

Package ‘MEMWAS’

August 8, 2026

Type Package

Title Mixed-Effects Models with Autocorrelation Structures

Version 0.9.3

Date 2026-08-04

Description Fits longitudinal generalized mixed-effects models through the 'MEMWAS' interface and a registered 'C++' numerical backend. Supported serial covariance structures include first-order autoregressive (AR(1)), exponential or Ornstein-Uhlenbeck, higher-order autoregressive (AR(p)), first-order autoregressive moving-average (ARMA(1,1)), compound symmetry, Toeplitz, and unstructured covariance. Serial processes can be unified or attached independently to numeric predictor loadings. Candidate temporal structures can be ranked on a common sample by primary-cluster grouped cross-validation, the Akaike information criterion, the Bayesian information criterion, or log-likelihood. Clustered, crossed, and nested random intercepts and slopes are assembled jointly with diagonal or term-specific unstructured covariance. Available approximation methods include Laplace, saddlepoint likelihood with latent Laplace integration, adaptive Gaussian quadrature, full-covariance Gaussian variational inference, and penalized quasi-likelihood. Subject-grouped tuning requires every validation fold to succeed and supports fold-local nonlinear screening, bootstrap inference, prediction inference, and effective degrees of freedom for penalized information criteria. The mixed-effects framework follows Laird and Ware (1982) <doi:10.2307/2529876>; generalized-model approximations follow Breslow and Clayton (1993) <doi:10.1080/01621459.1993.10594284>; and serial covariance formulations follow Pinheiro and Bates (2000) <doi:10.1007/b98882>. The run-time fitting interface imports no third-party 'R' packages.

License GPL (>= 3)

Depends R (>= 4.1.0)

Imports stats, utils

Config/roxygen2/version 8.0.0

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr, rmarkdown
Config/testthat/edition 3
NeedsCompilation yes
Language en-US
Encoding UTF-8
Author Enoch Kang [aut, cre, trl] (ORCID:
 <<https://orcid.org/0000-0002-4903-942X>>)
Maintainer Enoch Kang <y.enoch.kang@gmail.com>
Repository CRAN
Date/Publication 2026-08-08 14:00:02 UTC

Contents

check_MEMWAS_assumptions	2
coef.MEMWAS_fit	5
coef.tune_MEMWAS	7
compare_approximations	8
diagnose_approximation	9
fit_MEMWAS	10
MEMWAS_capabilities	17
predict.MEMWAS_fit	18
rank_autocorrelation_structures	19
screen_MEMWAS_nonlinearity	23
serial_component	26
Index	28

check_MEMWAS_assumptions
<i>Check assumptions for a fitted MEMWAS model</i>

Description

Runs selected diagnostic screens for temporal autocorrelation, response distribution and link adequacy, residual conditional independence, random-effect normality, random-effect-predictor independence, and variance homogeneity. Diagnostics are separate from ordinary model fitting and can be requested after inspecting the fitted model.

Usage

```
check_MEMWAS_assumptions(
  object,
  autocorrelation_check = "All",
  distribution_link_check = "All",
  conditional_independence_check = "All",
  random_effects_normality_check = "All",
  random_effects_predictor_independence_check = "All",
  homogeneity_variance_check = "All",
  alpha = 0.05,
  ...
)

## S3 method for class 'MEMWAS_assumption_checks'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

object	Object of class 'MEMWAS_fit'. Must be operationally converged and retain finite residuals, fitted values, response data, primary clusters, times, fitted random-effect contributions, and the fixed-effect design. No default.
autocorrelation_check	Character vector. "All", "None", or any of "WithinClusterDifferenceRatioScreen", "PooledWithinClusterPortmanteauScreen", "WithinClusterLag1CorrelationScreen", and "WithinClusterSignRunsScreen". Default is "All".
distribution_link_check	Character vector. "All", "None", or any of "PearsonDispersionScreen", "DevianceDispersionScreen", "PearsonResidualJarqueBeraScreen", "SquaredFittedPartialCorrelationScreen", and "BinomialGroupedCalibrationScreen". Default is "All".
conditional_independence_check	Character vector. "All", "None", or any of "WithinClusterLag1ResidualScreen", "ClusterMeanResidualHeterogeneityScreen", and "WithinClusterSignRunsScreen". Default is "All".
random_effects_normality_check	Character vector. "All", "None", or any of "PooledRandomModeNormalScoreScreen", "PooledRandomModeJarqueBeraScreen", and "PooledRandomModeSkewKurtosisScreen". Default is "All".
random_effects_predictor_independence_check	Character vector. "All", "None", or any of "MaxGroupMeanRandomContributionCorrelationScreen", "MaxPredictorRandomContributionPearsonScreen", and "MaxPredictorRandomContributionSpearmanScreen". Default is "All".
homogeneity_variance_check	Character vector. "All", "None", or any of "SquaredResidualFittedLinearScreen", "SquaredResidualFittedQuadraticScreen", "MedianSplitAbsoluteResidualScreen", and "GroupAbsoluteResidualHeterogeneityScreen". Default is "All".
alpha	Numeric scalar. Significance threshold in '(0, 0.5]' applied to Holm-adjusted p-values. Default is '0.05'.

...	Additional arguments. No additional arguments are accepted.
x	Object of class 'MEMWAS_assumption_checks'. No default.
digits	Integer scalar. Number of significant digits used for numeric output. Default is 'max(3L, getOption("digits") - 3L)'.

Details

Holm adjustment is applied jointly to every requested diagnostic with an available finite p-value. Structurally unavailable diagnostics remain in the output with 'available = FALSE'; an unavailable row is not evidence that its assumption holds. Method names describe the calculations actually performed and should not be interpreted as exact implementations of similarly named textbook tests unless the result detail explicitly says so.

Value

Object of class 'MEMWAS_assumption_checks'. It contains the normalized selector specification, requested-method table, full results, flagged rows, unavailable rows with explanations, temporal-autocorrelation subset, alpha, multiplicity method, settings, and call.

The input object, invisibly.

Examples

```
toy <- data.frame(
  id = rep(seq_len(6), each = 3),
  time = rep(0:2, times = 6),
  x = rep(c(-1, 0, 1), times = 6)
)
toy$y <- 1 + 0.6 * toy$x +
  rep(c(-0.15, -0.08, 0, 0.05, 0.10, 0.18), each = 3) +
  rep(c(-0.03, 0.04, -0.01), times = 6)
toy_fit <- fit_MEMWAS(
  y ~ x, data = toy, id = "id", time = "time",
  random = NULL, autocor = "NONE", se_method = "none",
  control = list(n_starts = 1L, cold_start_verification = FALSE),
  verbose = FALSE
)
check_MEMWAS_assumptions(
  toy_fit,
  autocorrelation_check = "WithinClusterLag1CorrelationScreen",
  distribution_link_check = "None",
  conditional_independence_check = "None",
  random_effects_normality_check = "None",
  random_effects_predictor_independence_check = "None",
  homogeneity_variance_check = "None"
)

dat <- data.frame(
  id = rep(seq_len(15), each = 4),
  time = rep(0:3, times = 15),
  x = stats::rnorm(60)
```

```

)
dat$y <- 0.5 + 0.7 * dat$x + stats::rnorm(60)
fit <- fit_MEMWAS(
  y ~ x, data = dat, id = "id", time = "time", random = ~ 1,
  autocor = "NONE", verbose = FALSE
)
checks <- check_MEMWAS_assumptions(fit)
print(checks)

```

coef.MEMWAS_fit

*Methods for fitted MEMWAS models***Description**

Extracts coefficient versions, covariance matrices, likelihood/objective values, information criteria, fitted values, residuals, observation counts, summaries, and printed representations from fitted MEMWAS models.

Usage

```

## S3 method for class 'MEMWAS_fit'
coef(object, type = c("adjusted", "original", "both"), ...)

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

## S3 method for class 'MEMWAS_fit'
vcov(object, full = FALSE, ...)

## S3 method for class 'MEMWAS_family_fit'
vcov(object, full = FALSE, ...)

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

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

## S3 method for class 'MEMWAS_fit'
AIC(object, ..., k = 2)

## S3 method for class 'MEMWAS_family_fit'
AIC(object, ..., k = 2)

## S3 method for class 'MEMWAS_fit'
fitted(object, ...)

```

```
## S3 method for class 'MEMWAS_family_fit'
fitted(object, ...)

## S3 method for class 'MEMWAS_fit'
residuals(object, type = c("response", "pearson"), ...)

## S3 method for class 'MEMWAS_family_fit'
residuals(object, type = c("response", "pearson"), ...)

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

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

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

## S3 method for class 'MEMWAS_fit'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

## S3 method for class 'summary.MEMWAS_fit'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

Arguments

object	Object of class ‘MEMWAS_fit’ or ‘MEMWAS_family_fit’. The fitted model to be queried. No default.
type	Character scalar. For ‘coef()’, options are “adjusted” for the final regularized coefficients, “original” for coefficients from the same selected model refitted without fixed-effect penalties, and “both” for the complete comparison table. For ‘residuals()’, options are “response” and “pearson”. The method-specific default is the first listed option.
...	Additional arguments. Methods reject unused arguments, except that the family-class wrappers pass arguments to the corresponding fit method.
full	Logical scalar. Whether ‘vcov()’ returns the full parameter covariance matrix instead of the fixed-effect block. Default is ‘FALSE’.
k	Numeric scalar. Positive multiplier used by ‘AIC()’; ‘2’ gives the ordinary Akaike criterion. Default is ‘2’.
x	Object of class ‘MEMWAS_fit’, ‘MEMWAS_family_fit’, ‘summary.MEMWAS_fit’, or ‘summary.MEMWAS_family_fit’. The object to print. No default.

digits Integer scalar. Number of significant digits used by print methods. Default is `'max(3L, getOption("digits") - 3L)'`.

Details

`'coef(type = "adjusted")'` returns the coefficients used for fitted values and predictions. `'coef(type = "original")'` returns the unpenalized reference coefficients for the same selected formula and dependence specification. `'coef(type = "both")'` reports both vectors, their signed adjustment, and a regularization indicator. For approximation-specific objectives that are not marginal likelihoods, `'logLik()'` returns a labelled objective value and `'AIC()'` returns `'NA'` with a warning rather than treating unlike objectives as likelihoods.

Value

The requested numeric vector, matrix, scalar, summary object, or the input object invisibly for print methods. `'coef(type = "both")'` returns a data frame containing original coefficients, adjusted coefficients, and regularization adjustments.

<code>coef.tune_MEMWAS</code>	<i>Methods for tuned MEMWAS objects</i>
-------------------------------	---

Description

Extracts selected coefficients or penalty parameters, creates a structured tuning summary, and prints tuning or summary objects.

Usage

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

## S3 method for class 'tune_MEMWAS'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

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

## S3 method for class 'summary.tune_MEMWAS'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

object	Object of class <code>'tune_MEMWAS'</code> . No default.
...	Additional arguments. <code>'coef()'</code> forwards them to the final fitted model when one exists; other methods reject unused arguments.
x	Object of class <code>'tune_MEMWAS'</code> or <code>'summary.tune_MEMWAS'</code> . No default.
digits	Integer scalar. Number of significant digits used for numeric output. Default is <code>'max(3L, getOption("digits") - 3L)'</code> .

Details

When a final refit exists, `'coef()'` supports the coefficient `'type'` choices documented for `'coef.MEMWAS_fit()'`. Without a final refit it returns the selected penalty parameters. The summary includes grouped-fold performance, stability, convergence, selected penalties, and the propagated Original Model and Final equations when available.

Value

`'coef()'` returns a coefficient vector, coefficient-version table, or selected penalty vector. `'summary()'` returns an object of class `'summary.tune_MEMWAS'`. Print methods return their input invisibly.

compare_approximations

Compare MEMWAS approximation requests

Description

Fits the same model under each requested approximation and reports the dispatched method, computational kernel, objective semantics, information criteria, and convergence state.

Usage

```
compare_approximations(approximations = c("laplace"), ...)

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

Arguments

approximations	Character vector. One or more of <code>"auto"</code> , <code>"laplace"</code> , <code>"saddlepoint"</code> , <code>"adaptive_gaussian_quadrature"</code> , <code>"variational_inference"</code> , and <code>"pql"</code> . Default is <code>'c("laplace")'</code> .
...	Named arguments passed to <code>'fit_MEMWAS()'</code> by <code>'compare_approximations()'</code> . At minimum these normally include <code>'formula'</code> , <code>'data'</code> , <code>'id'</code> , and <code>'time'</code> ; the wrapper introduces no separate defaults. For the print method, additional arguments are passed to <code>'print.data.frame()'</code> .
x	Object of class <code>'MEMWAS_approximation_comparison'</code> . Comparison object to print. No default.

Value

Object of class `'MEMWAS_approximation_comparison'` containing a summary table and the fitted model for every requested approximation.

Examples

```

toy <- data.frame(
  id = rep(seq_len(6), each = 3),
  time = rep(0:2, times = 6),
  x = rep(c(-1, 0, 1), times = 6)
)
toy$y <- 1 + 0.6 * toy$x +
  rep(c(-0.15, -0.08, 0, 0.05, 0.10, 0.18), each = 3) +
  rep(c(-0.03, 0.04, -0.01), times = 6)
one_method <- compare_approximations(
  "laplace", formula = y ~ x, data = toy, id = "id", time = "time",
  random = NULL, autocor = "NONE", se_method = "none",
  control = list(n_starts = 1L, cold_start_verification = FALSE)
)
one_method$table

dat <- data.frame(
  id = rep(seq_len(10), each = 4),
  time = rep(0:3, times = 10),
  x = rep(c(-1, -0.25, 0.25, 1), times = 10)
)
dat$y <- 1 + 0.5 * dat$x
comparison <- compare_approximations(
  c("laplace", "variational_inference"),
  formula = y ~ x, data = dat, id = "id", time = "time"
)
comparison$table

```

diagnose_approximation

Diagnose a MEMWAS approximation

Description

Summarizes objective semantics, optimizer stationarity, boundary conditions, design rank, covariance-Jacobian rank, and approximation-specific convergence.

Usage

```
diagnose_approximation(object)
```

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

Arguments

object	Object of class 'MEMWAS_fit'. Fitted model to diagnose. No default.
x	Object of class 'MEMWAS_approximation_diagnostics'. Diagnostic object to print. No default.
...	Additional arguments. No additional arguments are accepted by the diagnostic print method.

Value

Object of class 'MEMWAS_approximation_diagnostics' containing objective, optimizer, gradient, step, KKT, boundary, stationarity, rank, and approximation metadata.

Examples

```

toy <- data.frame(
  id = rep(seq_len(6), each = 3),
  time = rep(0:2, times = 6),
  x = rep(c(-1, 0, 1), times = 6)
)
toy$y <- 1 + 0.6 * toy$x +
  rep(c(-0.15, -0.08, 0, 0.05, 0.10, 0.18), each = 3) +
  rep(c(-0.03, 0.04, -0.01), times = 6)
toy_fit <- fit_MEMWAS(
  y ~ x, data = toy, id = "id", time = "time",
  random = NULL, autocor = "NONE", se_method = "none",
  control = list(n_starts = 1L, cold_start_verification = FALSE),
  verbose = FALSE
)
diagnose_approximation(toy_fit)

dat <- data.frame(
  id = rep(seq_len(10), each = 4),
  time = rep(0:3, times = 10),
  x = rep(c(-1, -0.25, 0.25, 1), times = 10)
)
dat$y <- 1 + 0.5 * dat$x
fit <- fit_MEMWAS(
  y ~ x, data = dat, id = "id", time = "time", verbose = FALSE
)
diagnose_approximation(fit)

```

Description

Fits longitudinal generalized mixed-effects models with clustered, crossed, or nested random effects and unified or predictor-specific serial processes. Nonlinearity screening and assumption diagnostics are separate optional stages and are disabled by default so that ordinary fitting avoids their additional computation.

Usage

```
fit_MEMWAS(  
  formula,  
  family = "gaussian",  
  data,  
  id,  
  time,  
  random = ~1,  
  autocor = "AR(1)",  
  serial = NULL,  
  residual_autocor = NULL,  
  predictor_autocor = NULL,  
  L1_penalty = 0,  
  L2_penalty = 0,  
  control = list(),  
  method = "ML",  
  random_cov = "unstructured",  
  approximation = "laplace",  
  init_approximation = "variational_inference",  
  quadrature_points = 7L,  
  se_method = "hessian",  
  link = NULL,  
  offset = NULL,  
  weights = NULL,  
  subset = NULL,  
  na.action = NULL,  
  theta = NULL,  
  shape = NULL,  
  start = NULL,  
  dot_predictors = NULL,  
  dot_alternative = NULL,  
  dot_threshold = 0,  
  dot_alpha = 0.05,  
  nonlinear_alpha = 0.05,  
  spline_knots = 4L,  
  spline_probs = NULL,  
  nonlinear_predictors = NULL,  
  screen_nonlinear = FALSE,  
  nonlinear_multiplicity = "holm",  
  nonlinear_screening_method = "nuisance_adjusted_score",  
  nonlinear_bootstrap_reps = 100L,
```

```

nonlinear_bootstrap_seed = NULL,
check_assumptions = FALSE,
run_checks_and_screening = NULL,
bootstrap_inference = FALSE,
bootstrap_reps = 100L,
bootstrap_conf_level = 0.95,
bootstrap_seed = NULL,
bootstrap_min_success_fraction = 0.8,
post_selection_inference = "none",
prediction_inference = FALSE,
prediction_level = 0.95,
prediction_interval = "confidence",
autocorrelation_check = "All",
distribution_link_check = "All",
conditional_independence_check = "All",
random_effects_normality_check = "All",
random_effects_predictor_independence_check = "All",
homogeneity_variance_check = "All",
verbose = TRUE,
...
)

```

Arguments

formula	Formula object. A two-sided model formula defining the response and fixed effects. No default.
family	Character scalar or family object. Supported families are "gaussian", "binomial", "poisson", "negative_binomial", "gamma", and "exponential". Default is "gaussian".
data	Data frame. Contains every variable used by the model, grouping, time, offset, weight, and subset expressions. No default.
id	Character scalar. Name of the primary independent-cluster column in 'data'. No default.
time	Character scalar. Name of the measurement-time column in 'data'. Values must be orderable within primary clusters. No default.
random	One-sided formula, character specification, structured term, list of terms, or 'NULL'. Defines random intercepts and slopes; crossed and nested terms are allowed. Default is '~ 1', interpreted at the primary ID.
autocor	Character scalar, named character vector, or 'NULL'. A scalar defines one outcome-loaded serial process; a named vector defines predictor-specific processes. Options include "NONE", "AR(1)", "OU", "expOU", "AR(p)", "ARMA(1,1)", "CS", "TOEP", and "UN". Default is "AR(1)"; use 'NULL' when supplying 'serial'.
serial	Object of class 'MEMWAS_serial_component', structured serial specification, list of components, or 'NULL'. Provides explicit unified or predictor-specific serial components. Default is 'NULL'.

residual_autocor	Character scalar, object of class 'MEMWAS_serial_component', structured component, or 'NULL'. Explicit outcome-loaded residual autocorrelation structure. When this argument or 'predictor_autocor' is supplied, do not also supply a non-NULL 'autocor' or 'serial'. 'NULL' adds no outcome-loaded serial component. Default is 'NULL'.
predictor_autocor	Named character vector, named list of structure specifications, or 'NULL'. Each name identifies a numeric predictor whose values load an independently parameterized residual serial covariance component. Entries may use the structures accepted by 'serial_component()'; "NONE" omits that predictor component. These components are conditional outcome-covariance terms, not joint stochastic models for the predictors. Default is 'NULL'.
L1_penalty	Numeric scalar. Non-negative L1 penalty applied to non-intercept fixed effects. Default is '0'.
L2_penalty	Numeric scalar. Non-negative L2 penalty applied to non-intercept fixed effects. Default is '0'.
control	Named list. Computational and model controls. Common entries include 'maxit', 'evalmax', 'reitol', 'mode_maxit', 'mode_tol', 'optimizer', 'n_starts', 'compute_hessian', 'hessian_strategy', 'quadrature_max_dimension', 'quadrature_max_nodes', 'vi_maxit', 'pql_maxit', 'autocor_regularization', 'serial_time_scale', and 'allow_negative_autocor'. Default is 'list()'.
method	Character scalar. Estimation option "ML" or, for an unpenalized Gaussian identity-link model, "REML". Default is "ML".
random_cov	Character scalar or term-wise character specification. Options are "diagonal" and "unstructured". Default is "unstructured".
approximation	Character scalar. Final approximation option: "auto", "laplace", "saddle-point", "adaptive_gaussian_quadrature", "variational_inference", or "pql". Default is "laplace".
init_approximation	Character scalar. Initialization option accepted by the approximation dispatcher. Default is "variational_inference".
quadrature_points	Integer scalar. Positive Gauss-Hermite order used by adaptive quadrature and predictive-density integration. Default is '7L'.
se_method	Character scalar. Options are "hessian" and "none". Default is "hessian".
link	Character scalar or 'NULL'. Optional supported link override for 'family'. 'NULL' uses the family default. Default is 'NULL'.
offset	Numeric vector, model-frame expression, or 'NULL'. Additive fixed offset on the linear-predictor scale. Default is 'NULL'.
weights	Numeric vector, model-frame expression, or 'NULL'. Non-negative observation weights; for binomial proportions, positive integer values are interpreted as trial totals. Default is 'NULL'.
subset	Logical/integer vector, model-frame expression, or 'NULL'. Selects analysis rows before missing-data handling. Default is 'NULL'.

na.action	Function, character name, or 'NULL'. Missing-data action used by model-frame preparation. 'NULL' uses the package default. Default is 'NULL'.
theta	Numeric scalar or 'NULL'. Positive negative-binomial dispersion parameter; 'NULL' estimates it when applicable. Default is 'NULL'.
shape	Numeric scalar or 'NULL'. Positive gamma shape parameter; 'NULL' estimates it when applicable. Default is 'NULL'.
start	Named numeric vector, complete numeric vector, or 'NULL'. Optional parameter starting values. Default is 'NULL'.
dot_predictors	Character vector, named specification, data frame, or 'NULL'. Selects fixed-effect terms for directional tests. 'NULL' requests no directional tests. Default is 'NULL'.
dot_alternative	Character scalar/vector, named specification, data frame, or 'NULL'. Options are "greater", "less", and "two.sided". Default is 'NULL', which uses two-sided testing for selected terms.
dot_threshold	Numeric scalar/vector or named specification. Defines the non-negative minimally important coefficient magnitude for directional tests. Default is '0'.
dot_alpha	Numeric scalar/vector or named specification. Significance level in '(0, 1)' for directional tests. Default is '0.05'.
nonlinear_alpha	Numeric scalar. Selection threshold in '(0, 1)' for nonlinear screening. Default is '0.05'.
spline_knots	Integer scalar. Number of restricted-cubic-spline knots; at least three are required. Default is '4L'.
spline_probs	Numeric vector or 'NULL'. Strictly increasing knot probabilities in '[0, 1]'; the length must equal 'spline_knots'. 'NULL' uses the package probabilities. Default is 'NULL'.
nonlinear_predictors	Character vector or 'NULL'. Numeric fixed-effect predictors eligible for screening. 'NULL' uses all eligible predictors. Default is 'NULL'.
screen_nonlinear	Logical scalar. Whether to call 'screen_MEMWAS_nonlinearity()' before the final fit. Default is 'FALSE'.
nonlinear_multiplicity	Character scalar. P-value adjustment used by likelihood-ratio screening. Options are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", and "none". Default is "holm".
nonlinear_screening_method	Character scalar. Options are "nuisance_adjusted_score" and "likelihood_ratio". Default is "nuisance_adjusted_score".
nonlinear_bootstrap_reps	Integer scalar. Number of primary-cluster multiplier-bootstrap replicates for score screening; at least 99 are required. Default is '100L'.
nonlinear_bootstrap_seed	Integer scalar or 'NULL'. Non-negative seed for a package-local native score-screening stream; it does not alter the global random-number state. 'NULL' uses and advances the current 'R' stream. Default is 'NULL'.

`check_assumptions`
 Logical scalar. Whether to call `'check_MEMWAS_assumptions()'` after fitting. Default is `'FALSE'`.

`run_checks_and_screening`
 Logical scalar or `'NULL'`. `'FALSE'` disables both optional stages; `'TRUE'` activates either stage whose individual flag was omitted; `'NULL'` respects `'screen_nonlinear'` and `'check_assumptions'`. Default is `'NULL'`.

`bootstrap_inference`
 Logical scalar. Whether to run a primary-ID cluster bootstrap after fitting. Default is `'FALSE'`.

`bootstrap_reps` Integer scalar. Number of cluster-bootstrap replicates; at least two are required when bootstrap inference is active. Default is `'100L'`.

`bootstrap_conf_level`
 Numeric scalar. Bootstrap confidence level in `'(0, 1)'`. Default is `'0.95'`.

`bootstrap_seed` Integer scalar or `'NULL'`. Non-negative seed for a package-local native bootstrap stream; it does not alter the global random-number state. `'NULL'` uses and advances the current 'R' stream. Default is `'NULL'`.

`bootstrap_min_success_fraction`
 Numeric scalar. Minimum acceptable fraction of successful bootstrap refits in `'(0, 1]'`. Default is `'0.80'`.

`post_selection_inference`
 Character scalar. Options are `"none"` and `"bootstrap_after_selection"`. Default is `"none"`.

`prediction_inference`
 Logical scalar. Whether to calculate fitted-row uncertainty after fitting. Default is `'FALSE'`.

`prediction_level`
 Numeric scalar. Prediction confidence level in `'(0, 1)'`. Default is `'0.95'`.

`prediction_interval`
 Character scalar. Options are `"confidence"` and `"prediction"`. Default is `"confidence"`.

`autocorrelation_check`
 Character vector. `"All"`, `"None"`, or method names accepted by `'check_MEMWAS_assumptions()'`. Default is `"All"`; active only when `'check_assumptions = TRUE'`.

`distribution_link_check`
 Character vector. `"All"`, `"None"`, or distribution/link method names accepted by `'check_MEMWAS_assumptions()'`. Default is `"All"`; active only when `'check_assumptions = TRUE'`.

`conditional_independence_check`
 Character vector. `"All"`, `"None"`, or conditional-independence method names accepted by `'check_MEMWAS_assumptions()'`. Default is `"All"`; active only when `'check_assumptions = TRUE'`.

`random_effects_normality_check`
 Character vector. `"All"`, `"None"`, or random-effect normality method names accepted by `'check_MEMWAS_assumptions()'`. Default is `"All"`; active only when `'check_assumptions = TRUE'`.

`random_effects_predictor_independence_check`
 Character vector. "All", "None", or random-effect-predictor method names accepted by 'check_MEMWAS_assumptions()'. Default is "All"; active only when 'check_assumptions = TRUE'.

`homogeneity_variance_check`
 Character vector. "All", "None", or variance-homogeneity method names accepted by 'check_MEMWAS_assumptions()'. Default is "All"; active only when 'check_assumptions = TRUE'.

`verbose`
 Logical scalar. Whether to report progress. Default is 'TRUE'.

`...`
 Additional arguments. No additional arguments are accepted.

Details

The fitted fixed-effect vector is the adjusted coefficient vector used by the final penalized objective. When a fixed-effect penalty is positive, the same selected model is also refitted without fixed-effect penalties to obtain the original coefficient vector. The fit stores both vectors, their differences, and two equations:

$$g\{E(Y_{it} | b_i, u_i)\} = X_{it}\beta^{(0)} + Z_{it}b_i + S_{it}u_i + o_{it}$$

for the Original Model equation and

$$g\{E(Y_{it} | b_i, u_i)\} = X_{it}\tilde{\beta} + Z_{it}b_i + S_{it}u_i + o_{it}$$

for the Final equation, where 'beta^(0)' is the unpenalized reference and 'tilde(beta)' is the adjusted vector from the selected penalized fit.

Score-based nonlinearity screening uses one unpenalized ML–Laplace null fit, nuisance residualization, and a shared primary-cluster maxT multiplier bootstrap. Likelihood-ratio screening rebuilds and refits each candidate design. Assumption diagnostics are evaluated only when requested.

Value

Object of class 'MEMWAS_fit'. It contains adjusted and original fixed-effect coefficients, coefficient adjustments, Original Model and Final equations, covariance parameters, random and serial modes, approximation metadata, inference, diagnostics, and convergence details.

Examples

```
toy <- data.frame(
  id = rep(seq_len(6), each = 3),
  time = rep(0:2, times = 6),
  x = rep(c(-1, 0, 1), times = 6)
)
toy$y <- 1 + 0.6 * toy$x +
  rep(c(-0.15, -0.08, 0, 0.05, 0.10, 0.18), each = 3) +
  rep(c(-0.03, 0.04, -0.01), times = 6)
toy_fit <- fit_MEMWAS(
  y ~ x, data = toy, id = "id", time = "time",
  random = NULL, autocor = "NONE", se_method = "none",
  control = list(n_starts = 1L, cold_start_verification = FALSE),
  verbose = FALSE
```

```

)
coef(toy_fit)

dat <- data.frame(
  id = rep(seq_len(12), each = 4),
  time = rep(0:3, times = 12),
  x = rep(c(-1, -0.25, 0.25, 1), times = 12)
)
dat$y <- 1 + 0.6 * dat$x + rep(seq(-0.4, 0.4, length.out = 12), each = 4)
fit <- fit_MEMWAS(
  y ~ x, data = dat, id = "id", time = "time", random = ~ 1,
  autocor = "AR(1)", L2_penalty = 0.1, verbose = FALSE
)
coef(fit, type = "both")
summary(fit)

```

MEMWAS_capabilities *Report MEMWAS fitting capabilities*

Description

Returns the model and approximation contract used by ‘fit_MEMWAS()’, ‘tune_MEMWAS()’, and ‘rank_autocorrelation_structures()’.

Usage

```
MEMWAS_capabilities()
```

Value

Data frame with one row per approximation and columns describing supported families and links, random effects, covariance structures, serial processes, penalties, estimation, initialization, standard-error methods, prediction modes, computational kernels, and unavailable combinations. The ‘registry_version’ attribute records the contract version.

Examples

```
MEMWAS_capabilities()
```

predict.MEMWAS_fit	<i>Predict from a fitted MEMWAS model</i>
--------------------	---

Description

Computes conditional, zero-latent, population-marginal, or new-cluster predictions from a fitted MEMWAS model or a tuned object with a final refit.

Usage

```
## S3 method for class 'MEMWAS_fit'
predict(
  object,
  newdata = NULL,
  type = c("link", "response"),
  mode = NULL,
  include_random = is.null(newdata),
  include_serial = is.null(newdata),
  allow_new_levels = TRUE,
  offset = NULL,
  interval = NULL,
  level = NULL,
  se.fit = FALSE,
  trials = NULL,
  ...
)

## S3 method for class 'MEMWAS_family_fit'
predict(object, ...)

## S3 method for class 'tune_MEMWAS'
predict(object, ...)
```

Arguments

object	Object of class 'MEMWAS_fit', 'MEMWAS_family_fit', or 'tune_MEMWAS'. A tuned object must contain a final refit. No default.
newdata	Data frame or 'NULL'. Prediction rows; 'NULL' uses retained fitted rows. Default is 'NULL'.
type	Character scalar. Options are "link" and "response". Population-marginal modes require "response". Default is "link".
mode	Character scalar or 'NULL'. Options are "fitted_cluster_conditional", "zero_random_effect", "population_marginal_mean", and "new_cluster_predictive_distribution". 'NULL' uses the legacy switches 'include_random' and 'include_serial'. Default is 'NULL'.
include_random	Logical scalar. Whether fitted random-effect contributions are included when 'mode = NULL'. Default is 'is.null(newdata)'.

include_serial	Logical scalar. Whether fitted serial-mode contributions are included when 'mode = NULL'. Default is 'is.null(newdata)'.
allow_new_levels	Logical scalar. Whether unseen grouping levels are accepted and assigned zero fitted-mode contribution. Default is 'TRUE'.
offset	Numeric vector, scalar, or 'NULL'. Explicit additive offset for 'newdata', in addition to formula offsets. Default is 'NULL'.
interval	Character scalar or 'NULL'. Options are "none", "confidence", and "prediction". 'NULL' requests point predictions, except that a new-cluster predictive mode selects a prediction interval. Default is 'NULL'.
level	Numeric scalar or 'NULL'. Confidence level in '(0, 1)'; 'NULL' uses the fitted prediction setting or '0.95'. Default is 'NULL'.
se.fit	Logical scalar. Whether to return delta-method standard errors when no interval is requested. Default is 'FALSE'.
trials	Numeric vector, scalar, or 'NULL'. Positive binomial trial totals for 'newdata'; fitted-row totals are reused automatically. Default is 'NULL'.
...	Additional arguments. Fit methods reject unused arguments; the tuned-object method forwards them to its final fit.

Value

A numeric prediction vector when neither an interval nor standard errors are requested. Otherwise, a data frame containing 'fit', 'se.fit', 'lower', and 'upper', with additional predictive-variance components when applicable.

rank_autocorrelation_structures

Rank temporal autocorrelation structures

Description

Fits and ranks candidate temporal covariance specifications by primary-ID grouped cross-validation, AIC, BIC, or reported log-likelihood. Candidate specifications may combine an outcome-loaded residual process with separate predictor-loaded residual processes. All viable candidates are evaluated on the same complete-case rows. Model approximation, covariance construction, likelihood evaluation, and optimization are performed by MEMWAS's registered C++ backend; grouped fold assignment and validation metrics are also native.

Usage

```
rank_autocorrelation_structures(
  formula,
  family = "gaussian",
  data,
  id,
```

```

time,
random = ~1,
candidates = c("NONE", "AR(1)", "OU", "CS"),
criterion = "grouped_cv",
K = 5L,
metric = "RMSE",
cv_aggregate = c("median", "mean"),
cv_prediction_mode = c("population_marginal_mean", "zero_random_effect"),
fold_accept = c("operational", "strict"),
random_cov = "diagonal",
control = list(),
method = "ML",
approximation = "laplace",
init_approximation = "variational_inference",
quadrature_points = 7L,
link = NULL,
offset = NULL,
weights = NULL,
subset = NULL,
na.action = NULL,
theta = NULL,
shape = NULL,
L1_penalty = 0,
L2_penalty = 0,
mape_epsilon = 1e-08,
seed = NULL,
refit_best = TRUE,
refit_se_method = c("hessian", "none"),
keep_fits = FALSE,
keep_fold_fits = FALSE,
verbose = TRUE,
...
)

## S3 method for class 'MEMWAS_autocorrelation_ranking'
print(x, ..., digits = 4L)

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

## S3 method for class 'summary.MEMWAS_autocorrelation_ranking'
print(x, ..., digits = 4L)

```

Arguments

formula	Formula object. Two-sided response and fixed-effect formula. No default.
family	Character scalar or family object. Family accepted by <code>'fit_MEMWAS()'</code> . Default is <code>"gaussian"</code> .

data	Data frame. Contains all model, grouping, time, offset, weight, subset, and serial-loading variables. No default.
id	Character scalar. Primary independent-cluster column; grouped CV keeps every value of this column in one fold. No default.
time	Character scalar. Measurement-time column. No default.
random	Random-effects specification accepted by <code>'fit_MEMWAS()'</code> . Default is <code>'~ 1'</code> .
candidates	Character vector or list. A character vector ranks outcome-loaded structures. List entries may be structure names, <code>'serial_component()'</code> objects, structured components, or lists containing <code>'autocor'</code> , <code>'serial'</code> , <code>'residual_autocor'</code> , and/or <code>'predictor_autocor'</code> . Named entries supply display labels. At least two are required. Default is <code>'c("NONE", "AR(1)", "OU", "CS")'</code> .
criterion	Character scalar. One of <code>'"grouped_cv"'</code> , <code>'"AIC"'</code> , <code>'"BIC"'</code> , or <code>'"logLik"'</code> . Default is <code>'"grouped_cv"'</code> .
K	Integer scalar. Number of primary-ID grouped folds. Default is <code>'5L'</code> .
metric	Character scalar. Grouped-CV metric: <code>'"RMSE"'</code> , <code>'"MAE"'</code> , <code>'"MSE"'</code> , <code>'"MAPE"'</code> , or <code>'"SMAPE"'</code> . Default is <code>'"RMSE"'</code> .
cv_aggregate	Character scalar. Use the fold <code>'"median"'</code> or <code>'"mean"'</code> as the grouped-CV ranking value. Default is <code>'"median"'</code> .
cv_prediction_mode	Character scalar. Held-out-cluster prediction mode: <code>'"population_marginal_mean"'</code> or <code>'"zero_random_effect"'</code> . Default is <code>'"population_marginal_mean"'</code> .
fold_accept	Character scalar. Accept <code>'"operational"'</code> or only <code>'"strict"'</code> convergence in each fold and criterion fit. Default is <code>'"operational"'</code> .
random_cov	Character scalar or term-wise covariance specification. Default is <code>'"diagonal"'</code> .
control	Named list. Computational controls accepted by <code>'fit_MEMWAS()'</code> . Ranking disables optional screening and diagnostics. Default is <code>'list()'</code> .
method	Character scalar. <code>'"ML"'</code> or an eligible <code>'"REML"'</code> fit. Default is <code>'"ML"'</code> .
approximation	Character scalar. Final approximation accepted by <code>'fit_MEMWAS()'</code> . Default is <code>'"laplace"'</code> .
init_approximation	Character scalar. Initialization approximation. Default is <code>'"variational_inference"'</code> .
quadrature_points	Integer scalar. Positive quadrature order. Default is <code>'7L'</code> .
link	Character scalar or <code>'NULL'</code> . Optional link override. Default is <code>'NULL'</code> .
offset	Numeric vector, expression, column name, or <code>'NULL'</code> . Explicit linear-predictor offset. Default is <code>'NULL'</code> .
weights	Numeric vector, expression, column name, or <code>'NULL'</code> . Non-negative observation weights. Default is <code>'NULL'</code> .
subset	Logical/integer expression or <code>'NULL'</code> . Initial row selection. Default is <code>'NULL'</code> .
na.action	Function, character scalar, or <code>'NULL'</code> . <code>'"na.fail"'</code> rejects incomplete candidate data; other supported actions use one common omitted sample across viable candidates. Default is <code>'NULL'</code> .

theta	Numeric scalar or 'NULL'. Negative-binomial dispersion. Default is 'NULL'.
shape	Numeric scalar or 'NULL'. Gamma shape. Default is 'NULL'.
L1_penalty	Numeric scalar. Common non-negative L1 penalty. Default is '0'.
L2_penalty	Numeric scalar. Common non-negative L2 penalty. Default is '0'.
mape_epsilon	Numeric scalar. Positive MAPE/SMAPE safeguard. Default is '1e-8'.
seed	Integer scalar or 'NULL'. Seed for a package-local native grouped-fold stream; it does not alter the global random-number state. 'NULL' uses and advances the current 'R' stream. Default is 'NULL'.
refit_best	Logical scalar. Whether to refit the selected candidate on all common rows. Default is 'TRUE'.
refit_se_method	Character scalar. "hessian" or "none" for the selected full-data refit. Default is "hessian".
keep_fits	Logical scalar. Whether to retain full criterion fits. Default is 'FALSE'.
keep_fold_fits	Logical scalar. Whether to retain accepted fold fits. Default is 'FALSE'.
verbose	Logical scalar. Whether to report progress. Default is 'TRUE'.
...	Additional arguments. No additional arguments are accepted.
x	Object of class 'MEMWAS_autocorrelation_ranking'. No default.
digits	Integer scalar. Number of significant digits printed. Default is '4L'.
object	Object of class 'MEMWAS_autocorrelation_ranking'. No default.

Details

For subject 'i', the native conditional covariance may contain

$$V_i = K_{0i}(\theta_0) + \sum_j \text{diag}(x_{ij}) K_{ji}(\theta_j) \text{diag}(x_{ij}),$$

where 'K0' is an optional outcome-loaded residual process and each 'Kj' is an independently parameterized predictor-loaded residual process. This models residual temporal heterogeneity associated with observed predictor loadings; it does not fit a joint stochastic time-series model for the predictors.

AIC, BIC, and log-likelihood are accepted only when the fit reports an available likelihood criterion. AIC/BIC candidates use the package's effective degrees of freedom when penalties are active. Grouped CV uses one native fold assignment for every candidate and requires every fold to fit, predict, and return a finite metric.

Value

Object of class 'MEMWAS_autocorrelation_ranking'. It contains a sorted 'ranking' table, fold-level results, common analysis-row indices, the selected candidate, an optional selected refit, and optionally retained criterion/fold fits.

'print()' returns 'x' invisibly. 'summary()' returns an object of class 'summary.MEMWAS_autocorrelation_ranking' containing ranking and fold-failure summaries.

Examples

```

toy <- data.frame(
  id = rep(seq_len(6), each = 3),
  time = rep(0:2, times = 6),
  x = rep(c(-1, 0, 1), times = 6)
)
toy$y <- 1 + 0.6 * toy$x +
  rep(c(-0.15, -0.08, 0, 0.05, 0.10, 0.18), each = 3) +
  rep(c(-0.03, 0.04, -0.01), times = 6)
toy_rank <- rank_autocorrelation_structures(
  y ~ x, data = toy, id = "id", time = "time", random = NULL,
  candidates = c(independent = "NONE", ar1 = "AR(1)"),
  criterion = "AIC", refit_best = FALSE,
  control = list(n_starts = 1L, cold_start_verification = FALSE),
  verbose = FALSE
)
toy_rank$ranking

dat <- data.frame(
  id = rep(seq_len(20), each = 5),
  time = rep(0:4, times = 20),
  x1 = stats::rnorm(100), x2 = stats::rnorm(100)
)
dat$y <- 0.5 + 0.6 * dat$x1 - 0.3 * dat$x2 + stats::rnorm(100)

rank_autocorrelation_structures(
  y ~ x1 + x2, data = dat, id = "id", time = "time",
  random = NULL,
  candidates = list(
    independent = "NONE",
    residual_ar1 = list(residual_autocor = "AR(1)"),
    predictor_specific = list(
      residual_autocor = "AR(1)",
      predictor_autocor = c(x1 = "OU", x2 = "AR(1)")
    )
  ),
  criterion = "grouped_cv", K = 5, verbose = FALSE
)

```

screen_MEMWAS_nonlinearity

Screen MEMWAS fixed effects for nonlinearity

Description

Evaluates restricted-cubic-spline additions to an unpenalized ML–Laplace null model. The function can be called independently of ‘fit_MEMWAS()’ so that screening is performed only when explicitly requested.

Usage

```
screen_MEMWAS_nonlinearity(
  object,
  nonlinear_predictors = NULL,
  nonlinear_alpha = 0.05,
  spline_knots = 4L,
  spline_probs = NULL,
  nonlinear_screening_method = "nuisance_adjusted_score",
  nonlinear_multiplicity = "holm",
  nonlinear_bootstrap_reps = 499L,
  nonlinear_bootstrap_seed = NULL,
  verbose = TRUE,
  ...
)

## S3 method for class 'MEMWAS_nonlinearity_screen'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

object	Object of class ‘MEMWAS_fit’. The null model must be operationally converged, unpenalized, fitted by ML with the Laplace approximation, retain a marginal log-likelihood, use no serial-covariance regularization, and not already contain terms selected by this nonlinear screening workflow. No default.
nonlinear_predictors	Character vector or ‘NULL’. Names of numeric fixed-effect predictors to screen. ‘NULL’ selects all eligible numeric predictors having at least five finite unique values. Default is ‘NULL’.
nonlinear_alpha	Numeric scalar. Family-wise selection threshold in ‘(0, 1)’. Default is ‘0.05’.
spline_knots	Integer scalar. Number of restricted-cubic-spline knots; at least three are required. Default is ‘4L’.
spline_probs	Numeric vector or ‘NULL’. Strictly increasing probabilities in ‘[0, 1]’; the length must equal ‘spline_knots’. ‘NULL’ uses the package probabilities. Default is ‘NULL’.
nonlinear_screening_method	Character scalar. Options are “nuisance_adjusted_score”, which uses one shared primary-cluster maxT multiplier bootstrap, and “likelihood_ratio”, which refits each candidate model. Default is “nuisance_adjusted_score”.
nonlinear_multiplicity	Character scalar. P-value adjustment for the likelihood-ratio option. Options are “holm”, “hochberg”, “hommel”, “bonferroni”, “BH”, “BY”, and “none”. It is not used by the shared maxT option. Default is “holm”.
nonlinear_bootstrap_reps	Integer scalar. Number of Rademacher multiplier-bootstrap replicates for score screening; at least 99 are required. Default is ‘499L’.

nonlinear_bootstrap_seed	Integer scalar or 'NULL'. Non-negative seed for a package-local native score-screening stream; it does not alter the global random-number state. 'NULL' uses and advances the current 'R' stream. Default is 'NULL'.
verbose	Logical scalar. Whether to report predictor-level screening progress. Default is 'TRUE'.
...	Additional arguments. No additional arguments are accepted.
x	Object of class 'MEMWAS_nonlinearity_screen'. No default.
digits	Integer scalar. Number of significant digits used for numeric output. Default is 'max(3L, getOption("digits") - 3L)'.

Details

The score method forms candidate spline blocks from retained analysis rows, residualizes them against the null fixed-effect design using model-based Fisher weights, and calibrates all candidate statistics with one shared maxT distribution. Its single-cluster bootstrap requires every random-effect grouping factor to be nested within the primary ID. For crossed grouping structures, use "likelihood_ratio".

The likelihood-ratio method reconstructs the fixed-effect design for every augmented formula and refits each nested candidate under unpenalized ML–Laplace estimation. Binomial trial totals, off-sets, subsets, and missing- data handling retained by the null fit are reused during reconstruction.

Value

Object of class 'MEMWAS_nonlinearity_screen'. It contains the complete screening table, selected predictors, knot and basis definitions, candidate names, selected formula, selected data, call, and method settings.

The input object, invisibly.

Examples

```
toy <- data.frame(
  id = rep(seq_len(6), each = 3),
  time = rep(0:2, times = 6),
  x = rep(c(-1, 0, 1), times = 6)
)
toy$y <- 1 + 0.6 * toy$x +
  rep(c(-0.15, -0.08, 0, 0.05, 0.10, 0.18), each = 3) +
  rep(c(-0.03, 0.04, -0.01), times = 6)
intercept_fit <- fit_MEMWAS(
  y ~ 1, data = toy, id = "id", time = "time",
  random = NULL, autocor = "NONE", se_method = "none",
  control = list(n_starts = 1L, cold_start_verification = FALSE),
  verbose = FALSE
)
empty_screen <- screen_MEMWAS_nonlinearity(intercept_fit, verbose = FALSE)
empty_screen$summary
```

```

dat <- data.frame(
  id = rep(seq_len(20), each = 5),
  time = rep(0:4, times = 20),
  x = rep(seq(-1, 1, length.out = 5), times = 20)
)
dat$y <- 0.4 + 1.2 * dat$x^2 + stats::rnorm(nrow(dat), sd = 0.2)
baseline <- fit_MEMWAS(
  y ~ x, data = dat, id = "id", time = "time", random = NULL,
  autocor = "NONE", se_method = "none", verbose = FALSE
)
screen <- screen_MEMWAS_nonlinearity(
  baseline, nonlinear_predictors = "x",
  nonlinear_bootstrap_reps = 99L, nonlinear_bootstrap_seed = 11L,
  verbose = FALSE
)
print(screen)

```

serial_component

Define a MEMWAS serial covariance component

Description

Describes one latent serial coefficient process for use in `'fit_MEMWAS()'` or `'tune_MEMWAS()'`. Components may load the outcome uniformly or load one numeric predictor/design column.

Usage

```

serial_component(
  structure = "AR(1)",
  predictor = NULL,
  design = NULL,
  order = NULL,
  name = NULL,
  control = list()
)

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

```

Arguments

structure	Character scalar. Options are <code>"NONE"</code> , <code>"AR(1)"</code> , <code>"OU"</code> , <code>"expOU"</code> , <code>"AR(p)"</code> , <code>"ARMA(1,1)"</code> , <code>"CS"</code> , <code>"TOEP"</code> , and <code>"UN"</code> . Orders may be encoded as <code>"AR(3)"</code> or <code>"TOEP(4)"</code> . Default is <code>"AR(1)"</code> .
predictor	Character scalar or <code>'NULL'</code> . Name of one numeric predictor whose values load the serial coefficient process. Supply either <code>'predictor'</code> or <code>'design'</code> . Default is <code>'NULL'</code> .

design	One-sided formula, character scalar, or 'NULL'. Its model matrix must contain exactly one numeric loading column. Supply either 'design' or 'predictor'. Default is 'NULL'.
order	Integer scalar or 'NULL'. Positive order for "AR(p)" or "TOEP"; it must agree with any order encoded in 'structure'. Default is 'NULL'.
name	Character scalar or 'NULL'. Optional non-empty component name used for covariance parameters, contributions, and latent modes. Default is 'NULL'.
control	Named list. Supported entries are 'serial_time_scale' (alias 'time_scale'), 'allow_negative_autocor' (alias 'allow_negative'), and 'max_unstructured_times'. Default is 'list()'.
x	Object of class 'MEMWAS_serial_component'. The component to print. No default.
...	Additional arguments. No additional arguments are accepted by the print method.

Value

Object of class 'MEMWAS_serial_component' containing the normalized structure, loading specification, order, component name, and controls.

Examples

```
ar1 <- serial_component("AR(1)", name = "within_subject")
exp_ou <- serial_component(
  "exp0U", predictor = "exposure", name = "exposure_persistence"
)
free_lag <- serial_component("TOEP", order = 4, name = "free_lag")
list(ar1 = ar1, exp_ou = exp_ou, free_lag = free_lag)
```

Index

AIC.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
AIC.MEMWAS_fit (coef.MEMWAS_fit), 5

check_MEMWAS_assumptions, 2
coef.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
coef.MEMWAS_fit, 5
coef.tune_MEMWAS, 7
compare_approximations, 8

diagnose_approximation, 9

fit_MEMWAS, 10
fitted.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
fitted.MEMWAS_fit (coef.MEMWAS_fit), 5

logLik.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
logLik.MEMWAS_fit (coef.MEMWAS_fit), 5

MEMWAS_capabilities, 17

nobs.MEMWAS_fit (coef.MEMWAS_fit), 5

predict.MEMWAS_family_fit
 (predict.MEMWAS_fit), 18
predict.MEMWAS_fit, 18
predict.tune_MEMWAS
 (predict.MEMWAS_fit), 18
print.MEMWAS_approximation_comparison
 (compare_approximations), 8
print.MEMWAS_approximation_diagnostics
 (diagnose_approximation), 9
print.MEMWAS_assumption_checks
 (check_MEMWAS_assumptions), 2
print.MEMWAS_autocorrelation_ranking
 (rank_autocorrelation_structures),
 19

print.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
print.MEMWAS_fit (coef.MEMWAS_fit), 5
print.MEMWAS_nonlinearity_screen
 (screen_MEMWAS_nonlinearity),
 23
print.MEMWAS_serial_component
 (serial_component), 26
print.summary.MEMWAS_autocorrelation_ranking
 (rank_autocorrelation_structures),
 19
print.summary.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
print.summary.MEMWAS_fit
 (coef.MEMWAS_fit), 5
print.summary.tune_MEMWAS
 (coef.tune_MEMWAS), 7
print.tune_MEMWAS (coef.tune_MEMWAS), 7

rank_autocorrelation_structures, 19
residuals.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
residuals.MEMWAS_fit (coef.MEMWAS_fit),
 5

screen_MEMWAS_nonlinearity, 23
serial_component, 26
summary.MEMWAS_autocorrelation_ranking
 (rank_autocorrelation_structures),
 19
summary.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
summary.MEMWAS_fit (coef.MEMWAS_fit), 5
summary.tune_MEMWAS (coef.tune_MEMWAS),
 7

vcov.MEMWAS_family_fit
 (coef.MEMWAS_fit), 5
vcov.MEMWAS_fit (coef.MEMWAS_fit), 5