

Package: jpndistrict (via r-universe)

August 10, 2024

Type Package

Title Create Japanese Administration Area and Office Maps

Version 0.3.9.9000

Date 2020-12-08

Maintainer Shinya Uryu <suika1127@gmail.com>

Description Utilizing the data that Japanese administration area provided by the National Land Numerical Information download service (<<https://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 <<https://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 (>= 2.0.0), leaflet (>= 2.0.3), magrittr (>= 1.5), memoise (>= 1.1.0), 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)

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

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

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

RemoteUrl <https://github.com/uribo/jpndistrict>

RemoteRef HEAD

RemoteSha 064b46ee1a33d9608b252d43e730a40c8ebdd078

Contents

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
jpnprefs	8
jpn_admins	8
jpn_cities	9
jpn_pref	10
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
Index	15

code_reform	<i>Reform input jis code as 2 or 5 character length.</i>
-------------	--

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

Examples

```
code_reform(c(1, "33", "08201"))
```

code_validate	<i>Administration code validation</i>
---------------	---------------------------------------

Description

Administration code validation

Usage

```
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

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

collect_cityarea	<i>Collect administration area</i>
------------------	------------------------------------

Description

Collect administration area

Usage

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

Arguments

path	path to N03 shapefile (if already exist)
------	--

collect_ksj_p34	<i>Collect administration office point datasets.</i>
-----------------	--

Description

Collect administration office point datasets.

Usage

```
collect_ksj_p34(path = NULL)
```

Arguments

path	path to P34 shapefile (if already exist)
------	--

collect_prefcode	<i>Get prefecture code (JIS X 0402)</i>
------------------	---

Description

Get prefecture code from prefecture of name or number.

Usage

```
collect_prefcode(code = NULL, admin_name = NULL)
```

Arguments

code	numeric
admin_name	prefecture code for Japanese (character)

district_viewer	<i>District Viewer</i>
-----------------	------------------------

Description

Interactive district map and information tool.

Usage

```
district_viewer(color = "red")
```

Arguments

color	polygon line color for leaflet
-------	--------------------------------

Examples

```
## Not run:  
district_viewer()  
  
## End(Not run)
```

find_city	<i>Detect city by coordinates</i>
-----------	-----------------------------------

Description

Detect city by coordinates

Usage

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

Arguments

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

Note

The find_city function was added in version 0.3.0

Examples

```
find_city(longitude = 140.1137418, latitude = 36.0533957)  
  
# Referenced by sf geometry  
library(sf)  
find_city(geometry = st_point(c(136.6833, 35.05)))
```

find_jis_code	<i>Find JIS city code</i>
---------------	---------------------------

Description

Find JIS city code

Usage

```
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

```
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
```

find_pref	<i>Detect prefecture by coordinates</i>
-----------	---

Description

Detect prefecture by coordinates

Usage

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

Arguments

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

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(130.4412895, 30.2984335)))

## End(Not run)
```

find_prefs	<i>Detect prefectures by coordinates</i>
------------	--

Description

Detect prefectures by coordinates

Usage

```
find_prefs(longitude, latitude, geometry = NULL)
```

Arguments

longitude	longitude
latitude	latitude
geometry	XY sfg object

Examples

```
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)))
```

jpnprefs	<i>Prefectural informations in Japan</i>
----------	--

Description

Prefectures dataset.

Usage

```
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 catital
- capital_longitude: longitude for catital

jpn_admins	<i>Simple features for administration office points</i>
------------	---

Description

Name and geolocations for administration offices in prefecture.

Usage

```
jpn_admins(jis_code)
```

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.
----------	---

Value

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

Examples

```
## 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

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

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.
admin_name	administration name

Examples

```
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

<code>pref_code</code>	jis code from 1 to 47
<code>admin_name</code>	prefecture names (string)
<code>district</code>	logical (default <i>TRUE</i>)
<code>download</code>	logical (default <i>FALSE</i>). IF <i>TRUE</i> , return raw data.
<code>drop_sinkyokyoku</code>	if <i>TRUE</i> , drop sichyo_sinkyokyoku variable (default <i>TRUE</i>)

Details

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

Examples

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

mesh_district	<i>Export district's mesh polygon</i>
---------------	---------------------------------------

Description

Export district's mesh polygon

Usage

```
mesh_district(jis_code = NULL, to_mesh_size = NULL)
```

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.
to_mesh_size	target mesh type. From 80km to 1km as numeric.

Examples

```
mesh_district(jis_code = "33101", to_mesh_size = 80)  
mesh_district(jis_code = "05", to_mesh_size = 80)
```

path_ksj_cityarea	<i>Download KSJ N03 zip files</i>
-------------------	-----------------------------------

Description

Download KSJ N03 zip files

Usage

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

Arguments

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

prefecture_mesh	<i>Prefecture's meshcode</i>
-----------------	------------------------------

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	<i>Intermediate function</i>
--------------------	------------------------------

Description

Intermediate function

Usage

```
raw_bind_cityareas(pref)
```

Arguments

pref	sf object (prefecture)
------	------------------------

read_ksj_cityarea	<i>Intermediate function</i>
-------------------	------------------------------

Description

Download N03 raw data files or loading if file exists.

Usage

```
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	<i>Intermediate function</i>
--------------	------------------------------

Description

Intermediate function

Usage

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

Arguments

pref_code	prefecture code (JIS X 0402)
path	path to P34 shapefile (if already exist)

<code>which_pol_min</code>	<i>Internal function</i>
----------------------------	--------------------------

Description

Internal function

Usage

```
which_pol_min(longitude, latitude, ...)
```

Arguments

<code>longitude</code>	<code>longitude</code>
<code>latitude</code>	<code>latitude</code>
<code>...</code>	export parameter to other functions

Index

* 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](#)