Package ‘SurrogateParadoxTest’

April 12, 2026

Type Package

Title Empirical Testing of Surrogate Paradox Assumptions

Version 2.2

Date 2026-04-11

Description Provides functions to nonparametrically assess assumptions sufficient to prevent the surrogate paradox through hypothesis tests of stochastic dominance, monotonicity of conditional mean functions, and non-negative residual treatment effect. Details are described in: Hsiao E, Tian L, and Parast L (2026). ``Avoiding the surrogate paradox: an empirical framework for assessing assumptions.'' Journal of Nonparametric Statistics <doi:10.1080/10485252.2025.2498609>. There are also functions to assess resilience to the surrogate paradox via calculation of the resilience probability, the resilience bound, and the resilience set. Details will be available in Hsiao E, Tian L, and Parast L, ``Resilience Measures for the Surrogate Paradox'' (Under Review). Lastly, there is a function to assess resilience to the surrogate paradox in the met-analytic setting, described in Hsiao E and Parast L, ``A Functional-Class Meta-Analytic Framework for Quantifying Surrogate Resilience'' (Under Review). A tutorial for this package can be found at <`https://www.laylaparast.com/surrogateparadoxtest`>.

License GPL

Imports stats, MonotonicityTest, MASS, ggplot2, Rcpp, splines, parallel, numDeriv, Matrix

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Emily Hsiao [aut], Layla Parast [aut, cre]

Maintainer Layla Parast <`parast@austin.utexas.edu`>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2026-04-12 15:50:02 UTC

Contents

`bootstrap_run_procedure_parallel` *Meta-Analytic Resilience Probability using Bootstrap*

Description

Calculates the meta-analytic resilience probability, estimating the standard error (SE) using the nonparametric bootstrap; used by main meta-analytic function, not intended to be called directly by the user.

Usage

```
bootstrap_run_procedure_parallel(data, s0.B, s1.B, degree = 3,
n.reps = 200,
use_spline = FALSE,
n.boot = 200,
n.cores = 2,
seed = NULL, knots0, knots1, boundary_knots0, boundary_knots1)
```

Arguments

`data`	Dataset in same format as required by meta_analytic_resilience function.
`s0.B`	Vector of S values in Study B in control group.
`s1.B`	Vector of S values in Study B in treatment group.
`degree`	Desired polynomial degree for mean function.
`n.reps`	Number of Delta_B samples to generate.
`use_spline`	Whether to use B-spline basis.
`n.boot`	Number of nonparametric bootstrap samples.
`n.cores`	Number of cores to use in parallelization.
`seed`	Seed for parallelization.

`knots0`	Knots for B-spline in control group.
`knots1`	Knots for B-spline in treatment group.
`boundary_knots0`	Boundary knots for spline in control group.
`boundary_knots1`	Boundary knots for spline in treatment group.

Value

`p_boot`	Estimated resilience probability.
`var_p`	Estimated variance of resilience probability.
`se_p`	Estimated SE of resilience probability.

`calculate_p_hat` *Calculate resilience probability.*

Description

Calculates resilience probability; used by main meta-analytic function, not intended to be called directly by the user.

Usage

```
calculate_p_hat(data, s0.B, s1.B, degree = 3, n.reps = 200, use_spline = FALSE,
knots0, knots1, boundary_knots0, boundary_knots1)
```

Arguments

`data`	Dataset in same format as required by meta_analytic_resilience function.
`s0.B`	Vector of S values in Study B in control group.
`s1.B`	Vector of Y values in Study B in treatment group.
`degree`	Degree of basis expansion.
`n.reps`	Number of Delta_B samples.
`use_spline`	Use B-spline basis.
`knots0`	Knots for B-spline in control group.
`knots1`	Knots for B-spline in treatment group.
`boundary_knots0`	Boundary knots for spline in control group.
`boundary_knots1`	Boundary knots for spline in treatment group.

Value

`p`	Estimated resilience probability.
`theta_hat0`	Estimated theta in control group.
`theta_hat1`	Estimated theta in treatment group.
`sigma2_hat0`	Estimated sigma^2 in control group.
`sigma2_hat1`	Estimated sigma^2 in treatment group.
`v2_hat0`	Estimated v^2 in control group.
`v2_hat1`	Estimated v^2 in treatment group.
`beta0_hat`	Estimated beta vector in control group.
`beta1_hat`	Estimated beta vector in treatment group.
`par0`	Estimated parameter vector for control group.
`par1`	Estimated parameter vector for treatment group.

`dataA` *Example dataset for meta-analytic analysis*

Description

Example dataset for meta-analytic analysis; this dataset contains the surrogate and primary outcome in the treatment and control groups, measured for multiple studies.

Usage

```
data("dataA")
```

Format

A data frame with 500 observations on the following 4 variables.

- `S` the surrogate value
- `Y` the primary outcome
- `G` the treatment indicator
- `study` the Study ID

Examples

```
data(dataA)
names(dataA)
head(dataA)
```

`dataB` *Example dataset for meta-analytic analysis, new study*

Description

Example dataset for meta-analytic analysis; this dataset contains the surrogate in the treatment (s1.B) and control group (s0.B) for the new study.

Usage

```
data("dataB")
```

Format

A list of 10 observations in each treatment group:

`s0.B` the surrogate values in the control group

`s1.B` the surrogate values in the treated group

Examples

```
data(dataB)
names(dataB)
head(dataB)
```

`fourier_interval` *Fourier Resilience Interval*

Description

Calculates the resilience probability and resilience bound, assuming that the mean function of Study B data is generated by adding a Fourier basis with random coefficients with specified parameters.

Usage

```
fourier_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, var_vec, period,
n.iter = 500, M = 100, q_quant = 0.1,      plot = FALSE, intervals
= TRUE, get_var = TRUE)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.
`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`var_vec`	Length-4 vector governing the variance of the random coefficients of the mean function for Study B.
`period`	Length-3 vector dictating the period of the Fourier basis for Study B. Each item represents a proportion of the range of the surrogate values of Study A.
`n.iter`	Number of Δ_B samples to generate; should be large.
`M`	Number of bootstrap iterations to estimate the SE of $\hat{p}_0$ and $\hat{q}_\alpha$.
`q_quant`	Desired quantile for the resilience bound. Default is 0.10.
`plot`	TRUE or FALSE; include plots of randomly generated functions in results. Default is FALSE.
`intervals`	TRUE or FALSE; return the inner 95% of Δ_B samples. Default is TRUE.
`get_var`	TRUE or FALSE; return an estimated variance of the $\hat{p}_0$ and $\hat{q}_\alpha$ parameters. Default is TRUE.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

`Delta_hats`	Vector of samples of Δ_B.
`Delta_estimate`	Mean of samples of Δ_B.
`p_hat`	Estimated value $\hat{p}_0$, the probability of the surrogate paradox.
`q_hat`	Estimated value $\hat{q}_\alpha$

, the resilience bound.

`p_se`	Estimated variance of $\hat{p}_0$; only returned if get_var = TRUE.
`q_se`	Estimated variance of $\hat{q}_\alpha$; only returned if get_var = TRUE.
`control_plot`	Plot including data points of control group of Study A and functions generated for Study B; only returned if plot = TRUE.
`treatment_plot`	Plot including data points of treatment group of Study A and functions generated for Study B; only returned if plot = TRUE.

Author(s)

Emily Hsiao

Examples

```
n.example = 200
s0.A <- rnorm(n.example, 3, sqrt(3))
y0.A <- sapply(s0.A, function(x) 2 * x - 1)
s1.A <- rnorm(n.example, 4, sqrt(3))
y1.A <- sapply(s1.A, function(x) x + 3)
s0.B <- rnorm(n.example, 4.75, sqrt(1))
s1.B <- rnorm(n.example, 5.25, sqrt(1))
result <- fourier_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
var_vec = c(0.25, 0.25, 0.1, 0.1), period = c(0.5, 0.25, 0.1), n.iter = 100, M = 20)
result$p_hat
```

---

fourier_resilience_set

Fourier Resilience Set

Description

Creates a plot of the resilience set i.e., the possible variance parameters of the coefficients of the Fourier terms of the mean function of Study B such that the probability of the surrogate paradox is below a threshold α. Note that this function assumes that the covariance matrix of the random coefficients takes the form $\Sigma = diag(\sigma_{11}^2, \sigma_{11}^2, \sigma_{22}^2, \sigma_{22}^2)$.

Usage

```
fourier_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
sig1_values, sig2_values, alpha = 0.10)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.
`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`sig1_values`	Vector of values of σ_{11}^2 for which the probability of surrogate paradox should be calculated with the given data.
`sig2_values`	Vector of values of σ_{22}^2 for which the probability of surrogate paradox should be calculated with the given data.
`alpha`	Threshold value of the surrogate paradox. Default is 0.10.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

Returns a ggplot object with values of σ^2_{11} on the x-axis and σ^2_{22} on the y-axis, and regions where $P(\Delta_B) < \alpha$ with given data and parameters highlighted in blue.

Author(s)

Emily Hsiao

Examples

```
n_A=200
n_B=200
s0.A <- rnorm(n_A, mean = 10, sd = 2)
y0.A <- 5 + 0.5 * s0.A + rnorm(n_A, mean = 0, sd = 0.5)

s1.A <- rnorm(n_A, mean = 12, sd = 2.5)
y1.A <- 5.5 + 0.6 * s1.A + rnorm(n_A, mean = 0, sd = 0.6)

 s0.B <- rnorm(n_B, 10, 2)
 s1.B <- rnorm(n_B, 11, 3)
 sig1_vals_example <- seq(0.1, 1, length.out = 30) # Sigma squared values
 sig2_vals_example <- seq(0.1, .5, length.out = 30)

 # Run the function with the generated example data

 fourier_resilience_set(
   s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
   sig1_values = sig1_vals_example,
   sig2_values = sig2_vals_example,
   alpha = 0.05
 )
```

`gaussian_process_interval` *Gaussian Process Resilience Interval*

Description

Calculates the resilience probability and resilience bound, assuming that the mean function of Study B data is generated according to a Gaussian Process with specified parameters.

Usage

```
gaussian_process_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2, theta,
  n.iter = 500, M = 100, q_quant = 0.1, plot = FALSE, intervals = TRUE, get_var = TRUE)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.
`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`sigma2`	Variance parameter of the Gaussian Process.
`theta`	Smoothness parameter of the Gaussian Process.
`n.iter`	Number of Δ_B samples to generate; should be large.
`M`	Number of bootstrap iterations to estimate the SE of $\hat{p}_0$ and $\hat{q}_\alpha$.
`q_quant`	Desired quantile for the resilience bound. Default is 0.10.
`plot`	TRUE or FALSE; include plots of randomly generated functions in results. Default is FALSE.
`intervals`	TRUE or FALSE; return the inner 95% of Δ_B samples. Default is TRUE.
`get_var`	TRUE or FALSE; return an estimated variance of the $\hat{p}_0$ and $\hat{q}_\alpha$ parameters. Default is TRUE.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

`Delta_hats`	Vector of samples of Δ_B.
`Delta_estimate`	Mean of samples of Δ_B.
`p_hat`	Estimated value $\hat{p}_0$, the probability of the surrogate paradox.
`q_hat`	Estimated value $\hat{q}_\alpha$

, the resilience bound.

`p_se`	Estimated variance of $\hat{p}_0$; only returned if get_var = TRUE.
`q_se`	Estimated variance of $\hat{q}_\alpha$; only returned if get_var = TRUE.
`control_plot`	Plot including data points of control group of Study A and functions generated for Study B; only returned if plot = TRUE.
`treatment_plot`	Plot including data points of treatment group of Study A and functions generated for Study B; only returned if plot = TRUE.
`p_interval`	Inner 95% of $\hat{p}_0$ samples from bootstrap; only returned if intervals = TRUE.
`q_interval`	Inner 95% of $\hat{q}_\alpha$ samples from bootstrap; only returned if intervals = TRUE.

Author(s)

Emily Hsiao

Examples

```
n.example = 200
s0.A <- rnorm(n.example, 3, sqrt(3))
y0.A <- sapply(s0.A, function(x) 2 * x - 1)
s1.A <- rnorm(n.example, 4, sqrt(3))
y1.A <- sapply(s1.A, function(x) x + 3)
s0.B <- rnorm(n.example, 4.75, sqrt(1))
s1.B <- rnorm(n.example, 5.25, sqrt(1))
result <- gaussian_process_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2 = 0.25,
  theta = 2, n.iter = 100, M = 20)
result$p_hat
```

`gp_resilience_set` *Gaussian Process Resilience Set*

Description

Creates a plot of the resilience set i.e., the possible parameters of the Gaussian Process such that the probability of the surrogate paradox is below a threshold α.

Usage

```
gp_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2_vals, theta_vals,
alpha = 0.10)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.
`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`sigma2_vals`	Vector of values of σ^2 for which the probability of surrogate paradox should be calculated with the given data.
`theta_vals`	Vector of values of θ for which the probability of surrogate paradox should be calculated with the given data.
`alpha`	Threshold value of the surrogate paradox. Default is 0.10.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

Returns a ggplot object with values of θ on the x-axis and σ^2 on the y-axis, and regions where $P(\Delta_B) < \alpha$ with given data and parameters highlighted in blue.

Author(s)

Emily Hsiao

Examples

```
n_A=200
n_B=200
s0.A <- rnorm(n_A, mean = 10, sd = 2)
y0.A <- 5 + 0.5 * s0.A + rnorm(n_A, mean = 0, sd = 0.5)

s1.A <- rnorm(n_A, mean = 12, sd = 2.5)
y1.A <- 5.5 + 0.6 * s1.A + rnorm(n_A, mean = 0, sd = 0.6)

 s0.B <- rnorm(n_B, 10, 2)
 s1.B <- rnorm(n_B, 11, 3)


  sigma2_values <- seq(0.01, 10, length.out = 30)
  theta_values <- seq(0.01, 10, length.out = 30)

  gp_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma2_values, theta_values,
  alpha = 0.05)
```

`make_basis` *Basis*

Description

Make basis matrix; used by main meta-analytic function, not intended to be called directly by the user.

Usage

```
make_basis(x, degree = 3, use_spline = FALSE, knots = NULL, boundary_knots = NULL)
```

Arguments

`x`	Vector of values.
`degree`	Polynomial Degree.
`use_spline`	Use B-spline or monomial.
`knots`	Knots.
`boundary_knots`	Boundary knots.

Value

Basis matrix.

`meta_analytic_resilience`

Meta-Analytic Resilience Probability

Description

This function calculates the resilience probability for the meta-analytic setting.

Usage

```
meta_analytic_resilience(data, s0.B, s1.B, degree = 3, use_spline = FALSE,
n.reps = 200, calculate_se = TRUE, try_analytic = TRUE, n_mc = 200, n_bootstrap = 200)
```

Arguments

`data`	Data from Study A. Needs to be in the specific format of a table with the columns: S, Y, G (treatment indicator), study (study index).
`s0.B`	Vector of surrogate values in Study B in control group.
`s1.B`	Vector of surrogate values in Study B in treatment group.
`degree`	The degree of polynomial desired for the basis expansion.
`use_spline`	TRUE or FALSE: whether to use a B-spline basis expansion for mean function.
`n.reps`	How many samples of Delta_K+1 are desired.
`calculate_se`	TRUE or FALSE: return SE of estimated p.
`try_analytic`	TRUE or FALSE: run PAB method or default to fully nonparametric bootstrap.
`n_mc`	Number of samples in PAB method.
`n_bootstrap`	Number of samples in nonparametric bootstrap.

Details

More details can be found in Hsiao, Parast. "A Functional-Class Meta-Analytic Framework for Quantifying Surrogate Resilience" (2026).

Value

`p_hat`	Estimated resilience probability.
`se_p`	Estimated SE of resilience probability; only if calculate_se = TRUE.
`conf_p`	95% confidence interval for resilience probability; only if calculate_se = TRUE.
`var_method`	Method used to estimate SE, bootstrap or analytic; only if calculate_se = TRUE.

Examples

```
#use example datasets
data(dataA)
names(dataA)
head(dataA)

data(dataB)
names(dataB)
head(dataB)


#apply function, no uncertainty quantification
set.seed(1)
result = meta_analytic_resilience(data=dataA, s0.B = dataB$s0.B, s1.B = dataB$s1.B,
degree = 1, use_spline = FALSE, calculate_se = FALSE)
result

#apply function, uncertainty quantification, analytic SE
set.seed(1)
result = meta_analytic_resilience(data=dataA, s0.B = dataB$s0.B, s1.B = dataB$s1.B,
degree = 1, use_spline = FALSE, calculate_se = TRUE, try_analytic = TRUE)
result

#apply function, uncertainty quantification, bootstrap SE
set.seed(1)
result = meta_analytic_resilience(data=dataA, s0.B = dataB$s0.B, s1.B = dataB$s1.B,
degree = 1, use_spline = FALSE, calculate_se = TRUE, try_analytic = FALSE)
result
```

`polynomial_interval` *Polynomial Resilience Interval*

Description

Calculates the resilience probability and resilience bound, assuming that the mean function of Study B data is generated by adding a polynomial basis with random coefficients with specified parameters.

Usage

```
polynomial_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, var_vec, n.iter = 500,
M = 100, q_quant = 0.1, plot =    FALSE, intervals = TRUE, get_var = TRUE)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.

`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`var_vec`	Length-4 vector governing the variance of the random coefficients of the mean function for Study B.
`n.iter`	Number of Δ_B samples to generate; should be large.
`M`	Number of bootstrap iterations to estimate the SE of $\hat{p}_0$ and $\hat{q}_\alpha$.
`q_quant`	Desired quantile for the resilience bound. Default is 0.10.
`plot`	TRUE or FALSE; include plots of randomly generated functions in results. Default is FALSE.
`intervals`	TRUE or FALSE; return the inner 95% of Δ_B samples. Default is TRUE.
`get_var`	TRUE or FALSE; return an estimated variance of the $\hat{p}_0$ and $\hat{q}_\alpha$ parameters. Defaults to TRUE.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

`Delta_hats`	Vector of samples of Δ_B.
`Delta_estimate`	Mean of samples of Δ_B.
`p_hat`	Estimated value $\hat{p}_0$, the probability of the surrogate paradox.
`q_hat`	Estimated value $\hat{q}_\alpha$

, the resilience bound.

`p_se`	Estimated variance of $\hat{p}_0$; only returned if get_var = TRUE.
`q_se`	Estimated variance of $\hat{q}_\alpha$; only returned if get_var = TRUE.
`control_plot`	Plot including data points of control group of Study A and functions generated for Study B; only returned if plot = TRUE.
`treatment_plot`	Plot including data points of treatment group of Study A and functions generated for Study B; only returned if plot = TRUE.

Author(s)

Emily Hsiao

Examples

```
n.example = 200
s0.A <- rnorm(n.example, 3, sqrt(3))
y0.A <- sapply(s0.A, function(x) 2 * x - 1)
s1.A <- rnorm(n.example, 4, sqrt(3))
y1.A <- sapply(s1.A, function(x) x + 3)
s0.B <- rnorm(n.example, 4.75, sqrt(1))
s1.B <- rnorm(n.example, 5.25, sqrt(1))
result <- polynomial_interval(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B,
var_vec = c(0.25, 0.25, 0.1, 0.1),n.iter = 100, M = 20)
result$p_hat
```

`polynomial_resilience_set`

Polynomial Resilience Set

Description

Creates a plot of the resilience set i.e., the possible variance parameters of the coefficients of the polynomial terms of the mean function of Study B such that the probability of the surrogate paradox is below a threshold α. Note that this function assumes that the covariance matrix of the random coefficients takes the form $\Sigma = diag(\sigma^2_{11}, \sigma^2_{11}, \sigma^2_{22}, \sigma^2_{22})$.

Usage

```
polynomial_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sig1_values,
sig2_values, alpha = 0.10)
```

Arguments

`s0.A`	Vector of surrogate values in the control group of Study A.
`y0.A`	Vector of primary outcome values in the control group of Study A.
`s1.A`	Vector of surrogate values in the treatment group of Study A.
`y1.A`	Vector of primary outcome values in the treatment group of Study A.
`s0.B`	Vector of surrogate values in the control group of Study B.
`s1.B`	Vector of surrogate values in the treatment group of Study B.
`sig1_values`	Vector of values of σ^2_{11} for which the probability of surrogate paradox should be calculated with the given data.
`sig2_values`	Vector of values of σ^2_{22} for which the probability of surrogate paradox should be calculated with the given data.
`alpha`	Threshold value of the surrogate paradox. Default is 0.10.

Details

More details can be found in Hsiao, Tian, Parast. "Resilience Measures for the Surrogate Paradox." (2025) Under Review.

Value

Returns a ggplot object with values of σ_{11}^2 on the x-axis and σ_{22}^2 on the y-axis, and regions where $P(\Delta_B) < \alpha$ with given data and parameters highlighted in blue.

Author(s)

Emily Hsiao

Examples

```
n_A=200
n_B=200
s0.A <- rnorm(n_A, mean = 10, sd = 2)
 y0.A <- 5 + 0.5 * s0.A + rnorm(n_A, mean = 0, sd = 0.5)

s1.A <- rnorm(n_A, mean = 12, sd = 2.5)
 y1.A <- 5.5 + 0.6 * s1.A + rnorm(n_A, mean = 0, sd = 0.6)

s0.B <- rnorm(n_B, 10, 2)
s1.B <- rnorm(n_B, 11, 3)


sigma1_values <- seq(0.01, 1.2, length.out = 30)
sigma2_values <- seq(0.01, .4, length.out = 30)

  polynomial_resilience_set(s0.A, y0.A, s1.A, y1.A, s0.B, s1.B, sigma1_values,
  sigma2_values, alpha = 0.05)
```

`run_procedure_pab` *Meta-Analytic Resilience Probability using PAB*

Description

Calculates the meta-analytic resilience probability, estimating SE using partially analytic bootstrap; used by main meta-analytic function, not intended to be called directly by the user.

Usage

```
run_procedure_pab(data, s0.B, s1.B, degree = 3, use_spline = FALSE, n_mc = 200,
knots0, knots1, boundary_knots0, boundary_knots1)
```

Arguments

`data`	Study A data.
`s0.B`	Vector of S values in Study B in control group.
`s1.B`	Vector of S values in Study B in treatment group.
`degree`	Desired polynomial degree for mean function.
`use_spline`	Whether to use B-spline basis.
`n_mc`	Number of Monte-Carlo iterations.
`knots0`	Knots for B-spline in control group.
`knots1`	Knots for B-spline in treatment group.
`boundary_knots0`	Boundary knots for spline in control group.
`boundary_knots1`	Boundary knots for spline in treatment group.

Value

`p_boot`	Estimated resilience probability.
`se_p`	Estimated SE of resilience probability.
`var_delta`	Estimated component of SE from Delta method.
`var_sb`	Estimated component of SE from resampling S.
`mDelta`	Mean for Delta method.
`vDelta`	Variance for Delta method.
`par0`	MLE parameters for control group.
`par1`	MLE parameters for treatment group.

`test_assumptions` *Test assumptions to prevent surrogate paradox*

Description

Tests the assumptions necessary to prevent the surrogate paradox: stochastic dominance of surrogate values in the treatment group over control group, monotonicity of the relationship between surrogate and primary endpoint in both treatment and control group, and non-negative residual treatment effect of the treatment group over the control group. For computational efficiency, Version 2.0 of this package uses the monotonicity_test function from the MonotonicityTest package.

Usage

```
test_assumptions(s0 = NULL, y0 = NULL, s1 = NULL, y1 = NULL, trim = 0.95,
alpha = 0.05, type = "all", all_results = TRUE, direction = "positive",
monotonicity_bootstrap_n = 100, nnr_bootstrap_n = 200)
```

Arguments

`s0`	Vector of surrogate values in control group.
`y0`	Vector of primary endpoint values in control group.
`s1`	Vector of surrogate values in treatment group.
`y1`	Vector of primary endpoint values in treatment group.
`trim`	Proportion of data to keep after trimming the outliers. Defaults to 95%. Trims data by sorting by surrogate value and removing (1 - trim)/2 % of the lowest and highest surrogate values with their corresponding primary endpoint values.
`alpha`	Desired alpha level of tests.
`type`	Type of test to run. Defaults to "all"; possible inputs are "sd" (stochastic dominance), "monotonicity" (monotonicity), and "nnr" (non-negative residual treatment effect).
`all_results`	TRUE or FALSE; return all outputs from hypothesis tests. Defaults to TRUE.
`direction`	Direction of the test. Defaults to "positive", which tests that the treatment group stochastically dominates the control group, that μ_0 and μ_1 are monotonically increasing, and that $\mu_0 \leq \mu_1 \forall s$. Parameter "negative" tests that the control group stochastically dominates the treatment group, that μ_0 and μ_1 are monotonically decreasing, and that $\mu_1 \leq \mu_0 \forall s$.
`monotonicity_bootstrap_n`	Number of bootstrap samples for monotonicity test.
`nnr_bootstrap_n`	Number of bootstrap samples for nnr test.

Value

`result`	Table or string of results of the tests
`sd_result`	Detailed results of stochastic dominance test; only returned if all_results is TRUE
`monotonicity0_result`	Detailed results of monotonicity test in control group; only returned if all_results is TRUE
`monotonicity1_result`	Detailed results of monotonicity test in treatment group; only returned if all_results is TRUE
`nnr_result`	Detailed results of nnr test; only returned if all_results is TRUE

Author(s)

Emily Hsiao

References

Barrett, Garry F., and Stephen G. Donald. "Consistent tests for stochastic dominance." Econometrica 71.1 (2003): 71-104.

Hall, Peter, and Nancy E. Heckman. "Testing for monotonicity of a regression mean by calibrating for linear functions." Annals of Statistics (2000): 20-39.

Examples

```
m_c <- function(s) 1 + 2 * s
m_t <- function(s) 1 + 2 * s

s_c <- rnorm(100, 3, 1)
y_c <- sapply(s_c, function(s) rnorm(1, m_c(s), 1))
s_t <- rnorm(100, 3, 1)
y_t <- sapply(s_t, function(s) rnorm(1, m_t(s), 1))

test_assumptions(
s0 = s_c, y0 = y_c, s1 = s_t, y1 = y_t, type = "sd"
)


test_assumptions(
s0 = s_c, y0 = y_c, s1 = s_t, y1 = y_t, type = "all")
```

Index