Package ‘RiskMap’

May 7, 2026

Type Package

Title Geostatistical Modeling of Spatially Referenced Data

Version 1.0.0

Description Geostatistical analysis of continuous and count data.
Implements stationary Gaussian processes with Matérn correlation for spatial prediction,
as described in Diggle and Giorgi (2019, ISBN: 978-1-138-06102-7).

License MIT + file LICENSE

URL https://claudiofronterre.github.io/RiskMap/

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports stats, methods, Matrix, terra, xtable, graphics, spatialEco,
spatialsample, Deriv, numDeriv, dplyr, ggplot2, ggpubr,
gridExtra, sf, sns, stars

NeedsCompilation no

Author Emanuele Giorgi [aut, cre] (ORCID:
<https://orcid.org/0000-0003-0640-181X>),
Claudio Fronterre [ctb] (ORCID:
<https://orcid.org/0000-0001-6723-9727>)

Maintainer Emanuele Giorgi <`e.giorgi@bham.ac.uk`>

Repository CRAN

Date/Publication 2025-10-09 16:30:02 UTC

Contents

`abund_sma` *Female Culex pipiens abundance (collections) in the Sacramento Metropolitan Area*

Description

A dataset of mosquito collection events used to quantify abundance of female Culex pipiens in the Sacramento Metropolitan Area (Sacramento, Placer, El Dorado counties, California, USA). Each row represents a single collection event at a specific location and date.

Usage

```
data(abund_sma)
```

Format

A data frame with one row per collection event (for a total of 1552 collection events) and 6 variables:

lon Numeric. Longitude in decimal degrees (WGS84).

lat Numeric. Latitude in decimal degrees (WGS84).

total_females Integer. Total number of female Cx. pipiens captured in the event.

date Date. Exact collection date (local).

trap_nights Integer. Number of trap-nights for the event.

trap_type Character. Acronym for trap type used to collect mosquitoes. These include: `"NJLT"` (New Jersey Light Trap), `"GRVD"` (Gravid Trap), and `"MMT"` (Mosquito Magnet Trap).

Details

Derived from the **vectorsurvR** sample datasets by:

1. Filtering to female *Culex pipiens* collection records within the Sacramento Metropolitan Area (SMA) using point-in-polygon against the union of Sacramento, Placer, and El Dorado county boundaries.
2. Summing counts and trap-nights per unique *location–date–trap type*.
3. Coordinates are kept as WGS84 (EPSG:4326).

This sample dataset is intended for examples, mapping, and vector-index workflows. It is not official surveillance output. Dates span the same period as the **vectorsurvR** sample (e.g., 2015–2021).

Source

Constructed from `sample_collections` in the `vectorsurvR` R package.

`anopheles` *Anopheles mosquitoes in Southern Cameroon*

Description

These data contain 116 georeferenced locations on the counts of Anopheles gambiae and Anopheles coluzzii in Southern Cameroon.

- web_x x-coordinate of the spatial locations.
- web_y y-coordinate of the spatial locations.
- Locality: name of the place of the sampled location.
- An.coluzzii: counts of Anopheles coluzzi.
- An.gambiae: counts of Anopheles gambiae.
- Total: total counts of Anopheles coluzzi and Anopheles gambiae.
- elevation: elevation in meters of the sampled location.

The coordinate reference system is 3857.

Usage

```
data(anopheles)
```

Format

A data frame with 116 rows and 7 variables

Source

Tene Fossog, B., Ayala, D., Acevedo, P., Kengne, P., Ngomo Abeso Mebuy, I., Makanga, B., et al. (2015) Habitat segregation and ecological character displacement in cryptic African malaria mosquitoes. Evolutionary Applications, 8 (4), 326-345.

`assess_pp` *Assess Predictive Performance via Spatial Cross-Validation*

Description

This function evaluates the predictive performance of spatial models fitted to ‘RiskMap‘ objects using cross-validation. It supports two classes of diagnostic tools:

- **Scoring rules**, including the Continuous Ranked Probability Score (CRPS) and its scaled version (SCRPS), which quantify the sharpness and calibration of probabilistic forecasts; - **Calibration diagnostics**, based on the Probability Integral Transform (PIT) for Gaussian outcomes and Aggregated nonparametric PIT (AnPIT) curves for discrete outcomes (e.g., Poisson or Binomial).

Cross-validation can be performed using either spatial clustering or regularized subsampling with a minimum inter-point distance. For each fold or subset, models can be refitted or evaluated with fixed parameters, offering flexibility in model validation. The function also provides visualizations of the spatial distribution of test folds.

Usage

```
assess_pp(
  object,
  keep_par_fixed = TRUE,
  iter = 1,
  fold = NULL,
  n_size = NULL,
  control_sim = set_control_sim(),
  method,
  min_dist = NULL,
  plot_fold = TRUE,
  messages = TRUE,
  which_metric = c("AnPIT", "CRPS", "SCRPS"),
  user_split = NULL,
  ...
)
```

Arguments

`object`	A list of ‘RiskMap‘ objects, each representing a model fitted with ‘glgpm‘.
`keep_par_fixed`	Logical; if ‘TRUE‘, parameters are kept fixed across folds, otherwise the model is re-estimated for each fold.
`iter`	Integer; number of times to repeat the cross-validation.
`fold`	Integer; number of folds for cross-validation (required if ‘method = "cluster"‘).
`n_size`	Optional; the size of the test set, required if ‘method = "regularized"‘.
`control_sim`	Control settings for simulation, an output from ‘set_control_sim‘.
`method`	Character; either ‘"cluster"‘ or ‘"regularized"‘ for the cross-validation method. The ‘"cluster"‘ method uses spatial clustering as implemented by the `spatial_clustering_cv` function from the ‘spatialEco‘ package, while the ‘"regularized"‘ method selects a subsample of the dataset by imposing a minimum distance, set by the ‘min_dist‘ argument, for a randomly selected subset of locations.
`min_dist`	Optional; minimum distance for regularized subsampling (required if ‘method = "regularized"‘).
`plot_fold`	Logical; if ‘TRUE‘, plots each fold’s test set.
`messages`	Logical; if ‘TRUE‘, displays progress messages.
`which_metric`	Character vector; one or more of ‘"CRPS"‘, ‘"SCRPS"‘, or ‘"AnPIT"‘, to specify the predictive performance metrics to compute.
`user_split`	A user-defined cross-validation split. Either: * a matrix with `nrow = n` (number of observations) and `ncol = iter` (number of iterations), where entries of `1` indicate membership in the test set for that iteration and `0` indicate training set; or * a list of length `iter`, where each element is either a vector of test indices, or a list with components `in_id` (training indices) and `out_id` (test indices). When supplied, `user_split` overrides the automatic clustering or regularized distance splitting defined by `method`.
`...`	Additional arguments passed to clustering or subsampling functions.

Value

A list of class ‘RiskMap.spatial.cv‘, containing:

test_set A list of test sets used for validation, each of class ‘’sf’‘.

model A named list, one per model, each containing:

  - **score** A list with CRPS and/or SCRPS scores for each fold if requested.
  - **PIT** (if ‘family = "gaussian"‘ and ‘which_metric‘ includes ‘"AnPIT"‘) A list of PIT values for test data.
  - **AnPIT** (if ‘family‘ is discrete and ‘which_metric‘ includes ‘"AnPIT"‘) A list of AnPIT curves for test data.

Author(s)

Emanuele Giorgi

References

Bolin, D., & Wallin, J. (2023). Local scale invariance and robustness of proper scoring rules. *Statistical Science*, 38(1), 140–159. doi:10.1214/22STS864.

See Also

`plot_AnPIT`

`assess_sim` *Assess Simulations*

Description

This function evaluates the performance of models based on simulation results from the ‘surf_sim‘ function.

Usage

```
assess_sim(
  obj_sim,
  models,
  control_mcmc = set_control_sim(),
  spatial_scale,
  messages = TRUE,
  f_grid_target = NULL,
  f_area_target = NULL,
  shp = NULL,
  col_names = NULL,
  pred_objective = c("mse", "classify"),
  categories = NULL
)
```

Arguments

`obj_sim`	An object of class ‘RiskMap.sim‘, obtained as an output from the ‘surf_sim‘ function.
`models`	A named list of models to be evaluated.
`control_mcmc`	A control object for MCMC sampling, created with ‘set_control_sim()‘. Default is ‘set_control_sim()‘.
`spatial_scale`	The scale at which predictions are assessed, either ‘"grid"‘ or ‘"area"‘.
`messages`	Logical, if ‘TRUE‘ messages will be displayed during processing. Default is ‘TRUE‘.
`f_grid_target`	A function for processing grid-level predictions.
`f_area_target`	A function for processing area-level predictions.
`shp`	A shapefile of class ‘sf‘ or ‘data.frame‘ for area-level analysis, required if ‘spatial_scale = "area"‘.
`col_names`	Column name in ‘shp‘ containing unique region names. If ‘NULL‘, defaults to ‘"region"‘.
`pred_objective`	A character vector specifying objectives, either ‘"mse"‘, ‘"classify"‘, or both.
`categories`	A numeric vector of thresholds defining categories for classification. Required if ‘pred_objective = "classify"‘.

Value

A list of class ‘RiskMap.sim.res‘ containing model evaluation results.

`check_mcmc` *Check MCMC Convergence for Spatial Random Effects*

Description

This function checks the Markov Chain Monte Carlo (MCMC) convergence of spatial random effects for either a `RiskMap` or `RiskMap.pred.re` object. It plots the trace plot and autocorrelation function (ACF) for the MCMC chain and calculates the effective sample size (ESS).

Usage

```
check_mcmc(object, check_mean = TRUE, component = NULL, ...)
```

Arguments

`object`	An object of class `RiskMap` or `RiskMap.pred.re`. `RiskMap` is the output from `glgpm` function, and `RiskMap.pred.re` is obtained from the `pred_over_grid` function.
`check_mean`	Logical. If `TRUE`, checks the MCMC chain for the mean of the spatial random effects. If `FALSE`, checks the chain for a specific component of the random effects vector.

`component`	Integer. The index of the spatial random effects component to check when `check_mean = FALSE`. Must be a positive integer corresponding to a location in the data. Ignored if `check_mean = TRUE`.
`...`	Additional arguments passed to the `acf` function for customizing the ACF plot.

Details

The function first checks that the input object is either of class `RiskMap` or `RiskMap.pred.re`. Depending on the value of `check_mean`, it either calculates the mean of the spatial random effects across all locations for each iteration or uses the specified component. It then generates two plots: - A trace plot of the selected spatial random effect over iterations. - An autocorrelation plot (ACF) with the effective sample size (ESS) displayed in the title.

The ESS is computed using the `ess` function, which provides a measure of the effective number of independent samples in the MCMC chain.

If `check_mean = TRUE`, the `component` argument is ignored, and a warning is issued. To specify a particular component of the random effects vector, set `check_mean = FALSE` and provide a valid `component` value.

Value

No return value, called for side effects (plots and warnings).

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

`coef.RiskMap` *Extract Parameter Estimates from a "RiskMap" Model Fit*

Description

This `coef` method for the "RiskMap" class extracts the maximum likelihood estimates from model fits obtained from the `glgpm` function.

Usage

```
## S3 method for class 'RiskMap'
coef(object, ...)
```

Arguments

`object`	An object of class "RiskMap" obtained as a result of a call to `glgpm`.
`...`	other parameters.

Details

The function processes the `RiskMap` object to extract and name the estimated parameters appropriately, transforming them if necessary.

Value

A list containing the maximum likelihood estimates:

`beta`	A vector of coefficient estimates.
`sigma2`	The estimate for the variance parameter σ^2.
`phi`	The estimate for the spatial range parameter ϕ.
`tau2`	The estimate for the nugget effect parameter τ^2, if applicable.
`sigma2_me`	The estimate for the measurement error variance σ^2_{me}, if applicable.
`sigma2_re`	A vector of variance estimates for the random effects, if applicable.

Note

This function handles both Gaussian and non-Gaussian families, and accounts for fixed and random effects in the model.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`glgpm`

`compute_ID_coords` *Compute Unique Coordinate Identifiers*

Description

This function identifies unique coordinates from a ‘sf‘ (simple feature) object and assigns an identifier to each coordinate occurrence. It returns a list containing the identifiers for each row and a vector of unique identifiers.

Usage

```
compute_ID_coords(data_sf)
```

Arguments

`data_sf`	An ‘sf‘ object containing geometrical data from which coordinates are extracted.

Details

The function extracts the coordinate pairs from the ‘sf‘ object and determines the unique coordinates. It then assigns each row in the input data an identifier corresponding to the unique coordinate it matches.

Value

A list with the following elements:

- **ID_coords** An integer vector where each element corresponds to a row in the input, indicating the index of the unique coordinate in the full set of unique coordinates.
- **s_unique** An integer vector containing the unique identifiers of all distinct coordinates.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

`convex_hull_sf` *Convex Hull of an sf Object*

Description

Computes the convex hull of an ‘sf‘ object, returning the boundaries of the smallest polygon that can enclose all geometries in the input.

Usage

```
convex_hull_sf(sf_object)
```

Arguments

`sf_object`	An ‘sf‘ data frame object containing geometries.

Details

The convex hull is the smallest convex polygon that encloses all points in the input ‘sf‘ object. This function computes the convex hull by first uniting all geometries in the input using ‘st_union()‘, and then applying ‘st_convex_hull()‘ to obtain the polygonal boundary. The result is returned as an ‘sf‘ object containing the convex hull geometry.

Value

An ‘sf‘ object representing the convex hull of the input geometries.

See Also

`st_convex_hull`, `st_union`

Examples

```
library(sf)

# Create example sf object
points <- st_sfc(st_point(c(0,0)), st_point(c(1,1)), st_point(c(2,2)), st_point(c(0,2)))
sf_points <- st_sf(geometry = points)

# Calculate the convex hull
convex_hull_result <- convex_hull_sf(sf_points)

# Plot the result
plot(sf_points, col = 'blue', pch = 19)
plot(convex_hull_result, add = TRUE, border = 'red')
```

`create_grid` *Create Grid of Points Within Shapefile*

Description

Generates a grid of points within a given shapefile. The grid points are created based on a specified spatial resolution.

Usage

```
create_grid(shp, spat_res, grid_crs = NULL)
```

Arguments

`shp`	An object of class ’sf’ representing the shapefile within which the grid of points will be created.
`spat_res`	Numeric value specifying the spatial resolution in kilometers for the grid.
`grid_crs`	Coordinate reference system for the grid. If NULL, the CRS of ’shp’ is used. The shapefile ’shp’ will be transformed to this CRS if specified.

Details

This function creates a grid of points within the boundaries of the provided shapefile (’shp’). The grid points are generated using the specified spatial resolution (’spat_res’). If a coordinate reference system (’grid_crs’) is provided, the shapefile is transformed to this CRS before creating the grid.

Value

An ’sf’ object containing the generated grid points within the shapefile.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`st_make_grid`, `st_intersects`, `st_transform`, `st_crs`

Examples

```
library(sf)

# Example shapefile data
nc <- st_read(system.file("shape/nc.shp", package="sf"))

# Create grid with 10 km spatial resolution
grid <- create_grid(nc, spat_res = 10)

# Plot the grid
plot(st_geometry(nc))
plot(grid, add = TRUE, col = 'red')
```

`dast` *Fitting of decay-adjusted spatio-temporal (DAST) model*

Description

The function fits a decay-adjusted spatio-temporal (DAST) model using Monte Carlo maximum likelihood. The DAST model allows for the incorporation of temporal decay in disease prevalence due to the impact of mass drug administration (MDA). The function requires the full MDA history as detailed in the arguments below.

Spatial and spatio-temporal dependence is specified through the `gp()` term in the model formula:

- `gp(x, y)` fits a purely spatial Gaussian process.
- `gp(x, y, t_gp)` fits a spatio-temporal Gaussian process, where `t_gp` is used as the GP temporal index.

In all cases, the `time` argument must be specified separately and provides the observation-level survey times used in modelling MDA impact. These survey times may differ from the GP temporal index.

Usage

```
dast(
  formula,
  data,
  den = NULL,
  time,
  mda_times,
  int_mat,
  penalty = NULL,
  drop = NULL,
```

```
  power_val,
  crs = NULL,
  convert_to_crs = NULL,
  scale_to_km = TRUE,
  control_mcmc = set_control_sim(),
  par0 = NULL,
  S_samples = NULL,
  return_samples = TRUE,
  messages = TRUE,
 start_pars = list(beta = NULL, sigma2 = NULL, tau2 = NULL, phi = NULL, psi = NULL,
    sigma2_re = NULL, gamma = NULL, alpha = NULL)
)
```

Arguments

`formula`	A model formula specifying the response variable, predictors, and the GP structure through `gp()`.
`data`	A `data.frame` or `sf` object containing the dataset.
`den`	The denominator for binomial models.
`time`	A variable in `data` giving the survey times of observations (required).
`mda_times`	A vector specifying the mass drug administration (MDA) times.
`int_mat`	Intervention matrix specifying the timing and coverage of MDA; the dimension of the matrix must be `n * n_mda`, where `n` is the number of rows of `data` and `n_mda` is the length of `mda_times`.
`penalty`	Optional list specifying penalty functions for regularization, used in the estimation of the "drop" parameter `alpha`.
`drop`	Optional value used for fixing the "drop" parameter of the MDA impact function.
`power_val`	Value expressing the power of the MDA impact function.
`crs`	Optional coordinate reference system (CRS) for spatial data.
`convert_to_crs`	CRS to which spatial data should be converted.
`scale_to_km`	Logical; whether to scale distances to kilometers (default: `TRUE`).
`control_mcmc`	A list of MCMC control parameters, typically from `set_control_sim()`.
`par0`	Optional list of initial parameter values.
`S_samples`	Number of posterior samples to retain.
`return_samples`	Logical; whether to return posterior samples (default: `TRUE`).
`messages`	Logical; whether to print messages (default: `TRUE`).
`start_pars`	List of starting values for parameters.

Value

A list containing model estimates, posterior samples, and metadata, including:

- `y`: Response variable values.
- `D`: Covariate matrix.

- `coords`: Unique spatial coordinates.
- `mda_times`: MDA time points.
- `survey_times_data`: Survey time data from the `time` argument.
- `time`: GP temporal index if specified in `gp(x,y,t_gp)`.
- `int_mat`: Intervention matrix.
- `ID_coords`: Indices of spatial locations (and time if spatio-temporal GP).
- `re`: Random effects levels (if applicable).
- `ID_re`: Indices of random effects (if applicable).
- `power_val`: Power of the MDA impact function.
- `fix_tau2`: Fixed tau-squared value (if applicable).
- `fix_alpha`: Fixed alpha value (if applicable).
- `formula`: Model formula.
- `crs`: Coordinate reference system.
- `scale_to_km`: Indicator of distance scaling.
- `data_sf`: Processed spatial dataset.
- `family`: Model family (e.g., "binomial").
- `sst`: Logical indicator of whether a spatio-temporal GP was used.
- `kappa`: Smoothness parameter.
- `units_m`: Denominator for binomial models.
- `cov_offset`: Offset for covariates.
- `call`: Function call.
- `penalty`: Penalty function details (if applicable).
- `posterior_samples`: Posterior samples if `return_samples = TRUE`.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`set_control_sim`, `summary.RiskMap`, `to_table`

`dist_summaries` *Summaries of the distances*

Description

Computes the distances between the locations in the data-set and returns summary statistics of these.

Usage

```
dist_summaries(data, convert_to_utm = TRUE, scale_to_km = FALSE)
```

Arguments

`data`	an object of class `sf` containing the variable for which the variogram is to be computed and the coordinates
`convert_to_utm`	a logical value, indicating if the conversion to UTM shuold be performed (`convert_to_utm = TRUE`) or the coordinate reference system of the data must be used without any conversion (`convert_to_utm = FALSE`). By default `convert_to_utm = TRUE`. Note: if `convert_to_utm = TRUE` the conversion to UTM is performed using the epsg provided by `propose_utm`.
`scale_to_km`	a logical value, indicating if the distances used in the variogram must be scaled to kilometers (`scale_to_km = TRUE`) or left in meters (`scale_to_km = FALSE`). By default `scale_to_km = FALSE`

Value

a list containing the following components

`min` the minimum distance

`max` the maximum distance

`mean` the mean distance

`median` the minimum distance

`galicia` *Heavy metal biomonitoring in Galicia*

Description

This data-set relates to two studies on lead concentration in moss samples, in micrograms per gram dry weight, collected in Galicia, norther Spain. The data are from two surveys, one conducted in July 2000. The variables are as follows:

- x x-coordinate of the spatial locations.
- y y-coordinate of the spatial locations.
- lead number of tested people for the presence nodules.

The coordinate reference system of the data is 32629.

Usage

```
data(galicia)
```

Format

A data frame with 195 rows and 4 variables

Source

Diggle, P.J., Menezes, R. and Su, T.-L. (2010). Geostatistical analysis under preferential sampling (with Discussion). Applied Statistics, 59, 191-232.

`glgpm` *Estimation of Generalized Linear Gaussian Process Models*

Description

Fits generalized linear Gaussian process models to spatial data, incorporating spatial Gaussian processes with a Matern correlation function. Supports Gaussian, binomial, and Poisson response families.

Usage

```
glgpm(
  formula,
  data,
  family,
  invlink = NULL,
  den = NULL,
  crs = NULL,
  convert_to_crs = NULL,
  scale_to_km = TRUE,
  control_mcmc = set_control_sim(),
  par0 = NULL,
  S_samples = NULL,
  return_samples = TRUE,
  messages = TRUE,
  fix_var_me = NULL,
 start_pars = list(beta = NULL, sigma2 = NULL, tau2 = NULL, phi = NULL, sigma2_me =
    NULL, sigma2_re = NULL)
)
```

Arguments

`formula`	A formula object specifying the model to be fitted. The formula should include fixed effects, random effects (specified using `re()`), and spatial effects (specified using `gp()`).

`data`	A data frame or sf object containing the variables in the model.
`family`	A character string specifying the distribution of the response variable. Must be one of "gaussian", "binomial", or "poisson".
`invlink`	A function that defines the inverse of the link function for the distribution of the data given the random effects.
`den`	Optional offset for binomial or Poisson distributions. If not provided, defaults to 1 for binomial.
`crs`	Optional integer specifying the Coordinate Reference System (CRS) if data is not an sf object. Defaults to 4326 (long/lat).
`convert_to_crs`	Optional integer specifying a CRS to convert the spatial coordinates.
`scale_to_km`	Logical indicating whether to scale coordinates to kilometers. Defaults to TRUE.
`control_mcmc`	Control parameters for MCMC sampling. Must be an object of class "mcmc.RiskMap" as returned by `set_control_sim`.
`par0`	Optional list of initial parameter values for the MCMC algorithm.
`S_samples`	Optional matrix of pre-specified sample paths for the spatial random effect.
`return_samples`	Logical indicating whether to return MCMC samples when fitting a Binomial or Poisson model. Defaults to FALSE.
`messages`	Logical indicating whether to print progress messages. Defaults to TRUE.
`fix_var_me`	Optional fixed value for the measurement error variance.
`start_pars`	Optional list of starting values for model parameters: beta (regression coefficients), sigma2 (spatial process variance), tau2 (nugget effect variance), phi (spatial correlation scale), sigma2_me (measurement error variance), and sigma2_re (random effects variances).

Details

Generalized linear Gaussian process models extend generalized linear models (GLMs) by incorporating spatial Gaussian processes to account for spatial correlation in the data. This function fits GLGPMs using maximum likelihood methods, allowing for Gaussian, binomial, and Poisson response families. In the case of the Binomial and Poisson families, a Monte Carlo maximum likelihood algorithm is used.

The spatial Gaussian process is modeled with a Matern correlation function, which is flexible and commonly used in geostatistical modeling. The function supports both spatial covariates and unstructured random effects, providing a comprehensive framework to analyze spatially correlated data across different response distributions.

Additionally, the function allows for the inclusion of unstructured random effects, specified through the `re()` term in the model formula. These random effects can capture unexplained variability at specific locations beyond the fixed and spatial covariate effects, enhancing the model's flexibility in capturing complex spatial patterns.

The `convert_to_crs` argument can be used to reproject the spatial coordinates to a different CRS. The `scale_to_km` argument scales the coordinates to kilometers if set to TRUE.

The `control_mcmc` argument specifies the control parameters for MCMC sampling. This argument must be an object returned by `set_control_sim`.

The `start_pars` argument allows for specifying starting values for the model parameters. If not provided, default starting values are used.

Value

An object of class "RiskMap" containing the fitted model and relevant information:

`y`	Response variable.
`D`	Covariate matrix.
`coords`	Unique spatial coordinates.
`ID_coords`	Index of coordinates.
`re`	Random effects.
`ID_re`	Index of random effects.
`fix_tau2`	Fixed nugget effect variance.
`fix_var_me`	Fixed measurement error variance.
`formula`	Model formula.
`family`	Response family.
`crs`	Coordinate Reference System.
`scale_to_km`	Indicator if coordinates are scaled to kilometers.
`data_sf`	Original data as an sf object.
`kappa`	Spatial correlation parameter.
`units_m`	Distribution offset for binomial/Poisson.
`cov_offset`	Covariate offset.
`call`	Matched call.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`set_control_sim`, `summary.RiskMap`, `to_table`

`glgpm_sim` *Simulation from Generalized Linear Gaussian Process Models*

Description

Simulates data from a fitted Generalized Linear Gaussian Process Model (GLGPM) or a specified model formula and data.

Usage

```
glgpm_sim(
  n_sim,
  model_fit = NULL,
  formula = NULL,
  data = NULL,
  family = NULL,
  den = NULL,
  cov_offset = NULL,
  crs = NULL,
  convert_to_crs = NULL,
  scale_to_km = TRUE,
 sim_pars = list(beta = NULL, sigma2 = NULL, tau2 = NULL, phi = NULL, sigma2_me = NULL,
    sigma2_re = NULL),
  messages = TRUE
)
```

Arguments

`n_sim`	Number of simulations to perform.
`model_fit`	Fitted GLGPM model object of class 'RiskMap'. If provided, overrides 'formula', 'data', 'family', 'crs', 'convert_to_crs', 'scale_to_km', and 'control_mcmc' arguments.
`formula`	Model formula indicating the variables of the model to be simulated.
`data`	Data frame or 'sf' object containing the variables in the model formula.
`family`	Distribution family for the response variable. Must be one of 'gaussian', 'binomial', or 'poisson'.
`den`	Required for 'binomial' to denote the denominator (i.e. number of trials) of the Binomial distribution. For the 'poisson' family, the argument is optional and is used a multiplicative term to express the mean counts.
`cov_offset`	Offset for the covariate part of the GLGPM.
`crs`	Coordinate reference system (CRS) code for spatial data.
`convert_to_crs`	CRS code to convert spatial data if different from 'crs'.
`scale_to_km`	Logical; if TRUE, distances between locations are computed in kilometers; if FALSE, in meters.
`sim_pars`	List of simulation parameters including 'beta', 'sigma2', 'tau2', 'phi', 'sigma2_me', and 'sigma2_re'.
`messages`	Logical; if TRUE, display progress and informative messages.

Details

Generalized Linear Gaussian Process Models (GLGPMs) extend generalized linear models (GLMs) by incorporating spatial Gaussian processes to model spatial correlation. This function simulates data from GLGPMs using Markov Chain Monte Carlo (MCMC) methods. It supports Gaussian, binomial, and Poisson response families, utilizing a Matern correlation function to model spatial dependence.

The simulation process involves generating spatially correlated random effects and simulating responses based on the fitted or specified model parameters. For 'gaussian' family, the function simulates response values by adding measurement error.

Additionally, GLGPMs can incorporate unstructured random effects specified through the `re()` term in the model formula, allowing for capturing additional variability beyond fixed and spatial covariate effects.

Value

A list containing simulated data, simulated spatial random effects (if applicable), and other simulation parameters.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`gp` *Gaussian Process Model Specification*

Description

Specifies the terms, smoothness, and nugget effect for a Gaussian Process (GP) model.

Usage

```
gp(..., kappa = 0.5, nugget = 0)
```

Arguments

`...`	Variables representing the spatial coordinates or covariates for the GP model.
`kappa`	The smoothness parameter κ. Default is 0.5.
`nugget`	The nugget effect, which represents the variance of the measurement error. Default is 0. A positive numeric value must be provided if not using the default.

Details

The function constructs a list that includes the specified terms (spatial coordinates or covariates), the smoothness parameter κ, and the nugget effect. This list can be used as a specification for a Gaussian Process model.

Value

A list of class `gp.spec` containing the following elements:

`term`	A character vector of the specified terms.
`kappa`	The smoothness parameter κ.
`nugget`	The nugget effect.
`dim`	The number of specified terms.
`label`	A character string representing the full call for the GP model.

Note

The nugget effect must be a positive real number if specified.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`infect_sma` *West Nile virus pool tests for female *Culex pipiens* in the Sacramento Metropolitan Area*

Description

A dataset of PCR-tested mosquito pools used to summarize infection for female *Culex pipiens* in the Sacramento Metropolitan Area (Sacramento, Placer, El Dorado counties, California, USA). Each row represents a tested pool at a specific location and date, with an estimated pool size.

Usage

```
data(infect_sma)
```

Format

A data frame with one row per tested pool (for a total of 596 pools) and 5 variables:

lon	Numeric. Longitude in decimal degrees (WGS84).
lat	Numeric. Latitude in decimal degrees (WGS84).
est_pool_n	Integer. Estimated number of mosquitoes in the pool (see Details).
wnv_pos	Logical. Whether the pool tested positive for West Nile virus (`TRUE`/`FALSE`).
date	Date. Pool collection date (local).

Details

Derived from the **vectorsurvR** sample datasets by:

1. Filtering to female *Culex pipiens* pools with WNV testing within the SMA using point-in-polygon against the union of Sacramento, Placer, and El Dorado counties.
2. Estimating pool size `est_pool_n` for each pool by:
   - Summing total female counts from nearby collection points within the *same week* and within a spatial radius (e.g., 2 km) to obtain T_{near}.
   - Counting nearby pools within the same week/radius to obtain m_{near}.
   - Setting `est_pool_n = round(max(1, T_near / m_near))`; if no nearby collections are found, a conservative fallback (default 25) is used.
3. Coordinates are kept as WGS84 (EPSG:4326).

Important: `est_pool_n` is an estimate for demonstration and pooled-modelling examples; it is not the recorded laboratory pool size.

Source

Constructed from `sample_collections` in the `vectorsurvR` R package.

`italy_sim` *Simulated data-set on the Italian peninsula*

Description

This is a simulated data-set over Italy for a continuous outcome. The data-set contains 10 repeated observations for each of the 200 geo-referenced locations. The variables are as follows:

- x1 ordinate of the spatial locations.
- x2 abscissa of the spatial locations.
- y simulated continuous outcome.
- region the name of the region within which a given observation falls.
- province the name of the province within which a given observation falls.
- pop_dens the population density at the location of the observation.
- ID_loc an ID identifying the location to which the observation belong.

The coordinate reference system of the data is 32634.

Usage

```
data(italy_sim)
```

Format

A data frame with 2000 rows and 7 variables

`Laplace_sampling_MCMC` *Laplace-sampling MCMC for Generalized Linear Gaussian Process Models*

Description

Runs Markov chain Monte Carlo (MCMC) sampling using a Laplace approximation for Generalized Linear Gaussian Process Models (GLGPMs). The latent Gaussian field is integrated via a second-order Taylor expansion around the mode, and a Gaussian proposal is used for Metropolis–Hastings updates with adaptive step-size tuning.

Usage

```
Laplace_sampling_MCMC(
  y,
  units_m,
  mu,
  Sigma,
  ID_coords,
  ID_re = NULL,
  sigma2_re = NULL,
  family,
  control_mcmc,
  invlink = NULL,
  Sigma_pd = NULL,
  mean_pd = NULL,
  messages = TRUE
)
```

Arguments

`y`	Numeric vector of responses of length n. For `family = "binomial"` this is the number of successes, for `family = "poisson"` counts, and for `family = "gaussian"` real values.
`units_m`	Numeric vector giving the binomial totals (number of trials) when `family = "binomial"`; ignored for other families (can be `NULL`).
`mu`	Numeric vector of length equal to the number of unique locations providing the mean of the latent spatial process on the link scale.
`Sigma`	Numeric positive-definite covariance matrix for the latent spatial process S at the unique locations referenced by `ID_coords`.
`ID_coords`	Integer vector of length n mapping each response in `y` to a row/column of `Sigma` (i.e., the index of the corresponding location).
`ID_re`	Optional matrix or data.frame with one column per unstructured random effect (RE). Each column is an integer vector of length n mapping observations in `y` to RE levels (e.g., cluster, survey, etc.). Use `NULL` to exclude REs.

`sigma2_re`	Optional named numeric vector of RE variances. Names must match the column names of `ID_re`. Ignored if `ID_re = NULL`.
`family`	Character string: one of `"gaussian"`, `"binomial"`, or `"poisson"`.
`control_mcmc`	List of control parameters: **n_sim** Total number of MCMC iterations (including burn-in). **burnin** Number of initial iterations to discard. **thin** Thinning interval for saving samples. **h** Initial step size for the Gaussian proposal. Defaults to $1.65/n_{\text{tot}}^{1/6}$ if not supplied. **c1.h, c2.h** Positive tuning constants for adaptive step-size updates.
`invlink`	Optional inverse-link function. If `NULL`, defaults are used: `identity` (gaussian), `plogis` (binomial), and `exp` (poisson).
`Sigma_pd`	Optional precision matrix used in the Laplace approximation. If `NULL`, it is obtained internally at the current mode.
`mean_pd`	Optional mean vector used in the Laplace approximation. If `NULL`, it is obtained internally as the mode of the integrand.
`messages`	Logical; if `TRUE`, prints progress and acceptance diagnostics.

Details

The algorithm alternates between:

1. Locating the mode of the joint integrand for the latent variables (via `maxim.integrand`) when `Sigma_pd` and `mean_pd` are not provided, yielding a Gaussian approximation.
2. Metropolis–Hastings updates using a Gaussian proposal centered at the current approximate mean with proposal variance governed by `h`. The step size is adapted based on empirical acceptance probability.

Dimensions must be consistent: `length(y) = n`, `nrow(Sigma) = ncol(Sigma) = n_loc`, and `length(ID_coords) = n` with entries in $1, \ldots, n_{\text{loc}}$. If `ID_re` is provided, each column must have length n; when `sigma2_re` is supplied, it must be named and match `colnames(ID_re)`.

Value

An object of class `"mcmc.RiskMap"` with components:

samples A list containing posterior draws. Always includes `$S` (latent spatial field). If `ID_re` is supplied, each unstructured RE is returned under `$<re_name>`.

tuning_par Numeric vector of step sizes (`h`) used over iterations.

acceptance_prob Numeric vector of Metropolis–Hastings acceptance probabilities.

Default links

The default inverse links are: identity (gaussian), logistic (binomial), and exponential (poisson). Supply `invlink` to override.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterre@lancaster.ac.uk`>

See Also

`maxim.integrand`

`liberia` *River-blindness in Liberia*

Description

This data-set contains counts of reported onchocerciasis (or riverblindess) cases from 91 villages sampled across Liberia. The variables are as follows:

- lat latitude of the of sampled villages.
- long longitude of the sampled villages.
- ntest number of tested people for the presence nodules.
- npos number of people that tested positive for the presence of nodules.
- elevation the elevation in meters of the sampled village.
- log_elevation the log-transformed elevation in meters of the sampled village.

Usage

```
data(liberia)
```

Format

A data frame with 90 rows and 6 variables

Source

Zouré, H. G. M., Noma, M., Tekle, Afework, H., Amazigo, U. V., Diggle, P. J., Giorgi, E., and Remme, J. H. F. (2014). The Geographic Distribution of Onchocerciasis in the 20 Participating Countries of the African Programme for Onchocerciasis Control: (2) Pre-Control Endemicity Levels and Estimated Number Infected. Parasites & Vectors, 7, 326.

`loaloa` *Loa loa prevalence data from 197 village surveys*

Description

This data-set relates to a study of the prevalence of Loa loa (eyeworm) in a series of surveys undertaken in 197 villages in west Africa (Cameroon and southern Nigeria). The variables are as follows:

- ROW row id: 1 to 197.
- VILLCODE village id.
- LONGITUDE Longitude in degrees.
- LATITUDE Latitude in degrees.
- NO_EXAM Number of people tested.
- NO_INF Number of positive test results.
- ELEVATION Height above sea-level in metres.
- MEAN9901 Mean of all NDVI values recorded at village location, 1999-2001
- MAX9901 Maximum of all NDVI values recorded at village location, 1999-2001
- MIN9901 Minimum of all NDVI values recorded at village location, 1999-2001
- MIN9901 Minimum of all NDVI values recorded at village location, 1999-2001
- STDEV9901 standard deviation of all NDVI values recorded at village location, 1999-2001

Usage

```
data(loaloa)
```

Format

A data frame with 197 rows and 11 variables

References

Diggle, P.J., Thomson, M.C., Christensen, O.F., Rowlingson, B., Obsomer, V., Gardon, J., Wanji, S., Takougang, I., Enyong, P., Kamgno, J., Remme, H., Boussinesq, M. and Molyneux, D.H. (2007). Spatial modelling and prediction of Loa loa risk: decision making under uncertainty. Annals of Tropical Medicine and Parasitology, 101, 499-509.

`malkenya` *Malaria Transmission in the Western Kenyan Highlands*

Description

The dataset contains information on 82014 individuals enrolled in concurrent school and community cross-sectional surveys, conducted in 46 school clusters in the western Kenyan highlands. Malaria was assessed by rapid diagnostic test (RDT).

The variables are as follows:

- Cluster: unique ID for each of the 46 school clusters.
- Long: longitude coordinate of the household location.
- Lat: latitude coordinate of the household location.
- RDT: binary variable indicating the outcome of the RDT: 1, if positive, and 0, if negative.
- Gender: factor variable indicating the gender of the sampled individual.
- Age: age in years of the sampled individual.
- NetUse: binary variable indicating whether the sampled individual slept under a bed net the previous night: 1, if yes, 0, if no.
- MosqCntl: binary variable indicating whether the household has used some kind of mosquito control, such as sprays and coils: 1, if yes, 0, if no.
- IRS: binary variables in indicating whether there has been indoor residual spraying (IRS) in the house in the last 12 months: 1, if yes, 0, if no.
- Travel: binary variable indicating whether the sampled individual has traveled outside the village in the last three months: 1, if yes, 0, if no.
- SES: ordinal variable indicating the socio-economic status (SES) of the household. The variables is an integer score from 1(=poor) to 5(=rich).
- District: factor variable indicating the village of the sampled individual, "Kisii Central" or "Rachuonyo".
- Survey: factor variable indicating the survey in which the participant was enrolled, "community" or "school".
- elevation: elevation, in meters, of the recorded household location

Usage

```
data(malkenya)
```

Format

A data frame with 82014 rows and 13 variables

Source

Stevenson, J.C., Stresman, G.H., Gitonga, C.W., Gillig, J., Owaga, C., et al. (2013). Reliability of School Surveys in Estimating Geographic Variation in Malaria Transmission in the Western Kenyan Highlands. PLOS ONE 8(10): e77641. doi: 10.1371/journal.pone.0077641

`malnutrition` *Malnutrition in Ghana*

Description

This geostatistical dataset was extracted from the Demographic and Health Survey 2014 conducted in Ghana.

- lng Longitude of the sampling cluster.
- lat Latitude of the sampling cluster.
- age age in months of the child.
- sex sex of the child.
- HAZ height-for-age Z-score.
- WAZ weight-for-age Z-score
- urb binary indicator: urban area=1; rural area=0.
- etn ethnic group.
- edu level of education of the mother, which takes integer values from 1="Poorly educated" to 3="Highly educated".
- wealth wealth score of the household, which takes integer values from 1="Poor" to 3="Rich".

The coordinate reference system is 3857.

Usage

```
data(malnutrition)
```

Format

A data frame with 2671 rows and 10 variables

Source

Demographic and Health Survey, dhsprogram.com

`matern.grad.phi` *First Derivative with Respect to ϕ*

Description

Computes the first derivative of the Matern correlation function with respect to ϕ.

Usage

```
matern.grad.phi(U, phi, kappa)
```

Arguments

`U`	A vector of distances between pairs of data locations.
`phi`	The scale parameter ϕ.
`kappa`	The smoothness parameter κ.

Value

A matrix with the values of the first derivative of the Matern function with respect to ϕ for the given distances.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`matern.hessian.phi` *Second Derivative with Respect to ϕ*

Description

Computes the second derivative of the Matern correlation function with respect to ϕ.

Usage

```
matern.hessian.phi(U, phi, kappa)
```

Arguments

`U`	A vector of distances between pairs of data locations.
`phi`	The scale parameter ϕ.
`kappa`	The smoothness parameter κ.

Value

A matrix with the values of the second derivative of the Matern function with respect to ϕ for the given distances.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`matern_cor` *Matern Correlation Function*

Description

Computes the Matern correlation function.

Usage

```
matern_cor(u, phi, kappa, return_sym_matrix = FALSE)
```

Arguments

`u`	A vector of distances between pairs of data locations.
`phi`	The scale parameter ϕ.
`kappa`	The smoothness parameter κ.
`return_sym_matrix`	A logical value indicating whether to return a symmetric correlation matrix. Defaults to `FALSE`.

Details

The Matern correlation function is defined as

Value

A vector of the same length as `u` with the values of the Matern correlation function for the given distances, if `return_sym_matrix=FALSE`. If `return_sym_matrix=TRUE`, a symmetric correlation matrix is returned.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

$$\rho(u;\phi;\kappa) = (2^{\kappa-1})^{-1}(u/\phi)^{\kappa}K_{\kappa}(u/\phi)$$

where ϕ and κ are the scale and smoothness parameters, and $K_{\kappa}(\cdot)$ denotes the modified Bessel function of the third kind of order κ. The parameters ϕ and κ must be positive.

`maxim.integrand` *Maximization of the Integrand for Generalized Linear Gaussian Process Models*

Description

Maximizes the integrand function for Generalized Linear Gaussian Process Models (GLGPMs), which involves the evaluation of likelihood functions with spatially correlated random effects.

Usage

```
maxim.integrand(
  y,
  units_m,
  mu,
  Sigma,
  ID_coords,
  ID_re = NULL,
  family,
  sigma2_re = NULL,
  hessian = FALSE,
  gradient = FALSE,
  invlink = NULL
)
```

Arguments

`y`	Response variable vector.
`units_m`	Units of measurement for the response variable.
`mu`	Mean vector of the response variable.
`Sigma`	Covariance matrix of the spatial process.
`ID_coords`	Indices mapping response to locations.
`ID_re`	Indices mapping response to unstructured random effects.
`family`	Distribution family for the response variable. Must be one of 'gaussian', 'binomial', or 'poisson'.
`sigma2_re`	Variance of the unstructured random effects.
`hessian`	Logical; if TRUE, compute the Hessian matrix.
`gradient`	Logical; if TRUE, compute the gradient vector.
`invlink`	A function that defines the inverse of the link function for the distribution of the data given the random effects.

Details

This function maximizes the integrand for GLGPMs using the Nelder-Mead optimization algorithm. It computes the likelihood function incorporating spatial covariance and unstructured random effects, if provided.

The integrand includes terms for the spatial process (Sigma), unstructured random effects (sigma2_re), and the likelihood function (llik) based on the specified distribution family ('gaussian', 'binomial', or 'poisson').

Value

A list containing the mode estimate, and optionally, the Hessian matrix and gradient vector.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`plot.RiskMap_pred_target_grid`

Plot Method for RiskMap_pred_target_grid Objects

Description

Generates a plot of the predicted values or summaries over the regular spatial grid from an object of class 'RiskMap_pred_target_grid'.

Usage

```
## S3 method for class 'RiskMap_pred_target_grid'
plot(x, which_target = "linear_target", which_summary = "mean", ...)
```

Arguments

`x`	An object of class 'RiskMap_pred_target_grid'.
`which_target`	Character string specifying which target prediction to plot.
`which_summary`	Character string specifying which summary statistic to plot (e.g., "mean", "sd").
`...`	Additional arguments passed to the `plot` function of the `terra` package.

Details

This function requires the 'terra' package for spatial data manipulation and plotting. It plots the values or summaries over a regular spatial grid, allowing for visual examination of spatial patterns.

Value

A `ggplot` object representing the specified prediction target or summary statistic over the spatial grid.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`pred_target_grid`

`plot.RiskMap_pred_target_shp`

Plot Method for RiskMap_pred_target_shp Objects

Description

Generates a plot of predictive target values or summaries over a shapefile.

Usage

```
## S3 method for class 'RiskMap_pred_target_shp'
plot(x, which_target = "linear_target", which_summary = "mean", ...)
```

Arguments

`x`	An object of class 'RiskMap_pred_target_shp' containing computed targets, summaries, and associated spatial data.
`which_target`	Character indicating the target type to plot (e.g., "linear_target").
`which_summary`	Character indicating the summary type to plot (e.g., "mean", "sd").
`...`	Additional arguments passed to 'scale_fill_distiller' in 'ggplot2'.

Details

This function plots the predictive target values or summaries over a shapefile. It requires the 'ggplot2' package for plotting and 'sf' objects for spatial data.

Value

A `ggplot` object showing the plot of the specified predictive target or summary.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`pred_target_shp`, `ggplot`, `geom_sf`, `aes`, `scale_fill_distiller`

`plot_AnPIT` *Plot Calibration Curves (AnPIT / PIT) from Spatial Cross-Validation*

Description

Produce calibration plots from a `RiskMap.spatial.cv` object returned by `assess_pp`. * For Binomial or Poisson models the function visualises the *Aggregated normalised Probability Integral Transform* (AnPIT) curves stored in `$AnPIT`. * For Gaussian models it detects the list `$PIT` and instead plots the empirical *Probability Integral Transform* curve (ECDF of PIT values) on the same u-grid.

A 45° dashed red line indicates perfect calibration.

Usage

```
plot_AnPIT(
  object,
  mode = "average",
  test_set = NULL,
  model_name = NULL,
  combine_panels = FALSE
)
```

Arguments

`object`	A `RiskMap.spatial.cv` object.
`mode`	One of `"average"` (average curve across test sets), `"single"` (a specific test set), or `"all"` (every test set separately).
`test_set`	Integer; required when `mode = "single"`.
`model_name`	Optional character string; if supplied, only that model is plotted.
`combine_panels`	Logical; when `mode = "average"`, draw all models in a single panel (`TRUE`) or one panel per model (`FALSE`, default).

Value

A **ggplot2** object (single plot) or a **ggpubr** grid.

`plot_mda` *Plot the estimated MDA impact function*

Description

Generate a plot of the estimated impact of mass drug administration (MDA) on infection prevalence, based on a fitted decay-adjusted spatio-temporal (DAST) model. The function simulates draws from the posterior distribution of model parameters, propagates them through the MDA effect function, and produces uncertainty bands around the estimated impact curve.

Usage

```
plot_mda(
  object,
  mda_history = NULL,
  n_sim = 1000,
  x_min = 1e-06,
  x_max = 10,
  conf_level = 0.95,
  lower_f = NULL,
  upper_f = NULL,
  mc_cores = 1,
  parallel_backend = c("none", "fork", "psock"),
  ...
)
```

Arguments

`object`	A fitted DAST model object, returned by `dast`.
`mda_history`	Specification of the MDA schedule. This can be either: • A numeric vector of event times (integers starting at 0, e.g. `c(0,1,2,6)`), • OR a 0/1 indicator vector on the yearly grid (e.g. `c(1,1,1,0,0,0,1)`), where position `i` corresponds to year `i-1`. If omitted, the default is a single MDA at time 0.
`n_sim`	Number of posterior draws used for uncertainty quantification (default: 1000).
`x_min`	Minimum value for the x-axis (default: `1e-6`).
`x_max`	Maximum value for the x-axis (default: `10`).
`conf_level`	Confidence level for the pointwise uncertainty interval (default: 0.95).
`lower_f`	Optional lower bound for the y-axis. If not provided, computed from the data.
`upper_f`	Optional upper bound for the y-axis. If not provided, computed from the data.
`mc_cores`	Number of CPU cores to use for parallel simulation. Default is 1 (serial).
`parallel_backend`	Parallelisation backend to use. Options are `"none"` (default), `"fork"` (Unix-like systems), or `"psock"` (cross-platform).
`...`	Additional arguments (currently unused).

Details

The time axis is assumed to start at 0 and increase in integer steps of 1 year. The argument `mda_history` allows the user to specify when MDAs occurred either by listing the years directly or by giving a binary indicator on the yearly grid. The function then evaluates the cumulative relative reduction $1 - \text{effect}(t)$ at a dense grid of time points between `x_min` and `x_max`, using the fitted parameters from the supplied DAST model.

Value

A `ggplot2` object showing the median estimated MDA impact function and the pointwise uncertainty band at the chosen confidence level.

`plot_score` *Plot Spatial Scores for a Specific Model and Metric*

Description

This function visualizes spatial scores for a specified model and metric. It combines test set data, handles duplicate locations by averaging scores, and creates a customizable map using ggplot2.

Usage

```
plot_score(object, which_score, which_model, ...)
```

Arguments

`object`	A list containing test sets and model scores. The structure should include ‘object$test_set‘ (list of sf objects) and ‘object$model[[which_model]]$score[[which_score]]‘.
`which_score`	A string specifying the score to visualize. Must match a score computed in the model.
`which_model`	A string specifying the model whose scores to visualize.
`...`	Additional arguments to customize ggplot, such as ‘scale_color_gradient‘ or ‘scale_color_manual‘.

Value

A ggplot object visualizing the spatial distribution of the specified score.

`plot_sim_surf` *Plot simulated surface data for a given simulation*

Description

This function plots the simulated surface data for a specific simulation from the result of ‘surf_sim‘. It visualizes the linear predictor values on a raster grid along with the actual data points.

Usage

```
plot_sim_surf(surf_obj, sim, ...)
```

Arguments

`surf_obj`	The output object from ‘surf_sim‘, containing both simulated data (‘data_sim‘) and predicted grid simulations (‘lp_grid_sim‘).
`sim`	The simulation index to plot.
`...`	Additional graphical parameters to be passed to the plotting function of the ‘terra‘ package.

Value

A plot of the simulation results.

`plot_s_variogram` *Plotting the empirical variogram*

Description

Plots the empirical variogram generated by `s_variogram`

Usage

```
plot_s_variogram(variog_output, plot_envelope = FALSE, color = "royalblue1")
```

Arguments

`variog_output`	The output generated by the function `s_variogram`.
`plot_envelope`	A logical value indicating if the envelope of spatial independence generated using the permutation test must be displayed (`plot_envelope = TRUE`) or not (`plot_envelope = FALSE`). By default `plot_envelope = FALSE`. Note: if `n_permutation = 0` when running the function `s_variogram`, the function will display an error message because no envelope can be generated.
`color`	If `plot_envelope = TRUE`, it sets the colour of the envelope; run `vignette("ggplot2-specs")` for more details on this argument.

Details

This function plots the empirical variogram, which shows the spatial dependence structure of the data. If `plot_envelope` is set to `TRUE`, the plot will also include an envelope indicating the range of values under spatial independence, based on a permutation test.

Value

A `ggplot` object representing the empirical variogram plot, optionally including the envelope of spatial independence.

See Also

`s_variogram`

`pred_over_grid`	*Prediction of the random effects components and covariates effects over a spatial grid using a fitted generalized linear Gaussian process model*

Description

This function computes predictions over a spatial grid using a fitted model obtained from the `glgpm` function. It provides point predictions and uncertainty estimates for the specified locations for each component of the model separately: the spatial random effects; the unstructured random effects (if included); and the covariates effects.

Usage

```
pred_over_grid(
  object,
  grid_pred = NULL,
  predictors = NULL,
  re_predictors = NULL,
  pred_cov_offset = NULL,
  control_sim = set_control_sim(),
  type = "marginal",
  messages = TRUE
)
```

Arguments

`object`	A RiskMap object obtained from the ‘glgpm‘ function.
`grid_pred`	An object of class ’sfc’, representing the spatial grid over which predictions are to be made. Must be in the same coordinate reference system (CRS) as the object passed to ’object’.
`predictors`	Optional. A data frame containing predictor variables used for prediction.

`re_predictors`	Optional. A data frame containing predictors for unstructured random effects, if applicable.
`pred_cov_offset`	Optional. A numeric vector specifying covariate offsets at prediction locations.
`control_sim`	Control parameters for MCMC sampling. Must be an object of class "mcmc.RiskMap" as returned by `set_control_sim`.
`type`	Type of prediction. "marginal" for marginal predictions, "joint" for joint predictions.
`messages`	Logical. If TRUE, display progress messages. Default is TRUE.

Value

An object of class 'RiskMap.pred.re' containing predicted values, uncertainty estimates, and additional information.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`pred_target_grid` *Predictive Target Over a Regular Spatial Grid*

Description

Computes predictions over a regular spatial grid using outputs from the `pred_over_grid` function. This function allows for incorporating covariates, offsets, MDA effects, and optional unstructured random effects into the predictive target.

Usage

```
pred_target_grid(
  object,
  include_covariates = TRUE,
  include_nugget = FALSE,
  include_cov_offset = FALSE,
  include_mda_effect = TRUE,
  mda_grid = NULL,
  time_pred = NULL,
  include_re = FALSE,
  f_target = NULL,
  pd_summary = NULL
)
```

Arguments

`object`	Output from ‘pred_over_grid‘, a RiskMap.pred.re object.
`include_covariates`	Logical. Include covariates in the predictive target.
`include_nugget`	Logical. Include the nugget effect in the predictive target.
`include_cov_offset`	Logical. Include the covariate offset in the predictive target.
`include_mda_effect`	Logical. Include the MDA effect in the predictive target using a DAST model; see `dast`.
`mda_grid`	Optional. Grid of MDA coverage values required for predictions using a DAST model; see `dast`.
`time_pred`	Optional. Time point for prediction required for predictions using a DAST model; see `dast`.
`include_re`	Logical. Include unstructured random effects in the predictive target.
`f_target`	Optional. List of functions to apply on the linear predictor samples.
`pd_summary`	Optional. List of summary functions to apply on the predicted values.

Value

An object of class ’RiskMap_pred_target_grid’ containing predicted values and summaries over the regular spatial grid.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`pred_over_grid`

`pred_target_shp` *Predictive Targets over a Shapefile (grid-aggregated)*

Description

Computes predictive targets over polygon features using joint prediction samples from `pred_over_grid`. Targets can incorporate covariates, offsets, optional unstructured random effects, and (if fitted) mass drug administration (MDA) effects from a DAST model.

Usage

```
pred_target_shp(
  object,
  shp,
  shp_target = mean,
  weights = NULL,
  standardize_weights = FALSE,
  col_names = NULL,
  include_covariates = TRUE,
  include_nugget = FALSE,
  include_cov_offset = FALSE,
  include_mda_effect = TRUE,
  return_shp = TRUE,
  time_pred = NULL,
  mda_grid = NULL,
  include_re = FALSE,
  f_target = NULL,
  pd_summary = NULL,
  messages = TRUE,
  return_target_samples = FALSE
)
```

Arguments

`object`	Output from `pred_over_grid` (class `RiskMap.pred.re`), typically fitted with `type = "joint"` so that linear predictor samples are available.
`shp`	An **sf** polygon object (preferred) or a `data.frame` with an attached geometry column, representing regions over which predictions are aggregated.
`shp_target`	A function that aggregates grid-cell values within each polygon to a single regional value (default `mean`). Examples: `mean`, `sum`, a custom weighted mean, etc.
`weights`	Optional numeric vector of weights used inside `shp_target`. If supplied with `standardize_weights = TRUE`, weights are normalized within each region.
`standardize_weights`	Logical; standardize `weights` within each region (`FALSE` by default).
`col_names`	Name or column index in `shp` containing region identifiers to use in outputs.
`include_covariates`	Logical; include fitted covariate effects in the linear predictor (default `TRUE`).
`include_nugget`	Logical; include the nugget (unstructured measurement error) in the linear predictor (default `FALSE`).
`include_cov_offset`	Logical; include any covariate offset term (default `FALSE`).
`include_mda_effect`	Logical; include the MDA effect as defined by the fitted DAST model (default `TRUE`). Requires `time_pred` and, when applicable, `mda_grid`.

`return_shp`	Logical; if `TRUE`, return the shapefile with appended summary columns defined by `pd_summary` (default `TRUE`).
`time_pred`	Optional numeric scalar (or time index) at which to evaluate the predictive target
`mda_grid`	Optional structure describing MDA schedules aligned with prediction grid cells (e.g., a `data.frame`/matrix/list). Used only when `include_mda_effect = TRUE`.
`include_re`	Logical; include unstructured random effects (RE) in the linear predictor (default `FALSE`).
`f_target`	List of target functions applied to linear predictor samples (e.g., `list(prev = plogis)` for prevalence on the probability scale). If `NULL`, the identity is used.
`pd_summary`	Named list of summary functions applied to each region's target samples (e.g., `list(mean = mean, sd = sd, q025 = function(x) quantile(x, 0.025), q975 = function(x) quantile(x, 0.975))`). Names are used as column suffixes in the outputs.
`messages`	Logical; if `TRUE`, print progress messages while computing regional targets.
`return_target_samples`	Logical; if `TRUE`, also return the raw target samples per region (default `FALSE`).

Details

For each polygon in `shp`, grid-cell samples of the linear predictor are transformed with `f_target`, optionally adjusted for covariates, offset, nugget, MDA effects and/or REs, and then aggregated via `shp_target` (optionally weighted). The list `pd_summary` is applied to each region's target samples to produce summary statistics.

Value

An object of class `RiskMap_pred_target_shp` with components:

- `target`: `data.frame` of region-level summaries (one row per region).
- `target_samples`: (optional) `list` with one element per region; each contains a `data.frame`/matrix of raw samples for each named target in `f_target`, if `return_target_samples = TRUE`.
- shp: (optional) the input `sf` object with appended summary columns, included if `return_shp = TRUE`.
- `f_target`, `pd_summary`, `grid_pred`: inputs echoed for reproducibility.

See Also

`pred_over_grid`, `pred_target_grid`

`print.summary.RiskMap` *Print Summary of RiskMap Model*

Description

Provides a `print` method for the summary of "RiskMap" objects, detailing the model type, parameter estimates, and other relevant statistics.

Usage

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

Arguments

`x`	An object of class "summary.RiskMap".
`...`	other parameters.

Details

This function prints a detailed summary of a fitted "RiskMap" model, including:

- The type of geostatistical model (e.g., Gaussian, Binomial, Poisson).
- Confidence intervals for parameter estimates.
- Regression coefficients with their standard errors and p-values.
- Measurement error variance, if applicable.
- Spatial process parameters, including the Matern covariance parameters.
- Variance of the nugget effect, if applicable.
- Unstructured random effects variances, if applicable.
- Log-likelihood of the model.
- Akaike Information Criterion (AIC) for Gaussian models.

Value

This function is used for its side effect of printing to the console. It does not return a value.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`print.summary.RiskMap.sim.res`
Print Simulation Results

Description

Prints a concise summary of simulation results from a ‘RiskMap.sim.res‘ object, including average metrics by category and a summary of overall correct classification (CC).

Usage

```
## S3 method for class 'summary.RiskMap.sim.res'
print(x, ...)
```

Arguments

`x`	An object of class ‘summary.RiskMap.sim.res‘, as returned by ‘summary.RiskMap.sim.res‘.
`...`	Additional arguments (not used).

Value

Invisibly returns ‘x‘.

Print Simulation Results

Prints a concise summary of simulation results from a ‘summary.RiskMap.sim.res‘ object, including average metrics by category and a summary of overall correct classification (CC).

Invisibly returns ‘x‘.

`print.summary.RiskMap.spatial.cv`
Print Summary of RiskMap Spatial Cross-Validation Scores

Description

This function prints the matrix of cross-validation scores produced by ‘summary.RiskMap.spatial.cv‘ in a readable format.

Usage

```
## S3 method for class 'summary.RiskMap.spatial.cv'
print(x, ...)
```

Arguments

`x`	An object of class ‘"summary.RiskMap.spatial.cv"‘, typically the output of ‘summary.RiskMap.spatial.cv‘.
`...`	Additional arguments passed to or from other methods.

Details

This method is primarily used to format and display the summary score matrix, printing it to the console. It provides a clear view of the cross-validation performance metrics across different spatial models.

Value

This function is used for its side effect of printing to the console. It does not return a value.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

`propose_utm` *EPSG of the UTM Zone*

Description

Suggests the EPSG code for the UTM zone where the majority of the data falls.

Usage

```
propose_utm(data)
```

Arguments

`data`	An object of class `sf` containing the coordinates.

Details

The function determines the UTM zone and hemisphere where the majority of the data points are located and proposes the corresponding EPSG code.

Value

An integer indicating the EPSG code of the UTM zone.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`> Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`re` *Random Effect Model Specification*

Description

Specifies the terms for a random effect model.

Usage

```
re(...)
```

Arguments

`...`	Variables representing the random effects in the model.

Details

The function constructs a list that includes the specified terms for the random effects. This list can be used as a specification for a random effect model.

Value

A list of class `re.spec` containing the following elements:

`term`	A character vector of the specified terms.
`dim`	The number of specified terms.
`label`	A character string representing the full call for the random effect model.

Note

At least one variable must be provided as input.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`set_control_sim` *Set Control Parameters for Simulation*

Description

This function sets control parameters for running simulations, particularly for MCMC methods. It allows users to specify the number of simulations, burn-in period, thinning interval, and various other parameters necessary for the simulation.

Usage

```
set_control_sim(
  n_sim = 12000,
  burnin = 2000,
  thin = 10,
  h = NULL,
  c1.h = 0.01,
  c2.h = 1e-04,
  linear_model = FALSE
)
```

Arguments

`n_sim`	Integer. The total number of simulations to run. Default is 12000.
`burnin`	Integer. The number of initial simulations to discard (burn-in period, used for the MCMC algorithm). Default is 2000.
`thin`	Integer. The interval at which simulations are recorded (thinning interval, used for the MCMC algorithm). Default is 10.
`h`	Numeric. An optional parameter. Must be non-negative if specified.
`c1.h`	Numeric. A control parameter for the simulation. Must be positive. Default is 0.01.
`c2.h`	Numeric. Another control parameter for the simulation. Must be between 0 and 1. Default is 1e-04.
`linear_model`	Logical. If TRUE, the function sets up parameters for a linear model and only returns `n_sim`. Default is FALSE.

Details

The function validates the input parameters and ensures they are appropriate for the simulation that is used in the `glgpm` fitting function. For non-linear models, it checks that `n_sim` is greater than `burnin`, that `thin` is positive and a divisor of `(n_sim - burnin)`, and that `h`, `c1.h`, and `c2.h` are within their respective valid ranges.

If `linear_model` is TRUE, only `n_sim` and `linear_model` are required, and the function returns a list containing these parameters.

If `linear_model` is FALSE, the function returns a list containing `n_sim`, `burnin`, `thin`, `h`, `c1.h`, `c2.h`, and `linear_model`.

Value

A list of control parameters for the simulation with class attribute "mcmc.RiskMap".

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`Matrix`, `forceSymmetric`

Examples

```
# Example with default parameters
control_params <- set_control_sim()

# Example with custom parameters
control_params <- set_control_sim(n_sim = 15000, burnin = 3000, thin = 20)
```

`summary.RiskMap` *Summarize Model Fits*

Description

Provides a `summary` method for the "RiskMap" class that computes the standard errors and p-values for likelihood-based model fits.

Usage

```
## S3 method for class 'RiskMap'
summary(object, ..., conf_level = 0.95)
```

Arguments

`object`	An object of class "RiskMap" obtained as a result of a call to `glgpm`.
`...`	other parameters.
`conf_level`	The confidence level for the intervals (default is 0.95).

Details

This function computes the standard errors and p-values for the parameters of a "RiskMap" model, adjusting for the covariance structure if needed.

Value

A list containing:

`reg_coef`	A matrix with the estimates, standard errors, z-values, p-values, and confidence intervals for the regression coefficients.
`me`	A matrix with the estimates and confidence intervals for the measurement error variance, if applicable.
`sp`	A matrix with the estimates and confidence intervals for the spatial process parameters.
`tau2`	The fixed nugget variance, if applicable.
`ranef`	A matrix with the estimates and confidence intervals for the random effects variances, if applicable.
`conf_level`	The confidence level used for the intervals.
`family`	The family of the model (e.g., "gaussian").
`kappa`	The kappa parameter of the model.
`log.lik`	The log-likelihood of the model fit.
`cov_offset_used`	A logical indicating if a covariance offset was used.
`aic`	The Akaike Information Criterion (AIC) for the model, if applicable.

Note

Handles both Gaussian and non-Gaussian families, and accounts for fixed and random effects in the model.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`glgpm`, `coef.RiskMap`

`summary.RiskMap.sim.res`

Summarize Simulation Results

Description

Summarizes the results of model evaluations from a ‘RiskMap.sim.res‘ object. Provides average metrics for classification by category and overall correct classification (CC) summary.

Usage

```
## S3 method for class 'RiskMap.sim.res'
summary(object, ...)
```

Arguments

`object`	An object of class ‘RiskMap.sim.res‘, as returned by ‘assess_sim‘.
`...`	Additional arguments (not used).

Value

A list containing summary data for each model: - ‘by_cat_summary‘: A data frame with average sensitivity, specificity, PPV, NPV, and CC by category. - ‘CC_summary‘: A numeric vector with mean, 2.5th percentile, and 97.5th percentile for CC across simulations.

---

summary.RiskMap.spatial.cv

Summarize Cross-Validation Scores for Spatial RiskMap Models

Description

This function summarizes cross-validation scores for different spatial models obtained from `assess_pp`.

Usage

```
## S3 method for class 'RiskMap.spatial.cv'
summary(object, view_all = TRUE, ...)
```

Arguments

`object`	A ‘RiskMap.spatial.cv‘ object containing cross-validation scores for each model, as obtained from `assess_pp`.
`view_all`	Logical. If ‘TRUE‘, stores the average scores across test sets for each model alongside the overall average across all models. Defaults to ‘TRUE‘.
`...`	Additional arguments passed to or from other methods.

Details

The function computes and returns a matrix where rows correspond to models and columns correspond to performance metrics (e.g., CRPS, SCRPS). Scores are weighted by subset sizes to compute averages. Attributes of the returned object include:

- ‘test_set_means‘: A list of average scores for each test set and model.
- ‘overall_averages‘: Overall averages for each metric across all models.
- ‘view_all‘: Indicates whether averages across test sets are available for visualization.

Value

A matrix of summary scores with models as rows and metrics as columns, with class ‘"summary.RiskMap.spatial.cv"‘.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

See Also

`assess_pp`

`surf_sim` *Simulate surface data based on a spatial model*

Description

This function simulates surface data based on a user-defined formula and other parameters. It allows for simulation of spatial data with various model families (Gaussian, Binomial, or Poisson). The simulation involves creating spatially correlated random fields and generating outcomes for data points in a given prediction grid.

Usage

```
surf_sim(
  n_sim,
  pred_grid,
  formula,
  sampling_f,
  family,
  scale_to_km = TRUE,
  control_mcmc = set_control_sim(),
  par0,
  nugget_over_grid = FALSE,
  include_covariates = TRUE,
  fix_var_me = NULL,
  messages = TRUE
)
```

Arguments

`n_sim`	The number of simulations to run.
`pred_grid`	A spatial object (either ‘sf‘ or ‘data.frame‘) representing the prediction grid where the simulation will take place.
`formula`	A formula object specifying the model to be fitted. It should include both fixed effects and random effects if applicable.

`sampling_f`	A function that returns a sampled dataset (of class ‘sf‘ or ‘data.frame‘) to simulate data from.
`family`	A character string specifying the family of the model. Must be one of "gaussian", "binomial", or "poisson".
`scale_to_km`	A logical indicating whether the coordinates should be scaled to kilometers. Defaults to ‘TRUE‘.
`control_mcmc`	A list of control parameters for MCMC (not used in this implementation but can be expanded later).
`par0`	A list containing initial parameter values for the simulation, including ‘beta‘, ‘sigma2‘, ‘phi‘, ‘tau2‘, and ‘sigma2_me‘.
`nugget_over_grid`	A logical indicating whether to include a nugget effect over the entire prediction grid.
`include_covariates`	A logical indicateing if the covariates (or the intercept if no covariates are used) should be included in the linear predictor. By default `include_covariates = TRUE`
`fix_var_me`	A parameter to fix the variance of the random effects for the measurement error. Defaults to ‘NULL‘.
`messages`	A logical value indicating whether to print messages during the simulation. Defaults to ‘TRUE‘.

Value

A list containing the simulated data (`data_sim`), the linear predictors (`lp_grid_sim`), a logical value indicating if covariates have been included in the linear predictor (`include_covariates`), a logical value indicating if the nugget has been included into the simulations of the linear predictor over the grid (`nugget_over_grid`), a logical indicating if a covariate offset has been included in the linear predictor (`include_cov_offset`), the model parameters set for the simulation (`par0`) and the family used in the model (`family`).

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

`s_variogram` *Empirical variogram*

Description

Computes the empirical variogram using “bins” of distance provided by the user.

Usage

```
s_variogram(
  data,
  variable,
  bins = NULL,
  n_permutation = 0,
  convert_to_utm = TRUE,
  scale_to_km = FALSE
)
```

Arguments

`data`	an object of class `sf` containing the variable for which the variogram is to be computed and the coordinates
`variable`	a character indicating the name of variable for which the variogram is to be computed.
`bins`	a vector indicating the 'bins' to be used to define the classes of distance used in the computation of the variogram. By default `bins=NULL` and bins are then computed as `seq(0, d_max/2, length=15)` where `d_max` is the maximum distance observed in the data.
`n_permutation`	a non-negative integer indicating the number of permutation used to compute the 95 level envelope under the assumption of spatial independence. By default `n_permutation=0`, and no envelope is generated.
`convert_to_utm`	a logical value, indicating if the conversion to UTM shuold be performed (`convert_to_utm = TRUE`) or the coordinate reference system of the data must be used without any conversion (`convert_to_utm = FALSE`). By default `convert_to_utm = TRUE`. Note: if `convert_to_utm = TRUE` the conversion to UTM is performed using the epsg provided by `propose_utm`.
`scale_to_km`	a logical value, indicating if the distances used in the variogram must be scaled to kilometers (`scale_to_km = TRUE`) or left in meters (`scale_to_km = FALSE`). By default `scale_to_km = FALSE`

Value

an object of class 'variogram' which is a list containing the following components

`variogram` a data-frame containing the following columns: `mid_points`, the middle points of the classes of distance provided by `bins`; `obs_vari` the values of the observed variogram; `obs_vari` the number of pairs. If `n_permutation > 0`, the data-frame also contains `lower_bound` and `upper_bound` corresponding to the lower and upper bounds of the 95 used to assess the departure of the observed variogram from the assumption of spatial independence.

`scale_to_km` the value passed to `scale_to_km`

`n_permutation` the number of permutations

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

`to_table` *Generate LaTeX Tables from RiskMap Model Fits and Validation*

Description

Converts a fitted "RiskMap" model or cross-validation results into an `xtable` object, formatted for easy export to LaTeX or HTML.

Usage

```
to_table(object, ...)
```

Arguments

`object`	An object of class "RiskMap" resulting from a call to `glgpm`, or a summary object of class "summary.RiskMap.spatial.cv" containing cross-validation results.
`...`	Additional arguments to be passed to `xtable` for customization.

Details

This function creates a summary table from a fitted "RiskMap" model or cross-validation results for multiple models, returning it as an `xtable` object.

When the input is a "RiskMap" model object, the table includes:

- Regression coefficients with their estimates, confidence intervals, and p-values.
- Parameters for the spatial process.
- Random effect variances.
- Measurement error variance, if applicable.

When the input is a cross-validation summary object ("summary.RiskMap.spatial.cv"), the table includes:

- A row for each model being compared.
- Performance metrics such as CRPS and SCRPS for each model.

The resulting `xtable` object can be further customized with additional formatting options and printed as a LaTeX or HTML table for reports or publications.

Value

An object of class "xtable", which contains the formatted table as a `data.frame` and several attributes specifying table formatting options.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

See Also

`glgpm`, `xtable`, `summary.RiskMap.spatial.cv`

`tz_covariates` *Covariates Dataset for Malaria Prediction in Tanzania*

Description

This dataset provides covariates over a 10 by 10 km regular grid covering Tanzania. It is intended to be used together with the 'tz_malaria' dataset for spatial prediction of malaria prevalence.

Population Population density in the area (in thousands).

ITN Percentage of households with at least one insecticide-treated net (ITN).

EVI Enhanced Vegetation Index, indicating vegetation density.

Temperature Average temperature in degrees Celsius.

NTL Nighttime light intensity, indicating urbanization and infrastructure.

Precipitation Total precipitation in millimeters.

utm_x UTM (Universal Transverse Mercator) x-coordinate of the grid point.

utm_y UTM (Universal Transverse Mercator) y-coordinate of the grid point.

Usage

```
data(tz_covariates)
```

Format

A data frame with 8740 observations of 8 variables. The CRS of the UTM coordinates is 32736.

Source

Giorgi E, Fronterrè C, Macharia PM, Alegana VA, Snow RW, Diggle PJ. 2021 Model building and assessment of the impact of covariates for disease prevalence mapping in low-resource settings: to explain and to predict. J. R. Soc. Interface 18: 20210104. doi:10.1098/rsif.2021.0104

`tz_malaria` *Malaria Dataset from Tanzania Demographic Health Surveys 2015*

Description

This dataset contains information on malaria prevalence and associated variables from the 2015 Tanzania Demographic Health Surveys. The data includes geographical, demographic, environmental, and health-related variables.

cluster.number Cluster number, identifying the survey cluster.
Lat Latitude of the survey cluster.
Long Longitude of the survey cluster.
MM Month of the survey (in two-digit format).
YY Year of the survey.
UpAge Upper age limit of the surveyed individuals in years.
LoAge Lower age limit of the surveyed individuals in years.
Ex Number of individuals examined for malaria.
Pf Number of individuals tested positive for Plasmodium falciparum (malaria parasite).
PfPR2.10 Plasmodium falciparum parasite rate in the population (aged 2-10 years).
Method Method used for malaria diagnosis (e.g., Rapid Diagnostic Test (RDT)).
EVI Enhanced Vegetation Index, indicating vegetation density.
Temperature Average temperature in degrees Celsius.
Precipitation Total precipitation in millimeters.
Population Population density in the area (in thousands).
ITN Percentage of households with at least one insecticide-treated net (ITN).
NTL Nighttime light intensity, indicating urbanization and infrastructure.
Urban.Rural Indicator of whether the area is urban ('U') or rural ('R').
utm_x UTM (Universal Transverse Mercator) x-coordinate of the survey cluster.
utm_y UTM (Universal Transverse Mercator) y-coordinate of the survey cluster.

Usage

```
data(tz_malaria)
```

Format

A data frame with 387 rows and 20 columns, containing the following variables: The CRS of the UTM coordinates is 32736.

Source

Tanzania Demographic Health Surveys 2015, Giorgi E, Fronterrè C, Macharia PM, Alegana VA, Snow RW, Diggle PJ. (2021) Model building and assessment of the impact of covariates for disease prevalence mapping in low-resource settings: to explain and to predict. J. R. Soc. Interface 18: 20210104. doi:10.1098/rsif.2021.0104

`update_predictors` *Update Predictors for a RiskMap Prediction Object*

Description

This function updates the predictors of a given RiskMap prediction object. It ensures that the new predictors match the original prediction grid and updates the relevant components of the object accordingly.

Usage

```
update_predictors(object, predictors)
```

Arguments

`object`	A ‘RiskMap.pred.re‘ object, which is the output of the `pred_over_grid` function.
`predictors`	A data frame containing the new predictor values. The number of rows must match the prediction grid in the ‘object‘.

Details

The function performs several checks and updates:

- Ensures that ‘object‘ is of class ‘RiskMap.pred.re‘.
- Ensures that the number of rows in ‘predictors‘ matches the prediction grid in ‘object‘.
- Removes any rows with missing values in ‘predictors‘ and updates the corresponding components of the ‘object‘.
- Updates the prediction locations, the predictive samples for the random effects, and the linear predictor.

Value

The updated ‘RiskMap.pred.re‘ object.

Author(s)

Emanuele Giorgi <`e.giorgi@lancaster.ac.uk`>

Claudio Fronterre <`c.fronterr@lancaster.ac.uk`>

Index