Package 'SCoRES'

November 17, 2025

Title Simultaneous Confidence Region Excursion Sets

Version 0.1.2

Author Zhuoran Yu [aut, cre],
Armin Schwartzman [aut],
Junting Ren [aut],
Julia Wrobel [aut]

Maintainer Zhuoran Yu <angela.yu@emory.edu>

Description Provides computational tools for estimating inverse regions and constructing the corresponding simultaneous outer and inner confidence regions. Acceptable input includes both one-dimensional and two-dimensional data for linear, logistic, functional, and spatial generalized least squares regression models. Functions are also available for constructing simultaneous confidence bands (SCBs) for these models. The definition of simultaneous confidence regions (SCRs) follows Sommerfeld et al. (2018) <doi:10.1080/01621459.2017.1341838>. Methods for estimating inverse regions, SCRs, and the nonparametric bootstrap are based on Ren et al. (2024) <doi:10.1093/jrsssc/qlae027>. Methods for constructing SCBs are described in Crainiceanu et al. (2024) <doi:10.1201/9781003278726> and Telschow et al. (2022) <doi:10.1016/j.jspi.2021.05.008>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

URL https://angelayustat.github.io/SCoRES/

BugReports https://github.com/AngelaYuStat/SCoRES/issues

Imports dplyr, forcats, ggplot2, grDevices, patchwork, MASS, Matrix, matrixStats, metR, nlme, stats, tidyr, refund, reshape, tibble, rlang, magrittr, utils

Depends R (>= 4.4)

LazyData true

Suggests knitr, mgcv, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr, rmarkdown

BuildVignettes true

LazyLoad no

ByteCompile false

NeedsCompilation no

Repository CRAN

Date/Publication 2025-11-17 21:20:12 UTC

Contents

`climate_data` *Historical and Future Summer Temperature Data in North America*

Description

A spatial dataset containing historical (1971-1999) and future (2041-2069) mean summer (June–August) surface temperatures over North America, used for evaluating increase of mean summer temperature between the 20th and 21st centuries in North America, and constructing simultaneous confidence bands via generalized least squares (GLS) modeling.

Usage

```
data(climate_data)
```

Format

A list with the following components:

`Z` A list containing spatial data with three components: `x` (longitude), `y` (latitude), and `obs`, a 3D array of observations with dimensions `[lon, lat, n]`. The first `na` slices of `z` come from mean summer temperature (June-August) in North America recorded from 1971 to 1999, and the last `nb` slices contain mean summer temperature from 2041 to 2069.

`mask` A logical or numeric matrix of dimensions `length(lon)` × `length(lat)`. Values are set to 1 for land and `NA` elsewhere based on the elevation matrix `orog > 0`.

`X` A numeric design matrix with `na + nb` rows and 4 columns, constructed for generalized least squares (GLS) regression. The rows correspond to spatial replicates from `na` current years and `nb` future years. The columns are:

1. `X1`: Group indicator (0 for current years (1971-1999), 1 for future years (2041-2069))
2. `X2`: Intercept
3. `X3`: Centered time variable `ta` for current years (1971-1999) (0 for future years (2041-2069))
4. `X4`: Centered time variable `tb` for future years (2041-2069) (0 for current years (1971-1999))

`correlation` A character string set to `"corAR1"`, indicating that an autoregressive correlation structure of order 1 (AR(1)) is used for GLS fitting.

Details

The data are arranged on a regular longitude–latitude grid, with spatial masking for land-only analysis. AR(1) correlation structure is assumed for statistical modeling.

Source

Processed from `data-raw/climate_data.R` using the `readr` package.

References

Sommerfeld, M., Sain, S., & Schwartzman, A. (2018). Confidence regions for spatial excursion sets from repeated random field observations, with an application to climate. *Journal of the American Statistical Association*, 113(523), 1327–1340. doi:10.1080/01621459.2017.1341838

`cma` *Construct CMA Confidence Intervals via Parametric Method*

Description

This function computes Correlation and Multiplicity Adjusted (CMA) confidence bands for a specified group in a functional outcome regression model using parameter simulations approach with Gaussian multiplier bootstrap.

Usage

```
cma(
  data_df,
  object,
  fitted = TRUE,
  alpha = 0.05,
  outcome,
  domain,
  subset = NULL,
  id,
  nboot = NULL
)
```

Arguments

data_df	A functional data frame that contains both name and values for variables including functional outcome, domain (e.g. time) and ID (e.g. subject names) used to fit the model object.
object	A fitted Function-on-Scalar Regression (FoSR) object (e.g., from mgcv::gam()/mgcv::bam()).
fitted	Logical. Whether to estimate the simultaneous confidence bands for the fitted mean function or the fitted parameter function • TRUE - Estimate the simultaneous confidence bands for the fitted mean outcome function. • FALSE - estimate the simultaneous confidence bands for the fitted parameter function. Default is TRUE.
alpha	Significance level for SCB. Default is 0.05.
outcome	A character string specifying the name of the outcome variable used in the model.
domain	A character string specifying the name of the domain variable (e.g. time) used in the model.
subset	An atomic character vector (e.g., c("user = 1", "age = 30")) specifying the target function for constructing the SCB. Each element must be of the form <name> = <value>, where <name> is the name of a scalar grouping variable and <value> is the desired value. Whitespace is ignored. Binary or categorical character variables should be transformed into numeric. Factors are not allowed here because if the input data contains factor variables, they will be automatically expanded into dummy (indicator) variables when constructing the design matrix, and the resulting variable names may differ from the original factor names. Default is NULL, representing the reference group.
id	A character string specifying the name of the ID variable.
nboot	An integer specifying the number of bootstrap samples used to construct the confidence bands. Default is 10,000.

Value

A list containing:

mu_hat	Estimated mean function for the group of interest.
domain	The domain used.
se_hat	Standard errors of the estimated means.
scb_low	Lower bound of the simultaneous confidence band.
scb_up	Upper bound of the simultaneous confidence band.
type	A character description of the output type.

References

Crainiceanu, C. M., Goldsmith, J., Leroux, A., & Cui, E. (2024). *Functional Data Analysis with R*. Chapman and Hall/CRC.

Examples

```
# example using pupil data
if (requireNamespace("mgcv", quietly = TRUE)) {
data(pupil)

pupil_fpca <- prepare_pupil_fpca(pupil)

fosr_mod <- mgcv::bam(percent_change ~ s(seconds, k=30, bs="cr") +
  s(seconds, by = use, k=30, bs = "cr") +
  s(id, by = Phi1, bs="re") +
  s(id, by = Phi2, bs="re")+
  s(id, by = Phi3, bs="re") +
  s(id, by = Phi4, bs="re"),
  method = "fREML", data = pupil_fpca, discrete = TRUE)

results <- cma(pupil_fpca, fosr_mod, fitted = TRUE, outcome = "percent_change",
               domain = "seconds", subset = c("use = 1"), id = "id")


mean_mod <- mgcv::gam(percent_change ~ s(seconds, k = 5, bs = "cr") +
s(seconds, by = use, k = 5, bs = "cr"),
data = pupil, method = "REML")

results <- cma(pupil, mean_mod, fitted = TRUE, outcome = "percent_change",
               domain = "seconds", subset = c("use = 1"), id = "id", nboot = 100)
}
```

expit *Expit (Inverse Logit) Function*

Description

Computes the inverse logit transformation.

Usage

```
expit(x)
```

Arguments

x	A numeric input.

Value

Value between 0 and 1.

Examples

```
expit(0)         # returns 0.5
expit(c(-2, 0, 2))
```

ipad *iPad task and physiology (40-minute timepoint)*

Description

The dataset ipad contains tablet-based task performance measures, pupillography features, blood cannabinoid metabolite concentrations, and cardiovascular measures collected **40 minutes after** smoking (or after a rest period for controls). Each row is one participant at this timepoint. The identifier id has been converted to a factor, and the data have been filtered to timept = 2 only.

Usage

```
data(ipad)
```

Format

A tibble, one row per participant at timepoint 2. Variables are grouped below.

Identifiers id Participant identifier (factor).

timept Timepoint indicator (fixed at 2 = 40 minutes).

use_group Participant use group: 1 = Daily, 2 = Occasional, 3 = No Use.

recent_smoke Recent use at this timepoint: 0 = No Use, 1 = Use.

Blood (metabolite concentrations) `t_thc`, `t_thc_oh`, `t_thc_cooh`, `t_thc_gluc`, `t_thc_cooh_gluc`, `t_cbg`, `t_cbd`, `t_cbn`, `t_mmr1`, `t_mmr2`.

Pupillography `p_fpc1`–`p_fpc6` (functional pupil components 1–6); `p_PMC_pctChg` (percent change at point of minimum constriction); `p_auc` (AUC of the pupillary constriction curve).

Tablet (task metrics) `i_prop_false_timeout`, `i_prop_failed1`, `i_prop_failed2`, `i_judgement_time1`, `i_judgement_time2`, `i_time_outside_reticle`, `i_time_on_edge`, `i_prop_hit`, `i_correct_reaction2`, `i_reaction_time_max2`, `i_reaction_time2`, `i_rep_shapes12`, `i_rep_shapes34`, `i_memory_time12`, `i_memory_time34`, `i_composite_score`.

Cardiovascular `h_hr` (heart rate), `h_dbp` (diastolic blood pressure), `h_sbp` (systolic blood pressure).

Details

Participants completed an iPad test assessing reaction time, decision making, working memory, and spatial-motor performance before and after cannabis use (or a rest period for controls). This dataset retains the **post** (40-minute) measurements only. Consider converting `use_group` and `recent_smoke` to factors with informative labels for modeling/plotting. Units for biochemical and physiological variables follow the original source.

References

Smith, S. J., Wrobel, J., Brooks-Russell, A., Kosnett, M. J., & Sammel, M. D. (2023). A Latent Variable Analysis of Psychomotor and Neurocognitive Performance After Acute Cannabis Smoking. *Cannabis (Albuquerque, N.M.)*, 6(2), 123–132. doi:10.26828/cannabis/2023/000156

Examples

```
data(ipad)
```

---

mean_response_predict *Functional Outcome Regression Prediction with Group-Specific Inference*

Description

This function is an internal function for constructing SCBs for functional data.

Usage

```
mean_response_predict(
  data_df,
  object,
  fitted = TRUE,
  outcome,
  domain,
  subset = NULL,
  id
)
```

Arguments

data_df	A functional data frame that contains both name and values for variables including functional outcome, domain (e.g. time) and ID (e.g. subject names) used to fit the model object.
object	A fitted Function-on-Scalar Regression (FoSR) model object (e.g., from mgcv::gam()/mgcv::bam()).
fitted	Logical. Whether to estimate the simultaneous confidence bands for the fitted mean function or the fitted parameter function • TRUE - Estimate the simultaneous confidence bands for the fitted mean outcome function. • FALSE - estimate the simultaneous confidence bands for the fitted parameter function. Default is TRUE.
outcome	A character string specifying the name of the outcome variable used in the model.
domain	A character string specifying the name of the domain variable (e.g. time) used in the model.
subset	An atomic character vector (e.g., c("user = 1", "age = 30")) specifying the target function for constructing the SCB. Each element must be of the form <name> = <value>, where <name> is the name of a scalar grouping variable and <value> is the desired value. Whitespace is ignored. Binary or categorical character variables should be transformed into numeric. Factors are not allowed here because if the input data contains factor variables, they will be automatically expanded into dummy (indicator) variables when constructing the design matrix, and the resulting variable names may differ from the original factor names. Default is NULL, which represents the reference group.
id	A character string specifying the name of the ID variable.

Value

A list containing the following elements:

s_pred Numeric vector of sorted unique domain used for prediction

pred_df Data frame with prediction results, containing:

- mean: Predicted mean values
- se: Standard errors

lpmat Linear predictor matrix (design matrix) used for confidence interval calculations

mod_coef Vector of model coefficients for selected group

mod_cov Variance-covariance matrix corresponding to the selected group coefficients

References

Crainiceanu, C. M., Goldsmith, J., Leroux, A., & Cui, E. (2024). *Functional Data Analysis with R*. Chapman and Hall/CRC.

Examples

```
if (requireNamespace("mgcv", quietly = TRUE)) {
data(pupil)


pupil_fpca <- prepare_pupil_fpca(pupil)

fosr_mod <- mgcv::bam(percent_change ~ s(seconds, k=30, bs="cr") +
  s(seconds, by = use, k=30, bs = "cr") +
  s(id, by = Phi1, bs="re") +
  s(id, by = Phi2, bs="re") +
  s(id, by = Phi3, bs="re") +
  s(id, by = Phi4, bs="re"),
  method = "fREML", data = pupil_fpca, discrete = TRUE)

results <- mean_response_predict(pupil_fpca, fosr_mod, fitted = TRUE,
outcome = "percent_change", domain = "seconds", subset = c("use = 1"), id = "id")

mean_mod <- mgcv::gam(percent_change ~ s(seconds, k = 5, bs = "cr") +
s(seconds, by = use, k = 5, bs = "cr"),
data = pupil, method = "REML")

results <- mean_response_predict(pupil, mean_mod, fitted = TRUE,
outcome = "percent_change", domain = "seconds", subset = c("use = 1"), id = "id")
}
```

---

plot_cs *Plot Inversion of Simultaneous Confidence Bands (SCBs) into Inner and Outer Simultaneous Confidence Regions (SCRs)*

Description

Visualizes simultaneous confidence regions of upper and lower excursion sets for discrete, 1D or 2D data, using contour or band plots. Supports plotting confidence regions at multiple levels and labeling contours.

Usage

```
plot_cs(
  SCB,
  levels,
  type = "upper",
  x,
  y = NULL,
  mu_hat = NULL,
  mu_true = NULL,
  together = TRUE,
```

```
  xlab = "X1",
  ylab = "X2",
  level_label = TRUE,
  min.size = 5,
  palette = "gray",
  color_level_label = "black"
)
```

Arguments

SCB	A numeric list returned by `regression_outcome_scb()`, `functional_outcome_scb()` or a custom list with two arrays of the same dimension: `scb_up` and `scb_low`, representing the upper and lower confidence bounds respectively. `SCB$scb_up` and `SCB$scb_low` should be numeric vectors (1D) or matrices (2D) containing the upper simultaneous confidence interval. Dimensions of `SCB$scb_up` and `SCB$scb_low` must match.
levels	A numeric vector or list of scalers for different levels or matrix containing interval sets to construct the confidence regions. If `type` = "upper" or "lower", `levels` should be a vector. "upper" represents upper excursion sets, and "lower" represents lower excursion sets.
type	A character specifying the type of inverse sets to fit. Choices are `"upper"` and `"lower"`. Default is `"upper"`.
x	A numerical vector of x-axis coordinates for 1D and 2D cases. For discrete coordinates, use a character vector. The order of x should correspond to the order of `scb_up` and `scb_low` in `SCB`.
y	Optional vector of y-axis coordinates for 2D data.
mu_hat	A numeric array (1D) or matrix (2D) of estimated means. If `mu_true` is provided, this will be overwritten by the true mean. Default is NULL. An input must be provided for either `mu_hat` or `mu_true`.
mu_true	Optional numeric array (1D) or matrix (2D) of true means, which overrides `mu_hat` if provided. Default is NULL.
together	Optional logical value for plotting option. If `TRUE`, plots all confidence levels on the same figure; otherwise, generates one plot per level. Default is `TRUE`.
xlab	Optional character for the label of the x-axis. Default is `"x1"`.
ylab	Optional character for the label of the y-axis. Default is `"x2"`.
level_label	Optional logical input for level displaying option. If `TRUE`, displays numeric level labels on contour lines for 2D confidence sets. Default is `TRUE`.
min.size	Optional logical input for minimum number of points required for a contour to be labeled. Default is 5.
palette	Optional character value for the name of the HCL color palette to use when plotting multiple levels together. Default is `"gray"`.
color_level_label	Optional character value for the color used for contour level labels. Default is `"black"`.

Value

A ggplot2 object that includes both simultaneous confidence intervals and simultaneous confidence region of excursion sets corresponding to levels assigned.

References

Ren, J., Telschow, F. J. E., & Schwartzman, A. (2024). Inverse set estimation and inversion of simultaneous confidence intervals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(4), 1082–1109. doi:10.1093/jrsssc/qlae027

Examples

```
if (requireNamespace("mgcv", quietly = TRUE)) {
# example using pupil data
data(pupil)
pupil_fpca <- prepare_pupil_fpca(pupil)

fosr_mod <- mgcv::bam(percent_change ~ s(seconds, k=30, bs="cr") +
  s(seconds, by = use, k=30, bs = "cr") +
  s(id, by = Phi1, bs="re") +
  s(id, by = Phi2, bs="re") +
  s(id, by = Phi3, bs="re") +
  s(id, by = Phi4, bs="re"),
  method = "fREML", data = pupil_fpca, discrete = TRUE)

pupil_multiplier <- SCB_functional_outcome(data = pupil_fpca, object = fosr_mod,
                                    method = "multiplier",
                                    outcome = "percent_change",
                                    domain = "seconds", subset= c("use = 1"),
                                    id = "id")

pupil_multiplier <- tibble::as_tibble(pupil_multiplier)

plot_cs(pupil_multiplier,levels = c(-18), x = pupil_multiplier$domain,
        mu_hat = pupil_multiplier$mu_hat, xlab = "", ylab = "",
        level_label = T, min.size = 40, palette = "Spectral",
        color_level_label = "black")
}


x <- rnorm(50)
epsilon <- rnorm(50,0,sqrt(2))
y <- -1 + x + epsilon
df <- data.frame(x = x, y = y)
grid <- data.frame(x = seq(-1, 1, length.out = 50))
model <- "y ~ x"
results <- SCB_linear_outcome(df_fit = df, model = model, grid_df = grid)

results <- tibble::as_tibble(results)
plot_cs(results, levels = c(0), x = seq(-1, 1, length.out = 50), mu_hat = results$Mean,
       xlab = "x1", ylab = "y", level_label = T, min.size = 40, palette = "Spectral",
```

```
        color_level_label = "black")
```

`prepare_pupil_fpca` *Prepare Pupil FPCA Dataset*

Description

Processes data by fitting a mean GAM model, extracting residuals, performing FPCA, and merging the results to create an enhanced dataset for functional regression analysis.

Usage

```
prepare_pupil_fpca(input_data, k_mean = 30, k_fpca = 15, example = "original")
```

Arguments

`input_data`	Raw pupil data
`k_mean`	Number of basis functions for mean model smooth terms (default: 30)
`k_fpca`	Number of knots for FPCA estimation (default: 15)
`example`	Choice for different model. If `example = "original"`, will only include use as the only covariate. If `example = "original"`, will include use, age and gender as covariates.

Value

A tibble containing:

- Original pupil variables
- FPCA eigenfunctions (Phi1, Phi2,...)
- Sorted by ID and domain

Examples

```
if (requireNamespace("mgcv", quietly = TRUE)) {
data(pupil)
processed_data <- prepare_pupil_fpca(pupil)

processed_data <- prepare_pupil_fpca(pupil, k_mean = 5, k_fpca = 5)
}
```

`pupil` *Trajectories of Pupil Response to Light after Cannabis Use*

Description

Dataset contains functional observation of pupil size percent change after a light stimulus. Participants in the cannabis use group smoked cannabis flower or concentrate 40 minutes prior to the pupillometry measurement. Goal of this data is to understand differences in pupil response to light driven by acute cannabis users. Measurements were collected on the right eye.

Usage

```
data(pupil)
```

Format

A tibble with 15000 rows and 10 variables:

id Factor. Subject identifier (127 unique levels).

use_group Character. Original usage group classification (e.g., "Daily - Flower", "No Use").

use Numeric. Binary indicator of cannabis use 40 minute prior to the light stimulus. (1 = user, 0 = non-user)

age Integer. Subject's age.

gender Numeric. Binary indicator of subject's gender: 1 = Female, 0 = Male.

bmi Numeric. Body Mass Index.

alcohol Numeric. Alcohol use score.

seconds Numeric. Time in seconds since light stimulus.

percent_change_baseline Numeric. Percent change relative to baseline.

percent_change Numeric. Percent change in the outcome of interest.

Source

Processed from `data-raw/pupil_load.R` using the `readr` and `dplyr` packages.

References

Godbole, S., Leroux, A., Brooks-Russell, A., Subramanian, P. S., Kosnett, M. J., & Wrobel, J. (2024). A Study of Pupil Response to Light as a Digital Biomarker of Recent Cannabis Use. *Digital biomarkers*, 8(1), 83–92. doi:10.1159/000538561

SCB_functional_outcome *Construct Simultaneous Confidence Bands (SCB) For One Dimensional Functional Data*

Description

This function builds simultaneous confidence bands through parametric and bootstrap approaches.

Usage

```
SCB_functional_outcome(
  data_df,
  object = NULL,
  method,
  fitted = TRUE,
  alpha = 0.05,
  outcome,
  domain,
  subset = NULL,
  id,
  nboot = NULL,
  method_SD = "t",
  weights = "rademacher"
)
```

Arguments

data_df	A functional data frame that contains both name and values for variables including functional outcome, domain (e.g. time) and ID (e.g. subject names) used to fit the model object.
object	A fitted Function-on-Scalar Regression (FoSR) object (e.g., from mgcv::gam()/mgcv::bam()). Default is NULL
method	A character string specifying the approach: • "cma" - Correlation and Multiplicity Adjusted (CMA) confidence bands via parametric approach (requires a fitted functional regression model) • "multiplier" - Dense confidence bands via Multiplier-t Bootstrap method For method = "multiplier", the outcome variable in data_df should not have all-zero entries within any specified domain (except for domain index zero, where this is allowed). Otherwise, the function will return an error. If missing values (NA) exist in the outcome variable in data_df, the function will impute them using fpca.face before performing the Multiplier Bootstrap.
fitted	Logical. Whether to estimate the simultaneous confidence bands for the fitted mean function or the fitted parameter function

- TRUE - Estimate the simultaneous confidence bands for the fitted mean outcome function.
- FALSE - estimate the simultaneous confidence bands for the fitted parameter function.

Default is TRUE.

alpha	Significance level for SCB. Default is 0.05.
outcome	A character string specifying the name of the outcome variable used in the model.
domain	A character string specifying the name of the domain variable (e.g. time) used in the model.
subset	An atomic character vector (e.g., c("user = 1", "age = 30")) specifying the target function for constructing the SCB. Each element must be of the form <name> = <value>, where <name> is the name of a scalar grouping variable and <value> is the desired value. Whitespace is ignored. Binary or categorical character variables should be transformed into numeric. Factors are not allowed here because if the input data contains factor variables, they will be automatically expanded into dummy (indicator) variables when constructing the design matrix, and the resulting variable names may differ from the original factor names. Default is NULL, representing the reference group.
id	A character string specifying the name of the ID variable.
nboot	An integer specifying the number of bootstrap samples used to construct the confidence bands. Default is 10,000 for cma, 5000 for Multiplier Bootstrap.
method_SD	Method for SD estimation: "t" or "regular". Default is "t".
weights	Multiplier type: "rademacher", "gaussian", or "mammen". Default is "rademacher".

Value

A list containing:

mu_hat Estimated mean function for the group of interest.

domain The domain used.

se_hat Standard errors of the estimated means.

scb_low Lower bound of the simultaneous confidence band.

scb_up Upper bound of the simultaneous confidence band.

type A character description of the output type.

Examples

```
# example using pupil data
if (requireNamespace("mgcv", quietly = TRUE)) {
data(pupil)

pupil_fpca <- prepare_pupil_fpca(pupil)

fosr_mod <- mgcv::bam(percent_change ~ s(seconds, k=30, bs="cr") +
```

```
  s(seconds, by = use, k=30, bs = "cr") +
  s(id, by = Phi1, bs="re") +
  s(id, by = Phi2, bs="re"),
  method = "fREML", data = pupil_fpca, discrete = TRUE)

# CMA approach
results <- SCB_functional_outcome(data_df = pupil, object = fosr_mod,
                                  method = "cma", fitted = TRUE,
                                  outcome = "percent_change", domain = "seconds",
                                  subset = c("use = 1"), id = "id")


# multiplier bootstrap
results <- SCB_functional_outcome(data_df = pupil, object = fosr_mod,
                                  method = "multiplier", fitted = TRUE,
                                  outcome = "percent_change", domain = "seconds",
                                  subset = c("use = 1"), id = "id")


mean_mod <- mgcv::gam(percent_change ~ s(seconds, k = 5, bs = "cr") +
s(seconds, by = use, k = 5, bs = "cr"),
data = pupil, method = "REML")

# multiplier bootstrap
pupil_multiplier <- SCB_functional_outcome(data = pupil, object = mean_mod, method = "multiplier",
                                  outcome = "percent_change",
                                  domain = "seconds", subset= c("use = 1"),
                                  id = "id")
}
```

SCB_gls_geospatial *Construct Simultaneous Confidence Bands for a Spatial Generalized Least Squares Model*

Description

Construct Simultaneous Confidence Bands for a Spatial Generalized Least Squares Model

Usage

```
SCB_gls_geospatial(
  sp_list,
  level = NULL,
  data_fit = NULL,
  w = NULL,
  correlation = NULL,
  corpar = NULL,
  groups = NULL,
```

```
  V = NULL,
  alpha = 0.1,
  N = 1000,
  mask = NULL
)
```

Arguments

sp_list	A list containing the spatial coordinates and the observations. Should include the following components: • x: A numeric vector of x-coordinates (e.g., longitude). • y: A numeric vector of y-coordinates (e.g., latitude). • obs: A 3D array of observations with dimensions length(x) × length(y) × n.
level	A optional numeric threshold value used to test whether the estimated mean surface significantly deviates from it. Default is NULL.
data_fit	A design matrix used to fit the generalized least squares (GLS) model. Each row corresponds to an observation, and each column to a covariate to be included in the model. Outcome/observation should not be included. The first column is typically an intercept column, which will contain only 1s, if an intercept is included in the model. Categorical variables in data_fit should be converted to dummy variables. Default is matrix(1, n, 1) (only keep the intercept term)
w	A numeric vector specifying the target function for constructing the SCB, by giving a linear combination of the regression coefficients in the GLS model. Default is matrix(1, 1, 1), will only construct the SCB for the first regression coefficient.
correlation	A character string specifying the name of the correlation structure (e.g., "corAR1", "corCompSymm") to be used in the GLS model. If NULL, no correlation structure is assumed.
corpar	A list containing parameters to be passed to the correlation structure function specified in correlation.
groups	A vector of group identifiers used to define the within-group correlation structure (e.g., repeated measures, time blocks). If not specified, defaults to rep(1, n), assuming all observations belong to a single group.
V	An optional array of known covariance matrices of shape [length(x), length(y), n, n], where each V[i,j,,] corresponds to the covariance matrix for the observations at spatial location (x[i], y[j]). If V is given, then the GLS model will be fitted based on V. Otherwise, the GLS model will be fitted based on correlation structure. If neither is provided, the model reduces to ordinary least squares (OLS) regression.
alpha	A numerical value specifying the confidence level for the Simultaneous Confidence Bands. Defalut is 0.1.
N	An integer specifying the number of bootstrap samples to construct the Simultaneous Confidence Bands. Default is 1000.

`mask` An optional logical matrix same dimensions as `c(length(sp_list$x), length(sp_list$y))`, indicating spatial locations to include in the SCB computation. Non-included locations (e.g., water areas) should be set to 0 or `NA`. Default is `array(1, dim = c(length(sp_list$x), length(sp_list$y)))`

Value

A list containing the following components:

`scb_up` A matrix of upper bounds for the simultaneous confidence bands at each spatial location corresponding to the target function specified by `w`.

`scb_low` A matrix of lower bounds for the simultaneous confidence bands at each spatial location corresponding to the target function specified by `w`.

`mu_hat` A matrix of estimated mean values at each spatial location corresponding to the target function specified by `w`.

`norm_est` A matrix of standardized test statistics `(mu_hat - level) / SE`.

`thres` The bootstrap threshold used to construct the confidence bands.

`x` The vector of x-coordinates corresponding to the columns of the spatial grid.

`y` The vector of y-coordinates corresponding to the rows of the spatial grid.

References

Sommerfeld, M., Sain, S., & Schwartzman, A. (2018). Confidence regions for spatial excursion sets from repeated random field observations, with an application to climate. *Journal of the American Statistical Association*, 113(523), 1327–1340. doi:10.1080/01621459.2017.1341838

Ren, J., Telschow, F. J. E., & Schwartzman, A. (2024). Inverse set estimation and inversion of simultaneous confidence intervals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(4), 1082–1109. doi:10.1093/jrsssc/qlae027

Examples

```
data(climate_data)
# Construct confidence sets for the increase of the mean temperature (June-August)
# in North America between the 20th and 21st centuries

temp = SCB_gls_geospatial(sp_list = climate_data$Z, level = 2, data_fit = climate_data$X,
                      w = c(1,0,0,0), correlation = climate_data$correlation,
                      mask = climate_data$mask, alpha = 0.1)


example_list <- list(x = climate_data$Z$x[50:60], y = climate_data$Z$y[40:50],
obs = climate_data$Z$obs[50:60, 40:50,])
temp = SCB_gls_geospatial(sp_list = example_list, level = 2, data_fit = climate_data$X,
                      w = c(1,0,0,0), correlation = NULL,
                      mask = NULL, alpha = 0.1, N = 50)
```

SCB_linear_outcome *Construct Simultaneous Confidence Bands for Linear Regression Outcome*

Description

This function fits a linear model and constructs simultaneous confidence bands (SCB) using a non-parametric bootstrap method for the mean outcome of regression on a fixed test set design matrix

Usage

```
SCB_linear_outcome(
  df_fit,
  model,
  grid_df = NULL,
  n_boot = 1000,
  alpha = 0.05,
  grid_df_boot = NULL
)
```

Arguments

df_fit	A data frame containing the training design matrix used to fit the linear model. Acceptable input format includes numeric and factor.
model	A character string representing the formula for the linear model (e.g., "y ~ x1 + x2").
grid_df	A data frame specifying the covariate settings that define the mean outcome for which simultaneous confidence bands (SCB) are constructed. Each row represents one covariate combination at which predictions and SCBs are evaluated. Column names should match variables in the fitted model, but grid_df may include only the subset of covariates of interest for the SCB (it is not required to cover all model variables). Default is NULL, in which case the SCB is constructed over the fitted values based on 'df_fit'.
n_boot	Number of bootstrap samples used in the non-parametric bootstrap procedure to generate the empirical distribution. Default is 1000.
alpha	Significance level for the confidence band (e.g., 0.05 for 95% confidence). Default is 0.05.
grid_df_boot	An optional data frame specifying the input grid at which predictions are evaluated during bootstrap resampling. This allows SCBs to be constructed on a denser set of covariate values if desired. If NULL, uses grid_df. If grid_df is set to NULL, grid_df_boot will also be set to NULL.

Value

A data frame with the following columns:

scb_low Lower bound of the simultaneous confidence band.

Mean Predicted mean response from the fitted model.

scb_up Upper bound of the simultaneous confidence band.

... All columns from `grid_df`, representing the prediction grid.

References

Ren, J., Telschow, F. J. E., & Schwartzman, A. (2024). Inverse set estimation and inversion of simultaneous confidence intervals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(4), 1082–1109. doi:10.1093/jrsssc/qlae027

Examples

```
set.seed(262)
x1 <- rnorm(100)
epsilon <- rnorm(100,0,sqrt(2))
y <- -1 + x1 + epsilon
df <- data.frame(x1 = x1, y = y)
grid <- data.frame(x1 = seq(-1, 1, length.out = 100))
model <- "y ~ x1"
results <- SCB_linear_outcome(df_fit = df, model = model, grid_df = grid, n_boot = 100)
```

SCB_logistic_outcome *Construct Simultaneous Confidence Bands for a Logistic Regression Outcome*

Description

This function fits a logistic regression model and constructs simultaneous confidence bands (SCB) using a non-parametric bootstrap method for the mean outcome of regression on a fixed test set design matrix

Usage

```
SCB_logistic_outcome(
  df_fit,
  model,
  grid_df = NULL,
  n_boot = 1000,
  alpha = 0.05
)
```

Arguments

df_fit	A data frame containing the training design matrix used to fit the logistic model.
model	A character string representing the formula for the logistic model (e.g., "y ~ x1 + x2").
grid_df	A data frame specifying the covariate settings that define the mean outcome for which simultaneous confidence bands (SCB) are constructed. Each row represents one covariate combination at which predictions and SCBs are evaluated. Column names should match variables in the fitted model, but grid_df may include only the subset of covariates of interest for the SCB (it is not required to cover all model variables). Default is NULL, in which case the SCB is constructed over the fitted values based on 'df_fit'.
n_boot	Number of bootstrap samples used in the non-parametric bootstrap procedure to generate the empirical distribution. Default is 1000.
alpha	Significance level for the confidence band (e.g., 0.05 for 95% confidence). Default is 0.05.

Value

A data frame with the following columns:

scb_low Lower bound of the simultaneous confidence band.

Mean Predicted mean response from the fitted model.

scb_up Upper bound of the simultaneous confidence band.

... All columns from grid_df, representing the prediction grid.

References

Ren, J., Telschow, F. J. E., & Schwartzman, A. (2024). Inverse set estimation and inversion of simultaneous confidence intervals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(4), 1082–1109. doi:10.1093/jrsssc/qlae027

Examples

```
set.seed(262)
x1 <- rnorm(100)
mu <- -1 + x1
p <- expit(mu)
y <- rbinom(100, size = 1, prob = p)
df <- data.frame(x1 = x1, y = y)
grid <- data.frame(x1 = seq(-1, 1, length.out = 100))
model <- "y ~ x1"
results <- SCB_logistic_outcome(df_fit = df, model = model, grid_df = grid, n_boot = 100)
```

`SCB_regression_coef` *Construct Simultaneous Confidence Bands for Regression Coefficients*

Description

This function fits either a linear or logistic regression model and computes simultaneous confidence bands (SCBs) for the model coefficients using a non-parametric bootstrap procedure.

Usage

```
SCB_regression_coef(
  df_fit,
  model,
  n_boot = 5000,
  alpha = 0.05,
  type = "linear"
)
```

Arguments

`df_fit`	A data frame containing the design matrix and response variable used to fit the model.
`model`	A character string specifying the regression formula (e.g., `"y ~ x1 + x2"`).
`n_boot`	Integer. Number of bootstrap samples to use for constructing the SCBs. Default is 5000.
`alpha`	Numeric. Significance level for the confidence bands (e.g., 0.05 for 95% SCBs). Default is 0.05.
`type`	A character string specifying the model type. Either `"linear"` (default) or `"logistic"`.

Value

A data frame with the following columns:

scb_low Lower bound of the simultaneous confidence band. The first row corresponds to the intercept, and subsequent rows correspond to regression coefficients.

Mean Estimated values. The first element is the intercept estimate, and the remaining are coefficient estimates.

scb_up Upper bound of the simultaneous confidence band. The first row corresponds to the intercept, and subsequent rows correspond to regression coefficients.

References

Ren, J., Telschow, F. J. E., & Schwartzman, A. (2024). Inverse set estimation and inversion of simultaneous confidence intervals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(4), 1082–1109. doi:10.1093/jrsssc/qlae027

Examples

```
library(MASS)
set.seed(262)
M <- 5
rho <- 0.4
n <- 100
beta <- rnorm(M, mean = 0, sd = 1)
Sigma <- outer(1:M, 1:M, function(i, j) rho^abs(i - j))
X <- MASS::mvrnorm(n = n, mu = rep(0, M), Sigma = Sigma)
epsilon <- rnorm(n, mean = 0, sd = 1)
y <- X %*% beta + epsilon
df <- as.data.frame(X)
names(df) <- paste0("x", 1:M)
df$y <- as.vector(y)
model <- "y ~ ."
results <- SCB_regression_coef(df, model, n_boot = 100)
```

SCB_regression_outcome

Simultaneous Confidence Bands for Regression Outcomes or Coefficients

---

Description

This function fits a user-specified regression model (linear or logistic), and constructs simultaneous confidence bands (SCBs) for either the mean outcome or the regression coefficients. SCBs are obtained using a nonparametric bootstrap procedure evaluated on a fixed test design matrix, providing simultaneous inference across the entire range of covariates or parameters of interest.

Usage

```
SCB_regression_outcome(
  df_fit,
  model,
  grid_df = NULL,
  n_boot = 1000,
  alpha = 0.05,
  grid_df_boot = NULL,
  type = "linear",
  fitted = TRUE,
  w = NULL
)
```

Arguments

df_fit	A data frame containing the training design matrix used to fit the linear model. Acceptable input format includes numeric and factor.

`model` A character string representing the formula for the linear model (e.g., `"y ~ x1 + x2"`).

`grid_df` A data frame specifying the covariate settings that define the mean outcome or linear combination for which simultaneous confidence bands (SCB) are constructed. Each row represents one covariate combination at which predictions and SCBs are evaluated. Column names should match variables in the fitted model, but `grid_df` may include only the subset of covariates of interest for the SCB (it is not required to cover all model variables). Default is `NULL`, in which case the SCB is constructed over the fitted values based on 'df_fit'.

`n_boot` Number of bootstrap samples used in the non-parametric bootstrap procedure to generate the empirical distribution. Default is 1000.

`alpha` Significance level for the confidence band (e.g., 0.05 for 95% confidence). Default is 0.05.

`grid_df_boot` An optional data frame specifying the input grid at which predictions are evaluated during bootstrap resampling. This allows SCBs to be constructed on a denser set of covariate values if desired. If NULL, uses `grid_df`. If `grid_df` is set to NULL, `grid_df_boot` will also be set to `NULL`. This argument is only for `type = linear`.

`type` A character string specifying the model type. Either `"linear"` (default) or `"logistic"`.

`fitted` Logical. Whether to estimate the simultaneous confidence bands for regression outcomes or coefficients.

- `TRUE` - Estimate the simultaneous confidence bands for regression outcomes.
- `FALSE` - estimate the simultaneous confidence bands for regression coefficients.

Default is `TRUE`.

`w` A numeric matrix that specifies the linear combinations of regression coefficients for which simultaneous confidence bands (SCBs) are to be constructed. The number of columns should be equal to the number of coefficients in the regression model fitted. Default is NULL, will return SCBs for all coefficients and the intercept. This argument is only for `fitted = FALSE`.

Value

A data frame with the following columns:

scb_low Lower bound of the simultaneous confidence band.

Mean Predicted mean response from the fitted model.

scb_up Upper bound of the simultaneous confidence band.

... All columns from `grid_df`, representing the prediction grid. Optional, if `fitted = TRUE`

References

Ren, J., Telschow, F. J. E., & Schwartzman, A. (2024). Inverse set estimation and inversion of simultaneous confidence intervals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(4), 1082–1109. doi:10.1093/jrsssc/qlae027

Examples

```
set.seed(262)
x1 <- rnorm(100)
x2 <- rnorm(100)
epsilon <- rnorm(100,0,sqrt(2))
y <- -1 + x1 + x2 + epsilon
df <- data.frame(x1 = x1, x2 = x2, y = y)
grid <- data.frame(x1 = seq(-1, 1, length.out = 100), x2 = seq(-1, 1, length.out = 100))
model <- "y ~ x1 + x2"
w <- matrix(c(0, 1, 1), ncol = 3)
results <- SCB_regression_outcome(df_fit = df, model = model, grid_df = grid,
                                  w = w, n_boot = 100)
```

---

scb_to_cs *Construct Simultaneous Confidence Region for Excursion/Interval Sets from Simultaneous Confidence Bands*

Description

This function constructs simultaneous confidence regions (SCRs) for upper and lower excursion sets, and interval sets from simultaneous confidence bands (SCBs). It allows estimation of inner and outer confidence region under single or multiple thresholds. Visualization of the confidence region is also included, along with a containment check for the coverage of true or estimated functions.

Usage

```
scb_to_cs(
  scb_up,
  scb_low,
  levels,
  true_mean = NULL,
  est_mean = NULL,
  x1 = NULL,
  x2 = NULL,
  type = "upper",
  return_contain_only = FALSE,
  return_plot = FALSE,
  xlab = NULL,
  ylab = NULL
)
```

Arguments

scb_up	A numeric vector (1D) or matrix (2D) containing the upper simultaneous confidence interval.
scb_low	A numeric vector (1D) or matrix (2D) containing the lower bounds of the simultaneous confidence bands. Dimensions of scb_up and scb_low must match.

`levels` A numeric vector or list of scalers for different levels or matrix containing interval sets to construct the confidence sets. If `type` = "upper", "lower", or "two-sided", `levels` should be a vector. "upper" represents upper excursion sets, and "lower" represents lower excursion sets. If "two-sided" option is chosen, will estimate only outer CSs for both upper and lower excursion sets. If `type` = `"interval"`, then `levels` should be a `list` with two named elements: `low` and `up`, corresponding to the bounds of the interval `[low, up]`.

`true_mean` Optional matrix of the true mean function. Should have the same dimension as `scb_up` and `scb_low`.

`est_mean` Optional matrix of the estimated mean function, used for plotting if `true_mean` is not available. Should have the same dimension as `scb_up` and `scb_low`.

`x1` A numeric vector of coordinates for the first dimension used for plotting the inner and outer confidence region. Default is NULL. Dimension of `x1` must match the first dimension of `scb_up` and `scb_low`.

`x2` A numeric vector of coordinates for the second dimension used for plotting inner and outer confidence region. Default is NULL. Dimension of `x1` must match the second dimension of `scb_up` and `scb_low`.

`type` A character string specifying the type of inverse set to construct if levels are not a matrix. Choices are `"upper"`, `"lower"`, `"two-sided"` or `"interval"`. Notice that `"two-sided"` and `"interval"` type is not available for plotting (`return_plot` = `TRUE`).

`return_contain_only`
Logical. If `TRUE`, only return a matrix/logical map indicating which point is contained within two types of CSs across all levels.

`return_plot` Logical. If `TRUE`, return a ggplot object for visualizing the inner and outer confidence region.

`xlab` A character for the name of the x axis used for plotting the inner and outer confidence region. Default is NULL.

`ylab` A character for the name of the y axis used for plotting the inner and outer confidence region. Default is NULL.

Value

A list containing the following components:

levels A vector (or list) of threshold levels used to define the confidence sets. Same as the input `levels`.

U_in (Optional) A list of logical matrices indicating whether each point is within the simultaneous inner confidence set for each level. Returned only when `return_contain_only` = `FALSE` and `type` != `"two-sided"`.

U_out (Optional) A list of logical matrices indicating whether each point is within the simultaneous outer confidence set for each level. Returned only when `return_contain_only` = `FALSE` and `type` != `"two-sided"`.

L_out (Two-sided only) A list of logical matrices indicating lower bound containment (for `type` = `"two-sided"` and `return_contain_only` = `FALSE`).

U_out (Two-sided only) A list of logical matrices indicating upper bound containment (for `type = "two-sided"` and `return_contain_only = FALSE`).

contain_individual A logical vector indicating whether the true mean is fully contained within each level's simultaneous inner and outer confidence region. Returned only if `true_mean` is provided.

contain_all A single logical value indicating whether the true mean is contained in all levels' simultaneous inner and outer confidence region. Returned only if `true_mean` is provided.

plot_cs (Optional) A list of ggplot2 objects for visualizing the SCBs and simultaneous confidence region across all levels, returned when `return_plot = TRUE`. Includes both a combined plot and individual plots per level.

References

Ren, J., Telschow, F. J. E., & Schwartzman, A. (2024). Inverse set estimation and inversion of simultaneous confidence intervals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 73(4), 1082–1109. doi:10.1093/jrsssc/qlae027

Examples

```
set.seed(262)
x1 <- rnorm(100)
x2 <- rnorm(100)
y <- -1 + x1 - 0.5 * x2 + rnorm(100,0,sqrt(2))
df <- data.frame(x1 = x1, x2 = x2, y = y)
grid <- data.frame(x1 = seq(-1, 1, length.out = 100), x2 = seq(-1, 1, length.out = 100))
model <- "y ~ x1 + x2 "
result <- SCB_linear_outcome(df_fit = df, model = model, grid_df = grid, n_boot = 100)
scb_to_cs(result$scb_up, result$scb_low, c(-1, -0.5, 0.5, 1),
x1 = grid$x1, x2 = grid$x2, est_mean = results$Mean)
```

Index