Package: uci (via r-universe)

October 30, 2026

Type Package

Title Urban Centrality Index

Version 1.0.0

Description Calculates the Urban Centrality Index (UCI) as in Pereira et al., (2013) <doi:10.1111/gean.12002>. The UCI measures the extent to which the spatial organization of a city or region varies from extreme polycentric to extreme monocentric in a continuous scale from 0 to 1. Values closer to 0 indicate more polycentric patterns and values closer to 1 indicate a more monocentric urban form.

License MIT + file LICENSE

URL https://github.com/ipea/uci, https://ipea.github.io/uci/

BugReports https://github.com/ipea/uci/issues

Depends R (>= 4.2.0)

Imports checkmate, cli (>= 3.2.0), cppRouting (>= 3.2), data.table, fields, rlang (>= 1.0.0), sf, spdep, stats, utils

Suggests covr, knitr, ggplot2, rmarkdown, testthat (>= 3.2.0)

VignetteBuilder knitr

Encoding UTF-8

Config/testthat/edition 3

NeedsCompilation no

Config/roxygen2/version 8.1.1

Author Rafael H. M. Pereira [aut, cre] (ORCID: <https://orcid.org/0000-0003-2125-7465>)

Maintainer Rafael H. M. Pereira <rafa.pereira.br@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-30 19:35:07 UTC

RemoteUrl https://github.com/cran/uci

RemoteRef HEAD

RemoteSha 80e35c69ddd783939474af5d2594780b0474d209

Contents

remove_islands 2
uci 3

Index **6**

`remove_islands` *Remove islands*

Description

Keeps only the largest group of neighbouring polygons (queen contiguity), i.e. the one with the most polygons, and drops the polygons outside it. This is useful before calling `uci()` with `dist_type = "spatial_link"`, which requires the polygons to form one contiguous area.

Usage

```
remove_islands(sf_object, var_name = NULL, showProgress = TRUE)
```

Arguments

`sf_object`	A `POLYGON sf data.frame`.
`var_name`	An optional `string`. The name of a numeric column in `sf_object`. If given, the message also reports how many activities were dropped (`NA` values are considered to be equal to `0`). Only used for the message. Defaults to `NULL`.
`showProgress`	A `logical`. If `TRUE` (Default), a message reports how many polygons were removed.

Details

If several groups tie for the most polygons, the group with the earliest row is kept. Objects with fewer than 2 rows are returned as they are. Contiguity is computed in the CRS of `sf_object`, which is not changed. `uci(remove_islands = TRUE)` applies this function after reprojecting to EPSG:3857, so for exactly the same result on geographic (lon/lat) data, reproject `sf_object` first.

Value

`sf_object` with only the polygons of the largest contiguous group (same CRS, columns, row order and row names).

See Also

Other urban centrality index: `uci()`

Examples

```
# load data
data_dir <- system.file("extdata", package = "uci")
# sf::st_as_sf() loads {sf}, whose methods subset and bind sf objects
grid <- sf::st_as_sf(readRDS(file.path(data_dir, "grid_bho.rds")))

# add an island: a copy of the first cell, shifted away from the grid
island <- grid[1, ]
sf::st_geometry(island) <- sf::st_geometry(island) + c(0.5, 0)
sf::st_crs(island) <- sf::st_crs(grid)
grid_island <- rbind(grid, island)
nrow(grid_island)

# remove the island
grid_main <- remove_islands(grid_island, var_name = 'jobs')
nrow(grid_main)
```

`uci` *Urban Centrality Index*

Description

Calculates the Urban Centrality Index (UCI) as in Pereira et al., (2013) doi:10.1111/gean.12002. The UCI measures the extent to which the spatial organization of a city or region varies from extreme monocentric to extreme polycentric in a continuous scale from 0 to 1. Values close to 0 indicate more polycentric patterns and values close to 1 indicate a more monocentric urban form.

Calculate Urban Centrality Index

Usage

```
uci(
  sf_object,
  var_name,
  dist_type = "euclidean",
  bootstrap_border = FALSE,
  showProgress = TRUE,
  parallel = FALSE,
  remove_islands = FALSE,
  max_method = NULL
)
```

Arguments

`sf_object`	A `POLYGON sf data.frame` of the study area, with at least two polygons. Distances and areas are computed in EPSG:3857 (Web Mercator), which distorts very large regions.

var_name
: A string. The name of the column in sf_object with the number of activities/opportunities/resources/services to be considered when calculating urban centrality levels. NA values are considered to be equal to 0.

dist_type
: A string indicating whether calculations should be based on "euclidean" distances (Default) or "spatial_link" distances. Spatial link distances consider Euclidean distances along the links of spatial neighbor links. In the case of areas with a concave shape (like a bay), it is strongly recommended to use "spatial_link" distances (even though they are computationally more costly) because simple Euclidean distances can bias UCI estimates in those cases. uci() warns when the study area is markedly concave or has a large gap inside it (e.g. a lake) and dist_type = "euclidean".

bootstrap_border
: A logical. Deprecated, use max_method = "bootstrap". If TRUE and max_method is NULL, the bootstrap is used; an explicit max_method takes precedence.

showProgress
: A logical. If TRUE (Default) and the session is interactive, uci() reports the progress of each step, with a progress bar for the spatial separation on large grids. It also controls the message on how many polygons were removed by remove_islands.

parallel
: A logical. Only used when dist_type = "spatial_link". If FALSE (Default), shortest-path distances are computed on a single thread (any thread setting made with RcppParallel::setThreadOptions() is restored on exit). If TRUE, they are computed in parallel using the number of threads set by {RcppParallel} (all available cores by default, see RcppParallel::setThreadOptions()).

remove_islands
: A logical. Defaults to FALSE. If TRUE, only the largest group of neighbouring polygons (queen contiguity), i.e. the one with the most polygons, is kept, and polygons outside it, along with their activities, are dropped before any calculation. When there are isolated islands, "spatial_link" distances only works with remove_islands = TRUE. Defaults to FALSE. See remove_islands().

max_method
: A string, or NULL (Default), which means "exact" unless bootstrap_border = TRUE. How to find the maximum value of the Venables spatial separation index that the study area admits (spatial_separation_max), which normalises the index. "exact" searches for the distribution of activities among the border polygons that maximises the spatial separation. The search is done on the distances between polygon centroids, where the maximum is global for Euclidean distances, and the result is then refined with the self distance of each polygon included in the distances, so that it is a stationary point of the same quantity as spatial_separation. UCI is 0 (up to the solver tolerance) when the activities sit on that maximising configuration. With "spatial_link" distances the same procedure gives a stationary point. "heuristic" is the approach of Pereira et al. (2013), which assumes that the maximum occurs when all activities are equally distributed along the border of the study area. It is fast, but it does not reach the maximum spatial separation, so it underestimates UCI by an amount that depends on the shape of the area. "bootstrap" simulates 20000 random distributions of activities along the border and uses the largest spatial separation found; it gets closer to the maximum than the heuristic but does not reach it either. Options "heuristic" and "bootstrap" two are kept so that results of previous versions can be reproduced.

Value

A one-row `data.frame` with the columns `UCI`, `location_coef` (LC), `spatial_separation` (the Venables index V of the observed distribution) and `spatial_separation_max` (its maximum for the study area, as found by `max_method`), so that `UCI = LC * (1 - V / V_max)`.

Usage

Please check the vignettes and data documentation on the website.

Author(s)

Maintainer: Rafael H. M. Pereira <rafa.pereira.br@gmail.com> (ORCID)

Authors:

- Rafael H. M. Pereira <rafa.pereira.br@gmail.com> (ORCID)

See Also

Useful links:

- https://github.com/ipea/uci
- https://ipea.github.io/uci/
- Report bugs at https://github.com/ipea/uci/issues

Other urban centrality index: `remove_islands()`

Examples

```
# load data
data_dir <- system.file("extdata", package = "uci")
grid <- readRDS(file.path(data_dir, "grid_bho.rds"))

# calculate UCI
df <- uci(
        sf_object = grid,
        var_name = 'jobs',
        dist_type = "spatial_link"
        )

head(df)
```

Index

∗ **urban centrality index**
remove_islands, 2
uci, 3

remove_islands, 2
remove_islands(), *4*, *5*

uci, 3
uci(), *2*
uci-package *(*uci*)*, 3