

Package ‘achieveGap’

May 7, 2026

Type Package

Title Modeling Achievement Gap Trajectories with Hierarchical Penalized Splines

Version 0.1.0

Description Implements a hierarchical penalized spline framework for estimating achievement gap trajectories in longitudinal educational data.

The achievement gap between two groups (e.g., low versus high socioeconomic status) is modeled directly as a smooth function of grade while the baseline trajectory is estimated simultaneously within a mixed-effects model. Smoothing parameters are selected using restricted maximum likelihood (REML), and simultaneous confidence bands with correct joint coverage are constructed using posterior simulation. The package also includes functions for simulation-based benchmarking, visualization of gap trajectories, and hypothesis testing for global and grade-specific differences. The modeling framework builds on penalized spline methods (Eilers and Marx, 1996, <doi:10.1214/ss/1038425655>) and generalized additive modeling approaches (Wood, 2017, <doi:10.1201/9781315370279>), with uncertainty quantification following Marra and Wood (2012, <doi:10.1111/j.1467-9469.2011.00760.x>).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports mgcv (>= 1.9-0), lme4 (>= 1.1-0), MASS (>= 7.3-0), ggplot2 (>= 3.4.0)

Suggests knitr (>= 1.36), rmarkdown (>= 2.11), testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://github.com/causalfragility-lab/achieveGap>

BugReports <https://github.com/causalfragility-lab/achieveGap/issues>

NeedsCompilation no

Author Subir Hait [aut, cre] (ORCID: <<https://orcid.org/0009-0004-9871-9677>>)

Maintainer Subir Hait <haitsubi@msu.edu>

Repository CRAN

Date/Publication 2026-03-19 14:10:02 UTC

Contents

achieveGap	2
achieve_gap	3
fit_separate	4
gap_trajectory	6
plot.achieveGap	8
print.achieveGap	9
run_simulation	9
simulate_gap	10
summarize_simulation	13
summary.achieveGap	13
test_gap	14

Index	15
--------------	-----------

achieveGap	<i>achieveGap: Modeling Achievement Gap Trajectories Using Hierarchical Penalized Splines</i>
------------	---

Description

The **achieveGap** package provides a joint hierarchical penalized spline framework for estimating achievement gap trajectories in longitudinal educational data. The gap between two groups (e.g., low vs. high socioeconomic status) is parameterized directly as a smooth function of grade, estimated simultaneously with the baseline trajectory within a mixed effects model. Smoothing parameters are selected via restricted maximum likelihood (REML), and simultaneous confidence bands with correct joint coverage are constructed via posterior simulation.

Main functions

`gap_trajectory` Fit the joint hierarchical spline model.
`plot.achieveGap` Plot the estimated gap trajectory.
`summary.achieveGap` Tabular summary of estimates.
`test_gap` Hypothesis tests for the gap trajectory.
`fit_separate` Separate-model benchmark.
`simulate_gap` Synthetic data generator.
`run_simulation` Benchmark simulation study.

Author(s)

Maintainer: Subir Hait <haitsubi@msu.edu> ([ORCID](#))

References

Eilers & Marx (1996); Marra & Wood (2012); Wood (2017); Raudenbush & Bryk (2002).

See Also

Useful links:

- <https://github.com/causalfragility-lab/achieveGap>
- Report bugs at <https://github.com/causalfragility-lab/achieveGap/issues>

 achieve_gap

Fit an achievement gap trajectory model (formula interface)

Description

Convenience wrapper around `gap_trajectory()` that provides a simple formula interface: `score ~ grade`. The group indicator and nested random effects are supplied via `group` and `random`.

Usage

```
achieve_gap(
  formula,
  group = NULL,
  random = ~1 | school/student,
  data,
  k = 6,
  bs = "cr",
  n_sim = 10000,
  conf_level = 0.95,
  grade_grid = NULL,
  verbose = TRUE
)
```

Arguments

<code>formula</code>	A two-sided formula of the form <code>score ~ grade</code> . Both sides must be single variable names (no transforms).
<code>group</code>	A single character string naming the binary group variable (0/1, FALSE/TRUE, or 2-level factor) indicating reference vs focal group.
<code>random</code>	Random intercept structure in lme4-style notation. Currently only nested intercepts are supported, with the default <code>~ 1 school/student</code> .
<code>data</code>	A <code>data.frame</code> containing all variables.

k	Basis dimension passed to <code>gap_trajectory()</code> .
bs	Basis type passed to <code>gap_trajectory()</code> .
n_sim	Number of posterior simulations used for simultaneous bands.
conf_level	Confidence level for bands (e.g., 0.95).
grade_grid	Optional numeric vector of grades/measurement occasions at which to evaluate trajectories.
verbose	Logical; if TRUE prints a compact model summary message.

Value

An object of class "achieveGap" as returned by `gap_trajectory()`.

Examples

```
sim <- simulate_gap(n_students = 200, n_schools = 20, seed = 1)
fit <- achieve_gap(
  score ~ grade,
  group = "SES_group",
  random = ~ 1 | school/student,
  data = sim$data,
  n_sim = 500,
  verbose = FALSE
)
summary(fit)
```

fit_separate

Fit Separate Spline Models per Group and Compute Post Hoc Gap

Description

Fits independent penalized spline mixed models to each group and computes the achievement gap as a post hoc difference between fitted curves. Pointwise standard errors are computed via a naive delta method assuming independence between the two fitted smooths:

$$SE\{\hat{g}(t)\} = \sqrt{SE\{\hat{f}_0(t)\}^2 + SE\{\hat{f}_1(t)\}^2}.$$

This is included for benchmarking against the proposed joint model `gap_trajectory`.

Usage

```
fit_separate(
  data,
  score,
  grade,
  group,
```

```

    school,
    student,
    k = 6,
    bs = "cr",
    conf_level = 0.95,
    grade_grid = NULL,
    verbose = TRUE
  )

```

Arguments

data	A data frame in long format.
score	Character string. Name of the outcome variable.
grade	Character string. Name of the grade/time variable.
group	Character string. Name of the binary group indicator.
school	Character string. Name of the school ID variable.
student	Character string. Name of the student ID variable.
k	Integer. Number of spline basis functions. Default is 6.
bs	Character string. Spline basis type. Default is "cr".
conf_level	Numeric. Confidence level for intervals. Default 0.95.
grade_grid	Numeric vector. Evaluation grid for the gap function. Defaults to 100 equally spaced points across the observed grade range.
verbose	Logical. Print progress. Default is TRUE.

Details

This function fits two separate models and subtracts fitted values. Because the two fits are obtained from disjoint subsets, the resulting uncertainty quantification is not directly comparable to the joint-model simultaneous bands (and can be inefficient for gap inference). It is provided as a simple baseline/benchmark.

Value

A named list with eight elements: `grade_grid` (numeric evaluation grid); `gap_hat` (estimated gap: reference minus focal); `gap_se` (delta-method pointwise standard errors); `ci_lower` and `ci_upper` (pointwise confidence bounds); `mod_ref` and `mod_focal` (fitted `mgcv::gamm` objects for each group); and `group_levels` (character vector `c(reference, focal)`).

See Also

[gap_trajectory](#)

Examples

```

sim <- simulate_gap(n_students = 300, n_schools = 25, seed = 42)
sep <- fit_separate(
  data   = sim$data,
  score  = "score",
  grade  = "grade",
  group  = "SES_group",
  school = "school",
  student = "student"
)
head(sep$gap_hat)

```

gap_trajectory	<i>Fit a Hierarchical Penalized Spline Model for Achievement Gap Trajectories</i>
----------------	---

Description

Fits a joint mixed-effects spline model in which the achievement gap between two groups is modeled directly as a smooth function of grade or time. The baseline trajectory and the group contrast trajectory are estimated simultaneously using penalized regression splines with restricted maximum likelihood (REML) smoothing parameter selection. Simultaneous confidence bands are constructed by posterior simulation from the approximate sampling distribution of the spline coefficients.

Usage

```

gap_trajectory(
  data,
  score,
  grade,
  group,
  school,
  student,
  covariates = NULL,
  k = 6,
  bs = "cr",
  n_sim = 10000,
  conf_level = 0.95,
  grade_grid = NULL,
  verbose = TRUE
)

```

Arguments

`data` A data frame in long format containing one row per observation.

score	Character string giving the outcome variable name.
grade	Character string giving the numeric grade or time variable name.
group	Character string giving the binary group indicator variable name.
school	Character string giving the school identifier variable name.
student	Character string giving the student identifier variable name.
covariates	Optional character vector of additional covariate names.
k	Integer basis dimension for each smooth term. Must be smaller than the number of unique observed grade values.
bs	Character string giving the spline basis type passed to <code>mgcv::s()</code> . Default is "cr".
n_sim	Integer number of posterior draws used to construct simultaneous confidence bands. Default is 10000.
conf_level	Numeric confidence level for pointwise and simultaneous intervals. Default is 0.95.
grade_grid	Optional numeric vector giving the grid of grade values at which the fitted gap trajectory is evaluated. If NULL, a regular grid of 100 points spanning the observed grade range is used.
verbose	Logical. If TRUE, prints progress messages. Default is TRUE.

Details

The estimated gap is defined as:

$$E[Y \mid \text{group} = \text{reference}] - E[Y \mid \text{group} = \text{focal}]$$

where the reference group is the first observed level of group and the focal group is the second observed level.

Value

An object of class "achieveGap" containing the estimated gap trajectory, pointwise and simultaneous confidence bands, fitted model objects, and supporting metadata.

Examples

```
sim <- simulate_gap(n_students = 20, n_schools = 5, seed = 1)

fit <- gap_trajectory(
  data = sim$data,
  score = "score",
  grade = "grade",
  group = "SES_group",
  school = "school",
  student = "student",
  k = 5,
  n_sim = 200,
  verbose = FALSE
```

```
)
summary(fit)
plot(fit)
```

plot.achieveGap

Plot Method for achieveGap Objects

Description

Plot the estimated achievement gap trajectory with pointwise and/or simultaneous confidence bands.

Usage

```
## S3 method for class 'achieveGap'
plot(
  x,
  band = c("both", "simultaneous", "pointwise"),
  true_gap = NULL,
  grade_labels = NULL,
  title = NULL,
  ...
)
```

Arguments

x	An object of class "achieveGap".
band	Which band(s) to display: "both" (default), "simultaneous", or "pointwise".
true_gap	Optional numeric vector of same length as x\$grade_grid (used in simulations to overlay the true gap).
grade_labels	Optional character labels for the x-axis tick marks. Three forms are accepted: (a) a <i>named</i> character vector mapping numeric grade values to labels (e.g. c("0" = "K", "1" = "G1")); (b) an <i>unnamed</i> character vector whose length equals the number of observed (discrete) grade values in the original data — labels are placed at those grade values in sorted order (e.g. 8 labels for grades 0–7); or (c) an unnamed vector whose length equals length(x\$grade_grid) — one label per evaluation grid point.
title	Optional plot title.
...	Additional arguments (ignored).

Value

A ggplot2 object.

print.achieveGap *Print Method for achieveGap Objects*

Description

Print Method for achieveGap Objects

Usage

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

Arguments

x An object of class "achieveGap".
... Additional arguments (ignored).

Value

Invisibly returns x.

run_simulation *Run a Benchmark Simulation Study*

Description

Runs a structured simulation study comparing the proposed joint spline model ([gap_trajectory](#)) against (1) a linear growth model and (2) separate splines with post hoc subtraction ([fit_separate](#)). Computes RMSE, bias, simultaneous band coverage, and pointwise coverage.

Usage

```
run_simulation(  
  n_reps = 100,  
  conditions = NULL,  
  k = 6,  
  n_sim = 3000,  
  alpha = 0.05,  
  seed = NULL,  
  verbose = TRUE  
)
```

Arguments

n_reps	Integer. Number of simulation replications. Default is 100.
conditions	A list of named lists specifying simulation conditions. If NULL (default), a standard 4-condition design is used.
k	Integer. Spline basis dimension. Default is 6.
n_sim	Integer. Posterior draws for simultaneous bands in the joint model. Default 3000.
alpha	Numeric. Significance level used only for linear-model pointwise intervals; default is 0.05 (95% CI).
seed	Integer or NULL. If provided, sets a seed for reproducible simulation across all replications/conditions.
verbose	Logical. Print progress. Default is TRUE.

Value

A data.frame with one row per replication-condition containing RMSE, bias, and coverage metrics for each method.

See Also

[simulate_gap](#), [gap_trajectory](#), [fit_separate](#)

Examples

```
results <- run_simulation(n_reps = 5, seed = 1)
summarize_simulation(results)
```

simulate_gap

Simulate Achievement Gap Data

Description

Generates synthetic longitudinal multilevel data with a known achievement gap trajectory, suitable for evaluating the performance of [gap_trajectory](#) and other methods.

Generates synthetic longitudinal multilevel data with a known achievement gap trajectory, suitable for evaluating the performance of [gap_trajectory](#) and other methods.

Usage

```
simulate_gap(
  n_students = 200,
  n_schools = 20,
  gap_shape = c("monotone", "nonmonotone"),
  grades = 0:7,
  sigma_u = 0.2,
  sigma_v = 0.3,
  sigma_e = 0.5,
  prop_low = 0.5,
  seed = NULL
)
```

```
simulate_gap(
  n_students = 200,
  n_schools = 20,
  gap_shape = c("monotone", "nonmonotone"),
  grades = 0:7,
  sigma_u = 0.2,
  sigma_v = 0.3,
  sigma_e = 0.5,
  prop_low = 0.5,
  seed = NULL
)
```

Arguments

n_students	Integer. Total number of students. Default is 200.
n_schools	Integer. Total number of schools. Default is 20.
gap_shape	Character string. Shape of the true gap function. One of "monotone" (default) or "nonmonotone".
grades	Numeric vector. Assessment grade points. Default is 0:7 (kindergarten through grade 7).
sigma_u	Numeric. School-level random effect standard deviation. Default is 0.20.
sigma_v	Numeric. Student-level random effect standard deviation. Default is 0.30.
sigma_e	Numeric. Residual standard deviation. Default is 0.50.
prop_low	Numeric. Proportion of students in the focal (low-SES) group. Default is 0.50.
seed	Integer or NULL. Random seed for reproducibility. Default is NULL.

Details

Data-generating model:

$$Y_{ijt} = f_0(t) - G_{ij}f_1(t) + u_j + v_i + \epsilon_{ijt}$$

where $f_1(t) > 0$ is the (positive) gap magnitude and the focal group has lower scores by construction.

Data-generating model:

$$Y_{ijt} = f_0(t) - G_{ij}f_1(t) + u_j + v_i + \epsilon_{ijt}$$

where $f_1(t) > 0$ is the (positive) gap magnitude and the focal group has lower scores by construction.

Value

A list with elements:

`data` A data frame in long format with columns: `student`, `grade`, `school`, `SES_group`, `score`.

`true_gap` A data frame with columns `grade` and `gap` containing the true (positive) gap function evaluated at each grade.

`f0_fun` The true baseline function.

`f1_fun` The true gap function (positive).

`params` List of simulation parameters.

A named list with five elements: `data` (a long-format data frame with columns `student`, `grade`, `school`, `SES_group`, and `score`); `true_gap` (a data frame with columns `grade` and `gap` giving the true gap at each grade); `f0_fun` (the true baseline function); `f1_fun` (the true gap function, always positive); and `params` (a list of the simulation parameters used).

See Also

[gap_trajectory](#), [run_simulation](#)

[gap_trajectory](#), [run_simulation](#)

Examples

```
sim <- simulate_gap(n_students = 200, n_schools = 20,
                  gap_shape = "monotone", seed = 123)
```

```
head(sim$data)
```

```
sim>true_gap
```

```
sim <- simulate_gap(n_students = 200, n_schools = 20,
                  gap_shape = "monotone", seed = 123)
```

```
head(sim$data)
```

```
sim>true_gap
```

summarize_simulation *Summarize Simulation Study Results*

Description

Prints formatted summary tables from a simulation study produced by [run_simulation](#) and returns them invisibly.

Usage

```
summarize_simulation(sim_results)
```

Arguments

`sim_results` A data.frame returned by [run_simulation](#).

Value

Invisibly returns a list with two data frames: `table1` (overall performance averaged across conditions) and `table2` (joint model coverage broken down by simulation condition).

See Also

[run_simulation](#)

Examples

```
results <- run_simulation(n_reps = 5, seed = 1)
summarize_simulation(results)
```

summary.achieveGap *Summary Method for achieveGap Objects*

Description

Prints a compact table of estimated gap values (with standard errors) and simultaneous confidence band bounds at selected points on the grade grid. Also reports the range of the estimated gap and the grade span where the simultaneous band excludes zero.

Usage

```
## S3 method for class 'achieveGap'
summary(object, n_points = 8, ...)
```

Arguments

object	An object of class "achieveGap".
n_points	Integer. Number of points from the grade grid to display. Default is 8.
...	Additional arguments (ignored).

Value

Invisibly returns a data.frame with the displayed summary rows.

test_gap

Hypothesis Tests for Achievement Gap Trajectories

Description

Provides (1) a global test of whether the gap trajectory is identically zero, and (2) identification of grade intervals where the gap is statistically different from zero using the simultaneous confidence band from [gap_trajectory](#).

Usage

```
test_gap(
  x,
  type = c("both", "global", "simultaneous"),
  alpha = 0.05,
  verbose = TRUE
)
```

Arguments

x	An object of class "achieveGap".
type	Character string. One of "global", "simultaneous", or "both".
alpha	Significance level. Default is 0.05. Used only to report decisions (the simultaneous band in x was built at x\$conf_level).
verbose	Logical; if TRUE prints a human-readable summary.

Value

A list with class "achieveGap_test" containing:

type Requested test type.

alpha Significance level.

global List with stat, df, p_value, reject.

simultaneous List with any_significant and a data.frame of significant intervals (if any).

Index

`achieve_gap`, [3](#)
`achieveGap`, [2](#)
`achieveGap`-package (`achieveGap`), [2](#)

`fit_separate`, [2](#), [4](#), [9](#), [10](#)

`gap_trajectory`, [2](#), [4](#), [5](#), [6](#), [9](#), [10](#), [12](#), [14](#)
`gap_trajectory()`, [3](#), [4](#)

`mgcv::s()`, [7](#)

`plot.achieveGap`, [2](#), [8](#)
`print.achieveGap`, [9](#)

`run_simulation`, [2](#), [9](#), [12](#), [13](#)

`simulate_gap`, [2](#), [10](#), [10](#)
`summarize_simulation`, [13](#)
`summary.achieveGap`, [2](#), [13](#)

`test_gap`, [2](#), [14](#)