------------	--

Description

A dataset containing codes/names of prefectures in Japan.

Usage

```
pref_code_table
```

Format

A data frame with 47 rows and 4 variables:

pref Prefecture codes

pref_name Prefecture names in Kanji

pref_kana Prefecture names in Hiragana

pref_romaji Prefecture names in Romaji

rcom_id_table	<i>Community ID/name correspondence table</i>
---------------	---

Description

A dataset containing codes/names of communities in Japan.

Usage

```
rcom_id_table
```

Format

A data frame with 154,069 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

kcitv_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
id ID

rcom_year_id	<i>Community year/ID correspondence table</i>
--------------	---

Description

A dataset containing year/ID of communities in Japan.

Usage

```
rcom_year_id
```

Format

A data frame with 448,296 rows and 2 variables:

rcom_year Agricultural community reference year
id Reference ID for rcom_id_table

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

Description

read_fude() reads MAFF Fude Polygon data from a ZIP file and returns the layers as a named list of `sf::sf()` objects. If path is not supplied, the function downloads the ZIP file for the specified prefecture and year using pref, year, and rcom_year.

The ZIP archive may contain one or more supported spatial files in GeoJSON (.json or .geojson) or FlatGeobuf (.fgb) format. The function also works with ZIP files created manually, provided that the original file names are preserved.

Usage

```

read_fude(
  path = NULL,
  pref = NULL,
  year = NULL,
  rcom_year = NULL,
  crs = NULL,
  supplementary = FALSE,
  quiet = FALSE
)

```

Arguments

path	Path to a ZIP file containing one or more supported spatial files. If NULL, the file is downloaded automatically from the MAFF website using pref, year, and rcom_year.
pref	Prefecture name or prefecture code used when downloading data. Ignored if path is supplied.
year	The Fude Polygon data year used in the download file name.
rcom_year	The agricultural community boundary year used in the download file name.
crs	Coordinate reference system to transform the output layers to. If NULL, the original CRS is kept.
supplementary	Logical. If TRUE, add supplementary columns such as land-use labels and polygon area.
quiet	Logical. If TRUE, suppress messages during download and reading.

Value

A named 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 and joins its tabular contents to agricultural community boundary data.

Usage

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

Arguments

boundary	Agricultural community boundary data, typically returned by <code>get_boundary()</code> . This can be a single boundary object or a list of boundary objects.
path	Path to an .xlsx file provided by MAFF or a .csv file.
na	Character vector of strings to interpret as missing values. Defaults to <code>c("-", "\u2026")</code> .
zero	Logical. If TRUE, treat masked values ("x" and "X") as zero before numeric conversion.

Value

An `sf::sf()` object created by joining the Excel data to boundary.

See Also

[get_boundary\(\)](#)

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

Description

`rename_fude()` renames the elements of a Fude Polygon data object returned by [read_fude\(\)](#) by replacing local government codes in the element names with corresponding municipality names, making the object easier to read.

Usage

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

Arguments

data	A Fude Polygon data object returned by read_fude() .
suffix	Logical. If FALSE, municipality suffixes are removed from renamed element names. For example, romaji suffixes such as "-shi" and "-ku" are removed when romaji is used.
romaji	Character scalar or NULL. If NULL, Japanese municipality names are used. Otherwise, municipality names are converted to romaji. Supported values are: "upper" for upper case, "title" for title case, and "lower" for lower case.
quiet	Logical. If FALSE, print the mapping from old names to new names.

Value

A Fude Polygon data object with renamed elements.

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 <- rename_fude(d, romaji = "upper")
```

`shiny_fude`*Prepare a Shiny Leaflet viewer for Fude Polygon data*

Description

`shiny_fude()` prepares a Shiny user interface and server function for interactive visualization of Fude Polygon data with `leaflet` and `DT`. The map supports polygon selection, optional agricultural community boundary overlays, and a linked attribute table.

Usage

```
shiny_fude(data, height = 1000, rcom = FALSE)
```

Arguments

<code>data</code>	A Fude Polygon data object, or a list containing <code>fude</code> and <code>rcom</code> elements. If <code>rcom = TRUE</code> , <code>data</code> must contain both polygon data in <code>data\$fude</code> and agricultural community boundary data in <code>data\$rcom</code> .
<code>height</code>	Height of the map passed to <code>leaflet::leafletOutput()</code> .
<code>rcom</code>	Logical. If <code>TRUE</code> , overlay agricultural community boundaries on the map.

Value

A list with two elements: `ui`, a Shiny UI object, and `server`, a Shiny server function.

Index

* datasets

- city_id_table, [3](#)
- city_year_id, [4](#)
- kcity_id_table, [8](#)
- kcity_year_id, [9](#)
- lg_code_table, [10](#)
- pref_code_table, [12](#)
- rcom_id_table, [12](#)
- rcom_year_id, [13](#)

bind_fude, [2](#)

cite_fude, [3](#)
city_id_table, [3](#)
city_year_id, [4](#)
combine_fude, [5](#)
combine_fude(), [2](#)

extract_boundary, [6](#)
extract_fude, [6](#)

get_boundary, [7](#)
get_boundary(), [5](#), [6](#), [14](#), [15](#)

kcity_id_table, [8](#)
kcity_year_id, [9](#)

leaflet::leafletOutput(), [16](#)
lg_code_table, [10](#)
ls_fude, [10](#)

merge_split_zip, [11](#)

pref_code_table, [12](#)

rcom_id_table, [12](#)
rcom_year_id, [13](#)
read_fude, [13](#)
read_fude(), [5–8](#), [10](#), [15](#)
read_ikasudb, [14](#)
rename_fude, [15](#)

sf::sf(), [2](#), [5–8](#), [13–15](#)
shiny_fude, [16](#)