

# Package ‘jpcity’

October 4, 2024

**Type** Package

**Title** Read and Convert Japanese Municipality Codes

**Version** 0.3.0

**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** 7.3.2

**Depends** R (>= 4.1)

**Imports** cli, dplyr, lifecycle, lubridate, pillar, purrr, rlang, stringr, tibble, vctrs

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**URL** <https://uchidamizuki.github.io/jpcity/>,  
<https://github.com/UchidaMizuki/jpcity>

**BugReports** <https://github.com/UchidaMizuki/jpcity/issues>

**NeedsCompilation** no

**Author** Mizuki Uchida [aut, cre]

**Maintainer** Mizuki Uchida <[uchidamizuki@vivaldi.net](mailto:uchidamizuki@vivaldi.net)>

**Repository** CRAN

**Date/Publication** 2024-10-04 14:30:02 UTC

## Contents

city_code . . . . .	2
city_convert . . . . .	3
city_data . . . . .	3

city_desig_merge . . . . .	4
city_desig_split . . . . .	4
city_interval . . . . .	5
city_name . . . . .	5
city_to_pref . . . . .	6
find_city . . . . .	6
get_city . . . . .	7
is_city . . . . .	7
is_city_desig . . . . .	8
is_pref . . . . .	8
parse_city . . . . .	9
parse_pref . . . . .	9
pref_code . . . . .	10
pref_data . . . . .	10
pref_name . . . . .	11
<b>Index</b>	<b>12</b>

---

city\_code

*Get city codes*

---

### Description

Get city codes

### Usage

city\_code(city)

### Arguments

city            A jpcity\_city object.

### Value

A character vector of city codes.

---

city_convert	<i>Convert to cities at different points in time</i>
--------------	--

---

**Description**

Convert to cities at different points in time

**Usage**

```
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**

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

---

city_data	<i>Get city data</i>
-----------	----------------------

---

**Description**

Get city data

**Usage**

```
city_data(city)
```

**Arguments**

city	A jpcity_city object.
------	-----------------------

**Value**

A data frame.

---

city_desig_merge	<i>Merge designated city wards</i>
------------------	------------------------------------

---

**Description**

Merge designated city wards

**Usage**

```
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**

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

---

city_desig_split	<i>Split designated cities into wards</i>
------------------	---

---

**Description**

Split designated cities into wards

**Usage**

```
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**

```
city <- parse_city(c("01100", "13100"))
city_desig_split(city)
city_desig_split(city,
  split_tokyo = FALSE)
```

---

city_interval	<i>Get city duration</i>
---------------	--------------------------

---

**Description**

Get city duration

**Usage**

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

**Arguments**

city	A <code>jpcity_city</code> object.
intersect	Whether to get the common part of the duration of cities.

**Value**

A interval vector of the duration of cities.

---

city_name	<i>Get city names</i>
-----------	-----------------------

---

**Description**

Get city names

**Usage**

```
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.

---

city_to_pref	<i>Convert city to prefecture</i>
--------------	-----------------------------------

---

**Description**

Convert city to prefecture

**Usage**

```
city_to_pref(city)
```

**Arguments**

city	A jpcity_city object.
------	-----------------------

**Value**

A jpcity\_pref object.

---

find_city	<i>Find cities by string patterns</i>
-----------	---------------------------------------

---

**Description**

Find cities by string patterns

**Usage**

```
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**

get\_city(when)

**Arguments**

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

**Value**

A jpcity\_city object.

**Examples**

```
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_city_desig	<i>Check if a city is a designated city or a ward of a designated city</i>
---------------	--

---

**Description**

Check if a city is a designated city or a ward of a designated city

**Usage**

```
is_city_desig(x, type = "city")
```

**Arguments**

x	A jpcity_city object.
type	A character vector of city types, "city" or "ward". By default, "city".

**Value**

A logical vector.

---

is_pref	<i>Test if the object is a jpcity_pref object</i>
---------	---

---

**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	<i>Parse city codes</i>
------------	-------------------------

---

**Description**

Parse city codes

**Usage**

```
parse_city(x, when = NULL, na = c("", "NA"))
```

**Arguments**

x	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 `jpccity_city` object.

---

parse_pref	<i>Parse prefecture codes or names</i>
------------	--

---

**Description**

Parse prefecture codes or names

**Usage**

```
parse_pref(x, strict = TRUE)
```

**Arguments**

x	A character vector of prefecture codes or names.
strict	A scalar logical. Whether to require the code to have 1 or 2 digits. By default, TRUE.

**Value**

A `jpccity_pref` object.

---

pref_code	<i>Get prefecture codes</i>
-----------	-----------------------------

---

**Description**

Get prefecture codes

**Usage**

```
pref_code(city)
```

**Arguments**

city            A jpcity\_city or jpcity\_pref object.

**Value**

A integer vector of prefecture codes.

---

pref_data	<i>Get pref data</i>
-----------	----------------------

---

**Description**

Get pref data

**Usage**

```
pref_data(pref)
```

**Arguments**

pref            A jpcity\_pref object.

**Value**

A data frame.

---

pref_name	<i>Get prefecture names</i>
-----------	-----------------------------

---

**Description**

Get prefecture names

**Usage**

pref\_name(city)

**Arguments**

city            A `jpcity_city` object.

**Value**

A character vector of prefecture names.

# Index

city\_code, 2  
city\_convert, 3  
city\_data, 3  
city\_desig\_merge, 4  
city\_desig\_split, 4  
city\_interval, 5  
city\_name, 5  
city\_to\_pref, 6  
  
find\_city, 6  
  
get\_city, 7  
  
is\_city, 7  
is\_city\_desig, 8  
is\_pref, 8  
  
parse\_city, 9  
parse\_pref, 9  
pref\_code, 10  
pref\_data, 10  
pref\_name, 11