

Package ‘nlpsem’

July 22, 2025

Type Package

Title Linear and Nonlinear Longitudinal Process in Structural Equation Modeling Framework

Version 0.3

Description Provides computational tools for nonlinear longitudinal models, in particular the intrinsically nonlinear models, in four scenarios: (1) univariate longitudinal processes with growth factors, with or without covariates including time-invariant covariates (TICs) and time-varying covariates (TVCs); (2) multivariate longitudinal processes that facilitate the assessment of correlation or causation between multiple longitudinal variables; (3) multiple-group models for scenarios (1) and (2) to evaluate differences among manifested groups, and (4) longitudinal mixture models for scenarios (1) and (2), with an assumption that trajectories are from multiple latent classes. The methods implemented are introduced in Jin Liu (2023) <[doi:10.48550/arXiv.2302.03237v2](https://doi.org/10.48550/arXiv.2302.03237v2)>.

License GPL (>= 3.0)

Encoding UTF-8

LazyData true

Depends R (>= 4.0.0), OpenMx (>= 2.21.8)

Imports ggplot2, dplyr, tidyr, stringr, Matrix, nnet, readr, methods

RoxygenNote 7.2.3

URL <https://github.com/Veronica0206/nlpsem>

BugReports <https://github.com/Veronica0206/nlpsem/issues>

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jin Liu [aut, cre]

Maintainer Jin Liu <Veronica.Liu0206@gmail.com>

Repository CRAN

Date/Publication 2023-09-12 22:20:02 UTC

Contents

figOutput-class	2
FSOutput-class	3
getEstimateStats	3
getFigure	5
getIndFS	8
getLatentKappa	10
getLCSM	12
getLGCM	14
getLRT	17
getMediation	18
getMGM	22
getMGroup	25
getMIX	29
getPosterior	33
getSummary	35
getTVCmodel	37
KappaOutput-class	40
ModelSummary	41
ModelSummary,myMxOutput-method	41
myMxOutput-class	41
postOutput-class	42
printTable	42
printTable,FSOutput-method	42
printTable,KappaOutput-method	43
printTable,myMxOutput-method	43
printTable,postOutput-method	44
printTable,StatsOutput-method	44
RMS_dat	45
show,figOutput-method	46
StatsOutput-class	46
Index	47

figOutput-class	<i>S4 Class for displaying figures</i>
-----------------	--

Description

S4 Class to hold the figures output from the getFigure function.

Slots

figures A list of lists containing figures for each specified sub_model and y_model (when applicable).

FSOutput-class	<i>S4 Class for estimated factor scores and their standard errors.</i>
----------------	--

Description

S4 Class for the output structure for the getIndFS() function.

Slots

scores_est A matrix of estimated factor scores.

scores_se A matrix of standard errors of estimated factor scores.

getEstimateStats	<i>Calculate p-Values and Confidence Intervals of Parameters for a Fitted Model</i>
------------------	---

Description

This function calculates p-values and confidence intervals (CIs) of parameters for a given model. It supports different types of CIs, including Wald CIs, likelihood-based CIs, bootstrap CIs, or all three.

Usage

```
getEstimateStats(
  model = NULL,
  est_in,
  p_values = TRUE,
  CI = TRUE,
  CI_type = "Wald",
  rep = NA,
  conf.level = 0.95
)
```

Arguments

model	A fitted mxModel object. Specifically, this should be the mxOutput slot from the result returned by one of the estimation functions provided by this package. The default value is NULL. Providing this parameter is essential when generating likelihood-based and bootstrap confidence intervals (CIs).
est_in	The Estimates slot from the result returned by one of the estimation functions provided by this package, which contains a dataframe with point estimates and standard errors.
p_values	A logical flag indicating whether to calculate p-values. Default is TRUE.
CI	A logical flag indicating whether to compute confidence intervals. Default is TRUE.

CI_type	A string specifying the type of confidence interval to compute. Supported options include "Wald", "likelihood", "bootstrap", or "all". Default is "Wald".
rep	An integer specifying the number of replications for bootstrap. This is applicable if CI_type is "bootstrap" or "all". Default is NA.
conf.level	A numeric value representing the confidence level for confidence interval calculation. Default is 0.95.

Value

An object of class StatsOutput with potential slots:

- `wald`: Contains a data frame with, point estimates, standard errors p-values, and Wald confidence intervals (when specified).
- `likelihood`: Contains a data frame with likelihood-based confidence intervals (when specified).
- `bootstrap`: Contains a data frame with bootstrap confidence intervals (when specified).

The content of these slots can be printed using the `printTable()` method for S4 objects.

References

- Casella, G. & Berger, R.L. (2002). Statistical Inference (2nd ed.). Duxbury Press.
- Madansky, A. (1965). Approximate Confidence Limits for the Reliability of Series and Parallel Systems. *Technometrics*, 7(4), 495-503. Taylor & Francis, Ltd. <https://www.jstor.org/stable/1266390>
- Matthews, D. E. (1988). Likelihood-Based Confidence Intervals for Functions of Many Parameters. *Biometrika*, 75(1), 139-144. Oxford University Press. <https://www.jstor.org/stable/2336444>
- Efron, B. & Tibshirani, R. J. (1994). An Introduction to the Bootstrap. CRC press.

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
# Standardized time-invariant covariates
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
```

```

RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)

# Fit bilinear spline latent growth curve model (fixed knots)
paraBLS_LGCM.r <- c(
  "mueta0", "mueta1", "mueta2", "knot",
  paste0("psi", c("00", "01", "02", "11", "12", "22")),
  "residuals"
)
BLS_LGCM_r <- getLGCM(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "BLS", intrinsic = FALSE,
  records = 1:9, res_scale = 0.1, paramOut = TRUE, names = paraBLS_LGCM.r)
## Generate P value and Wald confidence intervals
getEstimateStats(
  est_in = BLS_LGCM_r@Estimates, CI_type = "Wald"
)
# Fit bilinear spline latent growth curve model (random knots) with time-invariant covariates for
# mathematics development
## Define parameter names
paraBLS.TIC_LGCM.f <- c(
  "alpha0", "alpha1", "alpha2", "alphag",
  paste0("psi", c("00", "01", "02", "0g", "11", "12", "1g", "22", "2g", "gg")), "residuals",
  paste0("beta1", c(0:2, "g")), paste0("beta2", c(0:2, "g")), paste0("mux", 1:2),
  paste0("phi", c("11", "12", "22")), "mueta0", "mueta1", "mueta2", "mu_knot"
)
## Fit the model
BLS_LGCM.TIC_f <- getLGCM(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "BLS", intrinsic = TRUE, records = 1:9,
  growth_TIC = c("ex1", "ex2"), res_scale = 0.1, paramOut = TRUE, names = paraBLS.TIC_LGCM.f
)
## Change optimizer to "SLSQP" for getting likelihood-based confidence interval
mxOption(model = NULL, key = "Default optimizer", "SLSQP", reset = FALSE)
## Generate P value and all three types of confidence intervals
getEstimateStats(
  model = BLS_LGCM.TIC_f@mxOutput, est_in = BLS_LGCM.TIC_f@Estimates, CI_type = "all", rep = 1000
)

```

getFigure

Generate Visualization for Fitted Model

Description

This function generates visualizations for the output of a fitted model. When a Latent Growth Curve Model (LGCM) is fitted for the longitudinal process, it provides (class-specific) estimated growth status with 95 intervals. When a Latent Change Score Model (LCSM) is fitted for the longitudinal process, it provides (class-specific) estimated growth rate with 95 visualizations are particularly useful for understanding the results and trajectories of different classes or groups within the model.

Usage

```

getFigure(
  model,
  nClass = NULL,
  cluster_TIC = NULL,
  grp_var = NULL,
  sub_Model,
  y_var,
  curveFun,
  y_model = NULL,
  t_var,
  records,
  m_var = NULL,
  x_type = NULL,
  x_var = NULL,
  xstarts,
  xlab = "Time",
  outcome = "Process"
)

```

Arguments

model	A fitted mxModel object. Specifically, this should be the mxOutput slot from the result returned by one of the estimation functions provided by this package.
nClass	An integer specifying the number of latent classes for the mixture model or manifested classes for multiple group model. Default is NULL, indicating a single-group model.
cluster_TIC	A string or character vector representing the column name(s) for time-invariant covariate(s) indicating cluster formations. Default is NULL, indicating no such time-invariant covariates are present in the model.
grp_var	A string specifying the column that indicates manifested classes when applicable.
sub_Model	A string that specifies the (class-specific) model. Supported sub-models include "LGCM" (for latent growth curve models), "LCSM" (for latent change score models), "TVC" (for latent growth curve models or latent change score models with a time-varying covariate), "MGM" (for multivariate latent growth curve models or latent change score models), and "MED" (for longitudinal mediation models).
y_var	A string or character vector representing the prefix of the column names for the outcome variable(s) at each study wave.
curveFun	A string specifying the functional forms of the growth curve(s). Supported options for y_model = "LGCM" include: "linear" (or "LIN"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), "Jenss-Bayley" (or "JB"), and "bilinear spline" (or "BLS"). Supported options for y_model = "LCSM" include: "nonparametric" (or "NonP"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), and "Jenss-Bayley" (or "JB").

y_model	A string that specifies how to fit longitudinal outcomes. Supported values are "LGCM" and "LCSM". By default, this is NULL as this argument only requires when sub_Model is "TVC" or "MGM".
t_var	A string representing the prefix of the column names corresponding to the time variable at each study wave.
records	A numeric vector representing the indices of the study waves.
m_var	A string that specifies the prefix of the column names corresponding to the mediator variable at each time point. Default is NULL as this argument only requires when sub_Model is "MED".
x_type	A string indicating the type of predictor variable used in the model. Supported values are "baseline" and "longitudinal". Default is NULL as this argument only requires when sub_Model is "MED".
x_var	A string specifying the baseline predictor if x_type = "baseline", or the prefix of the column names corresponding to the predictor variable at each study wave if x_type = "longitudinal". Default is NULL as this argument only requires when sub_Model is "MED".
xstarts	A numeric value to indicate the starting time of the longitudinal process.
xlab	A string representing the time unit (e.g., "Week", "Month", or "Year") for the x-axis. Default is "Time".
outcome	A string or character vector representing the name(s) of the longitudinal process(es) under examination.

Value

An object of class `figOutput` containing a slot named `figures`. This slot holds a `ggplot` object or a list of `ggplot` objects, each representing a figure for the fitted model. If the `figures` slot contains a list of `ggplot` objects, individual figures can be visualized using the `show()` function.

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
xstarts <- mean(baseT)
```

```

# Plot single group LGCM model
set.seed(20191029)
BLS_LGCM1 <- getLGCM(dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "BLS",
                    intrinsic = FALSE, records = 1:9, res_scale = 0.1)
Figure1 <- getFigure(
  model = BLS_LGCM1@mxOutput, nClass = NULL, cluster_TIC = NULL, sub_Model = "LGCM",
  y_var = "M", curveFun = "BLS", y_model = "LGCM", t_var = "T", records = 1:9,
  m_var = NULL, x_var = NULL, x_type = NULL, xstarts = xstarts, xlab = "Month",
  outcome = "Mathematics"
)
show(Figure1)
# Plot mixture LGCM model
BLS_LGCM2 <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.45, 0.55), sub_Model = "LGCM",
  cluster_TIC = NULL, y_var = "M", t_var = "T", records = 1:9,
  curveFun = "BLS", intrinsic = FALSE, res_scale = list(0.3, 0.3)
)
Figure2 <- getFigure(
  model = BLS_LGCM2@mxOutput, nClass = 2, cluster_TIC = NULL, sub_Model = "LGCM",
  y_var = "M", curveFun = "BLS", y_model = "LGCM", t_var = "T", records = 1:9,
  m_var = NULL, x_var = NULL, x_type = NULL, xstarts = xstarts, xlab = "Month",
  outcome = "Mathematics"
)
show(Figure2)

```

getIndFS

Derive Individual Factor Scores for Each Latent Variable Included in Model

Description

This function computes individual factor scores for each latent variable in a given model. It supports three types of factor scores: maximum likelihood, weighted maximum likelihood, and regression.

Usage

```
getIndFS(model, FS_type = "Regression")
```

Arguments

model	A fitted mxModel object. Specifically, this should be the mxOutput slot from the result returned by one of the estimation functions provided by this package.
FS_type	A string specifying the type of factor scores to compute. Supported options include "ML" (for Maximum Likelihood), "WeightedML" (for Weighted Maximum Likelihood), and "Regression". Default is "Regression".

Value

An object of class FSOutput with two slots:

- `scores_est`: Contains the factor score estimates.
- `scores_se`: Contains the standard errors of the factor score estimates.

The content of these slots can be printed using the `printTable()` method for S4 objects.

References

- Estabrook, R. & Neale, M. C. (2013). A Comparison of Factor Score Estimation Methods in the Presence of Missing Data: Reliability and an Application to Nicotine Dependence. *Multivariate Behavioral Research*, 48, 1-27. doi:[10.1080/00273171.2012.730072](https://doi.org/10.1080/00273171.2012.730072)
- Priestley, M. & Subba Rao, T. (1975). The Estimation of Factor Scores and Kalman Filtering For Discrete Parameter Stationary Processes. *International Journal of Control*, 21, 971-975. doi:[10.1080/00207177508922050](https://doi.org/10.1080/00207177508922050)

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
# Standardized time-invariant covariates
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)

# Fit bilinear spline latent growth curve model (fixed knots)
LIN_LGCM <- getLGCM(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "linear",
  intrinsic = FALSE, records = 1:9, growth_TIC = NULL, res_scale = 0.1
)
getIndFS(model = LIN_LGCM@mxOutput, FS_type = "Regression")
# Fit bilinear spline latent growth curve model (random knots) with time-invariant covariates for
# mathematics development
## Fit the model
BLS_LGCM.TIC_f <- getLGCM(dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "BLS",
  intrinsic = TRUE, records = 1:9, growth_TIC = c("ex1", "ex2"),
  res_scale = 0.1)
getIndFS(model = BLS_LGCM.TIC_f@mxOutput, FS_type = "Regression")
```

getLatentKappa	<i>Compute Latent Kappa Coefficient for Agreement between Two Latent Label Sets</i>
----------------	---

Description

This function calculates the latent kappa, a measure of agreement between two sets of latent categorical labels. It also computes the confidence interval and provides a qualitative interpretation of the agreement level.

Usage

```
getLatentKappa(label1, label2, conf.level = 0.95)
```

Arguments

label1	A factor vector representing the first set of latent categorical labels.
label2	A factor vector representing the second set of latent categorical labels.
conf.level	A numeric value representing the confidence level for the confidence interval of the kappa statistic. The default value is 0.95.

Value

An object of class `KappaOutput` with the following slots:

- `kappa_value`: A string representing the kappa statistic along with its confidence interval.
- `judgment`: A string describing the level of agreement, such as "Perfect Agreement", "Slight Agreement", etc.

The content of these slots can be printed using the `printTable()` method for S4 objects.

References

- Dumenci, L. (2011). The Psychometric Latent Agreement Model (PLAM) for Discrete Latent Variables Measured by Multiple Items. *Organizational Research Methods*, 14(1), 91-115. SAGE Publications. doi:10.1177/1094428110374649
- Landis, J., & Koch, G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1), 159-174. doi:10.2307/2529310
- Agresti, A. (2012). Models for Matched Pairs. In *Categorical Data Analysis* (pp. 413-454). Wiley.

Examples

```

mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)
RMS_dat0$gx1 <- scale(RMS_dat0$INCOME)
RMS_dat0$gx2 <- scale(RMS_dat0$EDU)

## Fit a growth mixture model with no TICs
set.seed(20191029)
MIX_BLS_LGCM_r <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.33, 0.34, 0.33), sub_Model = "LGCM",
  cluster_TIC = NULL, y_var = "M", t_var = "T", records = 1:9,
  curveFun = "BLS", intrinsic = FALSE, res_scale = list(0.3, 0.3, 0.3),
  growth_TIC = NULL, tries = 10
)
## Membership of each individual from growth mixture model with no TICs
label1 <- getPosterior(
  model = MIX_BLS_LGCM_r@mxOutput, nClass = 3, label = FALSE, cluster_TIC = NULL
)
set.seed(20191029)
## Fit a growth mixture model with growth TICs and cluster TICs
MIX_BLS_LGCM.TIC_r <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.33, 0.34, 0.33), sub_Model = "LGCM",
  cluster_TIC = c("gx1", "gx2"), y_var = "M", t_var = "T", records = 1:9,
  curveFun = "BLS", intrinsic = FALSE, res_scale = list(0.3, 0.3, 0.3),
  growth_TIC = c("ex1", "ex2"), tries = 10
)
## Membership of each individual from growth mixture model with growth TICs and cluster TICs
label2 <- getPosterior(
  model = MIX_BLS_LGCM.TIC_r@mxOutput, nClass = 3, label = FALSE,
  cluster_TIC = c("gx1", "gx2")
)
## Calculate the agreement between two sets of membership labels
getLatentKappa(label1 = label1@membership, label2 = label2@membership)

```

getLCSM	<i>Fit a Latent Change Score Model with a Time-invariant Covariate (If Any)</i>
---------	---

Description

This function fits a latent change score model with or without time-invariant covariates to the provided data. It manages model setup, optimization, and if requested, outputs parameter estimates and standard errors.

Usage

```
getLCSM(
  dat,
  t_var,
  y_var,
  curveFun,
  intrinsic = TRUE,
  records,
  growth_TIC = NULL,
  starts = NULL,
  res_scale = NULL,
  tries = NULL,
  OKStatus = 0,
  jitterD = "runif",
  loc = 1,
  scale = 0.25,
  paramOut = FALSE,
  names = NULL
)
```

Arguments

dat	A wide-format data frame, with each row corresponding to a unique ID. It contains the observed variables with repeated measurements and occasions, and time-invariant covariates (TICs) if any.
t_var	A string specifying the prefix of the column names corresponding to the time variable at each study wave.
y_var	A string specifying the prefix of the column names corresponding to the outcome variable at each study wave.
curveFun	A string specifying the functional form of the growth curve. Supported options for latent change score models include: "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), "Jenss-Bayley" (or "JB"), and "nonparametric" (or "NonP").
intrinsic	A logical flag indicating whether to build an intrinsically nonlinear longitudinal model. Default is TRUE.

records	A numeric vector specifying indices of the study waves.
growth_TIC	A string or character vector specifying the column name(s) of time-invariant covariate(s) contributing to the variability of growth factors if any. Default is NULL, indicating no growth TICs are included in the model.
starts	A list containing initial values for the parameters. Default is NULL, indicating no user-specified initial values.
res_scale	A numeric value representing the scaling factor for the initial calculation of the residual variance. This value should be between 0 and 1, exclusive. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
tries	An integer specifying the number of additional optimization attempts. Default is NULL.
OKStatus	An integer (vector) specifying acceptable status codes for convergence. Default is 0.
jitterD	A string specifying the distribution for jitter. Supported values are: "runif" (uniform distribution), "rnorm" (normal distribution), and "rcauchy" (Cauchy distribution). Default is "runif".
loc	A numeric value representing the location parameter of the jitter distribution. Default is 1.
scale	A numeric value representing the scale parameter of the jitter distribution. Default is 0.25.
paramOut	A logical flag indicating whether to output the parameter estimates and standard errors. Default is FALSE.
names	A character vector specifying parameter names. Default is NULL.

Value

An object of class `myMxOutput`. Depending on the `paramOut` argument, the object may contain the following slots:

- `mxOutput`: This slot contains the fitted latent change score model. A summary of this model can be obtained using the `ModelSummary()` function.
- `Estimates` (optional): If `paramOut = TRUE`, a data frame with parameter estimates and standard errors. The content of this slot can be printed using the `printTable()` method for S4 objects.

References

- Liu, J., & Perera, R. A. (2023). Estimating Rate of Change for Nonlinear Trajectories in the Framework of Individual Measurement Occasions: A New Perspective on Growth Curves. *Behavior Research Methods*. doi:10.3758/s13428023020972
- Liu, J. (2022). "Jenss–Bayley Latent Change Score Model With Individual Ratio of the Growth Acceleration in the Framework of Individual Measurement Occasions." *Journal of Educational and Behavioral Statistics*, 47(5), 507–543. doi:10.3102/10769986221099919

- Grimm, K. J., Zhang, Z., Hamagami, F., & Mazzocco, M. (2013). "Modeling Nonlinear Change via Latent Change and Latent Acceleration Frameworks: Examining Velocity and Acceleration of Growth Trajectories." *Multivariate Behavioral Research*, 48(1), 117-143. doi:10.1080/00273171.2012.755111

Examples

```

mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- (RMS_dat0$T1 - baseT)/12
RMS_dat0$T2 <- (RMS_dat0$T2 - baseT)/12
RMS_dat0$T3 <- (RMS_dat0$T3 - baseT)/12
RMS_dat0$T4 <- (RMS_dat0$T4 - baseT)/12
RMS_dat0$T5 <- (RMS_dat0$T5 - baseT)/12
RMS_dat0$T6 <- (RMS_dat0$T6 - baseT)/12
RMS_dat0$T7 <- (RMS_dat0$T7 - baseT)/12
RMS_dat0$T8 <- (RMS_dat0$T8 - baseT)/12
RMS_dat0$T9 <- (RMS_dat0$T9 - baseT)/12
# Standardized time-invariant covariates
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)

# Fit nonparametric change score model for reading development
## Fit model
NonP_LCSM <- getLCSM(
  dat = RMS_dat0, t_var = "T", y_var = "R", curveFun = "nonparametric",
  intrinsic = FALSE, records = 1:9, res_scale = 0.1
)

```

getLGCM

Fit a Latent Growth Curve Model with Time-invariant Covariate (If Any)

Description

This function fits a latent growth curve model with or without time-invariant covariates to the provided data. It manages model setup, optimization, and if requested, outputs parameter estimates and standard errors.

Usage

```

getLGCM(
  dat,

```

```

    t_var,
    y_var,
    curveFun,
    intrinsic = TRUE,
    records,
    growth_TIC = NULL,
    starts = NULL,
    res_scale = NULL,
    tries = NULL,
    OKStatus = 0,
    jitterD = "runif",
    loc = 1,
    scale = 0.25,
    paramOut = FALSE,
    names = NULL
  )

```

Arguments

dat	A wide-format data frame, with each row corresponding to a unique ID. It contains the observed variables with repeated measurements and occasions, and time-invariant covariates (TICs) if any.
t_var	A string specifying the prefix of the column names corresponding to the time variable at each study wave.
y_var	A string specifying the prefix of the column names corresponding to the outcome variable at each study wave.
curveFun	A string specifying the functional form of the growth curve. Supported options for latent growth curve models are: "linear" (or "LIN"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), "Jenss-Bayley" (or "JB"), and "bilinear spline" (or "BLS").
intrinsic	A logical flag indicating whether to build an intrinsically nonlinear longitudinal model. Default is TRUE.
records	A numeric vector specifying indices of the study waves.
growth_TIC	A string or character vector specifying the column name(s) of time-invariant covariate(s) contributing to the variability of growth factors if any. Default is NULL, indicating no growth TICs are included in the model.
starts	A list containing initial values for the parameters. Default is NULL, indicating no user-specified initial values.
res_scale	A numeric value representing the scaling factor for the initial calculation of the residual variance. This value should be between 0 and 1, exclusive. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
tries	An integer specifying the number of additional optimization attempts. Default is NULL.
OKStatus	An integer (vector) specifying acceptable status codes for convergence. Default is 0.

jitterD	A string specifying the distribution for jitter. Supported values are: "runif" (uniform distribution), "rnorm" (normal distribution), and "rcauchy" (Cauchy distribution). Default is "runif".
loc	A numeric value representing the location parameter of the jitter distribution. Default is 1.
scale	A numeric value representing the scale parameter of the jitter distribution. Default is 0.25.
paramOut	A logical flag indicating whether to output the parameter estimates and standard errors. Default is FALSE.
names	A character vector specifying parameter names. Default is NULL.

Value

An object of class `myMxOutput`. Depending on the `paramOut` argument, the object may contain the following slots:

- `mxOutput`: This slot contains the fitted latent growth curve model. A summary of this model can be obtained using the `ModelSummary()` function.
- `Estimates (optional)`: If `paramOut = TRUE`, a data frame with parameter estimates and standard errors. The content of this slot can be printed using the `printTable()` method for `S4` objects.

References

- Liu, J., Perera, R. A., Kang, L., Kirkpatrick, R. M., & Sabo, R. T. (2021). "Obtaining Interpretable Parameters from Reparameterizing Longitudinal Models: Transformation Matrices between Growth Factors in Two Parameter Spaces". *Journal of Educational and Behavioral Statistics*. doi:10.3102/10769986211052009
- Sterba, S. K. (2014). "Fitting Nonlinear Latent Growth Curve Models With Individually Varying Time Points". *Structural Equation Modeling: A Multidisciplinary Journal*, 21(4), 630-647. doi:10.1080/10705511.2014.919828

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
```



```

# Standardized time-invariant covariates
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)

# Fit bilinear spline latent growth curve model (fixed knots)
BLS_LGCM_r <- getLGCM(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "bilinear spline",
  intrinsic = FALSE, records = 1:9, growth_TIC = NULL, res_scale = 0.1
)
# Fit bilinear spline latent growth curve model (random knots) with
# time-invariant covariates for mathematics development
## Define parameter names
paraBLS.TIC_LGCM.f <- c(
  "alpha0", "alpha1", "alpha2", "alphag",
  paste0("psi", c("00", "01", "02", "0g", "11", "12", "1g", "22", "2g", "gg")),
  "residuals", paste0("beta1", c(0:2, "g")), paste0("beta2", c(0:2, "g")),
  paste0("mux", 1:2), paste0("phi", c("11", "12", "22")),
  "mueta0", "mueta1", "mueta2", "mu_knot"
)
## Fit the model
BLS_LGCM.TIC_f <- getLGCM(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "bilinear spline",
  intrinsic = TRUE, records = 1:9, growth_TIC = c("ex1", "ex2"), res_scale = 0.1,
  paramOut = TRUE, names = paraBLS.TIC_LGCM.f
)
## Output point estimate and standard errors
printTable(BLS_LGCM.TIC_f)

```

getLRT

Perform Bootstrap Likelihood Ratio Test for Comparing Full and Reduced Models

Description

This function performs the likelihood ratio test (LRT) to compare a full model (an intrinsically nonlinear longitudinal model) with a corresponding parsimonious alternative (a non-intrinsically nonlinear longitudinal model). It also provides an option to perform bootstrapping for the comparison.

Usage

```
getLRT(full, reduced, boot = FALSE, rep = NA)
```

Arguments

full A fitted mxModel object for the full model. Specifically, this should be the mxOutput slot from the result returned by one of the estimation functions provided by this package.

reduced	A fitted mxModel object for the reduced model. Specifically, this should be the mxOutput slot from the result returned by one of the estimation functions provided by this package.
boot	A logical flag indicating whether to perform bootstrapping for the comparison. Default is FALSE.
rep	An integer specifying the number of bootstrap replications if boot is TRUE. Default is NA.

Value

A data frame containing the number of free parameters, estimated likelihood (-2ll), degrees of freedom, differences in log-likelihood and degrees of freedom, p-values, AIC, and BIC for both the full and reduced models.

Examples

```

mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT

# Fit bilinear spline growth model with random knot (intrinsically nonlinear model)
BLS_LGCM_f <- getLGCM(dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "bilinear spline",
  intrinsic = TRUE, records = 1:9, res_scale = 0.1)
# Fit bilinear spline growth model with fix knot (non-intrinsically nonlinear model)
BLS_LGCM_r <- getLGCM(dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "bilinear spline",
  intrinsic = FALSE, records = 1:9, res_scale = 0.1)
# Likelihood ratio test
getLRT(full = BLS_LGCM_f@mxOutput, reduced = BLS_LGCM_r@mxOutput, boot = FALSE, rep = NA)

```

Description

This function fits a longitudinal mediation model to the provided data. It manages model setup, optimization, and if requested, outputs parameter estimates and standard errors.

Usage

```
getMediation(
  dat,
  t_var,
  y_var,
  m_var,
  x_type,
  x_var,
  curveFun,
  records,
  starts = NULL,
  res_scale = NULL,
  res_cor = NULL,
  tries = NULL,
  OKStatus = 0,
  jitterD = "runif",
  loc = 1,
  scale = 0.25,
  paramOut = FALSE,
  names = NULL
)
```

Arguments

dat	A wide-format data frame, with each row corresponding to a unique ID. It contains the observed variables with repeated measurements and occasions for multiple longitudinal processes and a baseline predictor when applicable.
t_var	A vector of strings, with each element representing the prefix for column names related to the time variable for the corresponding longitudinal variable at each study wave.
y_var	A string specifying the prefix of the column names corresponding to the outcome variable at each study wave.
m_var	A string specifying the prefix of the column names corresponding to the mediator variable at each study wave.
x_type	A string indicating the type of predictor variable used in the model. Supported values are "baseline" and "longitudinal".
x_var	A string specifying the baseline predictor if x_type = "baseline", or the prefix of the column names corresponding to the predictor variable at each study wave if x_type = "longitudinal".
curveFun	A string specifying the functional form of the growth curve. Supported options include: "linear" (or "LIN"), and "bilinear spline" (or "BLS").

records	A list of numeric vectors, with each vector specifying the indices of the observed study waves for the corresponding longitudinal variable.
starts	A list containing initial values for the parameters. Default is NULL, indicating no user-specified initial values.
res_scale	A numeric vector with each element representing the scaling factor for the initial calculation of the residual variance. These values should be between 0 and 1, exclusive. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
res_cor	A numeric value or vector for user-specified residual correlation between any two longitudinal processes to calculate the corresponding initial value. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
tries	An integer specifying the number of additional optimization attempts. Default is NULL.
OKStatus	An integer (vector) specifying acceptable status codes for convergence. Default is 0.
jitterD	A string specifying the distribution for jitter. Supported values are: "runif" (uniform distribution), "rnorm" (normal distribution), and "rcauchy" (Cauchy distribution). Default is "runif".
loc	A numeric value representing the location parameter of the jitter distribution. Default is 1.
scale	A numeric value representing the scale parameter of the jitter distribution. Default is 0.25.
paramOut	A logical flag indicating whether to output the parameter estimates and standard errors. Default is FALSE.
names	A character vector specifying parameter names. Default is NULL.

Value

An object of class `myMxOutput`. Depending on the `paramOut` argument, the object may contain the following slots:

- `mxOutput`: This slot contains the fitted longitudinal mediation model. A summary of this model can be obtained using the `ModelSummary()` function.
- `Estimates (optional)`: If `paramOut = TRUE`, a data frame with parameter estimates and standard errors. The content of this slot can be printed using the `printTable()` method for S4 objects.

References

- Liu, J., & Perera, R.A. (2022). Assessing Mediation Processes Using Piecewise Linear Growth Curve Models with Individual Measurement Occasions. *Behavior Research Methods* (Advance online publication). doi:10.3758/s13428022019402
- MacKinnon, D. P. (2008). *Introduction to Statistical Mediation Analysis*. Taylor & Francis Group/Lawrence Erlbaum Associates.

- Cheong, J., Mackinnon, D. P., & Khoo, S. T. (2003). Investigation of Mediation Processes Using Parallel Process Latent Growth Curve Modeling. *Structural equation modeling: a multidisciplinary journal*, 10(2), 238-262. doi:10.1207/S15328007SEM1002_5
- Soest, T., & Hagtvet, K. A. (2011). Mediation Analysis in a Latent Growth Curve Modeling Framework. *Structural equation modeling: a multidisciplinary journal*, 18(2), 289-314. doi:10.1080/10705511.2011.557344

Examples

```

mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
# Standardized time-invariant covariates
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)

# Example 1: Baseline predictor, linear functional form
## Fit model
set.seed(20191029)
Med2_LGCM_LIN <- getMediation(
  dat = RMS_dat0, t_var = rep("T", 2), y_var = "M", m_var = "R", x_type = "baseline",
  x_var = "ex1", curveFun = "LIN", records = list(1:9, 1:9), res_scale = c(0.1, 0.1),
  res_cor = 0.3
)

# Example 2: Longitudinal predictor, bilinear spline functional form
## Define parameter names
paraMed3_BLS <- c(
  "muetaX1", "muetaXr", "muetaX2", "mugX",
  paste0("psi", c("X1X1", "X1Xr", "X1X2", "XrXr", "XrX2", "X2X2")),
  "alphaM1", "alphaMr", "alphaM2", "mugM",
  paste0("psi", c("M1M1", "M1Mr", "M1M2", "MrMr", "MrM2", "M2M2")), "_r"),
  "alphaY1", "alphaYr", "alphaY2", "mugY",
  paste0("psi", c("Y1Y1", "Y1Yr", "Y1Y2", "YrYr", "YrY2", "Y2Y2")), "_r"),
  paste0("beta", c("X1Y1", "X1Yr", "X1Y2", "XrYr", "XrY2", "X2Y2",
                  "X1M1", "X1Mr", "X1M2", "XrMr", "XrM2", "X2M2",
                  "M1Y1", "M1Yr", "M1Y2", "MrYr", "MrY2", "M2Y2")),
  "muetaM1", "muetaMr", "muetaM2", "muetaY1", "muetaYr", "muetaY2",
  paste0("mediator", c("111", "11r", "112", "1rr", "1r2",
                      "122", "1rr2", "r22", "rrr", "222")),

```

```

paste0("total", c("11", "1r", "12", "rr", "r2", "22")),
"residualsX", "residualsM", "residualsY", "residualsMX", "residualsYX", "residualsYM"
)

## Fit model
set.seed(20191029)
Med3_LGCM_BLS <- getMediation(
  dat = RMS_dat0, t_var = rep("T", 3), y_var = "S", m_var = "M", x_type = "longitudinal",
  x_var = "R", curveFun = "bilinear spline", records = list(2:9, 1:9, 1:9),
  res_scale = c(0.1, 0.1, 0.1), res_cor = c(0.3, 0.3), tries = 10, paramOut = TRUE,
  names = paraMed3_BLS
)
printTable(Med3_LGCM_BLS)

```

getMGM

Fit a Multivariate Latent Growth Curve Model or Multivariate Latent Change Score Model

Description

This function fits a multivariate latent growth curve model or a multivariate latent change score model with the provided data. It manages model setup, optimization, and if requested, outputs parameter estimates and standard errors.

Usage

```

getMGM(
  dat,
  t_var,
  y_var,
  curveFun,
  intrinsic = TRUE,
  records,
  y_model,
  starts = NULL,
  res_scale = NULL,
  res_cor = NULL,
  tries = NULL,
  OKStatus = 0,
  jitterD = "runif",
  loc = 1,
  scale = 0.25,
  paramOut = FALSE,
  names = NULL
)

```

Arguments

<code>dat</code>	A wide-format data frame, with each row corresponding to a unique ID. It contains the observed variables with repeated measurements and occasions for multiple longitudinal outcomes.
<code>t_var</code>	A vector of strings, with each element representing the prefix for column names related to the time variable for the corresponding outcome variable at each study wave.
<code>y_var</code>	A vector of strings, with each element representing the prefix for column names corresponding to a particular outcome variable at each study wave.
<code>curveFun</code>	A string specifying the functional forms of the growth curve(s). Supported options for <code>y_model = "LGCM"</code> include: "linear" (or "LIN"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), "Jenss-Bayley" (or "JB"), and "bilinear spline" (or "BLS"). Supported options for <code>y_model = "LCSM"</code> include: "nonparametric" (or "NonP"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), and "Jenss-Bayley" (or "JB").
<code>intrinsic</code>	A logical flag indicating whether to build an intrinsically nonlinear longitudinal model. Default is TRUE.
<code>records</code>	A list of numeric vectors, with each vector specifying the indices of the observed study waves for the corresponding outcome variable.
<code>y_model</code>	A string specifying how to fit the longitudinal outcome. Supported values are "LGCM" and "LCSM".
<code>starts</code>	A list containing initial values for the parameters. Default is NULL, indicating no user-specified initial values.
<code>res_scale</code>	A numeric vector with each element representing the scaling factor for the initial calculation of the residual variance. These values should be between 0 and 1, exclusive. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the <code>starts</code> argument.
<code>res_cor</code>	A numeric value or vector for user-specified residual correlation between any two longitudinal outcomes to calculate the corresponding initial value. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the <code>starts</code> argument.
<code>tries</code>	An integer specifying the number of additional optimization attempts. Default is NULL.
<code>OKStatus</code>	An integer (vector) specifying acceptable status codes for convergence. Default is 0.
<code>jitterD</code>	A string specifying the distribution for jitter. Supported values are: "runif" (uniform distribution), "rnorm" (normal distribution), and "rcauchy" (Cauchy distribution). Default is "runif".
<code>loc</code>	A numeric value representing the location parameter of the jitter distribution. Default is 1.
<code>scale</code>	A numeric value representing the scale parameter of the jitter distribution. Default is 0.25.
<code>paramOut</code>	A logical flag indicating whether to output the parameter estimates and standard errors. Default is FALSE.
<code>names</code>	A character vector specifying parameter names. Default is NULL.

Value

An object of class `myMxOutput`. Depending on the `paramOut` argument, the object may contain the following slots:

- `mxOutput`: This slot contains the fitted multivariate latent growth curve model or a multivariate latent change score model. A summary of this model can be obtained using the `ModelSummary()` function.
- `Estimates (optional)`: If `paramOut = TRUE`, a data frame with parameter estimates and standard errors. The content of this slot can be printed using the `printTable()` method for S4 objects.

References

- Liu, J., & Perera, R. A. (2021). "Estimating Knots and Their Association in Parallel Bilinear Spline Growth Curve Models in the Framework of Individual Measurement Occasions," *Psychological Methods* (Advance online publication). doi:10.1037/met0000309
- Blozis, S. A. (2004). "Structured Latent Curve Models for the Study of Change in Multivariate Repeated Measures," *Psychological Methods*, 9(3), 334-353. doi:10.1037/1082989X.9.3.334

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT

# Fit linear multivariate latent growth curve model
LIN_PLGCM_f <- getMGM(
  dat = RMS_dat0, t_var = c("T", "T"), y_var = c("R", "M"), curveFun = "LIN",
  intrinsic = FALSE, records = list(1:9, 1:9), y_model = "LGCM", res_scale = c(0.1, 0.1),
  res_cor = 0.3
)
# Fit bilinear spline multivariate latent growth curve model (random knots)
## Define parameter names
paraBLS_PLGCM.f <- c(
  "Y_mueta0", "Y_mueta1", "Y_mueta2", "Y_knot",
  paste0("Y_psi", c("00", "01", "02", "0g", "11", "12", "1g", "22", "2g", "gg")), "Y_res",
  "Z_mueta0", "Z_mueta1", "Z_mueta2", "Z_knot",
  paste0("Z_psi", c("00", "01", "02", "0g", "11", "12", "1g", "22", "2g", "gg")), "Z_res",
```



```

paste0("YZ_psi", c(c("00", "10", "20", "g0", "01", "11", "21", "g1",
                    "02", "12", "22", "g2", "0g", "1g", "2g", "gg"))), "YZ_res"
)
## Fit model
BLS_PLGCM_f <- getMGM(
  dat = RMS_dat0, t_var = c("T", "T"), y_var = c("R", "M"), curveFun = "BLS", intrinsic = TRUE,
  records = list(1:9, 1:9), y_model = "LGCM", res_scale = c(0.1, 0.1), res_cor = 0.3,
  paramOut = TRUE, names = paraBLS_PLGCM.f
)
printTable(BLS_PLGCM_f)

```

getMGroup

Fit a Longitudinal Multiple Group Model

Description

This function fits a longitudinal multiple group model based on the specified sub-model. Supported submodels include:

- Latent growth curve models,
- Latent change score models,
- Latent growth curve models or latent change score models with a time-varying covariate,
- Multivariate latent growth curve models or multivariate latent change score models,
- Longitudinal mediation models.

For the first three submodels, time-invariant covariates are allowed.

Usage

```

getMGroup(
  dat,
  grp_var,
  sub_Model,
  t_var,
  records,
  y_var,
  curveFun,
  intrinsic = NULL,
  y_model = NULL,
  m_var = NULL,
  x_type = NULL,
  x_var = NULL,
  TVC = NULL,
  decompose = NULL,
  growth_TIC = NULL,

```

```

starts = NULL,
res_scale = NULL,
res_cor = NULL,
tries = NULL,
OKStatus = 0,
jitterD = "runif",
loc = 1,
scale = 0.25,
paramOut = FALSE,
names = NULL
)

```

Arguments

<code>dat</code>	A wide-format data frame, with each row corresponding to a unique ID. It contains the observed variables with repeated measurements and occasions for each longitudinal process, time-invariant covariates (TICs) if any, and a variable that indicates manifested group.
<code>grp_var</code>	A string specifying the column that indicates manifested classes.
<code>sub_Model</code>	A string that specifies the sub-model for manifested classes. Supported sub-models include "LGCM" (for latent growth curve models), "LCSM" (for latent change score models), "TVC" (for latent growth curve models or latent change score models with a time-varying covariate), "MGM" (for multivariate latent growth curve models or latent change score models), and "MED" (for longitudinal mediation models).
<code>t_var</code>	A string specifying the prefix of the column names corresponding to the time variable for each study wave. This applies when <code>sub_Model</code> is "LGCM", "LCSM" or "TVC". For <code>sub_Model</code> being "MGM" or "MED", <code>t_var</code> should be a string vector where each element corresponds to the time variable prefix for each respective longitudinal process.
<code>records</code>	A numeric vector denoting the indices of the observed study waves. This applies when <code>sub_Model</code> is "LGCM", "LCSM" or "TVC". For <code>sub_Model</code> being "MGM" or "MED", <code>records</code> should be a list of numeric vectors, where each vector provides the indices of the observed study waves for each longitudinal process.
<code>y_var</code>	A string defining the prefix of the column names corresponding to the outcome variable for each study wave. This is applicable when <code>sub_Model</code> is not "MGM". For <code>sub_Model</code> being "MGM", <code>y_var</code> should be a string vector where each element corresponds to the prefix of the column names for each outcome variable across the study waves.
<code>curveFun</code>	A string specifying the functional forms of the growth curve(s). Supported options for <code>y_model</code> = "LGCM" include: "linear" (or "LIN"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), "Jenss-Bayley" (or "JB"), and "bilinear spline" (or "BLS"). Supported options for <code>y_model</code> = "LCSM" include: "nonparametric" (or "NonP"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), and "Jenss-Bayley" (or "JB").
<code>intrinsic</code>	A logical flag indicating whether to build an intrinsically nonlinear longitudinal model. By default, this is NULL, as it is unnecessary when <code>sub_Model</code> is "MED".

y_model	A string that specifies how to fit longitudinal outcomes. Supported values are "LGCM" and "LCSM". By default, this is NULL as this argument only requires when sub_Model is "TVC" or "MGM".
m_var	A string that specifies the prefix of the column names corresponding to the mediator variable at each study wave. By default, this is NULL as this argument only requires when sub_Model is "MED".
x_type	A string indicating the type of predictor variable used in the model. Supported values are "baseline" and "longitudinal". By default, this is NULL as this argument only requires when sub_Model is "MED".
x_var	A string specifying the baseline predictor if x_type = "baseline", or the prefix of the column names corresponding to the predictor variable at each study wave if x_type = "longitudinal". By default, this is NULL as this argument only requires when sub_Model is "MED".
TVC	A string that specifies the prefix of the column names corresponding to the time-varying covariate at each time point. By default, this is NULL as this argument only requires when sub_Model is "TVC".
decompose	An integer specifying the decomposition option for temporal states. Supported values include 0 (no decomposition), 1 (decomposition with interval-specific slopes as temporal states), 2 (decomposition with interval-specific changes as temporal states), and 3 (decomposition with change-from-baseline as temporal states). By default, this is NULL as this argument only requires when sub_Model is "TVC".
growth_TIC	A string or character vector of column names of time-invariant covariate(s) accounting for the variability of growth factors if any. Default is NULL, indicating no growth TICs present in the model.
starts	A list containing initial values for the parameters. Default is NULL, indicating no user-specified initial values.
res_scale	A list where each element is a (vector of) numeric scaling factor(s) for residual variance to calculate the corresponding initial value for a latent class, between 0 and 1 exclusive. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
res_cor	A list where each element is a (vector of) numeric initial value(s) for residual correlation in each class. It needs to be specified if the sub_Model is "TVC" (when decompose != 0), "MGM", or "MED". By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
tries	An integer specifying the number of additional optimization attempts. Default is NULL.
OKStatus	An integer (vector) specifying acceptable status codes for convergence. Default is 0.
jitterD	A string specifying the distribution for jitter. Supported values are: "runif" (uniform distribution), "rnorm" (normal distribution), and "rcauchy" (Cauchy distribution). Default is "runif".
loc	A numeric value representing the location parameter of the jitter distribution. Default is 1.

scale	A numeric value representing the scale parameter of the jitter distribution. Default is 0.25.
paramOut	A logical flag indicating whether to output the parameter estimates and standard errors. Default is FALSE.
names	A character vector specifying parameter names. Default is NULL.

Value

An object of class `myMxOutput`. Depending on the `paramOut` argument, the object may contain the following slots:

- `mxOutput`: This slot contains the fitted longitudinal multiple group model. A summary of this model can be obtained using the `ModelSummary()` function.
- `Estimates (optional)`: If `paramOut = TRUE`, a data frame with parameter estimates and standard errors. The content of this slot can be printed using the `printTable()` method for `S4` objects.

Examples

```

mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
data("RMS_dat")
# Re-baseline the data so that the estimated initial status is for the starting point of the study
RMS_dat0 <- RMS_dat
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)

# Fit longitudinal multiple group model of bilinear spline functional form with fixed knot
MGroup_BLS_LGCM.TIC_f <- getMGroup(
  dat = RMS_dat0, grp_var = "SEX", sub_Model = "LGCM", y_var = "M", t_var = "T",
  records = 1:9, curveFun = "BLS", intrinsic = FALSE, res_scale = list(0.3, 0.3)
)

# Fit longitudinal multiple group model of bilinear spline functional form with random knot
paraBLS.TIC_LGCM.f <- c(
  "alpha0", "alpha1", "alpha2", "alphag",
  paste0("psi", c("00", "01", "02", "0g", "11", "12", "1g", "22", "2g", "gg")),
  "residuals", paste0("beta1", c(0:2, "g")), paste0("beta2", c(0:2, "g")),
  paste0("mux", 1:2), paste0("phi", c("11", "12", "22")),
  "mueta0", "mueta1", "mueta2", "mu_knot"
)
set.seed(20191029)

```

```
MGroup_BLS_LGCM.TIC_f <- getMGroup(  
  dat = RMS_dat0, grp_var = "SEX", sub_Model = "LGCM", y_var = "M", t_var = "T",  
  records = 1:9, curveFun = "BLS", intrinsic = TRUE, res_scale = list(0.3, 0.3),  
  growth_TIC = c("ex1", "ex2"), tries = 10, paramOut = TRUE, names = paraBLS.TIC_LGCM.f  
)  
printTable(MGroup_BLS_LGCM.TIC_f)
```

getMIX

Fit a Longitudinal Mixture Model

Description

This function fits a longitudinal mixture model based on the specified sub-model. Supported sub-models include:

- Latent growth curve models,
- Latent change score models,
- Latent growth curve models or latent change score models with a time-varying covariate,
- Multivariate latent growth curve models or multivariate latent change score models,
- Longitudinal mediation models.

Time-invariant covariates are allowed for the first three submodels.

Usage

```
getMIX(  
  dat,  
  prop_starts,  
  sub_Model,  
  cluster_TIC = NULL,  
  t_var,  
  records,  
  y_var,  
  curveFun,  
  intrinsic = NULL,  
  y_model = NULL,  
  m_var = NULL,  
  x_type = NULL,  
  x_var = NULL,  
  TVC = NULL,  
  decompose = NULL,  
  growth_TIC = NULL,  
  starts = NULL,  
  res_scale = NULL,  
  res_cor = NULL,
```

```

    tries = NULL,
    OKStatus = 0,
    jitterD = "runif",
    loc = 1,
    scale = 0.25,
    paramOut = FALSE,
    names = NULL
)

```

Arguments

<code>dat</code>	A wide-format data frame, with each row corresponding to a unique ID. It contains the observed variables with repeated measurements and occasions for each longitudinal process, and time-invariant covariates (TICs) if any.
<code>prop_starts</code>	A numeric vector of user-specified initial component proportions of latent classes.
<code>sub_Model</code>	A string that specifies the sub-model for latent classes. Supported sub-models include "LGCM" (for latent growth curve models), "LCSM" (for latent change score models), "TVC" (for latent growth curve models or latent change score models with a time-varying covariate), "MGM" (for multivariate latent growth curve models or latent change score models), and "MED" (for longitudinal mediation models).
<code>cluster_TIC</code>	A string or character vector representing the column name(s) for time-invariant covariate(s) indicating cluster formations. Default is NULL, indicating no such time-invariant covariates are present in the model.
<code>t_var</code>	A string specifying the prefix of the column names corresponding to the time variable for each study wave. This applies when <code>sub_Model</code> is "LGCM", "LCSM" or "TVC". For <code>sub_Model</code> being "MGM" or "MED", <code>t_var</code> should be a string vector where each element corresponds to the time variable prefix for each respective longitudinal process.
<code>records</code>	A numeric vector denoting the indices of the observed study waves. This applies when <code>sub_Model</code> is "LGCM", "LCSM" or "TVC". For <code>sub_Model</code> being "MGM" or "MED", <code>records</code> should be a list of numeric vectors, where each vector provides the indices of the observed study waves for each longitudinal process.
<code>y_var</code>	A string defining the prefix of the column names corresponding to the outcome variable for each study wave. This is applicable when <code>sub_Model</code> is not "MGM". For <code>sub_Model</code> being "MGM", <code>y_var</code> should be a string vector where each element corresponds to the prefix of the column names for each outcome variable across the study waves.
<code>curveFun</code>	A string specifying the functional forms of the growth curve(s). Supported options for <code>y_model</code> = "LGCM" include: "linear" (or "LIN"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), "Jenss-Bayley" (or "JB"), and "bilinear spline" (or "BLS"). Supported options for <code>y_model</code> = "LCSM" include: "nonparametric" (or "NonP"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), and "Jenss-Bayley" (or "JB").
<code>intrinsic</code>	A logical flag indicating whether to build an intrinsically nonlinear longitudinal model. By default, this is NULL as it is unnecessary when <code>sub_Model</code> is "MED".

y_model	A string that specifies how to fit longitudinal outcomes. Supported values are "LGCM" and "LCSM". By default, this is NULL as this argument only requires when sub_Model is "TVC" or "MGM".
m_var	A string that specifies the prefix of the column names corresponding to the mediator variable at each study wave. By default, this is NULL as this argument only requires when sub_Model is "MED".
x_type	A string indicating the type of predictor variable used in the model. Supported values are "baseline" and "longitudinal". By default, this is NULL as this argument only requires when sub_Model is "MED".
x_var	A string specifying the baseline predictor if x_type = "baseline", or the prefix of the column names corresponding to the predictor variable at each study wave if x_type = "longitudinal". By default, this is NULL as this argument only requires when sub_Model is "MED".
TVC	A string that specifies the prefix of the column names corresponding to the time-varying covariate at each time point. By default, this is NULL as this argument only requires when sub_Model is "TVC".
decompose	An integer specifying the decomposition option for temporal states. Supported values include 0 (no decomposition), 1 (decomposition with interval-specific slopes as temporal states), 2 (decomposition with interval-specific changes as temporal states), and 3 (decomposition with change-from-baseline as temporal states). By default, this is NULL as this argument only requires when sub_Model is "TVC".
growth_TIC	A string or character vector of column names of time-invariant covariate(s) accounting for the variability of growth factors if any. Default is NULL, indicating no growth TICs present in the model.
starts	A list containing initial values for the parameters. Default is NULL, indicating no user-specified initial values.
res_scale	A list where each element is a (vector of) numeric scaling factor(s) for residual variance to calculate the corresponding initial value for a latent class, between 0 and 1 exclusive. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
res_cor	A list where each element is a (vector of) numeric initial value(s) for residual correlation in each class. It needs to be specified if the sub_Model is "TVC" (when decompose != 0), "MGM", or "MED". By default, this is NULL, as it is unnecessary when the user specifies the initial values using the starts argument.
tries	An integer specifying the number of additional optimization attempts. Default is NULL.
OKStatus	An integer (vector) specifying acceptable status codes for convergence. Default is 0.
jitterD	A string specifying the distribution for jitter. Supported values are: "runif" (uniform distribution), "rnorm" (normal distribution), and "rcauchy" (Cauchy distribution). Default is "runif".
loc	A numeric value representing the location parameter of the jitter distribution. Default is 1.

scale	A numeric value representing the scale parameter of the jitter distribution. Default is 0.25.
paramOut	A logical flag indicating whether to output the parameter estimates and standard errors. Default is FALSE.
names	A character vector specifying parameter names. Default is NULL.

Value

An object of class `myMxOutput`. Depending on the `paramOut` argument, the object may contain the following slots:

- `mxOutput`: This slot contains the fitted longitudinal mixture model. A summary of this model can be obtained using the `ModelSummary()` function.
- `Estimates (optional)`: If `paramOut = TRUE`, a data frame with parameter estimates and standard errors. The content of this slot can be printed using the `printTable()` method for S4 objects.

References

- Liu, J., & Perera, R. A. (2022). Extending Mixture of Experts Model to Investigate Heterogeneity of Trajectories: When, Where and How to Add Which Covariates. *Psychological Methods* (Advance online publication). doi:10.1037/met0000436
- Liu, J., & Perera, R. A. (2022). Extending Growth Mixture Model to Assess Heterogeneity in Joint Development with Piecewise Linear Trajectories in the Framework of Individual Measurement Occasions. *Psychological Methods* (Advance online publication). doi:10.1037/met0000500
- Liu, J., & Perera, R. A. (2023). Estimating Rate of Change for Nonlinear Trajectories in the Framework of Individual Measurement Occasions: A New Perspective on Growth Curves. *Behavior Research Methods*. doi:10.3758/s13428023020972
- Liu, J. (2023). Further Exploration of the Effects of Time-varying Covariate in Growth Mixture Models with Nonlinear Trajectories. *Behavior Research Methods*. doi:10.3758/s13428-023021835

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
```



```

RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)
RMS_dat0$gx1 <- scale(RMS_dat0$INCOME)
RMS_dat0$gx2 <- scale(RMS_dat0$EDU)

# Fit longitudinal mixture group model of bilinear spline functional form with fixed knot
# (2 classes)
MIX_BLS_LGCM.TIC_r <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.45, 0.55), sub_Model = "LGCM",
  cluster_TIC = NULL, y_var = "M", t_var = "T", records = 1:9,
  curveFun = "BLS", intrinsic = FALSE, res_scale = list(0.3, 0.3)
)
# Fit longitudinal mixture group model of bilinear spline functional form with fixed knot
# (3 classes)
paraBLS.TIC_LGCM.r <- c(
  "alpha0", "alpha1", "alpha2", "knot",
  paste0("psi", c("00", "01", "02", "11", "12", "22")), "residuals",
  paste0("beta1", 0:2), paste0("beta2", 0:2),
  paste0("mux", 1:2), paste0("phi", c("11", "12", "22")),
  "mueta0", "mueta1", "mueta2"
)
set.seed(20191029)
MIX_BLS_LGCM.TIC_r <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.33, 0.34, 0.33), sub_Model = "LGCM",
  cluster_TIC = c("gx1", "gx2"), y_var = "M", t_var = "T", records = 1:9,
  curveFun = "BLS", intrinsic = FALSE, res_scale = list(0.3, 0.3, 0.3),
  growth_TIC = c("ex1", "ex2"), tries = 10, paramOut = TRUE,
  names = paraBLS.TIC_LGCM.r
)
printTable(MIX_BLS_LGCM.TIC_r)

```

getPosterior

Compute Posterior Probabilities, Cluster Assignments, and Model Entropy for a Longitudinal Mixture Model

Description

This function computes posterior probabilities, cluster assignments, and model entropy for a given mixture model with a predefined number of classes. If the true labels are available, it can also compute the model accuracy.

Usage

```
getPosterior(model, nClass, label = FALSE, cluster_TIC = NULL)
```

Arguments

model	A fitted mxModel object. Specifically, this should be the mxOutput slot from the output of getMIX().
nClass	An integer representing the predefined number of latent classes in the model.
label	A logical value indicating whether the data contains true labels, which are often available in a simulated data set. Default is FALSE.
cluster_TIC	A string or character vector representing the column name(s) for time-invariant covariate(s) indicating cluster formations. Default is NULL, indicating that no such time-invariant covariates are present in the model.

Value

An object of class postOutput. Depending on the label argument, the object may contain the following slots:

- prob: A matrix of posterior probabilities.
- membership: A vector indicating class membership based on maximum posterior probability.
- entropy: The entropy of the model, a measure of uncertainty in class assignment.
- accuracy (optional): If label = TRUE, the model's accuracy based on true labels.

The content of these slots can be printed using the printTable() method for S4 objects.

References

- Peugh, J., & Fan, X. (2015). Enumeration Index Performance in Generalized Growth Mixture Models: A Monte Carlo Test of Muthén's (2003) Hypothesis. *Structural Equation Modeling: A Multidisciplinary Journal*, 22(1), 115-131. Routledge. doi:10.1080/10705511.2014.919823
- Lubke, G., & Muthén, B.O. (2007). Performance of Factor Mixture Models as a Function of Model Size, Covariate Effects, and Class-Specific Parameters. *Structural Equation Modeling: A Multidisciplinary Journal*, 14(1), 26-47. Routledge. doi:10.1080/10705510709336735

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
data("RMS_dat")
# Re-baseline the data so that the estimated initial status is for the starting point of the study
RMS_dat0 <- RMS_dat
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
```

```

RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)
RMS_dat0$gx1 <- scale(RMS_dat0$INCOME)
RMS_dat0$gx2 <- scale(RMS_dat0$EDU)

# Fit longitudinal mixture group model of bilinear spline functional form with fixed knot but no
# cluster TICs or growth TICs
set.seed(20191029)
MIX_BLS_LGCM_r <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.33, 0.34, 0.33), sub_Model = "LGCM",
  cluster_TIC = NULL, y_var = "M", t_var = "T", records = 1:9, curveFun = "BLS",
  intrinsic = FALSE, res_scale = list(0.3, 0.3, 0.3), growth_TIC = NULL, tries = 10
)
label1 <- getPosterior(
  model = MIX_BLS_LGCM_r@mxOutput, nClass = 3, label = FALSE, cluster_TIC = NULL
)
# Fit longitudinal mixture group model of bilinear spline functional form with fixed knot, cluster
# TICs, and growth TICs
set.seed(20191029)
MIX_BLS_LGCM.TIC_r <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.33, 0.34, 0.33), sub_Model = "LGCM",
  cluster_TIC = c("gx1", "gx2"), y_var = "M", t_var = "T", records = 1:9,
  curveFun = "BLS", intrinsic = FALSE, res_scale = list(0.3, 0.3, 0.3),
  growth_TIC = c("ex1", "ex2"), tries = 10
)
label2 <- getPosterior(
  model = MIX_BLS_LGCM.TIC_r@mxOutput, nClass = 3, label = FALSE, cluster_TIC = c("gx1", "gx2")
)

```

getSummary

Summarize Model Fit Statistics for Fitted Models

Description

This function summarizes the model fit statistics for a list of fitted models. The summary includes the number of parameters, estimated likelihood (-2ll), AIC, BIC, and other relevant statistics.

Usage

```
getSummary(model_list, HetModels = FALSE)
```

Arguments

model_list	A list of fitted mxModel objects. Specifically, each element of the list should be the mxOutput slot from the result returned by one of the estimation functions provided by this package.
HetModels	A logical flag indicating whether a mixture model or a multiple group model is included in the list. If set to TRUE, the function can also be used for the enumeration process, allowing the determination of the optimal number of latent classes based on model fit statistics such as BIC. The default value is FALSE.

Value

A data frame summarizing model fit statistics (number of parameters, estimated likelihood, AIC, BIC, etc.) for each model.

Examples

```

mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
# Load ECLS-K (2011) data
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- RMS_dat0$T1 - baseT
RMS_dat0$T2 <- RMS_dat0$T2 - baseT
RMS_dat0$T3 <- RMS_dat0$T3 - baseT
RMS_dat0$T4 <- RMS_dat0$T4 - baseT
RMS_dat0$T5 <- RMS_dat0$T5 - baseT
RMS_dat0$T6 <- RMS_dat0$T6 - baseT
RMS_dat0$T7 <- RMS_dat0$T7 - baseT
RMS_dat0$T8 <- RMS_dat0$T8 - baseT
RMS_dat0$T9 <- RMS_dat0$T9 - baseT

# Fit bilinear spline growth model with fix knot
## Single group model
BLS_LGCM1 <- getLGCM(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "BLS", intrinsic = FALSE,
  records = 1:9, res_scale = 0.1
)
getSummary(model_list = list(BLS_LGCM1@mxOutput), HetModels = FALSE)
## Mixture model with two latent classes
set.seed(20191029)
BLS_LGCM2 <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.45, 0.55), sub_Model = "LGCM", cluster_TIC = NULL,
  y_var = "M", t_var = "T", records = 1:9, curveFun = "BLS", intrinsic = FALSE,
  res_scale = list(0.3, 0.3), growth_TIC = NULL, tries = 10
)
## Mixture model with three latent classes
set.seed(20191029)
BLS_LGCM3 <- getMIX(
  dat = RMS_dat0, prop_starts = c(0.33, 0.34, 0.33), sub_Model = "LGCM", cluster_TIC = NULL,
  y_var = "M", t_var = "T", records = 1:9, curveFun = "BLS", intrinsic = FALSE,
  res_scale = list(0.3, 0.3, 0.3), growth_TIC = NULL, tries = 10
)

getSummary(model_list = list(BLS_LGCM1@mxOutput, BLS_LGCM2@mxOutput, BLS_LGCM3@mxOutput),
  HetModels = TRUE)

```

getTVCmodel	<i>Fit a Latent Growth Curve Model or Latent Change Score Model with Time-varying and Time-invariant Covariates</i>
-------------	---

Description

This function fits a latent growth curve model or latent change score model with a time-varying covariate and potential time-invariant covariates to the provided data. It manages model setup, optimization, and if requested, outputs parameter estimates and standard errors.

Usage

```
getTVCmodel(  
  dat,  
  t_var,  
  y_var,  
  curveFun,  
  intrinsic = TRUE,  
  records,  
  y_model,  
  TVC,  
  decompose,  
  growth_TIC = NULL,  
  starts = NULL,  
  res_scale = NULL,  
  res_cor = NULL,  
  tries = NULL,  
  OKStatus = 0,  
  jitterD = "runif",  
  loc = 1,  
  scale = 0.25,  
  paramOut = FALSE,  
  names = NULL  
)
```

Arguments

dat	A wide-format data frame, with each row corresponding to a unique ID. It contains the observed variables with repeated measurements (for the longitudinal outcome and time-varying covariates), occasions, and time-invariant covariates (TICs) if any.
t_var	A string specifying the prefix of the column names corresponding to the time variable at each study wave.
y_var	A string specifying the prefix of the column names corresponding to the outcome variable at each study wave.

curveFun	A string specifying the functional form of the growth curve. Supported options for <code>y_model = "LGCM"</code> include: "linear" (or "LIN"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), "Jenss-Bayley" (or "JB"), and "bilinear spline" (or "BLS"). Supported options for <code>y_model = "LCSM"</code> include: "nonparametric" (or "NonP"), "quadratic" (or "QUAD"), "negative exponential" (or "EXP"), and "Jenss-Bayley" (or "JB").
intrinsic	A logical flag indicating whether to build an intrinsically nonlinear longitudinal model. Default is TRUE.
records	A numeric vector specifying the indices of the observed study waves.
y_model	A string specifying how to fit the longitudinal outcome. Supported values are "LGCM" and "LCSM".
TVC	A string specifying the prefix of the column names corresponding to the time-varying covariate at each study wave.
decompose	An integer specifying the decomposition option for temporal states. Supported values include 0 (no decomposition), 1 (decomposition with interval-specific slopes as temporal states), 2 (decomposition with interval-specific changes as temporal states), and 3 (decomposition with change-from-baseline as temporal states).
growth_TIC	A string or character vector specifying the column name(s) of time-invariant covariate(s) that account for the variability of growth factors, if any. Default is NULL, indicating no growth TICs present in the model.
starts	A list containing initial values for the parameters. Default is NULL, indicating no user-specified initial values.
res_scale	A numeric value or numeric vector. For a model with <code>decompose = 0</code> , it is a numeric value representing the scaling factor used to calculate the initial value for the residual variance of the longitudinal outcome. In cases where <code>decompose != 0</code> , it is a numeric vector of user-specified scaling factors used to calculate the initial values for the residual variance of both the longitudinal outcome and the time-varying covariate. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the <code>starts</code> argument.
res_cor	A numeric value. When <code>decompose != 0</code> , this represents the user-specified residual correlation between the longitudinal outcome and the time-varying covariate, which is used to calculate the corresponding initial value. If <code>decompose = 0</code> , this should be NULL. By default, this is NULL, as it is unnecessary when the user specifies the initial values using the <code>starts</code> argument.
tries	An integer specifying the number of additional optimization attempts. Default is NULL.
OKStatus	An integer (vector) specifying acceptable status codes for convergence. Default is 0.
jitterD	A string specifying the distribution for jitter. Supported values are: "runif" (uniform distribution), "rnorm" (normal distribution), and "rcauchy" (Cauchy distribution). Default is "runif".
loc	A numeric value representing the location parameter of the jitter distribution. Default is 1.

scale	A numeric value representing the scale parameter of the jitter distribution. Default is 0.25.
paramOut	A logical flag indicating whether to output the parameter estimates and standard errors. Default is FALSE.
names	A character vector specifying parameter names. Default is NULL.

Value

An object of class myMxOutput. Depending on the paramOut argument, the object may contain the following slots:

- mxOutput: This slot contains the fitted latent growth curve model or latent change score model with a time-varying covariate. A summary of this model can be obtained using the ModelSummary() function.
- Estimates (optional): If paramOut = TRUE, a data frame with parameter estimates and standard errors. The content of this slot can be printed using the printTable() method for S4 objects.

References

- Liu, J., & Perera, R. A. (2023). Estimating Rate of Change for Nonlinear Trajectories in the Framework of Individual Measurement Occasions: A New Perspective on Growth Curves. Behavior Research Methods. doi:10.3758/s13428023020972
- Liu, J. (2022). "Decomposing Impact on Longitudinal Outcome of Time-varying Covariate into Baseline Effect and Temporal Effect." arXiv. <https://arxiv.org/abs/2210.16916>

Examples

```
mxOption(model = NULL, key = "Default optimizer", "CSOLNP", reset = FALSE)
data("RMS_dat")
RMS_dat0 <- RMS_dat
# Re-baseline the data so that the estimated initial status is for the starting point of the study
baseT <- RMS_dat0$T1
RMS_dat0$T1 <- (RMS_dat0$T1 - baseT)/12
RMS_dat0$T2 <- (RMS_dat0$T2 - baseT)/12
RMS_dat0$T3 <- (RMS_dat0$T3 - baseT)/12
RMS_dat0$T4 <- (RMS_dat0$T4 - baseT)/12
RMS_dat0$T5 <- (RMS_dat0$T5 - baseT)/12
RMS_dat0$T6 <- (RMS_dat0$T6 - baseT)/12
RMS_dat0$T7 <- (RMS_dat0$T7 - baseT)/12
RMS_dat0$T8 <- (RMS_dat0$T8 - baseT)/12
RMS_dat0$T9 <- (RMS_dat0$T9 - baseT)/12
RMS_dat0$ex1 <- scale(RMS_dat0$Approach_to_Learning)
RMS_dat0$ex2 <- scale(RMS_dat0$Attention_focus)

# Standardize reading ability over time with its baseline value
BL_mean <- mean(RMS_dat0[, "R1"])
BL_var <- var(RMS_dat0[, "R1"])
RMS_dat0$Rs1 <- (RMS_dat0$R1 - BL_mean)/sqrt(BL_var)
RMS_dat0$Rs2 <- (RMS_dat0$R2 - BL_mean)/sqrt(BL_var)
```

```

RMS_dat0$Rs3 <- (RMS_dat0$R3 - BL_mean)/sqrt(BL_var)
RMS_dat0$Rs4 <- (RMS_dat0$R4 - BL_mean)/sqrt(BL_var)
RMS_dat0$Rs5 <- (RMS_dat0$R5 - BL_mean)/sqrt(BL_var)
RMS_dat0$Rs6 <- (RMS_dat0$R6 - BL_mean)/sqrt(BL_var)
RMS_dat0$Rs7 <- (RMS_dat0$R7 - BL_mean)/sqrt(BL_var)
RMS_dat0$Rs8 <- (RMS_dat0$R8 - BL_mean)/sqrt(BL_var)
RMS_dat0$Rs9 <- (RMS_dat0$R9 - BL_mean)/sqrt(BL_var)

# Fit bilinear spline latent growth curve model (fixed knot) with a time-varying
# reading ability for mathematics development
BLS_TVC_LGCM1 <- getTVCmodel(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "BLS", intrinsic = FALSE,
  records = 1:9, y_model = "LGCM", TVC = "Rs", decompose = 0, growth_TIC = NULL,
  res_scale = 0.1
)

# Fit negative exponential latent growth curve model (random ratio) with a
# decomposed time-varying reading ability and time-invariant covariates for
# mathematics development
paraEXP_LGCM3.f <- c(
  "Y_alpha0", "Y_alpha1", "Y_alphag",
  paste0("Y_psi", c("00", "01", "0g", "11", "1g", "gg")), "Y_residuals",
  "X_mueta0", "X_mueta1", paste0("X_psi", c("00", "01", "11")),
  paste0("X_rel_rate", 2:8), paste0("X_abs_rate", 1:8), "X_residuals",
  paste0("betaTIC", c(0:1, "g")), paste0("betaTIC", c(0:1, "g")),
  paste0("betaTVC", c(0:1, "g")),
  "muTIC1", "muTIC2", "phiTIC11", "phiTIC12", "phiTIC22",
  "Y_mueta0", "Y_mueta1", "Y_mu_slp_ratio",
  "covBL1", "covBL2", "kappa", "Cov_XYres")
set.seed(20191029)
EXP_TVCSlp_LGCM3.f <- getTVCmodel(
  dat = RMS_dat0, t_var = "T", y_var = "M", curveFun = "EXP", intrinsic = TRUE,
  records = 1:9, y_model = "LGCM", TVC = "Rs", decompose = 1,
  growth_TIC = c("ex1", "ex2"), res_scale = c(0.1, 0.1),
  res_cor = 0.3, tries = 10, paramOut = TRUE, names = paraEXP_LGCM3.f
)
printTable(EXP_TVCSlp_LGCM3.f)

```

KappaOutput-class

S4 Class for kappa statistic with confidence interval and judgment.

Description

S4 Class for the output structure for the `getLatentKappa()` function.

Slots

kappa_value A character vector for the kappa statistic with 95% CI.
 judgment A character vector for the judgement for agreement.

ModelSummary *S4 Generic for summarizing an optimized MxModel.*

Description

Generic function for printing model summary of MxModel object.

Usage

ModelSummary(object)

Arguments

object An object of the appropriate class.

ModelSummary, myMxOutput-method
S4 Method for summarizing an optimized MxModel.

Description

Method for printing model summary of MxModel object.

Usage

```
## S4 method for signature 'myMxOutput'
ModelSummary(object)
```

Arguments

object An object of class "myMxOutput".

myMxOutput-class *Standard Methods (S4) for the package*

Description

S4 Class for the output structure for estimate functions.

Slots

mxOutput An object of class "MxModel".
 Estimates A data frame of estimates.

postOutput-class	<i>S4 Class for posterior probabilities, membership, entropy, and accuracy (when applicable)</i>
------------------	--

Description

S4 Class for the output structure for the `getPosterior()` function.

Slots

prob A matrix of posterior probabilities.
 membership A numeric vector for membership.
 entropy A numeric value for entropy.
 accuracy A numeric value for accuracy.

printTable	<i>S4 Generic for displaying output in a table format.</i>
------------	--

Description

Generic function for printing output that are tables.

Usage

```
printTable(object)
```

Arguments

object An object of the appropriate class.

printTable,FSOutput-method	<i>S4 Method for printing estimated factor scores and their standard errors</i>
----------------------------	---

Description

Method for printing estimated factor scores and their standard errors.

Usage

```
## S4 method for signature 'FSOutput'  
printTable(object)
```

Arguments

object An object of class "FSOutput".

printTable,KappaOutput-method

S4 Method for printing kappa statistic with 95% CI and judgement for agreement.

Description

Method for printing kappa statistic with 95% CI and judgement for agreement.

Usage

```
## S4 method for signature 'KappaOutput'  
printTable(object)
```

Arguments

object An object of class "KappaOutput".

printTable,myMxOutput-method

S4 Method for printing point estimates with standard errors

Description

Method for printing point estimates and standard errors.

Usage

```
## S4 method for signature 'myMxOutput'  
printTable(object)
```

Arguments

object An object of class "myMxOutput".

printTable,postOutput-method

S4 Method for printing posterior probabilities, membership, entropy, and accuracy.

Description

Method for printing posterior probabilities, membership, entropy, and accuracy.

Usage

```
## S4 method for signature 'postOutput'  
printTable(object)
```

Arguments

object An object of class "postOutput".

printTable,StatsOutput-method

S4 Method for printing p values and confidence intervals (when applicable)

Description

Method for printing p values and confidence intervals.

Usage

```
## S4 method for signature 'StatsOutput'  
printTable(object)
```

Arguments

object An object of class "StatsOutput".

RMS_dat

ECLS-K (2011) Sample Dataset for Demonstration

Description

A sample dataset extracted from the public-use Early Childhood Longitudinal Study, Kindergarten Class of 2010-11 (ECLS-K:2011) collected by the National Center for Education Statistics (NCES). This dataset is NOT a posting of the original data, and it has been processed and formatted for use in demonstration purposes within this package. For access to the original data, please visit the NCES data products page at <https://nces.ed.gov/ecls/dataproducts.asp>.

Usage

RMS_dat

Format

A data frame with 500 rows and 49 variables:

ID Identification number.

R1, R2, R3, R4, R5, R6, R7, R8, R9 Reading scores from 9 study waves.

M1, M2, M3, M4, M5, M6, M7, M8, M9 Math scores from 9 study waves.

S2, S3, S4, S5, S6, S7, S8, S9 Science scores from 8 study waves (starting from the second study wave).

T1, T2, T3, T4, T5, T6, T7, T8, T9 Children's age-in-month at 9 study waves.

SEX Sex of the child.

RACE Race of the child.

LOCALE Locale of the child's school.

INCOME Family income.

SCHOOL_TYPE Type of the child's school.

Approach_to_Learning Teacher's rating on the child's approach to learning.

Self_control Teacher's rating on the child's self-control.

Interpersonal Teacher's rating on the child's interpersonal skills.

External_prob_Behavior Teacher's rating on the child's external problem behaviors.

Internal_prob_Behavior Teacher's rating on the child's internal problem behaviors.

Attention_focus Teacher's rating on the child's attention focus.

Inhibitory_Ctrl Teacher's rating on the child's inhibitory control.

EDU Highest education level between the child's parents.

Details

The ECLS-K:2011 offers a comprehensive and detailed set of information about children's early life experiences, focusing on children's health, development, education, and experiences in the years leading up to kindergarten.

The sample dataset included in this package is used for demonstrating the functionality of the package's functions and it does not include survey weights. In real analysis, the complex survey weights provided by NCES should be utilized appropriately, for instance, as done in R packages such as `lavaan.survey` or `EdSurvey` if not using SEM.

Please note that this data must not be used to attempt to identify respondents. For detailed documentation and proper usage of the ECLS-K:2011 data, please refer to the original source at the National Center for Education Statistics (NCES) website: <https://nces.ed.gov/>.

Source

<https://nces.ed.gov/ecls/dataproducts.asp>

`show, figOutput-method` *S4 Method for displaying figures.*

Description

Method to display a summary of the `figOutput` object when printed.

Usage

```
## S4 method for signature 'figOutput'
show(object)
```

Arguments

`object` An object of class "figOutput".

`StatsOutput-class` *S4 Class for p values and confidence intervals (when specified).*

Description

S4 Class for the output structure for the `getEstimateStats()` function.

Slots

`wald` A data frame for p values and Wald confidence intervals (when specified).

`likelihood` A data frame for Likelihood confidence intervals (when specified).

`bootstrap` A data frame for Bootstrap confidence intervals (when specified).

Index

* datasets

RMS_dat, [45](#)

figOutput-class, [2](#)

FSOutput-class, [3](#)

getEstimateStats, [3](#)

getFigure, [5](#)

getIndFS, [8](#)

getLatentKappa, [10](#)

getLCSM, [12](#)

getLGCM, [14](#)

getLRT, [17](#)

getMediation, [18](#)

getMGM, [22](#)

getMGroup, [25](#)

getMIX, [29](#)

getPosterior, [33](#)

getSummary, [35](#)

getTVcmodel, [37](#)

KappaOutput-class, [40](#)

ModelSummary, [41](#)

ModelSummary, myMxOutput-method, [41](#)

myMxOutput-class, [41](#)

postOutput-class, [42](#)

printTable, [42](#)

printTable, FSOutput-method, [42](#)

printTable, KappaOutput-method, [43](#)

printTable, myMxOutput-method, [43](#)

printTable, postOutput-method, [44](#)

printTable, StatsOutput-method, [44](#)

RMS_dat, [45](#)

show, figOutput-method, [46](#)

StatsOutput-class, [46](#)