The Journal of Open Source Software

segregation: Segregation Analysis, Inference, and Decomposition in Python

Renan X. Cortes [1*], **Elijah Knaap** [2*], and **Sergio J. Rey** [3*]

1 Federal University of Rio Grande do Sul **2** University of California, Irvine **3** San Diego State University
***** These authors contributed equally.

DOI: 10.21105/joss.11126

Software

- Review
- Repository
- Archive

Editor: Henrik Finsberg

Reviewers:

- @mandarc64
- @Sidrcs

Submitted: 29 June 2026
Published: 16 September 2026

Summary

`segregation` is an open-source Python library for measuring, analyzing, inferring, and decomposing social and spatial segregation. Developed within the PySAL ecosystem, it provides a comprehensive framework for quantifying segregation across demographic, socioeconomic, and geographic contexts using both aspatial and spatially explicit measures. The package implements over 40 single-group, multigroup, local, and spatial segregation indices, capturing dimensions such as evenness, exposure, concentration, clustering, centralization, and spatial proximity. It supports multiple spatial representations, including adjacency, distance-based, and network-based relationships.

Beyond computing single index values (point estimation), `segregation` provides Monte Carlo inference for single-value and comparative hypothesis testing under alternative null models, as well as decomposition methods that separate differences in segregation into demographic composition and spatial structure. Built for Pandas and GeoPandas, `segregation` integrates seamlessly with the PySAL ecosystem (S. J. Rey et al., 2022; S. J. Rey & Anselin, 2007), providing transparent, reproducible, and extensible tools for segregation research and spatial data science.

Statement of need

Residential segregation is a central topic in urban studies, sociology, demography, geography, and public policy. Although demographic and spatial data are increasingly available, existing software often provides a limited set of segregation measures, relies on proprietary platforms, or lacks support for modern inference and decomposition methods. Within Python, segregation analysis has historically depended on custom implementations despite the widespread adoption of Pandas and GeoPandas.

`segregation` addresses these limitations by providing:

- A comprehensive, well-documented API for segregation analysis
- Native integration with Pandas, GeoPandas, and the PySAL ecosystem
- 27 single-group, 11 multigroup, and 7 local segregation indices (aspatial and spatial) at present, with new measures still being added
- Multiscalar segregation profiles
- Simulation-based inference for hypothesis testing
- Decomposition of segregation into demographic and spatial components
- Transparent, reproducible, and extensible implementations

These capabilities establish `segregation` as a unified, open-source framework for reproducible segregation analysis in Python.

State of the field

segregation is part of the PySAL ecosystem, where it provides tools for residential segregation analysis within the **Explore** family of spatial analysis libraries. Unlike desktop GIS software, it supports reproducible, scriptable workflows that integrate naturally with broader Python data science pipelines.

The most closely related tools are the R packages OasisR (Tivadar, 2019), seg (Hong, 2011), and segregation (Elbers, 2023). OasisR provides a broad catalog of aspatial and spatial evenness indices with simulation-based inference; seg targets spatial evenness and exposure measures; and the R segregation package focuses on the entropy family (the Mutual Information index and Theil's H) with within/between and temporal decomposition and Bayesian bias correction. PySAL's segregation is distinguished less by any single index than by combining, in one class-based API, index families that these packages cover separately. These include single-group, multigroup, local, and explicitly spatial measures of evenness, exposure, concentration, clustering, and centralization, together with Monte Carlo inference under several null models, Shapley decomposition of differences into demographic and spatial components, multiscalar profiles, and batch computation. It builds on the open-source Scientific Python and PySAL ecosystems, with no proprietary or desktop-GIS requirements. A native Python implementation, rather than a wrapper around the R packages, keeps these workflows within the PySAL ecosystem and lets them share a common index interface. The package has been under continuous development within PySAL since 2018 (Cortes et al., 2020).

Software design

segregation is organized into subpackages by analysis task (singlegroup, multigroup, local, inference, decomposition, batch, network, and dynamics[1]), so that the large collection of indices stays navigable and users import only what they need. Each index is implemented as a Python class rather than a bare function: fitting a class stores the input data, parameters, and result on one object, which inference and decomposition then consume directly. This estimator-style pattern follows a convention familiar from scikit-learn and avoids threading many arguments through a functional API. The design also distinguishes two kinds of indices. *Spatially-explicit* indices include space in their formula. *Spatially-implicit* indices instead accept spatial weights or a network and transform the data before estimation, following Reardon & O'Sullivan (2004); one class therefore serves both aspatial and spatial use, at the cost of expressing spatial behavior through parameters rather than through distinct types.

Figure 1 summarizes the resulting data flow: a pandas or geopandas object is passed to an index class, which returns a fitted object holding the data, parameters, and statistic; inference and decomposition consume that fitted object directly. batch, which covers both single-group and multigroup measures, instead takes the raw pandas/geopandas object directly, while the multiscalar-profile tool additionally takes an explicit index-class argument.

[1] Cortes et al. (2020) introduced segregation but many new implementations were developed recently and the API underwent a major revision.

The Journal of Open Source Software

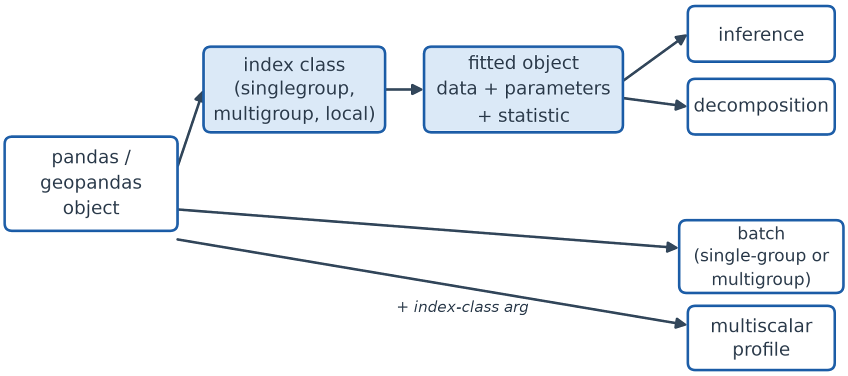

Figure 1: Data flow through the segregation API. A pandas or geopandas object is passed to an index class, which returns a fitted object holding the data, parameters, and computed statistic; inference and decomposition consume that fitted object, while batch (single-group or multigroup) and multiscalar-profile tools take the raw data directly, the latter also requiring an explicit index-class argument.

Additionally, `segregation` is developed with testing and documentation standards consistent with the Scientific Python ecosystem, ensuring reliability and maintainability.

Core functionality

`segregation` organizes its functionality around the type of segregation analysis the user is interested in, and each subpackage is explained as follows.

Single and multigroup indices

Single-group measures assess segregation between two different groups in a given location (i.e., one group vs. everyone else). Multigroup segregation evaluates the simultaneous separation of all groups in a population (e.g., the distribution of White, Black, Asian, and Hispanic residents) across areas.

Currently, `segregation` provides 27 single-group and 11 multigroup indices, which together are, to our knowledge, among the most extensive selections in any segregation software, and the `batch` subpackage can fit many of them at once.

Local indices

Unlike global indices that summarize an entire metropolitan area into a single value, local indices decompose segregation to the individual geographic unit level. Using these disaggregated measures helps identify precise spatial clusters where social isolation is most acute, uncovering micro-level dynamics that global metrics often mask. Currently, `segregation` has seven local indices.

Multiscalar

The multiscalar profile (Reardon et al., 2008) is a tool for measuring spatial segregation dynamics—the way that a segregation index changes values as the concept of a neighborhood changes, and what that tells us about macro versus micro patterns of segregation. The core idea is to calculate a segregation statistic, then expand the spatial scope of a neighborhood, recalculate the statistic, and repeat. The set of distance thresholds chosen sets the range of scales the profile can reveal, so it should reflect the neighborhood definitions relevant to the research question.

The wrapper `compute_multiscalar_profile` builds these profiles.

Simulation-based inference

PySAL's segregation module provides Monte Carlo inference for evaluating the statistical significance of segregation indices under different null hypotheses. For single-value inference, it supports resampling approaches such as bootstrap, systematic, evenness, and geographic_permutation. For comparative inference, it includes methods such as bootstrap and composition, which generate synthetic distributions through counterfactual estimates. Because different null hypotheses test distinct assumptions, their specification is critical and can lead to substantially different conclusions. Likewise, not all segregation indices are appropriate for every null hypothesis, particularly in comparative analyses, making careful selection of both the index and inference procedure essential.

Decomposition

The PySAL segregation module implements a decomposition framework for comparative segregation analysis that partitions differences in segregation into population composition and spatial structure. Building on S. Rey et al. (2021), it combines spatially explicit counterfactual distributions with Shapley value decomposition to quantify each component's contribution to differences in segregation across cities, time periods, and spatiotemporal contexts. Applicable to multiple segregation indices, the framework provides a richer interpretation than direct index comparisons by identifying whether observed differences primarily reflect demographic composition or neighborhood spatial organization. As in inference, the counterfactual approach chosen encodes an assumption about what is held fixed when the two contexts are compared, and different choices can shift the attribution between the demographic and spatial components. This functionality is available through the DecomposeSegregation class in the decomposition subpackage.

Example workflow

Assuming a pandas or geopandas DataFrame named data_1, a segregation index can be computed as:

```python
from segregation.singlegroup import Dissim

seg_index = Dissim(
    data_1,
    group_pop_var="group_A",
    total_pop_var="total_population",
)
```

The estimated value is available through the statistic attribute. Statistical significance can be evaluated using Monte Carlo inference:

```python
from segregation.inference import SingleValueTest

result = SingleValueTest(seg_index, null_approach="bootstrap")
```

The package also provides wrappers for computing all single-group indices (batch_compute_singlegroup), generating multiscalar profiles (compute_multiscalar_profile), comparing segregation measures (TwoValueTest), and decomposing differences into demographic and spatial components (DecomposeSegregation).

Research impact statement

segregation is actively used in research on urban inequality, spatial demography, and education policy. Applications include comparative analyses of racial segregation (Cortes et al., 2020), measurement of spatial information theory indices across U.S. metropolitan areas (Knaap &

Rey, 2024), studies of school-neighborhood segregation (S. J. Rey et al., 2024), optimization approaches for reducing school segregation (Wei et al., 2022), comparative segregation analytics (S. Rey et al., 2021), and analyses of historical redlining legacies (S. Rey & Knaap, 2022).

The package is also used in spatial data science education, including textbooks (Knaap, 2026) and instructional materials presented at conferences such as SciPy.[2]

AI usage disclosure

No generative AI or LLMs were used for code development in `segregation`. They were used for grammar, spelling, and consistency edits during preparation of this paper.

Acknowledgements

`segregation` is developed as part of the PySAL community, which brings together researchers and developers working on spatial analysis methods and software. The project builds on decades of research in segregation, urban, and spatial data science, and benefits from contributions across the open-source geospatial community.

Funding from National Science Foundation grants 2345820 and 1831615 has supported `segregation` development.

The following acknowledgement applies to Renan X. Cortes:

> Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), through process 88881.170553/2018-01, has supported `segregation` development.

References

Cortes, R. X., Rey, S., Knaap, E., & Wolf, L. J. (2020). An open-source framework for non-spatial and spatial segregation measures: The PySAL segregation module. *Journal of Computational Social Science*, *3*(1), 135–166. https://doi.org/10.1007/s42001-019-00059-3

Elbers, B. (2023). A method for studying differences in segregation across time and space. *Sociological Methods & Research*, *52*(1), 5–42. https://doi.org/10.1177/0049124121986204

Hong, S.-Y. (2011). *Seg: Measuring spatial segregation*. https://cran.r-project.org/web/packages/seg

Knaap, E. (2026). *Urban analysis & spatial science*. https://knaaptime.com/urban_analysis

Knaap, E., & Rey, S. J. (2024). Segregated by design? Street network topological structure and the measurement of urban segregation. *Environment and Planning B: Urban Analytics and City Science*, *51*(7), 1408–1429. https://doi.org/10.1177/23998083231197956

Reardon, S. F., Matthews, S. A., O'Sullivan, D., Lee, B. A., Firebaugh, G., Farrell, C. R., & Bischoff, K. (2008). The geographic scale of metropolitan racial segregation. *Demography*, *45*(3), 489–514. https://doi.org/10.1353/dem.0.0019

Reardon, S. F., & O'Sullivan, D. (2004). Measures of spatial segregation. *Sociological Methodology*, *34*(1), 121–162. https://doi.org/10.1111/j.0081-1750.2004.00150.x

Rey, S. J., & Anselin, L. (2007). PySAL: A Python Library of Spatial Analytical Methods. *The Review of Regional Studies*, *37*(1), 5–27. https://doi.org/10.52324/001c.8285

[2] https://www.youtube.com/watch?v=4AHJVMs7iH4

Rey, S. J., Anselin, L., Amaral, P., Arribas-Bel, D., Cortes, R. X., Gaboardi, J. D., Kang, W., Knaap, E., Li, Z., Lumnitz, S., Oshan, T. M., Shao, H., & Wolf, L. J. (2022). The **PySAL** Ecosystem: Philosophy and Implementation. *Geographical Analysis*, *54*(3), 467–487. https://doi.org/10.1111/gean.12276

Rey, S. J., Knaap, E., Wei, R., & Skrah, D. (2024). Measuring Spatial Congruence in the School-Neighborhood Nexus. *Applied Spatial Analysis and Policy*. https://doi.org/10.1007/s12061-024-09603-9

Rey, S., Cortes, R., & Knaap, E. (2021). Comparative spatial segregation analytics. *Spatial Demography*, *9*, 1–26. https://doi.org/10.1007/s40980-021-00075-w

Rey, S., & Knaap, E. (2022). The Legacy of Redlining: A Spatial Dynamics Perspective. *International Regional Science Review*, *0*(0). https://doi.org/10.1177/01600176221116566

Tivadar, M. (2019). OasisR: An R package to bring some order to the world of segregation measurement. *Journal of Statistical Software*, *89*(1), 1–39. https://doi.org/10.18637/jss.v089.i07

Wei, R., Feng, X., Rey, S., & Knaap, E. (2022). Reducing Racial Segregation of Public School Districts. *Socio-Economic Planning Sciences*, 101415. https://doi.org/10.1016/j.seps.2022.101415