Package 'fude'

May 8, 2026

Type Package

Title Utilities for Fude Polygon

Version 0.4.1

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 (>= 4.2.0)

Imports dplyr, readr, readxl, sf, units

Suggests DT, leaflet, shiny, 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 2026-04-05 04:50:02 UTC

Contents

`bind_fude` *Bind multiple Fude Polygon data objects*

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 *Generate citation text for Fude Polygon data*

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_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 *Combine Fude Polygon data with agricultural community boundary data*

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 *Extract specified agricultural community boundary data*

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 *Extract a subset of Fude Polygon data*

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 read_fude(). 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 *Get agricultural community boundary data*

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 = 2020,
  rcom_year = 2020,
  boundary_type = 1,
  path = NULL,
  suffix = FALSE,
  crs = NULL,
  encoding = "CP932",
  quiet = FALSE
)
```

Arguments

data	Either a Fude Polygon data object returned by read_fude(), or a prefecture code or Japanese prefecture name.
boundary_data_year	The year of the boundary dataset.
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_code_table *Kcity code/name correspondence table*

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 *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 *List the contents of Fude Polygon data*

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()`

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 *Read Fude Polygon data from a ZIP file*

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 = 2025,
  rcom_year = 2020,
  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 *Read a shuraku excel file*

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 get_boundary(). 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 c("-", "\u2026").
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

read_fude()

---

rename_fude *Rename Fude Polygon data*

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

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