

Package: spatgeom (via r-universe)

September 30, 2026

Type Package

Title Geometric Spatial Point Analysis

Version 0.3.0

Maintainer Maikol Solís <maikol.solis@ucr.ac.cr>

Description The implementation to perform the geometric spatial point analysis developed in Hernández & Solís (2022) <doi:10.1007/s00180-022-01244-1>. It estimates the geometric goodness-of-fit index for a set of variables against a response one based on the 'sf' package. The package has methods to print and plot the results.

License MIT + file LICENSE

URL <https://github.com/maikol-solis/spatgeom>

BugReports <https://github.com/maikol-solis/spatgeom/issues>

Encoding UTF-8

LazyData true

Imports ggplot2, scales, sf, dplyr, lwgeom, cowplot

Roxygen list(markdown=TRUE)

Depends R (>= 3.6.0)

Suggests rmarkdown, knitr, testthat (>= 2.1.0), uwot, Rtsne

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://maikol-solis.r-universe.dev>

Date/Publication 2026-06-02 10:57:07 UTC

RemoteUrl <https://github.com/maikol-solis/spatgeom>

RemoteRef HEAD

RemoteSha c79af1604f41f6d71d05525d6b435623997ed4c4

Contents

csr_test	2
donut_data	3
linear_data	4
plot_alpha_shape	4
plot_curve	5
plot_curve.spatgeom_group	6
print.spatgeom	7
print.spatgeom_group	8
reduce_dim	9
spatgeom	10
spatgeom_group	13
Index	15

csr_test	<i>CSR Hypothesis Testing for a spatgeom object</i>
----------	---

Description

Performs a global envelope test for Complete Spatial Randomness (CSR) on a spatgeom object that was computed with envelope = TRUE. Two test statistics are supported: Maximum Absolute Deviation (MAD) and Diggle-Cressie-Loosmore-Ford (DCLF).

Usage

```
csr_test(  
  spatgeom_obj,  
  significance_level = 0.05,  
  r = 0.5,  
  method = c("MAD", "DCLF")  
)
```

Arguments

- spatgeom_obj an object of class spatgeom computed with envelope = TRUE.
- significance_level a numeric value for the significance level of the test. Default 0.05. Currently used for documentation purposes; the p-value is computed from the Monte Carlo rank.
- r a numeric scaling parameter used in the theoretical CSR curve $\exp(-\lambda\pi(\alpha r)^2)$. Should match the value used when calling [spatgeom](#). Default 0.5.
- method a character string, one of "MAD" (Maximum Absolute Deviation) or "DCLF" (Diggle-Cressie-Loosmore-Ford), specifying the global envelope test statistic. Default "MAD".

Value

A named list with two elements:

hypothesis_testing_df A data frame (one row per alpha value per variable) with columns x (alpha grid), mean (mean of simulated CSR curves), alpha, geom_survival (observed), theor (theoretical CSR curve), upper_mean, lower_mean (confidence band around the mean curve), upper_theor, lower_theor (confidence band around the theoretical curve), and variable.

details A named list, one entry per variable, each a data frame summarising the test: null hypothesis, variable name, test type, number of Monte Carlo simulations, observed and maximum test statistics, and p-values against both the mean and theoretical reference curves.

Examples

```
xy <- donut_data(n = 30, a = -1, b = 1, theta = 2 * pi)

# Compute a spatgeom object with the Monte Carlo envelope:
est <- spatgeom(y = xy[, 1], x = xy[, -1], envelope = TRUE)

# Test with the MAD (Maximum Absolute Deviation) statistic:
result_mad <- csr_test(est, method = "MAD")
result_mad$details

# Test with the DCLF statistic:
result_dclf <- csr_test(est, method = "DCLF")
result_dclf$details
```

donut_data

Donut example

Description

Generate data points with the shape of a donut.

Usage

```
donut_data(n, a, b, theta)
```

Arguments

n	Number of data points.
a	Lower bound of the second variable.
b	Upper bound of the second variable.
theta	Angle of the donut.

Value

A data frame with three variables. Variable 'y' is the response, variable 'x1' makes the donut shape with 'y', and 'x2' is a uniform random variable between a and b. '

Examples

```
xy <- donut_data(n = 30, a = -1, b = 1, theta = 2 * pi)
```

linear_data

Linear example

Description

Generate data points with a linear relationship.

Usage

```
linear_data(n = 100, a = -3, b = 3)
```

Arguments

n	Number of data points.
a, b	Lower and upper bound of the uniform distribution.

Value

A data frame with four variables. Variable $y = 0.6 * x1 + 0.3 * x2 + 0.1 * x3$ is the response, and x1, x2 and x3 are independent uniform random variables between a and b.

Examples

```
xy <- linear_data(n = 30, a = -1, b = 1)
```

plot_alpha_shape

Plot alpha-shape for spatgeom objects

Description

Plot alpha-shape for spatgeom objects.

Usage

```
plot_alpha_shape(x, alpha, font_size = 12)
```

Arguments

x	an object of class spatgeom.
alpha	value of alpha determining the maximum length between points to build the alpha-shape.
font_size	a integer that increases the font size in the plot.

Value

a [ggplot](#) object with the raw alpha-shape for the original data at resolution alpha

Examples

```
xy <- donut_data(n = 30, a = -1, b = 1, theta = 2 * pi)

estimation <- spatgeom(y = xy[, 1], x = xy[, -1])

plot_alpha_shape(estimation, alpha = c(0.9, 1.2))
```

plot_curve	<i>plot spatgeom objects with hypothesis testing results and confidence intervals</i>
------------	---

Description

S3 generic for plotting spatial geometry analysis results. Dispatches to [plot_curve.spatgeom](#) for spatgeom objects (producing a ggplot of survival curves or their derivatives), and to [plot_curve.spatgeom_group](#) for spatgeom_group objects (producing a side-by-side cowplot grid, one panel per group).

Usage

```
plot_curve(x, ...)

## S3 method for class 'spatgeom'
plot_curve(x, type = "curve", font_size = 12, ...)
```

Arguments

x	an object of class spatgeom or spatgeom_group.
...	further arguments passed to the appropriate method.
type	a string: either "curve" (the default) or "deriv". The option "curve" plots the survival curve (with envelopes and, if available, confidence intervals). The option "deriv" plots its numerical derivative.
font_size	an integer controlling the font size in the plot.

Value

a ggplot object for spatgeom inputs, or a cowplot grid for spatgeom_group inputs.

Examples

```
xy <- donut_data(n = 30, a = -1, b = 1, theta = 2 * pi)

# Basic plots – no envelope, no hypothesis testing:
estimation <- spatgeom(y = xy[, 1], x = xy[, -1])
plot_curve(estimation, type = "curve")
plot_curve(estimation, type = "deriv")

# Curve with Monte Carlo envelope ribbon (grey) and theoretical CSR curve
# (red dashed):
est_env <- spatgeom(y = xy[, 1], x = xy[, -1], envelope = TRUE)
plot_curve(est_env, type = "curve")

# Curve with hypothesis testing results: additionally shows the CSR mean
# (blue dotted) and its confidence band (blue ribbon):
est_ht <- spatgeom(
  y = xy[, 1], x = xy[, -1],
  hypothesis_testing = TRUE, method = "MAD"
)
plot_curve(est_ht, type = "curve")
```

```
plot_curve.spatgeom_group
```

Plot a spatgeom_group object

Description

Plot method for objects of class `spatgeom_group`. Produces a side-by-side grid of survival curve (or derivative) panels, one column per group.

Usage

```
## S3 method for class 'spatgeom_group'
plot_curve(x, type = "curve", font_size = 12, ...)
```

Arguments

<code>x</code>	an object of class <code>spatgeom_group</code> .
<code>type</code>	a string: either <code>"curve"</code> (default) or <code>"deriv"</code> . Passed to <code>plot_curve.spatgeom</code> for each group.
<code>font_size</code>	an integer controlling the font size. Default 12.
<code>...</code>	further arguments passed to <code>plot_curve.spatgeom</code> .

Value

A cowplot grid object (produced by `cowplot::plot_grid`).

See Also[spatgeom_group](#), [plot_curve](#)**Examples**

```

set.seed(1)
xy <- donut_data(n = 60, a = -1, b = 1, theta = 2 * pi)
grp <- sample(c("A", "B", "C"), nrow(xy), replace = TRUE)
sg <- spatgeom_group(x = xy[, -1], by = grp, y = xy[, 1])
plot_curve(sg)
plot_curve(sg, type = "deriv")

```

print.spatgeom	<i>print a spatgeom object</i>
----------------	--------------------------------

Description

Print method for objects of class spatgeom.

Usage

```

## S3 method for class 'spatgeom'
print(x, return_table = FALSE, ...)

```

Arguments

x	an object of class spatgeom
return_table	if TRUE, returns a data frame with the estimated values. Otherwise, print the data frame in console. Defaults to FALSE
...	further arguments passed to the plot function

Value

Print the estimate given by [spatgeom](#).

Examples

```

xy <- donut_data(n = 30, a = -1, b = 1, theta = 2 * pi)

# Basic print - shows alpha and geom_survival ranges per variable:
estimation <- spatgeom(y = xy[, 1], x = xy[, -1])
print(estimation)

# Print a spatgeom object that includes hypothesis testing results:
est_ht <- spatgeom(
  y = xy[, 1], x = xy[, -1],

```

```
  hypothesis_testing = TRUE, method = "MAD"
)
print(est_ht)

# Return the underlying data frame instead of printing:
tbl <- print(est_ht, return_table = TRUE)
head(tbl)
```

`print.spatgeom_group` *Print a spatgeom_group object*

Description

Print method for objects of class `spatgeom_group`. Displays a summary for each group in turn.

Usage

```
## S3 method for class 'spatgeom_group'
print(x, ...)
```

Arguments

`x` an object of class `spatgeom_group`.
`...` further arguments passed to [print.spatgeom](#).

Value

`x` invisibly.

See Also

[spatgeom_group](#), [print.spatgeom](#)

Examples

```
set.seed(1)
xy <- donut_data(n = 60, a = -1, b = 1, theta = 2 * pi)
grp <- sample(c("A", "B", "C"), nrow(xy), replace = TRUE)
sg <- spatgeom_group(x = xy[, -1], by = grp, y = xy[, 1])
print(sg)
```

Description

Reduce an $n \times p$ matrix to $n \times d$ ($d = n_components$, typically 2) using a choice of static embedding methods. The result can be passed directly to [spatgeom](#) or [spatgeom_group](#) when the original feature matrix has more than 2 columns.

Usage

```
reduce_dim(x, method = c("pca", "umap", "tsne"), n_components = 2L, ...)
```

Arguments

<code>x</code>	a numeric matrix or data frame with n rows and p columns. For <code>method = "pca"</code> , $p \geq d$ is required; for <code>"umap"</code> and <code>"tsne"</code> , <code>n_components</code> may exceed <code>ncol(x)</code> .
<code>method</code>	character; the reduction method. One of: <code>"pca"</code> Principal Component Analysis via prcomp . No extra package required. <code>"umap"</code> Uniform Manifold Approximation and Projection via <code>uwot::umap</code> . Requires the uwot package. <code>"tsne"</code> t-distributed Stochastic Neighbour Embedding via <code>Rtsne::Rtsne</code> . Requires the Rtsne package.
<code>n_components</code>	integer; number of output dimensions. Default 2L.
<code>...</code>	additional arguments forwarded to the underlying reduction function (<code>prcomp</code> , <code>uwot::umap</code> , or <code>Rtsne::Rtsne</code>).

Value

A numeric matrix with `nrow(x)` rows and `n_components` columns. Column names are `Dim_1`, `Dim_2`, etc.

See Also

[spatgeom](#), [spatgeom_group](#)

Examples

```
set.seed(1)
xy <- donut_data(n = 50, a = -1, b = 1, theta = 2 * pi)

# PCA - no extra packages needed
emb <- reduce_dim(xy[, -1], method = "pca")
dim(emb) # 50 x 2

# UMAP - requires uwot
```

```

if (requireNamespace("uwot", quietly = TRUE)) {
  emb_umap <- reduce_dim(xy[, -1], method = "umap")
  dim(emb_umap)
}

# t-SNE - requires Rtsne; perplexity must be < n/3
if (requireNamespace("Rtsne", quietly = TRUE)) {
  emb_tsne <- reduce_dim(xy[, -1], method = "tsne", perplexity = 10)
  dim(emb_tsne)
}

```

spatgeom

Geometric Spatial Point Pattern Analysis

Description

Function to estimate the geometric correlation between variables.

Usage

```

spatgeom(
  x,
  y,
  scale_pts = FALSE,
  nalphas = 100,
  envelope = FALSE,
  domain_type = c("bounding-box", "convex-hull"),
  hypothesis_testing = FALSE,
  significance_level = 0.05,
  mc_cores = 1,
  r = 0.5,
  method = c("MAD", "DCLF"),
  reduce = c("none", "pca", "umap", "tsne"),
  reduce_args = list()
)

```

Arguments

- | | |
|-----------|---|
| x | numeric matrix or data.frame. Either a matrix of covariables (paired with y), or a single $n \times p$ matrix when y is omitted. In the latter case x must have at least 2 columns: if exactly 2, they are used directly as 2D coordinates; if more than 2, set reduce = "pca", "umap", or "tsne" to project to 2D first. |
| y | numeric vector of responses. Optional: when omitted, x is treated as a single point cloud (see reduce). |
| scale_pts | boolean to make the estimations with scaled variables. Default FALSE. |

<code>nalphas</code>	number of alphas generated for creating the geometric measure of fit index. Default 100.
<code>envelope</code>	boolean to determine if the Monte-Carlo should be estimated. Default FALSE.
<code>domain_type</code>	character with the type of domain to use. It can be either "bounding-box" or "convex-hull". Default "bounding-box".
<code>hypothesis_testing</code>	logical. If TRUE, performs a CSR hypothesis test using a global envelope approach after estimating the geometric survival curves. Automatically sets <code>envelope = TRUE</code> . Default FALSE.
<code>significance_level</code>	a numeric significance level passed to <code>csr_test</code> . Default 0.05.
<code>mc_cores</code>	integer with the number of parallel process to run (if available). Default 1.
<code>r</code>	numeric scaling parameter for the theoretical CSR curve used in <code>csr_test</code> : $\exp(-\lambda\pi(\alpha r)^2)$. Default 0.5.
<code>method</code>	character, one of "MAD" (Maximum Absolute Deviation) or "DCLF" (Diggle-Cressie-Loosmore-Ford). Specifies the global envelope test statistic passed to <code>csr_test</code> . Default "MAD".
<code>reduce</code>	character; dimensionality reduction method to apply when x has more than 2 columns. One of "none" (default, keeps all columns and pairs each with y as separate variables), "pca" (Principal Component Analysis, no extra package required), "umap" (requires uwot), or "tsne" (requires Rtsne). When <code>reduce != "none"</code> , x is projected to 2 dimensions via <code>reduce_dim</code> before the analysis.
<code>reduce_args</code>	a named list of additional arguments forwarded to <code>reduce_dim</code> (and in turn to the underlying method). For example, <code>list(n_neighbors = 30)</code> when <code>reduce = "umap"</code> . Default <code>list()</code> .

Value

A list of class `spatgeom` with the following elements:

call The function call.

x x input.

y y output.

results A list of size `ncol(x)` corresponding to each column of x. Each element of the list has:

triangles a data frame of class `sfc` (see `sf::st_sf()`) with columns `geometry`, `segments`, `max_length` and `alpha`. The data.frame contains the whole Delanauy triangulation for the corresponding column of x and y. The `segments` column are the segments of each individual triangle and `max_length` is the maximum length of them.

geom_indices a data frame with columns `alpha` and `geom_survival`. The `alpha` column is a numeric vector of size `nalphas` from the minimum to the maximum distance between points estimated in the data. The `geom_survival` column is the value $1 - (\text{alpha shape Area}) / (\text{containing box Area})$.

intensity the intensity estimated for the corresponding column of x and y.

mean_n the mean number of points in the point process.

envelope_data a data frame in tidy format with 40 runs of a CSR process, if `envelope=TRUE`, The CSR is created by generating n uniform points in the plane, where n is drawn from Poisson distribution with parameter `mean_n`.

hypothesis_testing_results Only present when `hypothesis_testing = TRUE`. A list returned by `csr_test` with elements `hypothesis_testing_df` (a data frame of test statistics and confidence bands for all variables) and `details` (per-variable summary tables of the test results).

reduce_embedding Only present when `reduce != "none"` and `x` had more than 2 columns. The $n \times 2$ numeric matrix produced by `reduce_dim`, whose columns were used as the 2D point cloud.

References

Hernández, A.J., Solís, M. Geometric goodness of fit measure to detect patterns in data point clouds. Comput Stat (2022). <https://doi.org/10.1007/s00180-022-01244-1>

Examples

```
xy <- donut_data(n = 30, a = -1, b = 1, theta = 2 * pi)

# Basic usage: estimate the geometric survival curves only.
estimation <- spatgeom(y = xy[, 1], x = xy[, -1])
print(estimation)

# With Monte Carlo envelope (takes a few seconds):
est_env <- spatgeom(y = xy[, 1], x = xy[, -1], envelope = TRUE)
plot_curve(est_env, type = "curve")

# With integrated CSR hypothesis testing using the MAD statistic:
est_ht <- spatgeom(
  y = xy[, 1], x = xy[, -1],
  hypothesis_testing = TRUE,
  method = "MAD"
)
print(est_ht)
plot_curve(est_ht, type = "curve")

# Inspect the per-variable test results:
est_ht$hypothesis_testing_results$details

# Wide matrix: reduce to 2D with PCA before running the analysis:
est_pca <- spatgeom(y = xy[, 1], x = xy[, -1], reduce = "pca")
plot_curve(est_pca, type = "curve")

# Wide matrix: reduce with UMAP (requires the 'uwot' package):
if (requireNamespace("uwot", quietly = TRUE)) {
  est_umap <- spatgeom(y = xy[, 1], x = xy[, -1], reduce = "umap")
  plot_curve(est_umap, type = "curve")
}
```

spatgeom_group

Geometric Spatial Point Pattern Analysis by Group

Description

Apply [spatgeom](#) independently to each level of a grouping variable. The result is an object of class `spatgeom_group` that bundles the per-group `spatgeom` objects and supports `print` and [plot_curve](#) methods for side-by-side comparison.

Usage

```
spatgeom_group(x, by, ...)
```

Arguments

<code>x</code>	a numeric matrix or data frame of covariates (passed as the <code>x</code> argument of spatgeom). Rows correspond to observations.
<code>by</code>	a vector of group labels with the same length as <code>nrow(x)</code> . Can be a character vector, factor, or any vector that can be coerced to character. Each unique value defines one group.
<code>...</code>	additional arguments forwarded verbatim to spatgeom for every group. This includes <code>y</code> , <code>nalphas</code> , <code>domain_type</code> , <code>envelope</code> , <code>hypothesis_testing</code> , <code>method</code> , <code>reduce</code> , <code>reduce_args</code> , <code>mc_cores</code> , etc. The subset of <code>x</code> rows belonging to each group is passed automatically; <i>do not</i> pass <code>x</code> again inside <code>...</code>

Details

When `y` is supplied via `...`, the corresponding rows of `y` are subsetting to match each group. When `y` is absent, each group's subset of `x` is treated as a 2-D point cloud (or reduced according to `reduce` with options supplied via `reduce_args`).

Groups are processed in the order returned by `unique(by)`, which preserves the order of first appearance in `by`.

Value

An object of class `spatgeom_group`, which is a list with:

call The matched function call.

groups Character vector of unique group labels in appearance order.

results Named list of `spatgeom` objects, one per group. Names match groups.

See Also

[spatgeom](#), [plot_curve](#), [print.spatgeom_group](#)

Examples

```
set.seed(1)
xy <- donut_data(n = 60, a = -1, b = 1, theta = 2 * pi)
grp <- sample(c("A", "B", "C"), nrow(xy), replace = TRUE)

sg <- spatgeom_group(x = xy[, -1], by = grp, y = xy[, 1])
print(sg)
plot_curve(sg)

# With hypothesis testing per group (slow – runs Monte Carlo per group):
sg_ht <- spatgeom_group(
  x = xy[, -1], by = grp, y = xy[, 1],
  hypothesis_testing = TRUE, method = "MAD"
)
plot_curve(sg_ht, type = "curve")
```

Index

`csr_test`, [2](#), [11](#), [12](#)

`donut_data`, [3](#)

`ggplot`, [5](#)

`linear_data`, [4](#)

`plot_alpha_shape`, [4](#)

`plot_curve`, [5](#), [7](#), [13](#)

`plot_curve.spatgeom`, [5](#), [6](#)

`plot_curve.spatgeom_group`, [5](#), [6](#)

`prcomp`, [9](#)

`print.spatgeom`, [7](#), [8](#)

`print.spatgeom_group`, [8](#), [13](#)

`reduce_dim`, [9](#), [11](#), [12](#)

`sf::st_sf()`, [11](#)

`spatgeom`, [2](#), [7](#), [9](#), [10](#), [13](#)

`spatgeom_group`, [7–9](#), [13](#)