

# Package ‘jpgrid’

May 8, 2026

**Type** Package

**Title** Functions for the Grid Square Codes in Japan

**Version** 0.4.0

**Description** Provides functions for grid square codes in Japan  
(<https://www.stat.go.jp/english/data/mesh/index.html>).  
Generates the grid square codes from longitude/latitude, geometries, and  
the grid square codes of different scales, and vice versa.

**License** MIT + file LICENSE

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

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

**Depends** R (>= 4.1.0)

**Imports** dplyr (>= 0.8.0), purrr (>= 1.0.0), rlang (>= 0.3.0), stars,  
sf, stringr (>= 1.4.0), tibble, tidyr (>= 1.3.0), units, vctrs,  
lifecycle, pillar, cli, tidygraph

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**NeedsCompilation** no

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

**Maintainer** Mizuki Uchida <uchidamizuki@vivaldi.net>

**Repository** CRAN

**Date/Publication** 2024-05-26 13:30:02 UTC

## Contents

bbox_to_grid . . . . .	2
coords . . . . .	3
geometry_to_grid . . . . .	3
grid_as_sf . . . . .	4
grid_as_stars . . . . .	4
grid_city_2020 . . . . .	5
grid_components . . . . .	6
grid_convert . . . . .	6
grid_distance . . . . .	7
grid_line . . . . .	7
grid_move . . . . .	8
grid_neighbor . . . . .	8
grid_neighborhood . . . . .	9
grid_subdivide . . . . .	10
is_grid . . . . .	10
jpgrid . . . . .	11
parse_grid . . . . .	11
<b>Index</b>	<b>12</b>

---

bbox_to_grid	<i>Converting bbox to grid square codes</i>
--------------	---

---

### Description

Converting bbox to grid square codes

### Usage

```
bbox_to_grid(bbox, grid_size)
```

### Arguments

bbox	A bbox.
grid_size	A grid size.

### Value

A grid vector.

---

coords	<i>Conversion between grid square codes and coordinates (longitude and latitude)</i>
--------	--

---

**Description**

Conversion between grid square codes and coordinates (longitude and latitude)

**Usage**

```
coords_to_grid(X, Y, grid_size)
```

```
grid_to_coords(grid, center = TRUE)
```

**Arguments**

X	A numeric vector of longitude.
Y	A numeric vector of latitude.
grid_size	A grid size.
grid	A grid class vector.
center	Should the center point of the grid be returned? Otherwise the end points will be returned. TRUE by default.

**Value**

coords\_to\_grid() returns a grid vector.

grid\_to\_coords() returns a tbl\_df.

---

geometry_to_grid	<i>Converting sfc geometries to grid square codes</i>
------------------	---

---

**Description**

Converting sfc geometries to grid square codes

**Usage**

```
geometry_to_grid(geometry, grid_size, options = "ALL_TOUCHED=TRUE", ...)
```

**Arguments**

geometry	A sfc vector.
grid_size	A grid size.
options	Options vector for GDALRasterize passed on to <a href="#">stars::st_rasterize()</a> .
...	Passed on to <a href="#">stars::st_rasterize()</a> .

**Value**

A list of grid vectors.

---

grid_as_sf	<i>Converting data frame containing grid square codes to sf</i>
------------	---

---

**Description**

Converting data frame containing grid square codes to sf

**Usage**

```
grid_as_sf(
  x,
  as_points = FALSE,
  crs = sf::NA_crs_,
  grid_column_name = NULL,
  ...
)
```

**Arguments**

x	A data frame or a grid.
as_points	Return the center points of the grids or not?
crs	Coordinate reference system.
grid_column_name	A scalar character.
...	passed on to <a href="#">sf::st_as_sf()</a> .

**Value**

A sf object.

---

grid_as_stars	<i>Converting data frame containing regional grids to stars</i>
---------------	---

---

**Description**

Converting data frame containing regional grids to stars

**Usage**

```
grid_as_stars(  
  x,  
  coords = NULL,  
  crs = sf::NA_crs_,  
  grid_column_name = NULL,  
  ...  
)
```

**Arguments**

x	A data frame or a grid.
coords	The column names or indices that form the cube dimensions.
crs	Coordinate reference system.
grid_column_name	A scalar character.
...	Passed on to <code>stars::st_as_stars()</code> .

**Value**

A stars object.

---

grid_city_2020	<i>List of grid square codes by Japanese municipalities</i>
----------------	---

---

**Description**

List of grid square codes by Japanese municipalities

**Usage**

```
grid_city_2020
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 462915 rows and 6 columns.

**Source**

[https://www.stat.go.jp/data/mesh/m\\_itiran.html](https://www.stat.go.jp/data/mesh/m_itiran.html)

---

grid_components	<i>Connected components of grid square codes</i>
-----------------	--

---

**Description**

Connected components of grid square codes

**Usage**

```
grid_components(grid, n = 0:1, type = NULL)
```

**Arguments**

grid	A grid vector.
n	A numeric vector of degrees. By default, 0:1.
type	A character vector of neighborhood types, "von_neumann" or "moore". By default, "von_neumann". (FALSE, default).

**Value**

A integer vector of group IDs.

---

grid_convert	<i>Convert the grid size of grid objects</i>
--------------	--

---

**Description**

Convert the grid size of grid objects

**Usage**

```
grid_convert(grid, grid_size)
```

**Arguments**

grid	A grid vector.
grid_size	A grid size.

**Value**

A grid vector.

**Examples**

```
grid_500m <- parse_grid(c("533945263", "533935863", "533945764"), "500m")
grid_convert(grid_500m, "10km")
```

---

grid_distance	<i>Distance between grid square codes</i>
---------------	---

---

**Description**

If grid and grid\_to are both vectors, the distance between grid and grid\_to is calculated. If grid is a list, The path distance of each element is calculated.

**Usage**

```
grid_distance(  
  grid,  
  grid_to = NULL,  
  close = FALSE,  
  type = c("keep_na", "ignore_na", "skip_na")  
)
```

**Arguments**

grid	A grid vector or a list of grid vector.
grid_to	A grid vector.
close	Should the path of each element be closed when grid is a list?
type	How is the NA grid treated when grid is a list? "skip_na" skips the NA grid and connects the paths. "keep_na" by default.

**Value**

A double vector.

---

grid_line	<i>Draw line segments between grid square codes</i>
-----------	---

---

**Description**

If grid and grid\_to are both vectors, the line between grid and grid\_to is drawn (using Bresenham's line algorithm). If grid is a list, The path lines for each element in the grid will be drawn.

**Usage**

```
grid_line(grid, grid_to = NULL, close = FALSE, skip_na = FALSE)
```

**Arguments**

grid	A grid vector or a list of grid vector.
grid_to	A grid vector.
close	Should the path of each element be closed when grid is a list?
skip_na	Should skip the NA grid and connects the paths? FALSE by default.

**Value**

A list of grid vectors.

---

grid_move	<i>Moving on grid square codes</i>
-----------	------------------------------------

---

**Description**

Moving on grid square codes

**Usage**

```
grid_move(grid, n_X, n_Y)
```

**Arguments**

grid	A grid vector.
n_X	Number of moving cells in the longitude direction.
n_Y	Number of moving cells in the latitude direction.

**Value**

A grid vector.

---

grid_neighbor	<i>Neighborhood grid square codes (Deprecated)</i>
---------------	--

---

**Description**

**[Deprecated]**

**Usage**

```
grid_neighbor(grid, n = 1L, moore = TRUE, simplify = TRUE)
```

**Arguments**

grid	A grid vector.
n	A numeric vector of degrees. By default, 1L.
moore	Moore neighborhood (TRUE, default) or Von Neumann neighborhood (FALSE).
simplify	Should simplify the format of the return?

**Value**

A list of grid vectors.

---

grid\_neighborhood      *Neighborhood grid square codes*

---

**Description**

Neighborhood grid square codes

**Usage**

```
grid_neighborhood(grid, n = 1L, type = NULL, simplify = TRUE)
```

**Arguments**

grid	A grid vector.
n	A numeric vector of degrees. By default, 1L.
type	A character vector of neighborhood types, "von_neumann" or "moore". By default, "von_neumann".
simplify	Should simplify the format of the return?

**Value**

A list of grid vectors.

---

grid_subdivide	<i>Subdivide grid square codes</i>
----------------	------------------------------------

---

**Description**

grid\_subdivide() makes the grid square codes finer.

**Usage**

```
grid_subdivide(grid, grid_size)
```

**Arguments**

grid	A grid vector.
grid_size	A grid size.

**Value**

A list of grid vector.

---

is_grid	<i>Test if the object is a grid</i>
---------	-------------------------------------

---

**Description**

Test if the object is a grid

**Usage**

```
is_grid(x, grid_size = NULL)
```

**Arguments**

x	An object.
grid_size	A grid size.

**Value**

TRUE if the object inherits from the grid class.

---

jpgrid

*Functions for the Grid Square Codes in Japan*

---

### Description

Provides functions for grid square codes in Japan (<https://www.stat.go.jp/english/data/mesh/index.html>). Generates the grid square codes from longitude/latitude, geometries, and the grid square codes of different scales, and vice versa.

### Author(s)

**Maintainer:** Mizuki Uchida <uchidamizuki@vivaldi.net>

### See Also

<https://www.stat.go.jp/english/data/mesh/index.html>

---

parse\_grid

*Parse grid square codes*

---

### Description

Parse grid square codes

### Usage

```
parse_grid(x, grid_size = NULL, strict = TRUE)
```

### Arguments

x	A character vector of grid square codes.
grid_size	A grid size.
strict	A scalar logical. Should the number of digits in the grid square code match a given number of digits? By default, TRUE.

### Examples

```
parse_grid("53394526313")
parse_grid("53394526313", "80km")
parse_grid("53394526313", "80km",
          strict = FALSE)
```

# Index

## \* datasets

grid\_city\_2020, 5

bbox\_to\_grid, 2

coords, 3

coords\_to\_grid (coords), 3

geometry\_to\_grid, 3

grid\_as\_sf, 4

grid\_as\_stars, 4

grid\_city\_2020, 5

grid\_components, 6

grid\_convert, 6

grid\_distance, 7

grid\_line, 7

grid\_move, 8

grid\_neighbor, 8

grid\_neighborhood, 9

grid\_subdivide, 10

grid\_to\_coords (coords), 3

is\_grid, 10

jpgrid, 11

jpgrid-package (jpgrid), 11

parse\_grid, 11

sf::st\_as\_sf(), 4

stars::st\_as\_stars(), 5

stars::st\_rasterize(), 3