Package 'modsem'

June 2, 2024

Type Package

Title Latent Interaction (and Moderation) Analysis in Structural Equation Models (SEM)

Version 0.1.4

Maintainer Kjell Solem Slupphaug <slupphaugkjell@gmail.com>

Description Estimation of interaction (i.e., moderation) effects between latent variables in structural equation models (SEM).
The supported methods are:
The constrained approach (Algina & Moulder, 2001).
The unconstrained approach (Marsh et al., 2004).
The residual centering approach (Little et al., 2006).
The double centering approach (Lin et al., 2010).
The latent moderated structural equations (LMS) approach (Klein & Moosbrugger, 2000).
The quasi-maximum likelihood (QML) approach (Klein & Muthén, 2007) (temporarily unavailable)
The constrained- unconstrained, residual- and double centering- approaches are estimated via 'lavaan' (Rosseel, 2012), whilst the LMS- and QML- approaches are estimated via by ModSEM it self. Alternatively model can be estimated via 'Mplus' (Muthén & Muthén, 1998-2017).
References:
Algina, J., & Moulder, B. C. (2001).
<doi:10.1207/S15328007SEM0801_3>.
``A note on estimating the Jöreskog-Yang model for latent variable interaction using 'LISREL' 8.3."
Klein, A., & Moosbrugger, H. (2000).
<doi:10.1007/BF02296338>.
``Maximum likelihood estimation of latent interaction effects with the LMS method."
Klein, A. G., & Muthén, B. O. (2007).
<doi:10.1080/00273170701710205>.
``Quasi-maximum likelihood estimation of structural equation models with multiple interaction and quadratic effects."
Lin, G. C., Wen, Z., Marsh, H. W., & Lin, H. S. (2010).
<doi:10.1080/10705511.2010.488999>.
``Structural equation models of latent interactions: Clarification of orthogonalizing and double-mean-centering strategies."

Little, T. D., Bovaird, J. A., & Widaman, K. F. (2006).
<doi:10.1207/s15328007sem1304_1>.
``On the merits of orthogonalizing powered and product terms: Implications for modeling interactions among latent variables."
Marsh, H. W., Wen, Z., & Hau, K. T. (2004).
<doi:10.1037/1082-989X.9.3.275>.
``Structural equation models of latent interactions: evaluation of alternative estimation strategies and indicator construction."
Muthén, L.K. and Muthén, B.O. (1998-2017).
``'Mplus' User's Guide. Eighth Edition."
<https://www.statmodel.com/>.
Rosseel Y (2012).
<doi:10.18637/jss.v048.i02>.
``'lavaan': An R Package for Structural Equation Modeling."

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

LinkingTo Rcpp, RcppArmadillo

Imports Rcpp, purrr, stringr, lavaan, rlang, MplusAutomation, nlme, R6, dplyr, mvnfast, stats, gaussquad, mvtnorm, ggplot2

Depends R (>= 3.50)

URL https://github.com/Kss2k/modsem

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Kjell Solem Slupphaug [aut, cre]
(<https://orcid.org/0009-0005-8324-2834>)

Repository CRAN

Date/Publication 2024-06-02 10:20:02 UTC

R topics documented:

`jordan` *Jordan subset of PISA 2006 data*

Description

The data stem from the large-scale assessment study PISA 2006 (Organisation for Economic Co-Operation and Development, 2009) where competencies of 15-year-old students in reading, mathematics, and science are assessed using nationally representative samples in 3-year cycles. In this eacademicample, data from the student background questionnaire from the Jordan sample of PISA 2006 were used. Only data of students with complete responses to all 15 items (N = 6,038) were considered.

Format

A data frame of fifteen variables and 6,038 observations:

enjoy1 indicator for enjoyment of science, item ST16Q01: I generally have fun when I am learning <broad science> topics.

enjoy2 indicator for enjoyment of science, item ST16Q02: I like reading about <broad science>.

enjoy3 indicator for enjoyment of science, item ST16Q03: I am happy doing <broad science> problems.

enjoy4 indicator for enjoyment of science, item ST16Q04: I enjoy acquiring new knowledge in <broad science>.

enjoy5 indicator for enjoyment of science, item ST16Q05: I am interested in learning about <broad science>.

academic1 indicator for academic self-concept in science, item ST37Q01: I can easily understand new ideas in <school science>.

academic2 indicator for academic self-concept in science, item ST37Q02: Learning advanced <school science> topics would be easy for me.

academic3 indicator for academic self-concept in science, item ST37Q03: I can usually give good answers to <test questions> on <school science> topics.

academic4 indicator for academic self-concept in science, item ST37Q04: I learn <school science> topics quickly.

academic5 indicator for academic self-concept in science, item ST37Q05: <School science> topics are easy for me.

academic6 indicator for academic self-concept in science, item ST37Q06: When I am being taught <school science>, I can understand the concepts very well.

career1 indicator for career aspirations in science, item ST29Q01: I would like to work in a career involving <broad science>.

career2 indicator for career aspirations in science, item ST29Q02: I would like to study <broad science> after <secondary school>.

career3 indicator for career aspirations in science, item ST29Q03: I would like to spend my life doing advanced <broad science>.

career4 indicator for career aspirations in science, item ST29Q04: I would like to work on <broad science> projects as an adult.

Source

This version of the dataset, as well as the description was gathered from the documentation of the 'nlsem' package (https://cran.r-project.org/package=nlsem), where the only difference is that the names of the variables were changed

Originally the dataset was gathered by the Organisation for Economic Co-Operation and Development (2009). Pisa 2006: Science competencies for tomorrow's world (Tech. Rep.). Paris, France. Obtained from: https://www.oecd.org/pisa/pisaproducts/database-pisa2006.htm

---

`modsem` *Interaction between latent variables*

Description

modsem is a function for estimating interaction effects between latent variables, in structural equation models (SEM's). Methods for estimating interaction effects in SEM's can basically be split into two frameworks: 1. Product Indicator based approaches ("dblcent", "rca", "uca", "ca", "pind"), and 2. Distributionally based approaches ("lms", "qml"). For the product indicator based approaces, modsem() is essentially a just a fancy wrapper for lavaan::sem() which generates the necessary syntax, and variables for the estimation of models with latent product indicators. The distributionally based approaches are implemented in seperately, and are are not estimated using lavaan::sem(), but rather using custom functions (largely) written in C++ for performance reasons. For greater control, it is advised that you use one of the sub-functions (modsem_pi, modsem_lms_qml, modsem_mplus) directly, as passing additional arguments to them via modsem() can lead to unexpected behavior.

Usage

```
modsem(
  modelSyntax = NULL,
  data = NULL,
  method = "dblcent",
  standardize = FALSE,
  center = FALSE,
  ...
)
```

Arguments

modelSyntax	lavaan syntax
data	dataframe
method	method to use: "rca" = residual centering approach (passed to lavaan), "uca" = unconstrained approach (passed to lavaan), "dblcent" = double centering approach (passed to lavaan), "pind" = prod ind approach, with no constraints or centering (passed to lavaan), "lms" = laten model structural equations (not passed to lavaan). "qml" = quasi maximum likelihood estimation of laten model structural equations (not passed to lavaan). "custom" = use parameters specified in the function call (passed to lavaan)
standardize	should data be scaled before fitting model
center	should data be centered before fitting model
...	arguments passed to other functions depending on method (see modsem_pi, modsem_lms_qml, and modsem_mplus)

Value

ModSEM object

Examples

```
library(modsem)
# For more examples check README and/or GitHub.
# One interaction
m1 <- '
  # Outer Model
  X =~ x1 + x2 +x3
  Y =~ y1 + y2 + y3
  Z =~ z1 + z2 + z3

  # Inner model
  Y ~ X + Z + X:Z
'

# Double centering approach
est1 <- modsem(m1, oneInt)
summary(est1)

## Not run:
# The Constrained Approach
est1Constrained <- modsem(m1, oneInt, method = "ca")
summary(est1Constrained)

# LMS approach
est1LMS <- modsem(m1, oneInt, method = "lms")
summary(est1LMS)

# QML approach
est1QML <- modsem(m1, oneInt, method = "qml")
```

```
summary(est1QML)


## End(Not run)

# Theory Of Planned Behavior
tpb <- '
# Outer Model (Based on Hagger et al., 2007)
  LATT =~ att1 + att2 + att3 + att4 + att5
  LSN =~ sn1 + sn2
  LPBC =~ pbc1 + pbc2 + pbc3
  LINT =~ int1 + int2 + int3
  LBEH =~ b1 + b2

# Inner Model (Based on Steinmetz et al., 2011)
  # Covariances
  LATT ~~ LSN + LPBC
  LPBC ~~ LSN
  # Causal Relationsships
  LINT ~ LATT + LSN + LPBC
  LBEH ~ LINT + LPBC
  LBEH ~ LINT:LPBC
'

# double centering approach
estTpb <- modsem(tpb, data = TPB)
summary(estTpb)

## Not run:
# The Constrained Approach
estTpbConstrained <- modsem(tpb, data = TPB, method = "ca")
summary(estTpbConstrained)

# LMS approach
estTpbLMS <- modsem(tpb, data = TPB, method = "lms")
summary(estTpbLMS)

## End(Not run)
```

`modsemify` *Generate parameter table for lavaan syntax*

Description

Generate parameter table for lavaan syntax

Usage

```
modsemify(syntax)
```

Arguments

syntax	model syntax

Value

data.frame with columns lhs, op, rhs, mod

Examples

```
library(modsem)
m1 <- '
  # Outer Model
  X =~ x1 + x2 +x3
  Y =~ y1 + y2 + y3
  Z =~ z1 + z2 + z3

  # Inner model
  Y ~ X + Z + X:Z
'
modsemify(m1)
```

---

modsem_lms_qml *Interaction between latent variables using lms and qml approaches*

Description

modsem_lms_qml is a function for estimating interaction effects between latent variables, in structural equation models (SEMs), using product indicators. Methods for estimating interaction effects in SEM's can basically be split into two frameworks: 1. Product Indicator based approaches ("dblcent", "rca", "uca", "ca", "pind"), and 2. Distributionally based approaches ("lms", "qml"). modsem_lms_qml() is essentially a just a fancy wrapper for lavaan::sem() which generates the necessary syntax, and variables for the estimation of models with latent product indicators.

Usage

```
modsem_lms_qml(
  modelSyntax = NULL,
  data = NULL,
  method = "lms",
  verbose = FALSE,
  optimize = TRUE,
  nodes = 16,
  convergence = 0.01,
  center = FALSE,
  standardize = FALSE,
  ...
)
```

Arguments

modelSyntax	lavaan syntax
data	dataframe
method	method to use: "lms" = laten model structural equations (not passed to lavaan). "qml" = quasi maximum likelihood estimation of laten model structural equations (not passed to lavaan).
verbose	should estimation progress be shown
optimize	should starting parameters be optimized
nodes	number of quadrature nodes (points of integration) used in lms
convergence	convergence criterion
center	should data be centered before fitting model
standardize	should data be scaled before fitting model
...	arguments passed to other functions

Value

modsem_lms or modsem_qml object

Examples

```
library(modsem)
# For more examples check README and/or GitHub.
# One interaction
m1 <- '
  # Outer Model
  X =~ x1 + x2 +x3
  Y =~ y1 + y2 + y3
  Z =~ z1 + z2 + z3

  # Inner model
  Y ~ X + Z + X:Z
'

## Not run:
# QML Approach
est1 <- modsem_lms_qml(m1, oneInt, method = "qml")
summary(est1)


# Theory Of Planned Behavior
tpb <- '
# Outer Model (Based on Hagger et al., 2007)
  LATT =~ att1 + att2 + att3 + att4 + att5
  LSN =~ sn1 + sn2
  LPBC =~ pbc1 + pbc2 + pbc3
  LINT =~ int1 + int2 + int3
  LBEH =~ b1 + b2
```

```
# Inner Model (Based on Steinmetz et al., 2011)
  # Covariances
  LATT ~~ LSN + LPBC
  LPBC ~~ LSN
  # Causal Relationsships
  LINT ~ LATT + LSN + LPBC
  LBEH ~ LINT + LPBC
  LBEH ~ LINT:LPBC
'

# lms approach
estTpb <- modsem_lms_qml(tpb, data = TPB, method = lms)
summary(estTpb)

## End(Not run)
```

`modsem_mplus` *Estimation latent interactions through mplus*

Description

Estimation latent interactions through mplus

Usage

```
modsem_mplus(
  modelSyntax,
  data,
  estimator = "ml",
  type = "random",
  algorithm = "integration",
  process = "8",
  ...
)
```

Arguments

`modelSyntax`	lavaan/modsem syntax
`data`	dataset
`estimator`	estimator argument passed to mplus
`type`	type argument passed to mplus
`algorithm`	algorithm argument passed to mplus
`process`	process argument passed to mplus
`...`	arguments passed to other functions

Value

modsem_mplus object

Examples

```
# Theory Of Planned Behavior
tpb <- '
# Outer Model (Based on Hagger et al., 2007)
  LATT =~ att1 + att2 + att3 + att4 + att5
  LSN =~ sn1 + sn2
  LPBC =~ pbc1 + pbc2 + pbc3
  LINT =~ int1 + int2 + int3
  LBEH =~ b1 + b2

# Inner Model (Based on Steinmetz et al., 2011)
  # Covariances
  LATT ~~ LSN + LPBC
  LPBC ~~ LSN
  # Causal Relationssships
  LINT ~ LATT + LSN + LPBC
  LBEH ~ LINT + LPBC
  LBEH ~ LINT:LPBC
'

## Not run:
estTpbMplus <- modsem_mplus(tpb, data = TPB)
summary(estTpbLMS)

## End(Not run)
```

---

modsem_pi *Interaction between latent variables using product indicators*

Description

modsem_pi is a function for estimating interaction effects between latent variables, in structural equation models (SEMs), using product indicators. Methods for estimating interaction effects in SEM's can basically be split into two frameworks: 1. Product Indicator based approaches ("dblcent", "rca", "uca", "ca", "pind"), and 2. Distributionally based approaches ("lms", "qml"). modsem_pi() is essentially a just a fancy wrapper for lavaan::sem() which generates the necessary syntax, and variables for the estimation of models with latent product indicators.

Usage

```
modsem_pi(
  modelSyntax = NULL,
  data = NULL,
  method = "dblcent",
```

```
  match = FALSE,
  standardizeData = FALSE,
  centerData = FALSE,
  firstLoadingFixed = TRUE,
  centerBefore = NULL,
  centerAfter = NULL,
  residualsProds = NULL,
  residualCovSyntax = NULL,
  constrainedProdMean = NULL,
  constrainedLoadings = NULL,
  constrainedVar = NULL,
  constrainedResCovMethod = NULL,
  auto.scale = "none",
  auto.center = "none",
  estimator = "ML",
  group = NULL,
  run = TRUE,
  ...
)
```

Arguments

modelSyntax	lavaan syntax
data	dataframe
method	method to use: "rca" = residual centering approach (passed to lavaan), "uca" = unconstrained approach (passed to lavaan), "dblcent" = double centering approach (passed to lavaan), "pind" = prod ind approach, with no constraints or centering (passed to lavaan), "custom" = use parameters specified in the function call (passed to lavaan)
match	should the product indicators be created by using the match-strategy
standardizeData	should data be scaled before fitting model
centerData	should data be centered before fitting model
firstLoadingFixed	Sould the first factorloading in the latent prod be fixed to one?
centerBefore	should inds in prods be centered before computing prods (overwritten by method, if method != NULL)
centerAfter	should ind prods be centered after they have been computed?
residualsProds	should ind prods be centered using residuals (overwritten by method, if method != NULL)
residualCovSyntax	should syntax for residual covariances be produced (overwritten by method, if method != NULL)
constrainedProdMean	should syntax prod mean be produced (overwritten by method, if method != NULL)

constrainedLoadings	should syntax for constrained loadings be produced (overwritten by method, if method != NULL)
constrainedVar	should syntax for constrained variances be produced (overwritten by method, if method != NULL)
constrainedResCovMethod	method for constraining residual covariances
auto.scale	methods which should be scaled automatically (usually not useful)
auto.center	methods which should be centered automatically (usually not useful)
estimator	estimator to use in lavaan
group	group variable for multigroup analysis
run	should the model be run via lavaan, if FALSE only modified syntax and data is returned
...	arguments passed to other functions, e.g,. lavaan

Value

ModSEM object

Examples

```
library(modsem)
# For more examples check README and/or GitHub.
# One interaction
m1 <- '
  # Outer Model
  X =~ x1 + x2 +x3
  Y =~ y1 + y2 + y3
  Z =~ z1 + z2 + z3

  # Inner model
  Y ~ X + Z + X:Z
'

# Double centering approach
est1 <- modsem_pi(m1, oneInt)
summary(est1)

## Not run:
# The Constrained Approach
est1Constrained <- modsem_pi(m1, oneInt, method = "ca")
summary(est1Constrained)

## End(Not run)

# Theory Of Planned Behavior
tpb <- '
# Outer Model (Based on Hagger et al., 2007)
  LATT =~ att1 + att2 + att3 + att4 + att5
```

```
  LSN =~ sn1 + sn2
  LPBC =~ pbc1 + pbc2 + pbc3
  LINT =~ int1 + int2 + int3
  LBEH =~ b1 + b2

# Inner Model (Based on Steinmetz et al., 2011)
  # Covariances
  LATT ~~ LSN + LPBC
  LPBC ~~ LSN
  # Causal Relationsships
  LINT ~ LATT + LSN + LPBC
  LBEH ~ LINT + LPBC
  LBEH ~ LINT:LPBC
'

# double centering approach
estTpb <- modsem_pi(tpb, data = TPB)
summary(estTpb)

## Not run:
# The Constrained Approach
estTpbConstrained <- modsem_pi(tpb, data = TPB, method = "ca")
summary(estTpbConstrained)

## End(Not run)
```

`multiplyIndicatorsCpp` *Multiply indicators*

Description

Multiply indicators

Usage

```
multiplyIndicatorsCpp(df)
```

Arguments

`df`	A data DataFrame

Value

A NumericVector

oneInt *oneInt*

Description

A simulated dataset with one interaction effect

parameter_estimates *Extract parameterEstimates from an estimated model*

Description

Extract parameterEstimates from an estimated model

Usage

```
parameter_estimates(object, ...)
```

Arguments

object	An object of class ‘modsem_pi‘, ‘modsem_lms_qml‘, or ‘modsem_mplus‘
...	Additional arguments passed to other functions

plot_interaction *Plot Interaction Effects*

Description

Plot Interaction Effects

Usage

```
plot_interaction(
  x,
  z,
  y,
  xz,
  vals_x = seq(-3, 3, 0.001),
  vals_z,
  model,
  alpha_se = 0.15,
  ...
)
```

Arguments

x	The name of the variable on the x-axis
z	The name of the moderator variable
y	The name of the outcome variable
xz	The name of the interaction term
vals_x	The values of the x variable to plot, the more values the smoother the std.error-area will be
vals_z	The values of the moderator variable to plot. A seperate regression line ("y ~ x | z") will be plotted for each value of the moderator variable
model	An object of class 'modsem_pi', 'modsem_lms_qml', or 'modsem_mplus'
alpha_se	The alpha level for the std.error area
...	Additional arguments passed to other functions

Value

A ggplot object

Examples

```
library(modsem)
## Not run:
m1 <- "
# Outer Model
  X =~ x1
  X =~ x2 + x3
  Z =~ z1 + z2 + z3
  Y =~ y1 + y2 + y3

# Inner model
  Y ~ X + Z + X:Z
"
est1 <- modsem(m1, data = oneInt)
plot_interaction("X", "Z", "Y", "X:Z", -3:3, c(-0.2, 0), est1)

tpb <- "
# Outer Model (Based on Hagger et al., 2007)
  ATT =~ att1 + att2 + att3 + att4 + att5
  LSN =~ sn1 + sn2
  PBC =~ pbc1 + pbc2 + pbc3
  INT =~ int1 + int2 + int3
  BEH =~ b1 + b2

# Inner Model (Based on Steinmetz et al., 2011)
  # Causal Relationsships
  INT ~ ATT + LSN + PBC
  BEH ~ INT + PBC
  # BEH ~ ATT:PBC
  BEH ~ PBC:INT
```

```
  # BEH ~ PBC:PBC
"

est2 <- modsem(tpb, TPB, method = "lms")
plot_interaction(x = "INT", z = "PBC", y = "BEH", xz = "PBC:INT",
                 vals_z = c(-0.5, 0.5), model = est2)

## End(Not run)
```

---

summary.modsem_lms *summary.modsem_lms*

Description

summary.modsem_lms

summary.modsem_qml

summary.modsem_mplus

summary.modsem_pi

Usage

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

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

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

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

Arguments

object	modsem object to summarized
...	arguments passed to lavaan::summary()

summaryLmsAndQml *summaryLmsAndQml*

Description

summaryLmsAndQml

Usage

```
summaryLmsAndQml(
  object,
  H0 = TRUE,
  verbose = TRUE,
  r.squared = TRUE,
  digits = 3,
  ...
)
```

Arguments

object	modsem object estimated with lms or qml
H0	should a null model be estimated (used for comparison)
verbose	print progress for the estimation of null model
r.squared	calculate R-squared
digits	number of digits to print
...	additional arguments

TPB *TPB*

Description

A simulated dataset based on the Theory of Planned Behaviora

tracePath — *Estimate formulas for (co-)variance paths using Wright's path tracing rules*

Description

This function estimates the path from x to y using the path tracing rules, note that it only works with structural parameters, so "=~" are ignored. If you you want to use the measurement model, it should work if you replace it "=~" with "~" in the mod column of pt.

Usage

```
tracePath(pt, x, y, parenthesis = TRUE, ...)
```

Arguments

pt	A data frame with columns lhs, op, rhs, and mod, from modsemify(syntax)
x	source variable
y	destination variable
parenthesis	if TRUE, the output will be enclosed in parenthesis
...	additional arguments passed to tracePath

Value

A string with the estimated path (simplified if possible)

Examples

```
library(modsem)
m1 <- '
  # Outer Model
  X =~ x1 + x2 +x3
  Y =~ y1 + y2 + y3
  Z =~ z1 + z2 + z3

  # Inner model
  Y ~ X + Z + X:Z
'
pt <- modsemify(m1)
tracePath(pt, "Y", "Y") # variance of Y
```

tripleInt — *tripleInt*

Description

A simulated dataset with three interaction effects

twoInt *twoInt*

Description

A simulated dataset with two interaction effects

Index