

# Package ‘spconform’

October 19, 2026

**Type** Package

**Title** Conformal Prediction for Spatially and Spatio-Temporally  
Dependent Data

**Version** 0.1.2

**Date** 2026-10-10

**Description** Provides distribution-free, model-agnostic prediction intervals for spatially and spatio-temporally dependent data using localized conformal calibration. Implements locally weighted split conformal prediction for geostatistical (point-referenced) data based on spatial-distance kernels, and a neighbourhood-weighted conformal procedure for areal (lattice) data based on graph adjacency structures. Relaxes the standard exchangeability assumption using spatial proximity, following the localized conformal framework of Mao, Martin and Reich (2024) <[doi:10.1080/01621459.2022.2147531](https://doi.org/10.1080/01621459.2022.2147531)>. Includes comprehensive spatial diagnostic tools to audit empirical coverage, conditional spatial strata, and boundary proximity effects.

**License** GPL-3

**URL** <https://amjed-droid.github.io/spconform/>,  
<https://github.com/amjed-droid/spconform>

**BugReports** <https://github.com/amjed-droid/spconform/issues>

**Encoding** UTF-8

**Depends** R (>= 3.6.0)

**Imports** stats

**Suggests** testthat (>= 3.0.0), knitr, rmarkdown, sp

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.1.0

**NeedsCompilation** no

**Author** Ahmed Sattar Jabbar [aut, cre] (ORCID:  
<<https://orcid.org/0009-0006-4614-735X>>)

**Maintainer** Ahmed Sattar Jabbar <ahmed.state.me@gmail.com>  
**Repository** CRAN  
**Date/Publication** 2026-10-19 12:00:02 UTC

**Contents**

|                                            |           |
|--------------------------------------------|-----------|
| areal_neighbor_weights . . . . .           | 2         |
| as.data.frame.spconform . . . . .          | 3         |
| as.data.frame.spconform_diagnose . . . . . | 4         |
| boundary_diagnostics . . . . .             | 4         |
| coverage_report . . . . .                  | 5         |
| diagnose . . . . .                         | 6         |
| plot.spconform . . . . .                   | 8         |
| plot.spconform_boundary . . . . .          | 9         |
| plot.spconform_diagnose . . . . .          | 9         |
| plot.spconform_strata . . . . .            | 10        |
| predict.spconform . . . . .                | 10        |
| print.spconform . . . . .                  | 11        |
| print.spconform_boundary . . . . .         | 11        |
| print.spconform_diagnose . . . . .         | 12        |
| print.spconform_moran . . . . .            | 12        |
| print.spconform_strata . . . . .           | 13        |
| residuals.spconform . . . . .              | 13        |
| scp_areal . . . . .                        | 14        |
| scp_geostatistical . . . . .               | 15        |
| spatial_kernel_weights . . . . .           | 17        |
| spatial_moran . . . . .                    | 19        |
| strata_diagnostics . . . . .               | 20        |
| summary.spconform . . . . .                | 21        |
| <b>Index</b>                               | <b>22</b> |

---

|                                                         |
|---------------------------------------------------------|
| areal_neighbor_weights                                  |
| <i>Areal (Lattice) Graph Distance Proximity Weights</i> |

---

**Description**

Computes neighbourhood-based weights for areal (lattice) data based on shortest-path graph distances.

**Usage**

```
areal_neighbor_weights(i0, adjacency, decay = 1)
```

**Arguments**

|           |                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| i0        | Integer index of the target (unobserved / held-out) areal unit.                                                                                                                         |
| adjacency | A square 0/1 (or weighted) adjacency matrix describing the neighbourhood structure of the areal units (e.g. a spatial contiguity matrix). Row/column i0 corresponds to the target unit. |
| decay     | Numeric decay rate applied to graph distance (number of hops) from i0; larger values down-weight distant neighbours more aggressively. Defaults to 1.                                   |

**Value**

A numeric vector of length nrow(adjacency) with weights based on graph distance from i0 (self-weight is 0, i.e. the target unit is excluded from its own calibration set).

---

as.data.frame.spconform

*Coerce an spconform Object to a Data Frame*

---

**Description**

Converts a conformal prediction output object of class "spconform" into a tidy data.frame containing coordinates (if present), point predictions, lower/upper bounds, and interval widths.

**Usage**

```
## S3 method for class 'spconform'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

**Arguments**

|           |                                              |
|-----------|----------------------------------------------|
| x         | An object of class "spconform".              |
| row.names | NULL or a character vector giving row names. |
| optional  | Logical; passed to as.data.frame.            |
| ...       | Additional arguments (currently unused).     |

**Value**

A data.frame with prediction columns pred, lower, upper, width, and any coordinate columns.

---

```
as.data.frame.spconform_diagnose
```

*Coerce an spconform\_diagnose Object to a Data Frame*

---

### Description

Extracts conditional spatial strata and diagnostic statistics as a tidy data frame for programmatic inspection.

### Usage

```
## S3 method for class 'spconform_diagnose'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

### Arguments

|                        |                                                 |
|------------------------|-------------------------------------------------|
| <code>x</code>         | An object of class "spconform_diagnose".        |
| <code>row.names</code> | Optional row names.                             |
| <code>optional</code>  | Logical; passed to <code>as.data.frame</code> . |
| <code>...</code>       | Additional arguments (unused).                  |

### Value

A tidy data frame of spatial diagnostic strata.

---

```
boundary_diagnostics
```

*Domain Boundary Proximity Diagnostics for Conformal Prediction*

---

### Description

Evaluates whether prediction intervals degrade near the convex-hull boundary of the spatial prediction domain versus interior locations.

### Usage

```
boundary_diagnostics(object, y_true, s_test)
```

### Arguments

|                     |                                                         |
|---------------------|---------------------------------------------------------|
| <code>object</code> | An object of class "spconform".                         |
| <code>y_true</code> | Numeric vector of true observed responses.              |
| <code>s_test</code> | Numeric matrix of spatial coordinates ( $m \times 2$ ). |

Value

An S3 object of class "spconform\_boundary" comparing empirical coverage, interval width, and Winkler scores across edge and interior subdomains.

References

Winkler, R. L. (1972). A Decision-Theoretic Approach to Interval Estimation. *Journal of the American Statistical Association*, 67(337), 187–191.

Examples

```
set.seed(42)
n <- 50
s <- matrix(runif(n * 2), n, 2)
y <- sin(3 * s[, 1]) + rnorm(n, sd = 0.2)
pfun <- function(s_tr, y_tr, s_new) rep(mean(y_tr), nrow(s_new))
fit <- scp_geostatistical(s[1:35, ], y[1:35], s[36:50, ], pfun, alpha = 0.1)
b_res <- boundary_diagnostics(fit, y[36:50], s[36:50, ])
print(b_res)
```

---

|                 |                                                                              |
|-----------------|------------------------------------------------------------------------------|
| coverage_report | <i>Empirical Coverage and Average Interval Width for an spconform Object</i> |
|-----------------|------------------------------------------------------------------------------|

---

Description

Evaluates empirical coverage, average prediction interval width, and the strictly proper Winkler interval score for an "spconform" object.

Usage

```
coverage_report(object, y_true, na.rm = TRUE)
```

Arguments

|        |                                                                                                               |
|--------|---------------------------------------------------------------------------------------------------------------|
| object | An object of class "spconform".                                                                               |
| y_true | Numeric vector of true observed values at the prediction locations, same length/order as object\$pred.        |
| na.rm  | Logical; if TRUE, missing values in y_true are removed before computing evaluation metrics. Defaults to TRUE. |

Value

A named list with:

coverage            Proportion of observations falling within [lower, upper]

.

mean\_width        Average prediction interval width

.

winkler\_score    Mean Winkler interval score across evaluation points

.

References

Winkler, R. L. (1972). A Decision-Theoretic Approach to Interval Estimation. *Journal of the American Statistical Association*, 67(337), 187–191.

Examples

```
set.seed(1)
n <- 20
adj <- matrix(0, n, n)
for (i in 1:(n - 1)) { adj[i, i + 1] <- 1; adj[i + 1, i] <- 1 }
y <- cumsum(rnorm(n))
out <- scp_areal(y, adjacency = adj, alpha = 0.2)
coverage_report(out, y)
```

---

|          |                                                                                |
|----------|--------------------------------------------------------------------------------|
| diagnose | <i>Comprehensive Diagnostic Suite for Spatial Conformal Prediction Objects</i> |
|----------|--------------------------------------------------------------------------------|

---

Description

Produces a comprehensive multi-panel diagnostic report evaluating marginal coverage validity, Winkler Interval Score (WIS) sharpness, conditional coverage across spatial strata, boundary proximity effects, spatial residual autocorrelation (Moran’s I), and nonconformity score distributions.

Usage

```
diagnose(object, y_true, s_test = NULL, n_bins = 4, plot = TRUE, ...)
```

**Arguments**

|                     |                                                                                                                                                                            |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>object</code> | An object of class "spconform", typically output from <a href="#">scp_geostatistical</a> or <a href="#">scp_areal</a> .                                                    |
| <code>y_true</code> | Numeric vector of true observed responses at target prediction locations. Must have the same length as <code>object\$pred</code> .                                         |
| <code>s_test</code> | Optional numeric matrix of target prediction coordinates ( $m \times p$ ). Required for spatial stratification, boundary effects, and spatial autocorrelation diagnostics. |
| <code>n_bins</code> | Integer; number of spatial bins per dimension for conditional coverage (default 4).                                                                                        |
| <code>plot</code>   | Logical; if TRUE (default), generates a multi-panel diagnostic plot.                                                                                                       |
| <code>...</code>    | Additional graphical parameters passed to internal plotting methods.                                                                                                       |

**Details**

The diagnostic suite coordinates modular audits across the prediction intervals:

1. **Marginal Validity & Sharpness:** Evaluates empirical coverage and the strictly proper Winkler Interval Score (WIS; Winkler 1972).
2. **Conditional Spatial Stratification:** Calls [strata\\_diagnostics](#) to evaluate local coverage across quadrants.
3. **Convex Hull Boundary Proximity:** Calls [boundary\\_diagnostics](#) to test whether edge-effect extrapolations degrade calibration.
4. **Spatial Autocorrelation (Moran's I):** Calls [spatial\\_moran](#) to test for residual spatial dependence.
5. **Score Distribution:** Analyzes empirical quantiles and normality/QQ structure.

**Value**

An S3 object of class "spconform\_diagnose" containing:

**marginal** List with coverage, mean width, median width, sd width, Winkler score, and sample size.

**conditional** Data frame of coverage and widths partitioned by spatial quadrant (class "spconform\_strata").

**boundary** Object of class "spconform\_boundary" comparing edge vs interior units.

**spatial\_autocorr** Object of class "spconform\_moran" with Moran's I statistics.

**scores** Summary of nonconformity score moments and quantiles.

**raw\_scores** Numeric vector of nonconformity scores.

**alpha** Miscoverage level of the evaluated object.

**References**

- Winkler, R. L. (1972). A Decision-Theoretic Approach to Interval Estimation. *Journal of the American Statistical Association*, 67(337), 187–191.
- Mao, H., Martin, R., and Reich, B. J. (2024). Valid Conformal Prediction for Dependent Data. *Journal of the American Statistical Association*, 119(546), 1269–1281. doi:10.1080/01621459.2022.2147531.
- Susmann, H. P. (2024). Adaptive Conformal Prediction for Spatial Data. *Computo*, doi:10.57750/edan5f53.

**See Also**

[spatial\\_moran](#), [boundary\\_diagnostics](#), [strata\\_diagnostics](#), [scp\\_geostatistical](#), [scp\\_areal](#)

**Examples**

```
set.seed(42)
n <- 100
s_tr <- matrix(runif(n * 2), n, 2)
y_tr <- sin(3 * s_tr[, 1]) + rnorm(n, sd = 0.2)
s_te <- matrix(runif(30 * 2), 30, 2)
y_te <- sin(3 * s_te[, 1]) + rnorm(30, sd = 0.2)

pfun <- function(s_tr, y_tr, s_new) rep(mean(y_tr), nrow(s_new))
out <- scp_geostatistical(s_tr, y_tr, s_te, pfun, alpha = 0.1)

diag_res <- diagnose(out, y_true = y_te, s_test = s_te, plot = FALSE)
print(diag_res)

# Programmatic access to modular components
as.data.frame(diag_res)
```

---

plot.spconform

---

*Plot Prediction Intervals for spconform Objects*


---

**Description**

Plots point predictions and conformal prediction intervals across observation indices, optionally overlaying true target values for visual evaluation.

**Usage**

```
## S3 method for class 'spconform'
plot(x, y_true = NULL, ...)
```

**Arguments**

|        |                                                                             |
|--------|-----------------------------------------------------------------------------|
| x      | An object of class "spconform".                                             |
| y_true | Optional numeric vector of true observed responses at prediction locations. |
| ...    | Further graphical parameters passed to <a href="#">plot</a> .               |

**Value**

Invisibly returns the input object x.

---

`plot.spconform_boundary`*Plot Method for spconform\_boundary Objects*

---

**Description**

Plot Method for spconform\_boundary Objects

**Usage**

```
## S3 method for class 'spconform_boundary'  
plot(x, ...)
```

**Arguments**

|     |                                          |
|-----|------------------------------------------|
| x   | An object of class "spconform_boundary". |
| ... | Further graphical arguments.             |

---

`plot.spconform_diagnose`*Plot Diagnostic Audit for an spconform\_diagnose Object*

---

**Description**

Generates a multi-panel visual audit of marginal coverage, nonconformity score QQ-distribution, conditional spatial strata coverage, and boundary proximity effects.

**Usage**

```
## S3 method for class 'spconform_diagnose'  
plot(x, ...)
```

**Arguments**

|     |                                                    |
|-----|----------------------------------------------------|
| x   | An object of class "spconform_diagnose".           |
| ... | Additional arguments passed to plotting functions. |

**Value**

Invisibly returns the input object x.

---

`plot.spconform_strata` *Plot Method for spconform\_strata Objects*

---

### Description

Plot Method for spconform\_strata Objects

### Usage

```
## S3 method for class 'spconform_strata'
plot(x, ...)
```

### Arguments

`x`                    An object of class "spconform\_strata".  
`...`                Further graphical arguments.

---

`predict.spconform`        *Extract Predictions and Intervals from an spconform Object*

---

### Description

Extracts point predictions or prediction interval bounds from a fitted spconform object.

### Usage

```
## S3 method for class 'spconform'
predict(object, interval = c("none", "prediction"), ...)
```

### Arguments

`object`                An object of class "spconform".  
`interval`            Character string specifying the type of prediction intervals to extract: "none" (default, point predictions only) or "prediction" (matrix with fit, lwr, upr).  
`...`                Further arguments passed to or from other methods.

### Value

If `interval = "none"`, a numeric vector of point predictions. If `interval = "prediction"`, a numeric matrix with columns `fit`, `lwr`, and `upr`.

---

|                 |                                           |
|-----------------|-------------------------------------------|
| print.spconform | <i>Print Method for spconform Objects</i> |
|-----------------|-------------------------------------------|

---

**Description**

Displays a concise summary of the conformal prediction object including model type, nominal target coverage, and a preview table of prediction intervals.

**Usage**

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

**Arguments**

|     |                                                    |
|-----|----------------------------------------------------|
| x   | An object of class "spconform".                    |
| ... | Further arguments passed to or from other methods. |

**Value**

Invisibly returns the input object x.

---

|                          |                                                    |
|--------------------------|----------------------------------------------------|
| print.spconform_boundary | <i>Print Method for spconform_boundary Objects</i> |
|--------------------------|----------------------------------------------------|

---

**Description**

Print Method for spconform\_boundary Objects

**Usage**

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

**Arguments**

|     |                                          |
|-----|------------------------------------------|
| x   | An object of class "spconform_boundary". |
| ... | Further arguments (unused).              |

---

`print.spconform_diagnose`*Print Diagnostic Audit for an spconform\_diagnose Object*

---

**Description**

Print Diagnostic Audit for an spconform\_diagnose Object

**Usage**

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

**Arguments**

|     |                                          |
|-----|------------------------------------------|
| x   | An object of class "spconform_diagnose". |
| ... | Further arguments (unused).              |

---

`print.spconform_moran` *Print Method for spconform\_moran Objects*

---

**Description**

Print Method for spconform\_moran Objects

**Usage**

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

**Arguments**

|     |                                       |
|-----|---------------------------------------|
| x   | An object of class "spconform_moran". |
| ... | Further arguments (unused).           |

---

```
print.spconform_strata
```

*Print Method for spconform\_strata Objects*

---

## Description

Print Method for spconform\_strata Objects

## Usage

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

## Arguments

|     |                                        |
|-----|----------------------------------------|
| x   | An object of class "spconform_strata". |
| ... | Further arguments (unused).            |

---

```
residuals.spconform
```

*Extract Residuals from an spconform Object*

---

## Description

Calculates prediction residuals (observed minus fitted values) or nonconformity scores for an spconform object given true responses.

## Usage

```
## S3 method for class 'spconform'
residuals(object, y_true, type = c("response", "abs"), ...)
```

## Arguments

|        |                                                                                                                                                 |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| object | An object of class "spconform".                                                                                                                 |
| y_true | Numeric vector of true observed responses at prediction locations.                                                                              |
| type   | Character string indicating residual type: "response" (raw residuals $y - \hat{y}$ ) or "abs" (absolute nonconformity scores $ y - \hat{y} $ ). |
| ...    | Further arguments passed to or from other methods.                                                                                              |

## Value

A numeric vector of residuals.

---

|           |                                                                             |
|-----------|-----------------------------------------------------------------------------|
| scp_areal | <i>Neighbourhood-Weighted Conformal Prediction for Areal (Lattice) Data</i> |
|-----------|-----------------------------------------------------------------------------|

---

## Description

Constructs distribution-free prediction intervals for areal units (e.g. counties, census tracts, grid cells) using a leave-one-unit-out conformal procedure in which nonconformity scores are weighted by graph (neighbourhood) distance to the held-out unit, via its adjacency structure.

## Usage

```
scp_areal(y, X = NULL, adjacency, pred_fun = NULL, alpha = 0.1, decay = 1)
```

## Arguments

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| y         | Numeric vector of observed responses, one per areal unit.                                                                                                                                                                                                                                                                                                                                                                             |
| X         | Optional numeric design matrix of covariates (one row per areal unit); passed to pred_fun if supplied. May be NULL for purely spatial (neighbourhood-based) prediction.                                                                                                                                                                                                                                                               |
| adjacency | Square adjacency (contiguity) matrix describing the neighbourhood structure among the length(y) areal units. Non-zero entries are treated as neighbours (coerced to binary).                                                                                                                                                                                                                                                          |
| pred_fun  | A function with signature function(y_train, X_train, idx_train, idx_target, adjacency) returning a single numeric point prediction for the areal unit indexed by idx_target, fitted using the training units idx_train. If NULL (default), a simple neighbourhood-mean predictor is used (the mean of y_train over the target's graph neighbours, falling back to the global training mean if the target has no observed neighbours). |
| alpha     | Miscoverage level; intervals target $1 - \alpha$ coverage. Default 0.1.                                                                                                                                                                                                                                                                                                                                                               |
| decay     | Decay rate for neighbourhood weights; see <a href="#">areal_neighbor_weights</a> .                                                                                                                                                                                                                                                                                                                                                    |

## Details

Uses a full leave-one-out (jackknife-style) conformal scheme: for each areal unit  $i$ , a model is fit on all other units and used to predict unit  $i$ ; the resulting nonconformity scores across all units are combined into a graph-distance-weighted quantile specific to each target unit, so that the calibration set is dominated by spatially/graph-proximate units rather than treating all units as exchangeable.

If the effective neighbourhood weight for a target unit is vanishingly small (e.g., an isolated unit with no neighbours), the procedure falls back to an unweighted quantile over all other units and issues a warning.

## Value

An object of class "spconform" (areal variant) with the same structure as [scp\\_geostatistical](#), using unit indices in place of coordinates.

## References

- Mao, H., Martin, R., and Reich, B. J. (2024). Valid Conformal Prediction for Dependent Data. *Journal of the American Statistical Association*, 119(546), 1269–1281. doi:10.1080/01621459.2022.2147531.
- Susmann, H. P. (2024). Adaptive Conformal Prediction for Spatial Data. *Computo*, doi:10.57750/edan5f53.

## Examples

```
set.seed(1)
n <- 25
adj <- matrix(0, n, n)
for (i in 1:(n - 1)) { adj[i, i + 1] <- 1; adj[i + 1, i] <- 1 }
y <- cumsum(rnorm(n)) + rnorm(n, sd = 0.2)

out <- scp_areal(y, adjacency = adj, alpha = 0.2)
print(out)
```

---

|                    |                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------|
| scp_geostatistical | <i>Locally Weighted Split Conformal Prediction for Geostatistical (Point-Referenced) Data</i> |
|--------------------|-----------------------------------------------------------------------------------------------|

---

## Description

Constructs distribution-free prediction intervals at new spatial (or spatio-temporal) locations by combining split conformal prediction with spatial-distance kernel weights, relaxing the exchangeability assumption that standard conformal prediction relies on.

## Usage

```
scp_geostatistical(
  s_train,
  y_train,
  s0,
  pred_fun,
  alpha = 0.1,
  split = 0.5,
  bandwidth = NULL,
  t_train = NULL,
  t0 = NULL,
  temporal_bandwidth = NULL,
  k_neighbors = NULL,
  seed = NULL
)
```

**Arguments**

|                                              |                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>s_train</code>                         | Numeric matrix of training coordinates, one row per observation (e.g. longitude/latitude, or projected easting/northing).                                                                                                                                                                                                                                                    |
| <code>y_train</code>                         | Numeric vector of training responses, same length as <code>nrow(s_train)</code> .                                                                                                                                                                                                                                                                                            |
| <code>s0</code>                              | Numeric matrix of coordinates at which prediction intervals are desired, one row per target location.                                                                                                                                                                                                                                                                        |
| <code>pred_fun</code>                        | A function with signature <code>function(s_train, y_train, s_new)</code> that returns a numeric vector of point predictions for the locations in <code>s_new</code> , fitted using <code>s_train/y_train</code> . This lets the user plug in any spatial predictor (kriging, random forest, GAM, etc.); <code>spconform</code> handles only the conformal calibration layer. |
| <code>alpha</code>                           | Miscoverage level; prediction intervals target $1 - \alpha$ coverage. Default 0.1 (90 percent intervals).                                                                                                                                                                                                                                                                    |
| <code>split</code>                           | Proportion of the training data used for model fitting; the remainder is used as the calibration set for conformal scores. Default 0.5.                                                                                                                                                                                                                                      |
| <code>bandwidth</code>                       | Optional spatial kernel bandwidth passed to <a href="#">spatial_kernel_weights</a> ; if NULL, chosen automatically per target location.                                                                                                                                                                                                                                      |
| <code>t_train, t0, temporal_bandwidth</code> | Optional time indices (and temporal bandwidth) for spatio-temporal weighting; see <a href="#">spatial_kernel_weights</a> .                                                                                                                                                                                                                                                   |
| <code>k_neighbors</code>                     | Optional integer; number of nearest calibration neighbours to retain for localized calibration. Defaults to NULL (all units).                                                                                                                                                                                                                                                |
| <code>seed</code>                            | Optional integer seed for the train/calibration split, for reproducibility.                                                                                                                                                                                                                                                                                                  |

**Details**

This implements a localized split-conformal procedure in the spirit of Mao, Martin and Reich (2020, "Valid model-free spatial prediction") and subsequent spatio-temporal extensions: nonconformity scores from a held-out calibration set are combined into a weighted empirical quantile where weights decay with distance (in space, and optionally time) from the target location, so nearby, more relevant observations dominate the calibration of the interval while validity is retained under a local exchangeability argument.

**Value**

An object of class "spconform", a list with components:

**pred** Point predictions at `s0`.

**lower, upper** Lower/upper prediction interval bounds at `s0`.

**alpha** The requested miscoverage level.

**s0** The target coordinates.

**call** The matched call.

## References

- Mao, H., Martin, R., and Reich, B. J. (2024). Valid Conformal Prediction for Dependent Data. *Journal of the American Statistical Association*, 119(546), 1269–1281. doi:10.1080/01621459.2022.2147531.
- Shafer, G. and Vovk, V. (2008). A Tutorial on Conformal Prediction. *Journal of Machine Learning Research*, 9, 371–421.

## Examples

```
set.seed(42)
n <- 200
s <- matrix(runif(2 * n), ncol = 2)
f <- function(s) sin(4 * s[, 1]) + cos(3 * s[, 2])
y <- f(s) + rnorm(n, sd = 0.3)

idx <- sample(n, 150)
s_train <- s[idx, ]; y_train <- y[idx]
s0 <- s[-idx, ][1:10, ]

pred_fun <- function(s_train, y_train, s_new) {
  fit <- lm(y_train ~ s_train[, 1] + s_train[, 2] +
            I(s_train[, 1]^2) + I(s_train[, 2]^2))
  nd <- data.frame(s_train = I(s_new))
  as.numeric(cbind(1, s_new[, 1], s_new[, 2], s_new[, 1]^2, s_new[, 2]^2) %*%
             coef(fit))
}

out <- scp_geostatistical(s_train, y_train, s0, pred_fun, alpha = 0.1, seed = 1)
print(out)
```

---

spatial\_kernel\_weights

*Spatial Gaussian Kernel Proximity Weights*


---

## Description

Computes Gaussian-kernel spatial weights between a target location and a set of reference locations.

## Usage

```
spatial_kernel_weights(
  s0,
  s,
  bandwidth = NULL,
  t0 = NULL,
  t = NULL,
  temporal_bandwidth = NULL
)
```

**Arguments**

|                                 |                                                                                                                                                                                         |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>s0</code>                 | A numeric vector (or 1-row matrix) giving the coordinates of the target (prediction) location.                                                                                          |
| <code>s</code>                  | A numeric matrix of coordinates (one row per observation) for the reference (calibration) set.                                                                                          |
| <code>bandwidth</code>          | Positive numeric bandwidth of the Gaussian kernel. If NULL (default), the bandwidth is chosen automatically as the median pairwise distance among <code>s</code> (Silverman-type rule). |
| <code>t0</code>                 | Optional numeric scalar: time index of the target observation, for spatio-temporal weighting. If supplied, <code>t</code> must be supplied too.                                         |
| <code>t</code>                  | Optional numeric vector: time indices for the reference set, same length as <code>nrow(s)</code> .                                                                                      |
| <code>temporal_bandwidth</code> | Positive numeric bandwidth for the temporal kernel; only used if <code>t0/t</code> are supplied. Defaults to the spatial bandwidth logic applied to <code>t</code> .                    |

**Details**

Weights are computed as

$$w_i = \exp(-\|s_0 - s_i\|^2 / (2h^2))$$

for the spatial-only case, and multiplied by an analogous temporal kernel term when `t0/t` are provided. This is the core device used to relax the exchangeability assumption required by standard conformal prediction: observations located near the point to be predicted contribute more to the calibration of the prediction interval than distant ones (Mao, Martin and Reich, 2024).

**Value**

A numeric vector of weights (not necessarily summing to one), one per row of `s`, giving higher weight to reference points that are spatially (and, if requested, temporally) closer to `s0`.

**References**

Mao, H., Martin, R., and Reich, B. J. (2024). Valid Conformal Prediction for Dependent Data. *Journal of the American Statistical Association*, 119(546), 1269–1281. doi:10.1080/01621459.2022.2147531.

**Examples**

```
set.seed(1)
s <- matrix(runif(20), ncol = 2)
s0 <- c(0.5, 0.5)
w <- spatial_kernel_weights(s0, s)
round(w, 3)
```

---

|               |                                                                                |
|---------------|--------------------------------------------------------------------------------|
| spatial_moran | <i>Spatial Autocorrelation Diagnostics (Moran's I) for Conformal Residuals</i> |
|---------------|--------------------------------------------------------------------------------|

---

### Description

Evaluates spatial autocorrelation and clustering among prediction residuals and binary miscoverage indicators using Moran's  $I$  statistic.

### Usage

```
spatial_moran(object, y_true, s_test)
```

### Arguments

|               |                                                                  |
|---------------|------------------------------------------------------------------|
| <b>object</b> | An object of class "spconform" or numeric vector of predictions. |
| <b>y_true</b> | Numeric vector of true observed responses.                       |
| <b>s_test</b> | Numeric matrix of spatial coordinates ( $m \times p$ ).          |

### Value

An S3 object of class "spconform\_moran" containing:

**moran\_i** Observed Moran's  $I$  statistic for prediction residuals.  
**expected** Theoretical expectation  $-1/(n - 1)$  under spatial independence.  
**z\_score** Standardized normal z-score under the randomization assumption.  
**p\_value** Two-sided p-value against the null hypothesis of spatial randomness.  
**is\_random** Logical; TRUE if p-value  $\geq 0.05$ .  
**moran\_i\_miscoverage** Moran's  $I$  on binary hit/miss indicators if available.

### References

Cliff, A. D. and Ord, J. K. (1981). *Spatial Processes: Models & Applications*. Pion, London.  
Mao, H., Martin, R., and Reich, B. J. (2024). Valid Conformal Prediction for Dependent Data. *Journal of the American Statistical Association*, 119(546), 1269–1281. doi:10.1080/01621459.2022.2147531.

### Examples

```
set.seed(42)
n <- 50
s <- matrix(runif(n * 2), n, 2)
y <- sin(3 * s[, 1]) + rnorm(n, sd = 0.2)
pfun <- function(s_tr, y_tr, s_new) rep(mean(y_tr), nrow(s_new))
fit <- scp_geostatistical(s[1:35, ], y[1:35], s[36:50, ], pfun, alpha = 0.1)
m_res <- spatial_moran(fit, y[36:50], s[36:50, ])
print(m_res)
```

**Description**

Partitions the spatial domain into coordinate grid strata/quadrants and audits local empirical coverage, interval widths, and Winkler scores.

**Usage**

```
strata_diagnostics(object, y_true, s_test, n_bins = 4)
```

**Arguments**

|        |                                                                  |
|--------|------------------------------------------------------------------|
| object | An object of class "spconform".                                  |
| y_true | Numeric vector of true observed responses.                       |
| s_test | Numeric matrix of spatial coordinates ( $m \times 2$ ).          |
| n_bins | Integer; number of spatial bins per coordinate axis (default 4). |

**Value**

An S3 object of class "spconform\_strata" containing a data frame of strata evaluations and summary metrics.

**References**

Winkler, R. L. (1972). A Decision-Theoretic Approach to Interval Estimation. *Journal of the American Statistical Association*, 67(337), 187–191.

**Examples**

```
set.seed(42)
n <- 50
s <- matrix(runif(n * 2), n, 2)
y <- sin(3 * s[, 1]) + rnorm(n, sd = 0.2)
pfun <- function(s_tr, y_tr, s_new) rep(mean(y_tr), nrow(s_new))
fit <- scp_geostatistical(s[1:35, ], y[1:35], s[36:50, ], pfun, alpha = 0.1)
s_res <- strata_diagnostics(fit, y[36:50], s[36:50, ], n_bins = 2)
print(s_res)
```

---

|                   |                                             |
|-------------------|---------------------------------------------|
| summary.spconform | <i>Summary Method for spconform Objects</i> |
|-------------------|---------------------------------------------|

---

**Description**

Computes and displays summary statistics of the prediction interval widths generated by an spconform object.

**Usage**

```
## S3 method for class 'spconform'  
summary(object, ...)
```

**Arguments**

|        |                                                    |
|--------|----------------------------------------------------|
| object | An object of class "spconform".                    |
| ...    | Further arguments passed to or from other methods. |

**Value**

Invisibly returns the input object object.

# Index

areal\_neighbor\_weights, [2](#), [14](#)  
as.data.frame.spconform, [3](#)  
as.data.frame.spconform\_diagnose, [4](#)  
  
boundary\_diagnostics, [4](#), [7](#), [8](#)  
  
coverage\_report, [5](#)  
  
diagnose, [6](#)  
  
plot, [8](#)  
plot.spconform, [8](#)  
plot.spconform\_boundary, [9](#)  
plot.spconform\_diagnose, [9](#)  
plot.spconform\_strata, [10](#)  
predict.spconform, [10](#)  
print.spconform, [11](#)  
print.spconform\_boundary, [11](#)  
print.spconform\_diagnose, [12](#)  
print.spconform\_moran, [12](#)  
print.spconform\_strata, [13](#)  
  
residuals.spconform, [13](#)  
  
scp\_areal, [7](#), [8](#), [14](#)  
scp\_geostatistical, [7](#), [8](#), [14](#), [15](#)  
spatial\_kernel\_weights, [16](#), [17](#)  
spatial\_moran, [7](#), [8](#), [19](#)  
strata\_diagnostics, [7](#), [8](#), [20](#)  
summary.spconform, [21](#)