

# Package ‘jpndistrict’

December 7, 2017

**Type** Package

**Title** Create Japanese Administration Area and Office Maps

**Version** 0.3.0

**Date** 2017-12-06

**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>>).

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

**Imports** dplyr (>= 0.7.4), leaflet (>= 1.1.0), magrittr (>= 1.5), miniUI (>= 0.1.1), purrr (>= 0.2.4), readr (>= 1.1.1), rlang (>= 0.1.4), sf (>= 0.5-5), shiny (>= 0.13), stringi (>= 1.1.6), tibble (>= 1.3.4), tidyr (>= 0.7.2)

**Suggests** devtools, kokudosuuchi, knitr, rmarkdown, rvest, testthat (>= 1.0.2)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**URL** <https://github.com/uribo/jpndistrict>

**BugReports** <https://github.com/uribo/jpndistrict/issues>

**NeedsCompilation** no

**Author** Shinya Uryu [aut, cre] (0000-0002-0493-6186),  
Geospatial Information Authority of Japan [dct] (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** 2017-12-07 09:38:52 UTC

**R topics documented:**

bind_cityareas . . . . .	2
collect_cityarea . . . . .	3
collect_ksj_p34 . . . . .	3
collect_prefcode . . . . .	3
district_viewer . . . . .	4
find_city . . . . .	4
find_pref . . . . .	5
find_prefs . . . . .	5
jpnprefs . . . . .	6
jpn_admins . . . . .	7
jpn_cities . . . . .	7
jpn_pref . . . . .	8
mesh_district . . . . .	9
path_ksj_cityarea . . . . .	9
prefecture_mesh . . . . .	10
pref_code . . . . .	10
raw_bind_cityareas . . . . .	11
read_ksj_cityarea . . . . .	11
read_ksj_p34 . . . . .	11
which_pol_min . . . . .	12
<b>Index</b>	<b>13</b>

---

bind_cityareas	<i>Bind city area polygons to prefecture polygon</i>
----------------	--

---

**Description**

Bind city area polygons to prefecture polygon

**Usage**

```
bind_cityareas(path = NULL)
```

**Arguments**

path            path to N03 shapefile (if already exist)

---

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(pref_code = 33, color = "red")
```

**Arguments**

pref_code	prefecture code (default 33)
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, ...)
```

**Arguments**

longitude	longitude
latitude	latitude
...	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)  
  
## End(Not run)
```

---

find\_pref                      *Detect prefecture by coordinates*

---

**Description**

Detect prefecture by coordinates

**Usage**

```
find_pref(longitude, latitude, ...)
```

**Arguments**

longitude	longitude
latitude	latitude
...	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)  
  
## End(Not run)
```

---

find\_prefs                      *Detect prefectures by coordinates*

---

**Description**

Detect prefectures by coordinates

**Usage**

```
find_prefs(longitude, latitude)
```

**Arguments**

longitude	longitude
latitude	latitude

**Examples**

```
## Not run:  
find_prefs(longitude = 122.940625, latitude = 24.4520833334)  
find_prefs(longitude = 140.1137418, latitude = 36.0533957)  
  
## 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(path = NULL, jis_code)
```

**Arguments**

<code>path</code>	shapefile path
<code>jis_code</code>	jis code for prefecture and city identical number

**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**

<code>jis_code</code>	jis code for prefecture and city identical number
<code>admin_name</code>	administration name

## Examples

```
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 (integer)
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	<i>Export district's mesh polygon</i>
---------------	---------------------------------------

---

**Description**

Export district's mesh polygon

**Usage**

```
mesh_district(jis_code)
```

**Arguments**

jis_code	jis code for prefecture and city identificial number
----------	--

**Examples**

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

---

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 6 variables:

- pref\_code: prefecture code
- prefecture: name
- city\_code: city code (JIS code)
- city: name
- geometry

---

pref_code	<i>Get prefecture code from jis code</i>
-----------	--

---

**Description**

Get prefecture code from jis code

**Usage**

pref\_code(jis\_code)

**Arguments**

jis\_code      jis code

---

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

## \*Topic **datasets**

- [jpnprefs, 6](#)
  - [prefecture\\_mesh, 10](#)
- [bind\\_cityareas, 2](#)
- [collect\\_cityarea, 3](#)
- [collect\\_ksj\\_p34, 3](#)
- [collect\\_prefcode, 3](#)
- [district\\_viewer, 4](#)
- [find\\_city, 4](#)
- [find\\_pref, 5](#)
- [find\\_prefs, 5](#)
- [jpn\\_admins, 7](#)
- [jpn\\_cities, 7](#)
- [jpn\\_pref, 8](#)
- [jpnprefs, 6](#)
- [mesh\\_district, 9](#)
- [path\\_ksj\\_cityarea, 9](#)
- [pref\\_code, 10](#)
- [prefecture\\_mesh, 10](#)
- [raw\\_bind\\_cityareas, 11](#)
- [read\\_ksj\\_cityarea, 11](#)
- [read\\_ksj\\_p34, 11](#)
- [which\\_pol\\_min, 12](#)