Package: GPvam (via r-universe)

October 23, 2026

Type Package

Title Maximum Likelihood Estimation of Multiple Membership Mixed Models Used in Value-Added Modeling

Version 3.4-0

Date 2026-10-22

Description An EM algorithm, Karl et al. (2013) <doi:10.1016/j.csda.2012.10.004>, estimates generalized, reduced generalized, variable, complete, and zero persistence models by maximum likelihood or restricted maximum likelihood (the default). These are multiple-membership linear mixed models with teachers modeled as ``G-side'' effects and students modeled with either ``G-side'' or ``R-side'' effects; see Mariano et al. (2010) <doi:10.3102/1076998609346967>.

Depends R (>= 3.2.0), Matrix

Imports numDeriv, rlang, Rcpp (>= 0.11.2), graphics, grDevices, methods, stats, utils, ggplot2, patchwork, MASS

LinkingTo Rcpp, RcppArmadillo

ByteCompile yes

NeedsCompilation yes

LazyData yes

License GPL-2

Suggests testthat (>= 3.1.7)

Config/testthat/edition 3

RoxygenNote 8.0.0

Encoding UTF-8

Author Andrew Karl [cre, aut] (ORCID: <https://orcid.org/0000-0002-5933-8706>), Yan Yang [aut], Sharon Lohr [aut]

Maintainer Andrew Karl <akarl@asu.edu>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-22 23:50:02 UTC

RemoteUrl https://github.com/cran/GPvam

RemoteRef HEAD

RemoteSha 9b16f37eb1fdd8bd131e02e8c4b92a4b495fce98

Contents

GPvam-package 2
bias.test.custom 3
GPvam 6
GPvam-methods 10
GPvam.benchmark 12
plot.GPvam 13
print.GPvam 14
print.GPvam_simulation 15
simulate_bias 15
summary.GPvam 18
vam_data 19

Index **21**

`GPvam-package` *Maximum Likelihood Estimation of Multiple Membership Mixed Models*

Description

An EM algorithm fits multiple-membership linear mixed models used in educational value-added assessment. Teachers are modeled with random effects; dependence among a student's observations is modeled through either the residual covariance or a random student effect. Restricted maximum likelihood (REML) is used by default; set `REML = FALSE` in `GPvam` for maximum likelihood (ML).

Details

Use `GPvam` to fit generalized, reduced generalized, variable, complete, or zero persistence models. `summary.GPvam` summarizes a fitted model and `plot.GPvam` draws diagnostic plots. `bias.test.custom` provides an internal alignment diagnostic with a permutation reference. It omits the residual-weighting contribution to bias from covariance fitting and is not an automatic bias correction. `simulate_bias` estimates latent and residual contributions under a specified conditional generating model by refitting the response, with Monte Carlo uncertainty and explicit fit-failure reporting.

The package includes simulated example data in `vam_data` and historical runtime and memory measurements in `GPvam.benchmark`.

Author(s)

Andrew Karl <akarl@asu.edu>, Yan Yang and Sharon Lohr.

References

Karl, A. T. (2026). Motivating REML via Prediction-Error Covariances in EM Updates for Linear Mixed Models. arXiv:2602.09247v1. https://arxiv.org/abs/2602.09247.

Karl, A., Yang, Y. and Lohr, S. (2013). Efficient Maximum Likelihood Estimation of Multiple Membership Linear Mixed Models, with an Application to Educational Value-Added Assessments. *Computational Statistics & Data Analysis* **59**, 13–27. doi:10.1016/j.csda.2012.10.004.

Karl, A., Yang, Y. and Lohr, S. (2014). Computation of Maximum Likelihood Estimates for Multiresponse Generalized Linear Mixed Models with Non-nested, Correlated Random Effects. *Computational Statistics & Data Analysis* **73**, 146–162.

Karl, A., Yang, Y. and Lohr, S. (2013). A Correlated Random Effects Model for Nonignorable Missing Data in Value-Added Assessment of Teacher Effects. *Journal of Educational and Behavioral Statistics* **38**, 577–603. doi:10.3102/1076998613494819.

Karl, A. and Zimmerman, D. (2021). A diagnostic for bias in linear mixed model estimators induced by dependence between the random effects and the corresponding model matrix. *Journal of Statistical Planning and Inference* **211**, 107–118. doi:10.1016/j.jspi.2020.06.004.

Lockwood, J., McCaffrey, D., Mariano, L. and Setodji, C. (2007). Bayesian Methods for Scalable Multivariate Value-Added Assessment. *Journal of Educational and Behavioral Statistics* **32**, 125–150.

Mariano, L., McCaffrey, D. and Lockwood, J. (2010). A Model for Teacher Effects From Longitudinal Data Without Assuming Vertical Scaling. *Journal of Educational and Behavioral Statistics* **35**, 253–279. doi:10.3102/1076998609346967.

McCaffrey, D. and Lockwood, J. (2011). Missing Data in Value-Added Modeling of Teacher Effects. *Annals of Applied Statistics* **5**, 773–797.

`bias.test.custom` *Inspect Fixed-Effect Alignment with a Permutation Diagnostic*

Description

Summarize the alignment of fitted random effects with the weights of a fixed-effect estimate or contrast, following Karl and Zimmerman (2021). This internal diagnostic addresses one contribution to bias; it does not estimate the complete repeated-sampling bias or provide a bias correction.

Usage

```
bias.test.custom(
  result,
  k_vectors = NULL,
  n_perms = 1e+05,
  plot = TRUE,
  verbose = TRUE
)
```

Arguments

result	A fitted GPvam object containing X, Z, vinv, eta.hat, G, num.teach, and persistence. G-side student models also require their student count and student-side metadata.
k_vectors	A numeric contrast vector, or a list of such vectors. Each vector must have one finite coefficient per column of X. If NULL, examine each fixed effect separately using the standard basis vectors.
n_perms	Positive integer number of random permutations for each contrast. The default is 1e5; larger values improve Monte Carlo precision at a greater computational cost.
plot	Logical; display the permutation histograms and their combined plot. Histogram objects are returned even when FALSE.
verbose	Logical; print progress and the results table.

Details

For a contrast k, the observed diagnostic is $\nu'\hat{\eta}$, with

$$\nu' = k'(X'V^{-1}X)^{-1}X'V^{-1}Z.$$

Here X and Z are the fixed- and random-effects design matrices, V is the fitted marginal covariance, and $\hat{\eta}$ is the vector of predicted random effects.

Write $A = (X'V^{-1}X)^{-1}X'V^{-1}$ for the fitted GLS weights. Under $y = X\beta + Z\eta + \epsilon$, the pointwise identity is

$$k'(\hat{\beta} - \beta) = k'AZ\eta + k'A\epsilon.$$

Expectations therefore give a random-effects contribution and a residual contribution $C_k = E[k'A\epsilon]$. Covariance fitting makes A depend on the response, so mean-zero errors independent of the design do not, by themselves, imply $C_k = 0$. The returned diagnostic substitutes predicted effects for η in the first contribution and omits C_k. Shrinkage and estimation also separate this plug-in value from the expectation of the first contribution. For random X, the pointwise identity still holds for full-rank realized designs, but its expectation must use the specified joint law of the designs and effects. None of these expectations is identified from a single fitted model alone.

Teacher effects are permuted among teachers from the same originating year. For generalized and reduced generalized persistence models, each teacher's whole vector of current and future effects moves together, preserving the within-teacher covariance structure. Random student intercepts in G-side models are permuted separately among students. This block permutation specializes the diagnostic to GPvam's covariance structure; it does not implement the rank-based sampler in Karl and Zimmerman (2021).

The main directional summary is Permutation_Percentile, on a scale from 0 to 100. It is 100 times the proportion of simulated diagnostics below the observed diagnostic, plus half the proportion tied with it (a midrank convention). Thus a completely degenerate distribution has percentile 50. Numerical ties use a tolerance of 16 machine epsilons times the largest absolute observed or simulated diagnostic. Small and large percentiles indicate the lower and upper tails, respectively; this percentile is not a p-value. Permutation_Mean reports the Monte Carlo reference mean. Inspect it together with the signed diagnostic: the reference distribution need not be centered at zero. Percentiles can be 0 or 100 with finite simulation and do not establish zero tail probability.

The reported two-sided Monte Carlo p-value is $(1 + B)/(1 + \texttt{n_perms})$, where B counts permutations whose absolute diagnostic is at least as large as the observed value. Including the observed arrangement avoids zero p-values from a finite random sample of permutations (Phipson and Smyth, 2010). Because covariance parameters and random effects are estimated, these are approximate fitted-model diagnostics, not exact finite-sample tests of fixed-effect bias. Fitted predictions can have different precisions and need not be exchangeable even when the latent teacher effects are exchangeable. The absolute-value tail probability is measured about zero, not the reference mean; retain the signed diagnostic, percentile and reference mean when interpreting it.

Permutations use R's current random-number generator. Call `set.seed()` before the diagnostic to reproduce the results.

Value

A list with:

`permutation_results`
: A data frame with columns `Fixed_Effect` (effect or contrast name), `Nu_Prime_Eta` (observed diagnostic), `Permutation_P_Value` (legacy two-sided Monte Carlo p-value), `Permutation_Percentile` (midrank percentile from 0 to 100), and `Permutation_Mean` (Monte Carlo reference mean). The original three columns are retained in their original order for compatibility.

`plot_list`
: A list of `ggplot2` histogram objects.

Full-refit sensitivity simulations

`simulate_bias` implements this conditional response-refit experiment for every supported GPvam persistence/student-side structure. It requires the original input data and a current converged fitted object. A simulation that includes the residual contribution must generate a new response and re-estimate the covariance parameters in each replication. For a specified generating β, retain the generated vectors $Z\eta$ and ϵ. After refitting, calculate $k'AZ\eta$ and $k'A\epsilon$; their sum must equal the simulated estimation error. Average each component and report its Monte Carlo standard error, together with convergence failures. The internal diagnostic $k'AZ\hat{\eta}$ can be recorded separately but cannot replace the generated latent term.

Holding the fitted designs and predicted teacher effects fixed and drawing errors from a chosen residual covariance gives a conditional plug-in sensitivity analysis. It does not recover an unknown classroom-assignment mechanism or establish the unconditional bias. To examine stochastic or endogenous covariates or assignments, generate their joint relationship with the latent effects before generating responses and refitting. Increasing `n_perms` only improves the existing permutation reference; it does not carry out these simulations or estimate C_k. In custom simulations use `result$observation` to align generated observation-level vectors with the internal fitting order, and inspect `result$converged` before using a refit. With response-dependent persistence estimates, distinguish the generating random-effect design from the design estimated in the refit.

References

Karl, A. and Zimmerman, D. (2021). A diagnostic for bias in linear mixed model estimators induced by dependence between the random effects and the corresponding model matrix. *Journal of Statistical Planning and Inference* **211**, 107–118. doi:10.1016/j.jspi.2020.06.004. Corrected author preprint: https://arxiv.org/abs/2003.08087.

Phipson, B. and Smyth, G. K. (2010). Permutation P-values Should Never Be Zero: Calculating Exact P-values When Permutations Are Randomly Drawn. *Statistical Applications in Genetics and Molecular Biology* **9**(1), Article 39. doi:10.2202/15446115.1585.

See Also

GPvam

Examples

```
data(vam_data)
result <- GPvam(vam_data, persistence = "CP",
                fixed_effects = ~ as.factor(year) + cont_var + 0,
                verbose = FALSE)
set.seed(2026)
bias <- bias.test.custom(result, k_vectors = c(0, 0, 0, 1),
                         n_perms = 999, plot = FALSE, verbose = FALSE)
bias$permutation_results
```

GPvam *Fit a Generalized or Variable Persistence Model*

Description

Fit a multiple-membership linear mixed model by an EM algorithm. Teacher effects may persist into future years, and dependence among a student's observations is represented by a residual covariance matrix or a random student effect. See Karl, Yang and Lohr (2013) and Mariano et al. (2010).

Usage

```
GPvam(
  vam_data,
  fixed_effects = formula(~as.factor(year) + 0),
  student.side = "R",
  persistence = "GP",
  max.iter.EM = 1000,
  tol1 = 1e-07,
  hessian = FALSE,
  hes.method = "simple",
  REML = TRUE,
  verbose = TRUE
)
```

Arguments

vam_data	A data frame containing y (numeric student score), student (student identifier), teacher (teacher identifier), and year (numeric year or other ordered time period), together with any fixed-effect covariates. There must be at most one row per student and year. Student identifiers and years must not be missing. Scores and teacher identifiers may be NA; fixed-effect covariates must be complete on rows with observed scores. Unused columns are ignored. See 'Missing data' below.
fixed_effects	A one-sided formula specifying fixed effects. The default estimates a separate mean for each year, or an intercept if there is only one year. Use factors for categorical covariates; for example, ~ as.factor(year) + cont_var + 0 also includes a continuous covariate. Offsets and the reserved variable names mis, pat, and r are not supported in this formula.
student.side	Character string, either "R" for an unstructured within-student residual covariance or "G" for a random student intercept and year-specific residual variances. A G-side model requires repeated observed scores for at least one student.
persistence	Character string selecting "GP", "rGP", "VP", "CP", or "ZP". Only "GP" is supported with student.side = "G". See 'Persistence models' below.
max.iter.EM	Positive integer giving the maximum number of EM iterations. A fit that reaches this limit without convergence is returned with a warning and converged = FALSE.
tol1	Positive finite convergence tolerance for the relative change in log likelihood. The default is 1e-7.
hessian	Logical; calculate the observed information for covariance parameters and, for persistence = "VP", persistence parameters. This supplies their standard errors. Fixed-effect standard errors and teacher prediction-error standard deviations are computed regardless. Covariance/persistence standard errors are NA when the information matrix cannot be computed or is not positive definite.
hes.method	Numerical differentiation method for the information matrix: "simple" or "richardson".
REML	Logical; use restricted maximum likelihood (the default) rather than maximum likelihood. Supported for every available persistence/student-side combination.
verbose	Logical; print model information and iteration progress.

Details

The model is fitted by an EM algorithm based on Karl, Yang and Lohr (2013). Covariance updates use conditional random-effect moments for ML and prediction-error covariance adjustments for REML; see Karl (2026). With incomplete observation patterns, the unstructured residual covariance update uses a Cholesky parameterization to retain positive definiteness. Setting hessian = TRUE adds numerical differentiation after fitting and can substantially increase runtime. Historical timings are available in GPvam.benchmark.

Fixed-effect standard errors and teacher prediction-error standard deviations use the inverse mixed-model coefficient matrix with estimated covariance parameters treated as known. They do not include an additional adjustment for uncertainty in those covariance estimates.

Value

An object of class `GPvam`, a list that includes:

`loglik`	Log likelihood (restricted log likelihood for REML) at the returned fit. Meeting the relative-change convergence criterion does not guarantee a global maximum.
`converged`	Whether the convergence criterion was met.
`iter`, `loglik_history`	Number of evaluated iterations and the corresponding sequence of log likelihoods.
`teach.effects`	Data frame containing teacher identifiers, originating and effect years, predicted effects (`EBLUP`), and prediction-error standard deviations (`std_error`).
`parameters`	Matrix of fixed-effect, covariance, and applicable persistence estimates with their standard errors. Covariance and persistence standard errors are `NA` unless `hessian = TRUE`.
`Hessian`	Observed information matrix for covariance and persistence parameters when requested, otherwise `NA`.
`R_i`	Within-student residual covariance for R-side models; `NA` for G-side models.
`teach.cov`	List of teacher covariance blocks, one per originating year.
`stu.cov`	Student random-effect variance for a G-side model; `NA` for an R-side model.
`mresid`, `cresid`, `sresid`	Marginal residuals, conditional residuals, and conditional residuals scaled using the residual precision matrix.
`y`, `yhat`, `yhat.m`, `yhat.s`	Observed responses, conditional fitted values, marginal fitted values, and scaled conditional fitted values, in the internal fitting order.
`X`, `Z`, `G`, `R.inv`, `vinv`	Fixed- and random-effects design matrices, random-effects covariance, residual precision, and marginal precision.
`ybetas`, `eta.hat`	Unrounded fixed-effect estimates and random-effect predictions.
`num.obs`, `num.student`, `num.year`, `num.teach`	Observation, student, year, and per-year teacher counts.
`persistence`, `student.side`, `REML`	The fitted model and estimation options.
`key`	Mapping between original year values and internal year indices.
`observation`	Input-row indices in the internal fitting order.
`simulation_spec`	Original student/year/teacher row keys, observed-score pattern, collation locale, and resolved numerical controls used by `simulate_bias`. These keys are private input metadata, not an aggregate summary.
`call`, `fixed_effects`	Original model call and fixed-effects formula.

Use the standard `GPvam-methods` to extract full-precision fixed-effect estimates and covariances, fitted values, residuals, likelihood, and observation count. The fitted-value and residual methods return observations in input order with missing responses omitted.

Persistence models

The generalized persistence model (`"GP"`) assigns each teacher correlated current-year and future-year effects. The reduced model (`"rGP"`) combines a teacher's future-year effects into one shared effect. The variable persistence model (`"VP"`) makes a teacher's future-year effects multiples of the current-year effect and estimates those multipliers. Complete (`"CP"`) and zero (`"ZP"`) persistence fix the multipliers at one and zero, respectively.

With `student.side = "R"`, an unstructured covariance matrix models within-student dependence. With `student.side = "G"`, a random student intercept represents that dependence and the remaining errors are independent, with a separate variance for each year. The latter formulation assumes that scores across years use a common scale and can have substantial memory requirements for large numbers of random effects.

Missing data

A missing score is omitted from the response used for fitting. A known teacher assignment on that row is retained in the student's teaching history and can contribute persistent effects to later observed scores. A missing teacher identifier represents an unknown assignment, not a teacher named `"NA"`; no teacher effect is attached to that assignment.

Note

Each teacher-year is a separate unit. If the same teacher identifier appears in different years, the corresponding effects are modeled independently. Internally, a year suffix is added to each known teacher identifier, and the original year labels are retained in the output. Before evaluating `fixed_effects`, years are recoded to consecutive numeric indices in increasing order. Thus `~ year` specifies a linear trend across those indices. To use the original numerical spacing, copy the original years into a separate covariate and use that covariate in the formula.

The fixed-effects design must have full column rank. Use `~ as.factor(year) * cont_var + 0` to include year-specific slopes; see `formula` for formula syntax.

Author(s)

Andrew Karl <akarl@asu.edu>, Yan Yang and Sharon Lohr.

References

Karl, A. T. (2026). Motivating REML via Prediction-Error Covariances in EM Updates for Linear Mixed Models. arXiv:2602.09247v1. https://arxiv.org/abs/2602.09247.

Karl, A., Yang, Y. and Lohr, S. (2013). Efficient Maximum Likelihood Estimation of Multiple Membership Linear Mixed Models, with an Application to Educational Value-Added Assessments. *Computational Statistics & Data Analysis* **59**, 13–27. doi:10.1016/j.csda.2012.10.004.

Karl, A., Yang, Y. and Lohr, S. (2014). Computation of Maximum Likelihood Estimates for Multiresponse Generalized Linear Mixed Models with Non-nested, Correlated Random Effects. *Computational Statistics & Data Analysis* **73**, 146–162.

Karl, A., Yang, Y. and Lohr, S. (2013). A Correlated Random Effects Model for Nonignorable Missing Data in Value-Added Assessment of Teacher Effects. *Journal of Educational and Behavioral Statistics* **38**, 577–603. doi:10.3102/1076998613494819.

Karl, A. and Zimmerman, D. (2021). A diagnostic for bias in linear mixed model estimators induced by dependence between the random effects and the corresponding model matrix. *Journal of Statistical Planning and Inference* **211**, 107–118. doi:10.1016/j.jspi.2020.06.004.

Lockwood, J., McCaffrey, D., Mariano, L. and Setodji, C. (2007). Bayesian Methods for Scalable Multivariate Value-Added Assessment. *Journal of Educational and Behavioral Statistics* **32**, 125–150.

Mariano, L., McCaffrey, D. and Lockwood, J. (2010). A Model for Teacher Effects From Longitudinal Data Without Assuming Vertical Scaling. *Journal of Educational and Behavioral Statistics* **35**, 253–279. doi:10.3102/1076998609346967.

McCaffrey, D. and Lockwood, J. (2011). Missing Data in Value-Added Modeling of Teacher Effects. *Annals of Applied Statistics* **5**, 773–797.

See Also

`plot.GPvam`, `summary.GPvam`, `bias.test.custom`, `vam_data`

Examples

```
data(vam_data)

result <- GPvam(vam_data, persistence = "CP",
                fixed_effects = ~ as.factor(year) + cont_var + 0,
                verbose = FALSE)
summary(result)
plot(result, ask = FALSE)
```

`GPvam-methods` *Extract components of a fitted GPvam model*

Description

Standard model methods expose the unrounded estimates used by the fitting algorithm. The displayed `parameters` table is rounded for presentation.

Usage

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

## S3 method for class 'GPvam'
vcov(object, ...)

## S3 method for class 'GPvam'
fitted(object, type = c("conditional", "marginal"), ...)

## S3 method for class 'GPvam'
```

```
residuals(object, type = c("conditional", "marginal", "scaled"), ...)

## S3 method for class 'GPvam'
nobs(object, ...)

## S3 method for class 'GPvam'
logLik(object, ...)
```

Arguments

object	A fitted GPvam object.
...	Further arguments, currently unused.
type	For fitted, "conditional" includes the predicted random effects and "marginal" uses only the fixed effects. For residuals, these choices subtract the corresponding fitted values; "scaled" returns the conditional residuals transformed by a Cholesky factor of the inverse residual covariance.

Details

logLik returns the likelihood criterion actually fitted (ML or REML). It does not refit or convert between likelihood types. The parameter-count convention agrees with common R mixed-model packages. Compare REML likelihoods or information criteria only for models with the same fixed-effect design and the same observations; ML and REML criteria are not directly comparable.

These methods apply to objects fitted by GPvam 3.3-0 or later, which retain full-precision matrices and an input-row map for all model settings.

Value

coef returns a named fixed-effect coefficient vector; vcov returns its estimated covariance matrix. fitted and residuals return named numeric vectors in the order of the input rows with observed responses. Rows with missing responses are omitted. nobs returns the number of observed responses. logLik returns an object of class logLik, with degrees of freedom equal to the number of fixed-effect and covariance/persistence parameters.

Examples

```
data(vam_data)
fit <- GPvam(vam_data, persistence = "CP", verbose = FALSE)
coef(fit)
vcov(fit)
head(fitted(fit))
head(residuals(fit, type = "marginal"))
logLik(fit)
nobs(fit)
```

`GPvam.benchmark` *Historical Runtime and Memory Benchmarks*

Description

Runtime and peak memory measurements for different persistence models and simulated data sets. The data sets vary in their numbers of years, teachers per year, and students per teacher. These are historical measurements, not predictions for a current computer or package version.

Usage

```
data(GPvam.benchmark)
```

Format

A data frame with 160 rows and nine variables:

Persistence Persistence model and, where indicated, student side.

Years Number of years.

Students.per.Teacher Number of students per teacher.

Teachers.per.Year Number of teachers per year.

Total.Students Total student-year records, equal to `Years * Students.per.Teacher * Teachers.per.Year`; the original column name is retained.

Total.Teachers Total number of teacher-year effects.

Number.of.Random.Effects.in.Model Number of random effects.

PEAK.MEMORY..MB. Peak memory usage in megabytes.

RUNTIME..min. Runtime in minutes.

Details

All benchmarks used `hessian = TRUE`. Numerical Hessian calculation accounted for 20% to 75% of the recorded runtimes. Use `hessian = FALSE` when standard errors of covariance and persistence parameters are not required. The G-side student model can have large memory requirements as the numbers of students and teachers increase.

Examples

```
data(GPvam.benchmark)
head(GPvam.benchmark)
```

plot.GPvam *Plot Teacher Effects and Residuals from a GPvam Model*

Description

Draw caterpillar plots of teacher effects with normal-approximation intervals, normal Q-Q plots of conditional residuals, and residuals against fitted values.

Usage

```
## S3 method for class 'GPvam'
plot(x, ..., alpha = 0.1, ask = interactive())
```

Arguments

x	An object of class GPvam.
...	Additional graphical arguments passed to the plotting functions.
alpha	Significance level for the caterpillar intervals, a finite number strictly between zero and one. The default 0.1 produces 90% intervals.
ask	Logical; ask before advancing to the next plot. Defaults to interactive(). The previous graphics-device setting is restored when the method returns.

Details

Caterpillar intervals use the teacher prediction-error standard deviations returned by GPvam, treating estimated covariance parameters as known. They do not include an adjustment for estimation of the covariance parameters. Separate caterpillar plots are drawn for each available originating-year/effect-year pair using the original year labels.

Earlier versions of GPvam used caterpillar code adapted from plotCI in package **gplots**. Its original version was posted to R-help by Bill Venables on September 20, 1997; an enhanced version was posted by Ben Bolker on April 16, 2001. Gregory R. Warnes modified and extended that version, and changes suggested by Martin Maechler were integrated on July 29, 2004.

Value

The input object x, invisibly. Plots are drawn on the active graphics device.

Author(s)

Andrew Karl <akarl@asu.edu>, Yan Yang and Sharon Lohr. Caterpillar plotting contributors are listed in 'Details'.

See Also

GPvam, summary.GPvam

Examples

```
data(vam_data)
result <- GPvam(vam_data, persistence = "CP", verbose = FALSE)
plot(result, alpha = 0.05, ask = FALSE)
```

`print.GPvam` *Print a Fitted GPvam Model*

Description

Print the fitting iteration count, likelihood, estimated parameters, and teacher effects. The summary print method also displays covariance matrices, information criteria, and residual summaries.

Usage

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

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

Arguments

`x`	An object of class `GPvam` or, for the summary print method, `summary.GPvam`.
`...`	Additional arguments; currently unused.

Details

Teacher effects and covariance matrices are formatted using six significant digits for display. The fitted object retains their full numerical precision, and correlations use the unrounded covariances.

Value

The input object `x`, invisibly.

See Also

`summary.GPvam`, `GPvam`

print.GPvam_simulation *Print a GPvam response-refit simulation*

Description

Print a GPvam response-refit simulation

Usage

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

Arguments

x	An object returned by simulate_bias.
...	Additional arguments passed to print.

Value

The object, invisibly.

simulate_bias *Response-refit simulations for a fitted GPvam model*

Description

Quantify fixed-effect estimation error under a specified conditional generating model, re-estimating covariance and persistence parameters for every response.

Usage

```
simulate_bias(
  model,
  data,
  nsim = 100,
  beta = NULL,
  k_list = NULL,
  generator = NULL,
  centered_errors = is.null(generator),
  refit_control = list(),
  verbose = FALSE
)
```

Arguments

model	A converged GPvam fit created with version 3.4-0 or later.
data	The original input data frame, including missing-score rows. Student, teacher and year keys, row names, observed-score pattern, fitted responses and fixed-effect design must match the supplied fit.
nsim	Number of independent simulations, an integer at least 2.
beta	Generating fixed-effect vector, defaulting to the fitted vector. Named vectors must have exactly the fitted names and order.
k_list	Named list of contrast vectors in the fitted coefficient order. A single numeric vector is also accepted. NULL examines every coefficient.
generator	Optional function taking model and replicate, returning finite numeric vectors random_effects and residual_errors. Their orders are those of model$eta.hat and model$observation, respectively. The default holds fitted random effects fixed and draws centered Gaussian residuals with the original fitted residual covariance on observed scores.
centered_errors	Logical declaration that generating residuals have mean zero, default TRUE for the default generator and FALSE otherwise.
refit_control	Named list with optional max.iter.EM and tol1 overrides. Defaults are the original fit's resolved values. Model structure, formula, and ML/REML choice cannot be changed. Hessians are not recomputed.
verbose	Print one progress line per replicate.

Details

All R-side persistence types (GP, rGP, VP, CP, ZP) and the supported G-side GP model are handled. The original design, score-missingness pattern, and known teacher assignments on missing-score rows are fixed. Missing scores are not imputed; their known assignments can still contribute to later scores. For R-side models the original within-student residual correlations are retained. G-side residuals have the fitted year-specific variances; the default also holds the fitted student random effects fixed, along with teacher effects. These fixed predictions are shrunken; refitted covariance estimates can differ systematically from the original fit as part of this conditional target. R-side defaults redraw within-student dependence through residuals, whereas G-side defaults hold student effects fixed, so their targets differ.

Responses are generated as y_s = X beta + h_s + e_s, where h_s = Z_0 eta_s uses the ORIGINAL fitted random-effect design. Every refit uses the same formula, persistence type, student side and ML/REML setting. For VP, the persistence multipliers and hence the refitted design Z_s may change. With fitted GLS map W_s, Alignment is k'W_s h_s, Residual is k'W_s e_s, and Total is their sum. Estimation_Error records k'(beta_hat_s - beta), and the pointwise decomposition is checked numerically. Internal_Diagnostic is k'W_s Z_s eta_hat_s, using the REFITTED design and effects; it is not the generated latent contribution. Alignment_Teacher and Alignment_Student split Alignment for G-side fits. The student term is zero for R-side fits.

If centered_errors is TRUE, Residual_CV is k'(W_s - W_0)e_s, where W_0 uses the original fitted covariance. Its expectation equals that of Residual because W_0 is fixed and the declared residual mean is zero. Total_CV is Alignment + Residual_CV. These paired control variates often reduce

Monte Carlo uncertainty, but variance reduction is not guaranteed. Raw contributions are always retained. MCSE is the empirical standard deviation of the actual replicate values divided by the square root of the successful replicate count.

Simulations are sequential and use R's RNG; use set.seed(). Custom generators must produce independent replicates with finite variances for this MCSE interpretation. A generator error or invalid return aborts the entire call; completed replicates are not returned after this contract violation. Refit error and nonconvergence rows remain explicit with missing contributions. After failures, raw summaries describe successful refits only; ALL control-variate summary means and MCSEs are withheld, because selection of successful refits can invalidate their equal-expectation argument. An Ill_Conditioned flag records a covariance block with eigenvalue ratio below 1e-8 or a nonpositive eigenvalue; those converged refits are retained. This flag does not detect every variance boundary (in particular, a tiny scalar variance has ratio one). GLS and decomposition checks use a tolerance of 1e-6 times the larger of one and the absolute values being compared; discrepancies are retained in fit_status, and values beyond tolerance make the replicate a failure.

This is a conditional sensitivity experiment under chosen generating values, not an estimate of unconditional population bias or a calibrated bias test. It does not generate an endogenous classroom-assignment or stochastic-covariate mechanism. No automatic coefficient adjustment is supplied. A custom generator changes latent/error draws at a fixed design; stochastic designs require an external joint generating and refitting workflow.

The original data are required explicitly and are never recovered by evaluating the stored call. Older fitted objects must be refitted once with the current GPvam to obtain validated simulation metadata. Formula dependence on y is rejected, as are formula variables absent from data. Use the same collation locale as for the original fit; differing locales are rejected before refits. Output contains coefficient-level summaries and statuses, not input rows, generated response vectors, fitted objects, or original identifiers.

Value

A list of class GPvam_simulation with summary, replicates, fit_status, counts, beta, original_diagnostic, centered_errors, generator_type, and model_spec. Use qualified function names GPvam::simulate_bias and mixedbiastest::simulate_bias when both are attached.

See Also

GPvam, bias.test.custom

Examples

```
data(vam_data)
fit <- GPvam(vam_data, ~as.factor(year) + cont_var + 0,
             persistence = "CP", verbose = FALSE)
set.seed(20261022)
# Two draws illustrate the API; use more to obtain useful Monte Carlo precision.
sim <- simulate_bias(fit, vam_data, nsim = 2)
sim$summary
```

summary.GPvam *Summarize a Fitted GPvam Model*

Description

Add information criteria to a fitted model for display by its summary print method.

Usage

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

Arguments

object	An object of class GPvam.
...	Additional arguments; currently unused.

Details

AIC is $-2\ell + 2k$, where ℓ is the reported likelihood and k counts estimated fixed effects and covariance/persistence parameters. Restricted-likelihood criteria should only compare models with the same fixed-effects design and response data. Do not compare ML and REML criteria directly.

For ML fits, AICc is the conventional adjustment $\mathrm{AIC} + 2k(k+1)/(n-k-1)$, using the number of observed scores for n. It is a heuristic for these correlated-data models, not a model-specific finite-sample correction. It is returned as NA for REML fits or when $n \leq k+1$.

Value

An object of class summary.GPvam, containing the fitted object's components plus AIC and AICc.

Author(s)

Andrew Karl <akarl@asu.edu>, Yan Yang and Sharon Lohr.

See Also

GPvam, plot.GPvam

Examples

```
data(vam_data)
result <- GPvam(vam_data, persistence = "CP", verbose = FALSE)
summary(result)
```

vam_data *Simulated Value-Added Data*

Description

A data set simulated from a variable persistence model, illustrating a model with perfectly correlated current and future effects for each teacher.

Usage

```
data(vam_data)
```

Format

A data frame with 3,750 observations on 1,250 students over three years, with 50 teachers in each year, and five variables:

student Numeric student identifier.

teacher Numeric teacher identifier, interpreted within year.

year Numeric observation year, from 1 to 3.

y Numeric student score.

cont_var Numeric continuous covariate.

Details

Scores combine a student effect, current and persistent teacher effects, the continuous covariate, and independent errors. The persistence multiplier is 0.4. The student, teacher, and residual standard deviations are each 5; the covariate standard deviation is 4 and its coefficient is 1.

The original simulation used the following code and R's pre-3.6.0 `"Rounding"` sampler. On R 3.6.0 or later, first select `RNGkind(sample.kind = "Rounding")` to reproduce the supplied data, and restore your preferred RNG settings afterward. With that sampler, the code reproduces the stored data exactly.

```
set.seed(0)
years <- 3
teachers <- 50
students <- 25
alpha <- .4
eta.stu <- rnorm(students * teachers, 0, 5)
z1 <- rep(1:teachers, each = students)
z2 <- sample(rep(1:teachers, each = students))
z3 <- sample(rep(1:teachers, each = students))
cont_var1 <- rnorm(students * teachers, 0, 4)
cont_var2 <- rnorm(students * teachers, 0, 4)
cont_var3 <- rnorm(students * teachers, 0, 4)
gam1 <- rnorm(teachers, 0, 5)
```

```
gam2 <- rnorm(teachers, 0, 5)
gam3 <- rnorm(teachers, 0, 5)
eps1 <- rnorm(students * teachers, 0, 5)
eps2 <- rnorm(students * teachers, 0, 5)
eps3 <- rnorm(students * teachers, 0, 5)
y1 <- eta.stu + gam1[z1] + cont_var1 + eps1
y2 <- eta.stu + gam1[z1] * alpha + gam2[z2] + cont_var2 + eps2
y3 <- eta.stu + gam1[z1] * alpha + gam2[z2] * alpha + gam3[z3] +
  cont_var3 + eps3
student <- 1:(students * teachers)
teacher <- c(z1, z2, z3)
cont_var <- c(cont_var1, cont_var2, cont_var3)
year <- rep(1:3, each = students * teachers)
y <- c(y1, y2, y3)
vam_data <- as.data.frame(cbind(student, teacher, year, y, cont_var))
```

Examples

```
data(vam_data)
head(vam_data)
```

Index

∗ **datasets**
GPvam.benchmark, 12
vam_data, 19
∗ **regression**
GPvam, 6
plot.GPvam, 13
summary.GPvam, 18

bias.test.custom, *2*, 3, *10*, *17*

coef.GPvam *(*GPvam-methods*)*, 10

fitted.GPvam *(*GPvam-methods*)*, 10
formula, *9*

GPvam, *2*, 6, *6*, *13, 14*, *17, 18*
GPvam-methods, 10
GPvam-package, 2
GPvam.benchmark, *2*, *7*, 12

logLik.GPvam *(*GPvam-methods*)*, 10

nobs.GPvam *(*GPvam-methods*)*, 10

plot.GPvam, *2*, *10*, 13, *18*
print.GPvam, 14
print.GPvam_simulation, 15
print.summary.GPvam *(*print.GPvam*)*, 14

residuals.GPvam *(*GPvam-methods*)*, 10

simulate_bias, *2*, *5*, *8*, 15, *15*
summary.GPvam, *2*, *10*, *13, 14*, 18

vam_data, *2*, *10*, 19
vcov.GPvam *(*GPvam-methods*)*, 10