Package ‘populR’

March 13, 2023

Type Package
Title Population Downscaling Using Areal Interpolation
Version 0.2.1
Author Marios Batsaris
Maintainer Marios Batsaris <m.batsaris@aegean.gr>
Description Given a set of source zone polygons such as census tracts or city blocks alongside with population counts and a target zone of incogruent yet superimposed polygon features (such as individual buildings) populR transforms population counts from the former to the latter using Areal Interpolation methods.
License GPL-3
URL https://github.com/mbatsaris/populR/
BugReports https://github.com/mbatsaris/populR/issues/
Encoding UTF-8
LazyData true
Imports sf, rlang, Metrics, usethis, osmdata, dplyr, units
Depends R (>= 3.3.0)
RoxygenNote 7.2.3
Suggests rmarkdown, microbenchmark, areal, knitr, testthat (>= 3.0.0)
Config/testthat/edition 3
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2023-03-13 13:10:02 UTC
R topics documented:

- pp_ancillary ................................................................. 2
- pp_compare ................................................................. 3
- pp_estimate ................................................................. 4
- pp_round ................................................................. 5
- pp_vgi ................................................................. 6
- src ................................................................. 7
- trg ................................................................. 7

Index

---

**pp_ancillary**

A **Ancillary Information from OSM Features**

Description

Ancillary Information from OSM Features

Usage

```r
pp_ancillary(x, volume = NULL, key)
```

Arguments

- `x`
  - an object of class `sf` that is used to associate OSM features to. Usually, `x` may include polygon features representing building units
- `volume`
  - `x` volume information (height or number of floors) useful for float ancillary information
- `key`
  - OSM feature keys or values available in `x`

Value

an object of class `sf` including ancillary information either for night or day estimates

Examples

```r
## Not run:
data('trg')

# Download OSM amenities
dt <- pp_vgi(trg, key = amenity)

# create binary ancillary information
dt <- pp_ancillary(dt, 'amenity')

# create ancillary information both binary and float
dt <- pp_ancillary(dt, floors, 'amenity')
```
**Description**

Comparison to Other Data

**Usage**

```r
pp_compare(x, estimated, actual, title)
```

**Arguments**

- `x`: An object of class `sf` or `data.frame` including estimated and actual values
- `estimated`: Population estimates using `pp_estimate` function
- `actual`: Actual population values
- `title`: Scatterplot title string

**Value**

A list including rmse, mae, linear model details and correlation coefficient

**Examples**

```r
# read lib data
data('src')
data('trg')

# areal weighting interpolation - awi
awi <- pp_estimate(trg, src, sid = sid, spop = pop, method = awi)

# volume weighting interpolation - vwi
vwi <- pp_estimate(trg, src, sid = sid, spop = pop, method = vwi, volume = floors)

# awi - rmse
pp_compare(awi, estimated = pp_est, actual = rf, title = 'awi')

# vwi - rmse
pp_compare(vwi, estimated = pp_est, actual = rf, title = 'vwi')
```
Description

Areal Interpolation of Population Data

Usage

```r
pp_estimate(target, source, sid, spop, volume = NULL, ancillary = NULL, point = FALSE, method)
```

Arguments

- **target**: An object of class `sf` that is used to interpolate data to. Usually, target may include polygon features representing building units.
- **source**: An object of class `sf` including data to be interpolated. Source may be a set of coarse polygon features such as city blocks or census tracts.
- **sid**: Source identification number.
- **spop**: Source population values to be interpolated.
- **volume**: Target feature volume information (height or number of floors). Required when `method = vwi`.
- **ancillary**: Ancillary information.
- **point**: Whether to return point geometries (FALSE by default).
- **method**: Two methods provided: `awi` (areal weighting interpolation) and `vwi` (volume weighting interpolation). `awi` proportionately interpolates the population values based on areal weights calculated by the area of intersection between the source and target zones. `vwi` proportionately interpolates the population values based on areal weights calculated by the area of intersection between the source and target zones multiplied by the volume information (height or number of floors).

Value

An object of class `sf` including estimated population counts for target features using either `awi` or `vwi` methods. The estimated population counts are stored in a new column called `pp_est`. 

---

### pp_estimate

*Areal Interpolation of Population Data*

**Description**

Areal Interpolation of Population Data

**Usage**

```r
pp_estimate(target, source, sid, spop, volume = NULL, ancillary = NULL, point = FALSE, method)
```

**Arguments**

- **target**: An object of class `sf` that is used to interpolate data to. Usually, target may include polygon features representing building units.
- **source**: An object of class `sf` including data to be interpolated. Source may be a set of coarse polygon features such as city blocks or census tracts.
- **sid**: Source identification number.
- **spop**: Source population values to be interpolated.
- **volume**: Target feature volume information (height or number of floors). Required when `method = vwi`.
- **ancillary**: Ancillary information.
- **point**: Whether to return point geometries (FALSE by default).
- **method**: Two methods provided: `awi` (areal weighting interpolation) and `vwi` (volume weighting interpolation). `awi` proportionately interpolates the population values based on areal weights calculated by the area of intersection between the source and target zones. `vwi` proportionately interpolates the population values based on areal weights calculated by the area of intersection between the source and target zones multiplied by the volume information (height or number of floors).

**Value**

An object of class `sf` including estimated population counts for target features using either `awi` or `vwi` methods. The estimated population counts are stored in a new column called `pp_est`. 

---
Examples

# read lib data
data('src')
data('trg')

# areal weighted interpolation - awi
pp_estimate(trg, src, sid = sid, spop = pop, method = awi)

# areal weighted interpolation - awi using point geometries
pp_estimate(trg, src, sid = sid, spop = pop, method = awi, point = TRUE)

# volume weighted interpolation - vwi
pp_estimate(trg, src, sid = sid, spop = pop, method = vwi, volume = floors)

# volume weighted interpolation - vwi using point geometries
pp_estimate(trg, src, sid = sid, spop = pop, method = vwi, volume = floors, point = TRUE)

---

pp_round

<table>
<thead>
<tr>
<th>Rounding Function</th>
</tr>
</thead>
</table>

Description

Rounding Function

Usage

pp_round(x, tpop, spop, sid)

Arguments

- **x**: An object of class sf obtained by the *pp_estimate* function
- **tpop**: Target population estimates obtained by the *pp_estimate* function
- **spop**: Initial source population values (included after the implementation of the *pp_estimate* function)
- **sid**: Source identification number

Value

An object of class sf including rounded population counts stored in a new column called *pp_int*
Examples

```r
# read lib data
data('src')
data('trg')

# areal weighted interpolation - awi
awi <- pp_estimate(trg, src, sid = sid, spop = pop, method = awi)

# volume weighted interpolation - vwi
vwi <- pp_estimate(trg, src, sid = sid, spop = pop, method = vwi, volume = floors)

# awi - round
pp_round(awi, tpop = pp_est, spop = pop, sid = sid)

# vwi - round
pp_round(vwi, tpop = pp_est, spop = pop, sid = sid)
```

---

**pp_vgi**

*Download and Count OSM Features Over Target*

**Description**

Download and Count OSM Features Over Target

**Usage**

```r
pp_vgi(x, key)
```

**Arguments**

- `x`: an object of class `sf` that is used to interpolate data to. Usually, `x` may include polygon features representing building units
- `key`: osm feature key (quoted) see available_features

**Value**

an object of class `sf` including OSM features

**Examples**

```r
## Not run:
data('trg')

# example using just a key
pp_vgi(trg, key = 'amenity')
```
# example using two keys
pp_vgi(trg, key = c('amenity', 'shop'))

## End(Not run)

---

**src**  
*Source (src)*

**Description**

Object of `sf` class representing the blocks of a fictional area

**Usage**

```
src
```

**Format**

Object of `sf` class with 9 rows and 3 columns:

- **sid**: Source identification number
- **pop**: Source population values to be interpolated
- **geometry**: Geometry

**Source**


---

**trg**  
*Target (trg)*

**Description**

An object of `sf` class representing the buildings of a subset area of the city of Mytilini, Greece. The data set contains 179 building units along with the number of floors and residential use in binary format where 0 for non-residential floors and 1 for residential floors.

**Usage**

```
trg
```
Format

object of sf class with 179 rows and 12 columns:

- tid Target identification number
- floors Number of floors
- rf Reference population estimates
- geometry Geometry

Source

http://mbatsaris.gr/
Index

* datasets
  src, 7
  trg, 7

available_features, 6

pp_ancillary, 2
pp_compare, 3
pp_estimate, 3, 4, 5
pp_round, 5
pp_vgi, 6

src, 7

trg, 7