

# Package ‘pyncogrid’

July 10, 2026

**Title** Flexible Pyncophylactic Interpolation to Discrete Global and Local Grid Systems

**Version** 0.2.0

**Description** Provides tools for pyncophylactic interpolation of polygon totals to discrete global and local grid systems. The method follows Tobler (1979) <[doi:10.1080/01621459.1979.10481647](https://doi.org/10.1080/01621459.1979.10481647)>, preserving source-zone totals while smoothing values across neighboring target cells.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Depends** R (>= 4.1)

**Imports** a5R, dplyr, h3o, hexify, Matrix, rlang, s2, sf, sfdep, terra, tibble

**Suggests** forcats, gt, knitr, purrr, quarto, testthat (>= 3.0.0), tidy, tmap

**VignetteBuilder** quarto

**Config/testthat/edition** 3

**LazyData** true

**Config/roxygen2/version** 8.0.0

**URL** <https://higgicd.github.io/pyncogrid/>,  
<https://github.com/higgicd/pyncogrid>

**BugReports** <https://github.com/higgicd/pyncogrid/issues>

**NeedsCompilation** no

**Author** Christopher Higgins [aut, cre, cph]

**Maintainer** Christopher Higgins <[cd.higgins@utoronto.ca](mailto:cd.higgins@utoronto.ca)>

**Repository** CRAN

**Date/Publication** 2026-07-10 20:30:13 UTC

## Contents

pyncogrid-package . . . . .	2
nyc_ct . . . . .	3
nyc_ct_small . . . . .	3
to_a5 . . . . .	4
to_grid . . . . .	5
to_h3 . . . . .	6
to_hex . . . . .	7
to_isea . . . . .	8
to_raster . . . . .	9
to_s2 . . . . .	10
<b>Index</b>	<b>12</b>

---

pyncogrid-package      *pyncogrid*

---

## Description

Provides tools for pyncophylactic interpolation of polygon totals to discrete global and local grid systems. The method follows Tobler (1979) [doi:10.1080/01621459.1979.10481647](https://doi.org/10.1080/01621459.1979.10481647), preserving source-zone totals while smoothing values across neighboring target cells.

## Author(s)

**Maintainer:** Christopher Higgins <cd.higgins@utoronto.ca> [copyright holder]

Authors:

- Christopher Higgins <cd.higgins@utoronto.ca> [copyright holder]

## See Also

Useful links:

- <https://higgicd.github.io/pyncogrid/>
- <https://github.com/higgicd/pyncogrid>
- Report bugs at <https://github.com/higgicd/pyncogrid/issues>

---

nyc_ct	<i>New York City census tract example data</i>
--------	--

---

**Description**

Census tract population and employment data for New York City.

**Usage**

nyc\_ct

**Format**

An sf object with:

**id** Census tract GEOID

**populationE** Estimated total population

**employment** Total employment

**geometry** MULTIPOLYGON geometry

**Details**

Population data were obtained from the American Community Survey using the tidycensus package. Employment data were obtained from LEHD LODES using the lehdr package.

**Source**

U.S. Census Bureau American Community Survey 2023 5-year estimates; LEHD LODES 2022. This product uses Census Bureau data but is not endorsed or certified by the Census Bureau.

---

nyc_ct_small	<i>Small New York City census tract example data</i>
--------------	--

---

**Description**

A small contiguous subset of NYC census tracts for examples and testing.

**Usage**

nyc\_ct\_small

**Format**

An sf object with the same columns as nyc\_ct.

**Source**

American Community Survey 2023; LEHD LODES 2022. This product uses Census Bureau data but is not endorsed or certified by the Census Bureau.

---

to\_a5

*Pycnophylactic interpolation to an A5 grid*

---

**Description**

Interpolates polygon counts to an A5 grid while preserving source-zone totals.

**Usage**

```
to_a5(source, value_col, resolution, ...)
```

**Arguments**

source	An sf polygon object in a projected CRS.
value_col	Column containing the values to interpolate. May be supplied as an unquoted column name or a character string.
resolution	A5 resolution level (0–30). Higher values produce smaller cells and many more output polygons. For typical urban analyses, values around 8–16 are usually more practical.
...	Additional arguments passed to <code>to_grid()</code> .

**Details**

A5 is a hierarchical global discrete grid system in which each increase in resolution subdivides cells by a factor of five. Consequently, the number of cells grows exponentially with resolution and very high resolutions may be impractical for pycnophylactic interpolation.

Calls `to_grid()` with `grid_type = "a5"`.

**Value**

An sf object containing A5 cells and interpolated values.

**Examples**

```
out <- to_a5(
  source = nyc_ct_small,
  value_col = populationE,
  resolution = 14,
  max_iter = 5
)
```

---

to_grid	<i>Pycnophylactic interpolation to a grid</i>
---------	---

---

### Description

Interpolates polygon counts to a regular or global discrete grid while preserving source-zone totals.

### Usage

```
to_grid(
  source,
  value_col,
  id_col = NULL,
  grid_type = c("h3", "a5", "s2", "isea3h", "isea4h", "raster", "hex"),
  resolution,
  cell_inclusion = c("intersect", "centroid"),
  cell_allocation = c("area", "centroid"),
  nb_order = 1,
  max_iter = 500,
  tolerance = 1e-04,
  include_self = TRUE,
  missing_policy = c("abort", "warn", "ignore")
)
```

### Arguments

source	An sf polygon object in a projected CRS.
value_col	Column containing the values to interpolate. May be supplied as an unquoted column name or a character string.
id_col	Optional unique identifier column for source polygons.
grid_type	Grid system to interpolate to. One of "h3", "a5", "s2", "isea3h", "isea4h", "raster", or "hex".
resolution	Grid resolution. For H3, A5, S2, and ISEA grids this is the grid level. For local raster and hex grids this is the cell size in the linear units of the input projected CRS.
cell_inclusion	Method used to determine which grid cells are included. One of "intersect" or "centroid".
cell_allocation	Method used to allocate source values to grid cells. One of "area" or "centroid".
nb_order	Neighbourhood order used for smoothing. A value of 1 uses first-order neighbours; larger values include higher-order neighbours.
max_iter	Maximum number of pycnophylactic smoothing iterations. If 0, returns the initial allocation without smoothing.
tolerance	Convergence tolerance based on relative mean density change.

`include_self` Logical. Should each cell include itself when smoothing?

`missing_policy` How to handle source polygons that receive no grid cells. One of "abort", "warn", or "ignore".

### Value

An sf object containing grid-cell geometries and interpolated values.

### Examples

```
out <- to_grid(
  source = nyc_ct_small,
  value_col = populationE,
  grid_type = "h3",
  resolution = 9,
  max_iter = 5
)
```

---

to\_h3

*Pycnophylactic interpolation to an H3 grid*

---

### Description

Interpolates polygon counts to an H3 grid while preserving source-zone totals.

### Usage

```
to_h3(source, value_col, resolution, ...)
```

### Arguments

`source` An sf polygon object in a projected CRS.

`value_col` Column containing the values to interpolate. May be supplied as an unquoted column name or a character string.

`resolution` H3 resolution level (0–15). Higher values produce smaller cells and many more output polygons. For typical urban analyses, values around 9–12 are usually more practical.

`...` Additional arguments passed to `to_grid()`.

### Details

H3 is a hierarchical global discrete grid system composed primarily of hexagonal cells. Higher resolution levels produce smaller cells and a larger number of output cells. H3 resolutions range from 0 to 15, but high resolutions may be impractical for large study areas.

Calls `to_grid()` with `grid_type = "h3"`.

**Value**

An sf object containing H3 cells and interpolated values.

**Examples**

```
out <- to_h3(
  source = nyc_ct_small,
  value_col = populationE,
  resolution = 9,
  max_iter = 5
)
```

---

to\_hex

*Pycnophylactic interpolation to a local hexagonal grid*


---

**Description**

Interpolates polygon counts to a local hexagonal grid while preserving source-zone totals.

**Usage**

```
to_hex(source, value_col, resolution, ...)
```

**Arguments**

source	An sf polygon object in a projected CRS.
value_col	Column containing the values to interpolate. May be supplied as an unquoted column name or a character string.
resolution	Hexagonal cell size specified in the linear units of the input projected CRS.
...	Additional arguments passed to <code>to_grid()</code> .

**Details**

Hex grids divide the study area into hexagonal cells of a fixed size. Smaller cell sizes produce more output cells and may substantially increase computation time.

**Value**

An sf object containing hexagonal grid-cell polygons and interpolated values.

**Examples**

```
out <- to_hex(
  source = nyc_ct_small,
  value_col = populationE,
  resolution = 500,
  max_iter = 5
)
```

---

to\_isea

*Pycnophylactic interpolation to an ISEA grid*


---

**Description**

Interpolates polygon counts to an ISEA discrete global grid while preserving source-zone totals.

**Usage**

```
to_isea(source, value_col, resolution, aperture = c(3, 4), ...)
```

**Arguments**

source	An sf polygon object in a projected CRS.
value_col	Column containing the values to interpolate. May be supplied as an unquoted column name or a character string.
resolution	ISEA resolution level. Higher values produce smaller cells and substantially more output polygons.
aperture	ISEA aperture: one of 3 or 4.
...	Additional arguments passed to <a href="#">to_grid()</a> .

**Details**

ISEA grids are equal-area hierarchical discrete global grid systems based on the Icosahedral Snyder Equal Area projection. The aperture determines the refinement factor between successive resolutions: aperture 3, 4, or 7.

Calls [to\\_grid\(\)](#) with `grid_type` set to "isea3h" or "isea4h" according to aperture.

**Value**

An sf object containing ISEA cells and interpolated values.

**Examples**

```

out <- to_isea(
  source = nyc_ct_small,
  value_col = populationE,
  resolution = 18,
  aperture = 3,
  max_iter = 5
)

out <- to_isea(
  source = nyc_ct_small,
  value_col = populationE,
  resolution = 14,
  aperture = 4,
  max_iter = 5
)

```

to\_raster

*Pycnophylactic interpolation to a raster grid***Description**

Interpolates polygon counts to a regular raster grid while preserving source-zone totals.

**Usage**

```
to_raster(source, value_col, resolution, ...)
```

**Arguments**

source	An sf polygon object in a projected CRS.
value_col	Column containing the values to interpolate. May be supplied as an unquoted column name or a character string.
resolution	Raster cell size specified in the linear units of the input projected CRS.
...	Additional arguments passed to <code>to_grid()</code> .

**Details**

Raster grids divide the study area into regular cells of a fixed size. In `pyncogrid`, interpolation is performed using a raster grid, but results are returned as polygon grid cells in an sf object. Smaller cell sizes produce more output cells and may substantially increase computation time.

**Value**

An sf object containing raster grid-cell polygons and interpolated values.

**Examples**

```
out <- to_raster(
  source = nyc_ct_small,
  value_col = populationE,
  resolution = 500,
  max_iter = 5
)
```

---

to\_s2

*Pycnophylactic interpolation to an S2 grid*


---

**Description**

Interpolates polygon counts to an S2 grid while preserving source-zone totals.

**Usage**

```
to_s2(source, value_col, resolution, ...)
```

**Arguments**

source	An sf polygon object in a projected CRS.
value_col	Column containing the values to interpolate. May be supplied as an unquoted column name or a character string.
resolution	S2 resolution level (0–30). Higher values produce smaller cells and many more output polygons. For typical urban analyses, values around 8–16 are usually more practical.
...	Additional arguments passed to <a href="#">to_grid()</a> .

**Details**

S2 is a hierarchical global discrete grid system based on recursively subdividing the six faces of a cube projected onto the sphere. Higher cell levels produce smaller cells and a larger number of output cells. Very high levels may be impractical for pycnophylactic interpolation over large study areas.

Calls [to\\_grid\(\)](#) with `grid_type = "s2"`.

**Value**

An sf object containing S2 cells and interpolated values.

**Examples**

```
out <- to_s2(  
  source = nyc_ct_small,  
  value_col = populationE,  
  resolution = 15,  
  max_iter = 5  
)
```

# Index

## \* datasets

nyc\_ct, 3

nyc\_ct\_small, 3

nyc\_ct, 3

nyc\_ct\_small, 3

pyncogrid (pyncogrid-package), 2

pyncogrid-package, 2

to\_a5, 4

to\_grid, 5

to\_grid(), 4, 6–10

to\_h3, 6

to\_hex, 7

to\_isea, 8

to\_raster, 9

to\_s2, 10