Package ‘jpcity’

April 26, 2023

Type Package
Title Read and Convert Japanese Municipality Codes
Version 0.1.1
Description Read Japanese city codes (<https://www.e-stat.go.jp/municipalities/cities>) to get city and prefecture names, or convert to city codes at different points in time. In addition, it merges or splits wards of designated cities and gets all city codes at a specific point in time.
License MIT + file LICENSE
Encoding UTF-8
RoxygenNote 7.2.3
Depends R (>= 4.1)
Imports cli, dplyr, lubridate, pillar, purrr, rlang, stringr, tibble, vctrs
Suggests testthat (>= 3.0.0)
Config/testthat/edition 3
URL https://uchidamizuki.github.io/jpcity/
NeedsCompilation no
Author Mizuki Uchida [aut, cre]
Maintainer Mizuki Uchida <uchidamizuki@vivaldi.net>
Repository CRAN
Date/Publication 2023-04-26 14:20:02 UTC

R topics documented:

as_pref .......................................................... 2
city_code ...................................................... 2
city_convert ................................................... 3
city_desig_merge ............................................ 3
city_desig_split ............................................. 4
city_interval .................................................. 5
as_pref

**Description**

Coerce an object to a `jpcity_pref` object

**Usage**

```r
as_pref(x)
```

**Arguments**

- `x` A integer vector, a `jpcity_city` object or a `jpcity_pref` object.

**Value**

A `jpcity_pref` object.

city_code

**Description**

Get city codes

**Usage**

```r
city_code(city)
```

**Arguments**

- `city` A `jpcity_city` object.

**Value**

A character vector of city codes.
**city_convert**  
*Convert to cities at different points in time*

**Description**  
Convert to cities at different points in time

**Usage**  
```r  
city_convert(city, from, to)  
```

**Arguments**

- `city`  
  A `jpcity_city` object.

- `from`  
  A character (year, month, and day components) or date-time object of the starting date.

- `to`  
  A character (year, month, and day components) or date-time object of the ending date.

**Value**  
A list of a `jpcity_city` object.

**Examples**

```r  
city <- parse_city(c("01201", "01202"))  
city_convert(city,  
  from = "1970-04-01",  
  to = "2020-01-01")  
```

---

**city_desig_merge**  
*Merge designated city wards*

**Description**  
Merge designated city wards

**Usage**  
```r  
city_desig_merge(city, merge_tokyo = FALSE)  
```

**Arguments**

- `city`  
  A `jpcity_city` object.

- `merge_tokyo`  
  Whether to merge Tokyo special wards?
Value

A jpcity_city object.

Examples

```r
city <- parse_city(c("01101", "13101"))
city_desig_merge(city)
city_desig_merge(city, merge_tokyo = TRUE)
```

---

`city_desig_split`  
_Split designated cities into wards_

**Description**

Split designated cities into wards

**Usage**

```r
city_desig_split(city, split_tokyo = TRUE)
```

**Arguments**

- `city`  
  A jpcity_city object.
- `split_tokyo`  
  Whether to split into Tokyo special wards?

**Value**

A list of a jpcity_city object.

**Examples**

```r
city <- parse_city(c("01100", "13100"))
city_desig_split(city)
city_desig_split(city, split_tokyo = FALSE)
```
**city_interval**  
*Get city duration*

**Description**
Get city duration

**Usage**

```r
city_interval(city, intersect = FALSE)
```

**Arguments**

- `city` A `jpcity_city` object.
- `intersect` Whether to get the common part of the duration of cities.

**Value**
A `interval` vector of the duration of cities.

**city_name**  
*Get city names*

**Description**
Get city names

**Usage**

```r
city_name(city, type = c("city_desig", "city"), sep = "", kana = FALSE)
```

**Arguments**

- `city` A `jpcity_city` object.
- `type` Types of city names. By default, returns both designated city names ("city_desig") and city names ("city").
- `sep` Separator for city names.
- `kana` Whether to use hiragana or not?

**Value**
A character vector of city names.
find_city

*Find cities by string patterns*

**Description**
Find cities by string patterns

**Usage**
```r
find_city(patterns, when = NULL)
```

**Arguments**
- **patterns**
  Patterns to look for. If multiple patterns are given, find the cities that match all patterns.
- **when**
  A character (year, month, and day components) or date-time object.

**Value**
A `jpcity_city` object.

---

get_city

*Get cities at a specific point in time*

**Description**
Get cities at a specific point in time

**Usage**
```r
get_city(when)
```

**Arguments**
- **when**
  A character (year, month, and day components) or date-time object.

**Value**
A `jpcity_city` object.

**Examples**
```r
get_city("2020-01-01")
```
**is_city**  
*Test if the object is a jpcity_city object*

**Description**  
Test if the object is a jpcity_city object

**Usage**  
`is_city(x)`

**Arguments**  
`x`  
An object.

**Value**  
TRUE if the object inherits from the jpcity_city class.

**is_pref**  
*Test if the object is a jpcity_pref object*

**Description**  
Test if the object is a jpcity_pref object

**Usage**  
`is_pref(x)`

**Arguments**  
`x`  
An object.

**Value**  
TRUE if the object inherits from the jpcity_pref class.
parse_city

Parse city codes

Description
Parse city codes

Usage
parse_city(city_code, when = NULL, na = c("", "NA"))

Arguments
- city_code: A character vector of city codes.
- when: A character (year, month, and day components) or date-time object.
- na: A character vector to be treated as missing values.

Value
A jpcity_city object.

parse_pref

Parse prefecture codes or names

Description
Parse prefecture codes or names

Usage
parse_pref(pref_name)

Arguments
- pref_name: A character vector of prefecture codes or names.

Value
A jpcity_pref object.
### pref_code

<table>
<thead>
<tr>
<th>pref_code</th>
<th>Get prefecture codes</th>
</tr>
</thead>
</table>

**Description**

Get prefecture codes

**Usage**

```r
pref_code(city)
```

**Arguments**

- `city` A `jpcity_city` or `jpcity_pref` object.

**Value**

A integer vector of prefecture codes.

### pref_name

<table>
<thead>
<tr>
<th>pref_name</th>
<th>Get prefecture names</th>
</tr>
</thead>
</table>

**Description**

Get prefecture names

**Usage**

```r
pref_name(city)
```

**Arguments**

- `city` A `jpcity_city` object.

**Value**

A character vector of prefecture names.
Index

as_pref, 2

city_code, 2
city_convert, 3
city_desig_merge, 3
city_desig_split, 4
city_interval, 5
city_name, 5

find_city, 6

get_city, 6

is_city, 7
is_pref, 7

parse_city, 8
parse_pref, 8
pref_code, 9
pref_name, 9