Package 'jpndistrict'

June 14, 2018

Type Package

Title Create Japanese Administration Area and Office Maps

Version 0.3.2

Date 2018-06-14

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://github.com/uribo/jpndistrict#readme

BugReports https://github.com/uribo/jpndistrict/issues

Depends jpmesh (>= 1.0.0), R (>= 3.1.2)

Imports dplyr (>= 0.7.4), leaflet (>= 1.1.0), magrittr (>= 1.5), miniUI (>= 0.1.1), purrr (>= 0.2.4), rlang (>= 0.1.4), sf (>= 0.5-5), shiny (>= 0.13), tibble (>= 1.3.4), tidyr (>= 0.7.2)

Suggests covr, knitr, rvest, testthat

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

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))

Repository CRAN

Date/Publication 2018-06-14 13:59:57 UTC

R topics documented:

`admins_code_validate` *Administration code varidation*

Description

Administration code varidation

Usage

```
admins_code_validate(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.

collect_cityarea *Collect administration area*

Description

Collect administration area

Usage

```
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

```
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

```
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

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

Arguments

color	polygon line color for leaflet

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	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

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

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

## End(Not run)
```

`find_pref` *Detect prefecture by coordinates*

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)

# Refrenced 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)

# Refrenced 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

```
jpnprefs
```

Format

A data frame with 47 rows 7 variables:

- jis_code: jis code
- prefecture: prefecture names
- capital: capital name for prefecture
- region: region
- major_island:
- capital_latitude: latitude for catital
- capital_longitude: longitude for catital

`jpn_admins` *Simple features for administration office points*

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 identifical 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

pref_code	jis code from 1 to 47
admin_name	prefecture names (string)
district	logical (default *TRUE*)
download	logical (default *FALSE*).
drop_sinkyokyoku	if *TRUE*, drop sichyo_sinkyokyoku variable (default *TRUE*)

Details

Collect unit of prefecture simple feature data.frame objects.. If downalod 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 *Export district's mesh polygon*

Description

Export district's mesh polygon

Usage

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

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.

Examples

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

## End(Not run)
```

path_ksj_cityarea *Download KSJ N03 zip files*

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 *Prefecture's meshcode*

Description

Prefectures dataset.

Usage

```
prefecture_mesh
```

Format

A simple feature data frame with 314 rows 6 variables:

- pref_code: prefecture code
- prefecture: name
- city_code: city code (JIS code)
- city: name
- 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

```
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

```
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
`latitude`	latitude
`...`	export parameter to other functions

Index