Package ‘jpndistrict’

June 12, 2020

Type Package
Title Create Japanese Administration Area and Office Maps
Version 0.3.7
Date 2020-06-12
Maintainer Shinya Uryu <suika1127@gmail.com>
Description Utilizing the data that Japanese administration area provided by the National Land Numerical Information download service (<http://nlftp.mlit.go.jp/ksj/index.html>).
This package provide map data is based on the Digital Map 25000 (Map Image) published by Geospatial Information Authority of Japan (Approval No.603FY2017 information usage <http://www.gsi.go.jp>).
License MIT + file LICENSE

URL https://uribo.github.io/jpndistrict
BugReports https://github.com/uribo/jpndistrict/issues
Depends R (>= 3.5.0)
Imports curl (>= 4.3.0), googlePolylines (>= 0.7.2), dplyr (>= 1.0.0), jpmesh (>= 1.2.0), leaflet (>= 2.0.3), magrittr (>= 1.5), miniUI (>= 0.1.1), purrr (>= 0.3.3), rlang (>= 0.4.5), sf (>= 0.9.0), shiny (>= 1.4.0.2), tibble (>= 2.1.3), tidyselect (>= 0.2.5), tidyr (>= 1.0.0)
Suggests covr (>= 3.4.0), knitr (>= 1.26), lwgeom (>= 0.2-1), rvest (>= 0.3.5), testthat (>= 2.3.2)
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
NeedsCompilation no
Author Shinya Uryu [aut, cre] (<https://orcid.org/0000-0002-0493-6186>), Geospatial Information Authority of Japan [dtc] (This package data sets, National Land numerical information by the Geographical Survey Institute with the approval of Geographical Survey Institute Head (Approval No.603FY2017 information usage))
Reform input JIS code as 2 or 5 character length.

description

Reform input JIS code as 2 or 5 character length.

usage

code_reform(jis_code)

arguments

jis_code jis code for prefecture and city identical number. If prefecture, must be from 1 to 47. If city, range of 5 digits.

note

The code_reform function was added in version 0.3.2.9000
code_validate

**Examples**

```r
code_validate(jis_code = "05")
code_validate(jis_code = 33101)
code_validate(jis_code = c("01", "33101"))
```

**Description**

Administration code validation

**Usage**

```r
code_validate(jis_code)
```

**Arguments**

- `jis_code` : jis code for prefecture and city identical number. If prefecture, must be from 1 to 47. If city, range of 5 digits.

**Note**

The `code_validate` function was added in version 0.3.2.9000

**Examples**

```r
code_validate(jis_code = "05")
code_validate(jis_code = 33101)
code_validate(jis_code = c("01", "33101"))
```

collect_cityarea

**Description**

Collect administration area

**Usage**

```r
collect_cityarea(path = NULL)
```

**Arguments**

- `path` : path to N03 shapefile (if already exist)
**collect_ksj_p34**  
*Collect administration office point datasets.*

**Description**
Collect administration office point datasets.

**Usage**
```r
collect_ksj_p34(path = NULL)
```

**Arguments**
- `path` : path to P34 shapefile (if already exist)

---

**collect_prefcode**  
*Get prefecture code (JIS X 0402)*

**Description**
Get prefecture code from prefecture of name or number.

**Usage**
```r
collect_prefcode(code = NULL, admin_name = NULL)
```

**Arguments**
- `code` : numeric
- `admin_name` : prefecture code for Japanese (character)

---

**district_viewer**  
*District Viewer*

**Description**
Interactive district map and information tool.

**Usage**
```r
district_viewer(color = "red")
```

**Arguments**
- `color` : polygon line color for leaflet
find_city

Examples

## Not run:
district_viewer()

## End(Not run)

---

find_city  Detect city by coordinates

Description

Detect city by coordinates

Usage

find_city(longitude, latitude, geometry = NULL, ...)

Arguments

- longitude
- latitude
- geometry: XY sfg object
- ... export parameter to other functions

Note

The find_city function was added in version 0.3.0

Examples

## Not run:
find_city(longitude = 140.1137418, latitude = 36.0533957)

# Referenced by sf geometry
library(sf)
find_city(geometry = st_point(c(136.6833, 35.05)))

## End(Not run)
find_jis_code  
*Find jis city code*

**Description**

Find jis city code

**Usage**

```r
test <- find_jis_code(pref_code, admin_name, strict = TRUE)
```

**Arguments**

- `pref_code`  
  jis code from 1 to 47
- `admin_name`  
  prefecture names (string)
- `strict`  
  matching patterns

**Value**

Identification code for cities, towns and villages (JIS X 0402:2010)

**Examples**

```r
## Not run:
find_jis_code(33,
  intToUtf8(c(20489, 25975, 24066), multiple = FALSE))
find_jis_code(33,
  enc2native(intToUtf8(c(20489, 25975, 24066), multiple = FALSE)),
  strict = FALSE)
find_jis_code(14,
  c(enc2native(intToUtf8(c(37772, 20489, 24066), multiple = FALSE)),
    enc2native(intToUtf8(c(23567, 30000, 21407, 24066), multiple = FALSE))),
  strict = FALSE) # nolint
## End(Not run)
```

find_pref  
*Detect prefecture by coordinates*

**Description**

Detect prefecture by coordinates

**Usage**

```r
find_pref(longitude, latitude, geometry = NULL, ...)
```
find_prefs

Arguments

longitude longitude
latitude latitude
geometry XY sfg object
...

Note

The find_pref function was added in version 0.3.0

Examples

## Not run:
find_pref(longitude = 130.4412895, latitude = 30.2984335)

# Referenced by sf geometry
library(sf)
find_pref(geometry = st_point(c(136.6833, 35.05)))

## End(Not run)

find_prefs

Detect prefectures by coordinates

Description

Detect prefectures by coordinates

Usage

find_prefs(longitude, latitude, geometry = NULL)

Arguments

longitude longitude
latitude latitude
geometry XY sfg object

Examples

## Not run:
find_prefs(longitude = 122.940625, latitude = 24.4520833334)
find_prefs(longitude = 140.1137418, latitude = 36.0533957)

# Referenced by sf geometry
library(sf)
find_pref(geometry = st_point(c(136.6833, 35.05)))

## End(Not run)
### jpnprefs

**Prefectural informations in Japan**

**Description**
Prefectures dataset.

**Usage**

```r
jpnprefs
```

**Format**
A data frame with 47 rows 11 variables:

- `jis_code`: jis code
- `prefecture`: prefecture names
- `capital`: capital name for prefecture
- `region`: region
- `major_island`: 
- `prefecture_en`: 
- `capital_en`: 
- `region_en`: 
- `major_island_en`: 
- `capital_latitude`: latitude for capital
- `capital_longitude`: longitude for capital

### jpn_admins

**Simple features for administration office points**

**Description**
Name and geolocations for administration offices in prefecture.

**Usage**

```r
jpn_admins(jis_code)
```

**Arguments**

- `jis_code`: jis code for prefecture and city identifiable number. If prefecture, must be from 1 to 47. If city, range of 5 digits.
**Value**

data.frame. contains follow columns jis_code, type, name, address, longitude and latitude.

**Examples**

```r
## Not run:
jpn_admins(jis_code = 17)

## End(Not run)
```

---

**jpn_cities**

Simple features for city area polygons

**Description**

City area polygon data. When an administrative name (jis_code_city) or code (jis_code_city) is specified as an argument, the target city data is extracted. If neither is given, it becomes the data of the target prefecture.

**Usage**

```r
jpn_cities(jis_code, admin_name)
```

**Arguments**

- `jis_code` : jis code for prefecture and city identifical number. If prefecture, must be from 1 to 47. If city, range of 5 digits.
- `admin_name` : administration name

**Examples**

```r
jpn_cities(jis_code = "08",
           admin_name = intToUtf8(c(12388, 12367, 12400, 24066)))

jpn_cities(jis_code = 33103)

jpn_cities(jis_code = "33103")

jpn_cities(jis_code = c(33103, 33104, 33205))

jpn_cities(jis_code = c(33103, 34107))
```
jpn_pref

Simple features for prefecture area polygon

Description

Prefecture polygon data.

Usage

jpn_pref(
  pref_code,
  admin_name,
  district = TRUE,
  download = FALSE,
  drop_sinkyokyoku = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pref_code</td>
<td>jis code from 1 to 47</td>
</tr>
<tr>
<td>admin_name</td>
<td>prefecture names (string)</td>
</tr>
<tr>
<td>district</td>
<td>logical (default \texttt{TRUE})</td>
</tr>
<tr>
<td>download</td>
<td>logical (default \texttt{FALSE}). IF \texttt{TRUE}, return raw data.</td>
</tr>
<tr>
<td>drop_sinkyokyoku</td>
<td>if \texttt{TRUE}, drop sichyo_sinkyokyoku variable (default \texttt{TRUE})</td>
</tr>
</tbody>
</table>

Details

Collect unit of prefecture simple feature data.frame objects. If download argument is \texttt{TRUE}, download administrative area data from the National Land Numeral Information Download Service (for law data).

Examples

```r
## Not run:
jpn_pref(pref_code = 33, district = FALSE)
jpn_pref(pref_code = 14, district = TRUE)
```

## End(Not run)
mesh_district

Export district’s mesh polygon

**Description**

Export district’s mesh polygon

**Usage**

```r
mesh_district(jis_code = NULL)
```

**Arguments**

- **jis_code**: JIS code for prefecture and city identification number. If prefecture, must be from 1 to 47. If city, range of 5 digits.

**Examples**

```r
## Not run:
mesh_district(jis_code = "33101")
mesh_district(jis_code = "05")
## End(Not run)
```

path_ksj_cityarea

Download KSJ N03 zip files

**Description**

Download KSJ N03 zip files

**Usage**

```r
path_ksj_cityarea(code = NULL, path = NULL)
```

**Arguments**

- **code**: prefecture code (JIS X 0402)
- **path**: path to N03 shapefile (if already exist)
prefecture_mesh  

Prefecture’s meshcode

Description

Prefectures dataset.

Usage

prefecture_mesh

Format

A simple feature data frame with 314 rows 5 variables:

- prefcode: prefecture code
- meshcode
- name
- type
- geometry

raw_bind_cityareas  

Intermediate function

Description

Intermediate function

Usage

raw_bind_cityareas(pref)

Arguments

pref  sf object (prefecture)
**read_ksj_cityarea**  Intermediate function

**Description**

Download N03 raw data files or loading if file exists.

**Usage**

```r
read_ksj_cityarea(code = NULL, path = NULL)
```

**Arguments**

- `code`: prefecture code (JIS X 0402)
- `path`: path to N03 shapefile (if already exist)

---

**read_ksj_p34**  Intermediate function

**Description**

Intermediate function

**Usage**

```r
read_ksj_p34(pref_code = NULL, path = NULL)
```

**Arguments**

- `pref_code`: prefecture code (JIS X 0402)
- `path`: path to P34 shapefile (if already exist)
which_pol_min

Internal function

Description

Internal function

Usage

which_pol_min(longitude, latitude, ...)

Arguments

| longitude   | longitude |
| latituide   | latitude  |
| ...         | export parameter to other functions |
Index

*Topic datasets
  jpnprefs, 8
  prefecture_mesh, 12

code_reform, 2
code_validate, 3
collect_cityarea, 3
collect_ksj_p34, 4
collect_prefcode, 4
district_viewer, 4

find_city, 5
find_jis_code, 6
find_pref, 6
find_prefs, 7

jpn_admins, 8
jpn_cities, 9
jpn_pref, 10
jpnprefs, 8

mesh_district, 11

path_ksj_cityarea, 11
prefecture_mesh, 12

raw_bind_cityareas, 12
read_ksj_cityarea, 13
read_ksj_p34, 13

which_pol_min, 14