Package ‘gesca’

July 22, 2025

Type Package

Title Generalized Structured Component Analysis Structural Equation Modeling

Version 1.0.5

Description Implementing generalized structured component analysis (GSCA) and its basic extensions, including constrained single and multiple group analysis, and second order latent variable modeling. For a comprehensive overview of GSCA, see Hwang & Takane (2014, ISBN: 9780367738754).

License GPL (>= 2)

Encoding UTF-8

Language en-US

URL https://quantmm.github.io/gesca/

NeedsCompilation no

Author Heungsun Hwang [aut] (ORCID: <https://orcid.org/0000-0002-5057-7479>), Sunmee Kim [aut, cre] (ORCID: <https://orcid.org/0000-0003-3723-9974>), Sungyoung Lee [aut] (ORCID: <https://orcid.org/0000-0003-3458-1440>), Taesung Park [aut] (ORCID: <https://orcid.org/0000-0002-8294-590X>)

Maintainer Sunmee Kim <sunmee.kim@umanitoba.ca>

Repository CRAN

Date/Publication 2024-10-22 08:20:02 UTC

Contents

effectmeasures *Total and Indirect Effects of Variables*

Description

In the user-specified structural equation model, this function calculates the total and indirect effects of (1) latent variables on other latent variables in the structural model and (2) latent variables on their indicators in the measurement model.

Usage

```
effectmeasures(object)
```

Arguments

object	An object of class. This can be created via the gesca.run function.

Details

The bootstrapped standard errors will be provided for all total and indirect effects. The user can specify the number of bootstrap samples via the gesca.run function. If the number of bootstrap samples is equal to 0, these standard errors will not be provided.

Value

Four matrices whose row and column names are the same as the names of indicators and latent variables.

References

Hwang, H., & Takane, Y. (2014). Generalized Structured Component Analysis: A Component-Based Approach to Structural Equation Modeling (pp.119-123). Boca Raton, FL: Chapman & Hall/CRC Press.

See Also

gesca.run

Examples

```
library(gesca)
data(gesca.rick2) # Organizational identification example of Bagozzi

# Model specification
myModel <- "
# Measurement model
OP =~ cei1 + cei2 + cei3
OI =~ ma1 + ma2 + ma3
```

```
AC_J =~ orgcmt1 + orgcmt2 + orgcmt3
AC_L =~ orgcmt5 + orgcmt6 + orgcmt8

# Structural model
OI ~ OP
AC_J ~ OI
AC_L ~ OI
"

# Run a multiple-group GSCA with the grouping variable gender:
GSCA.group <- gesca.run(myModel, gesca.rick2, group.name = "gender")
summary(GSCA.group) # default output
effectmeasures(GSCA.group) # total and indirect effects per group
GSCA.group.nbt0 <- gesca.run(myModel, gesca.rick2, group.name = "gender", nbt = 0)
effectmeasures(GSCA.group.nbt0) # no bootstrapped standard error of parameters

# Second-order latent model with formative indicators
SndForm <- "
# Measurement model
OP(0) =~ cei1 + cei2 + cei3 + cei4 + cei5 + cei6 + cei7 + cei8
OI(0) =~ ma1 + ma2 + ma3 + ma4 + ma5 + ma6
AC_J(0) =~ orgcmt1 + orgcmt2 + orgcmt3 + orgcmt7
AC_L(0) =~ orgcmt5 + orgcmt6 + orgcmt8

# Second-order latents
AC(0) =: AC_J + AC_L

# Structural model
OI ~ OP
AC ~ OI
"

SndFormFit <- gesca.run(SndForm, gesca.rick2, "gender")
summary(SndFormFit)
effectmeasures(SndFormFit)
# Total/Indirect effects of latent variables on indicators
# will not be provided and instead, be reported as "NULL".
```

`fitmeasures` *Indices for Assessment of Model Fit*

Description

This function provides six model fit indices (FIT, Adjusted FIT, GFI, Standardized Root Mean Square, FIT_M, and FIT_S) for the user-specified model.

Usage

```
fitmeasures(object)
```

Arguments

object An object of class. This can be created via the gesca.run function.

Details

FIT indicates how much variance of both indicators and latent variables is explained by the user-specified model. AFIT takes into account the number of parameters when calculating the explained variance. GFI and SRMR measure the closeness between the sample covariances and the covariances reproduced by model parameter estimates. In GSCA, separate model fit measures for the measurement and structural models are available: FIT_M and FIT_S. FIT_M assesses how much variance of indicators is explained by a measurement model, whereas FIT_S calculates how much variance of latent variables is accounted for by a structural model. All six indices are given with their bootstrapped standard errors and confidence intervals. When the number of bootstrap samples is equal to 0, the standard errors will not be reported. The user can specify the number of bootstrap samples via the gesca.run function.

Value

Numeric values of six model fit indices: the numbers of parameters and bootstrap samples, and six fit indices with their bootstrapped standard errors and confidence intervals.

References

Hwang, H., & Takane, Y. (2014). Generalized structured component analysis: A Component-Based Approach to Structural Equation Modeling (pp.26-32). Boca Raton, FL: Chapman & Hall/CRC Press.

See Also

gesca.run

Examples

```
library(gesca)
data(gesca.rick2) # Organizational identification example of Bagozzi

# Model specification
myModel <- "
# Measurement model
OP =~ cei1 + cei2 + cei3
OI =~ ma1 + ma2 + ma3
AC_J =~ orgcmt1 + orgcmt2 + orgcmt3
AC_L =~ orgcmt5 + orgcmt6 + orgcmt8

# Structural model
OI ~ OP
AC_J ~ OI
AC_L ~ OI
"
```

```
# Run a multiple-group GSCA with the grouping variable gender:
GSCA.group <- gesca.run(myModel, gesca.rick2, group.name = "gender", nbt=50)
summary(GSCA.group) # default output shows model fit indices as well, but
fitmeasures(GSCA.group) # this gives bootstrapped std errors and CI
```

`gesca.rick2` *Bagozzi's organizational identification example*

Description

This example is part of the organizational identification data used in Bergami and Bagozzi (2000). The data are used in Hwang and Takane (2014) (see Chapter 3). The data contain a sample of 305 employees (male = 157 and female = 148) and 22 indicators.

Usage

```
data(gesca.rick2)
```

Format

A data frame with 305 observations on the following 22 variables.

`gender` grouping variable - gender (1 = male, 2 = female)
`cei1` First indicator associated with the first latent variable - Organizational Prestige
`cei2` Second indicator associated with the first latent variable - Organizational Prestige
`cei3` Third indicator associated with the first latent variable - Organizational Prestige
`cei4` Fourth indicator associated with the first latent variable - Organizational Prestige
`cei5` Fifth indicator associated with the first latent variable - Organizational Prestige
`cei6` Sixth indicator associated with the first latent variable - Organizational Prestige
`cei7` Seventh indicator associated with the first latent variable - Organizational Prestige
`cei8` Eighth indicator associated with the first latent variable - Organizational Prestige
`ma1` First indicator associated with the second latent variable - Organizational Identification
`ma2` Second indicator associated with the second latent variable - Organizational Identification
`ma3` Third indicator associated with the second latent variable - Organizational Identification
`ma4` Fourth indicator associated with the second latent variable - Organizational Identification
`ma5` Fifth indicator associated with the second latent variable - Organizational Identification
`ma6` Sixth indicator associated with the second latent variable - Organizational Identification
`orgcmt1` First indicator associated with the third latent variable - Affective Commitment-Joy
`orgcmt2` Second indicator associated with the third latent variable - Affective Commitment-Joy
`orgcmt3` Third indicator associated with the third latent variable - Affective Commitment-Joy
`orgcmt7` Fourth indicator associated with the third latent variable - Affective Commitment-Joy
`orgcmt5` First indicator associated with the fourth latent variable - Affective Commitment-Love
`orgcmt6` Second indicator associated with the fourth latent variable - Affective Commitment-Love
`orgcmt8` Third indicator associated with the fourth latent variable - Affective Commitment-Love

References

Bergami, M., & Bagozzi, R. P. (2000). Self-categorization, affective commitment and group self-esteem as distinct aspects of social identity in the organization. *British Journal of Social Psychology*, *39*(4), 555-577.

Hwang, H., & Takane, Y. (2014). Generalized Structured Component Analysis: A Component-Based Approach to Structural Equation Modeling (pp.112-119). Boca Raton, FL: Chapman & Hall/CRC Press.

Examples

```
data(gesca.rick2)
str(gesca.rick2)
```

`gesca.run` *Fit Structural Equation Models using Generalized Structural Component Analysis*

Description

Fit a generalized structured component analysis (GSCA) model. gesca 1.0 can handle various extensions of GSCA, including imposing constraints on parameters, second-order latent variable modeling, multiple-group analysis, the calculation of total and indirect effects, and missing value imputation.

Usage

```
gesca.run(myModel, data, group.name = NULL, group.equal = NULL,
          nbt = 100, itmax = 100, ceps = 1e-5,
          moption = 0, missingvalue = -9999)
```

Arguments

`myModel` A simple description of the user-specified model using the gesca model syntax. The model can be described by strings or multiple lines of formulas. See Details and Examples below.

`data` A data matrix or frame including the observed variables used in `myModel`.

`group.name` A character string only used in a multiple group analysis. A variable name in the data frame which defines the group membership of each case. The user can specify which variable is used to provide group memberships for a multiple group analysis. If `NULL` (the default), a single-group GSCA will be run.

`group.equal` A vector of character strings only used in a multiple group analysis. It can be one or both of the following: `"loadings"` or `"paths"`, specifying the pattern of equality constraints across multiple groups. The default is set to `NULL` (the default), assuming that all parameters are freely estimated in each group. See Details and Examples below.

`nbt` Denote the number of bootstrap samples. In gesca 1.0, the bootstrap method is employed to obtain the standard errors of parameter estimates. By default, `nbt` is set to 100, but any non-negative numeric value can be used. If the number of bootstrap samples is equal to 0, only the values of parameter estimates obtained from the original dataset will be reported without their standard errors and CIs.

`itmax` The maximum number of iterations for the alternating least-squares (ALS) algorithm. The default value is set to 100.

`ceps` Convergence threshold for the ALS algorithm. The default is set to 0.00001.

`moption` Options of handling missing data. By default, it is assumed that no value is missing in all variables. gesca 1.0 currently offers three options for handling missing data and the choice should be one of the following: list-wise deletion (`moption = 1`), mean substitution (`moption = 2`), and least-squares imputation (`moption = 3`).

`missingvalue` A missing value identifier. Only used if `moption` is specified. By default, every element of `-9999` in `data` is defined as a missing value. Users can use any numeric or integer value to specify missing values in data. See Examples below.

Details

General Overview of the gesca Model Syntax:

In gesca 1.0, the user can write one or more formula lines to specify a structural equation model. This is the gesca model syntax for the `gesca.run` function. The model syntax consists of character strings or multiple lines of formulas as follows:

1) To specify the relationships between indicators and latent variables, the user can use the `=~` operator. For example, the model syntax `Y =~ X1 + X2 + X3` means the latent variable `Y` is related to the indicators `X1`, `X2`, and `X3`.

2) The relationships among latent variables can be expressed with the `~` operator. For example, the syntax `Y1 ~ Y2 + Y3` shows that `Y1` is regressed on `Y2` and `Y3`.

3) The user can consider a second-order latent variable that is linked to a set of first-order latent variables. This can be described by the `=:` operator. For example, the syntax `H1 =: Y1+Y2+Y3` indicates that a second-order latent variable, `H1`, is related to its lower-order latent variables `Y1`, `Y2`, and `Y3`.

Importantly, the variable names in the gesca model syntax should correspond to the variable names in the data frame. If the option `group.name` is used for a multiple-group GSCA, the same model structure is fitted across groups by default.

Constraining Parameters in the gesca Model Syntax:

If the user wants to fix or constrain parameters to constants, it can be done in the gesca model syntax by adding values or labels followed by the `*` operator. For example, in the following model syntax, the path coefficients of `L1` regressed on `L4` and of `L4` regressed on `L5` are fixed at 0.01 and 0.6, respectively:

```
myModel <- "
 L1 =~ z1 + z2 + z3
 L2 =~ z4 + z5 + z6
 L3 =~ z7 + z8
```

```
 L4 =~ z9 + z10 + z11
 L5 =~ z12
 L2 ~ L1
 L3 ~ L1 + L2
 L4 ~ 0.01*L1 + L2 + L3
 L5 ~ 0.6*L4
"
```

The above model is identical to the following model using labels (instead of values):

```
myModel <- "
 L1 =~ z1 + z2 + z3
 L2 =~ z4 + z5 + z6
 L3 =~ z7 + z8
 L4 =~ z9 + z10 + z11
 L5 =~ z12
 L2 ~ L1
 L3 ~ L1 + L2
 L4 ~ c1*L1 + L2 + L3
 L5 ~ c2*L4
 c1 == 0.01
 c2 == 0.6
"
```

Constraining Parameters in a Multiple-Group GSCA:

Any across-group equality constraints can be imposed on parameters in the same way as above. In addition, the `group.equal` option can be used to constrain all loadings and/or all path coefficients across groups simultaneously. To demonstrate the imposition of across-group equality constraints on all loadings and path coefficients, the user can simply input `group.equal = c("loadings", "paths")`, then the `gesca.run` function estimates a single set of loadings and path coefficients that is fixed to be equal across groups. Similarly, `group.equal = c("loadings")` treats all loadings to be fixed across all groups, and `group.equal = c("paths")` returns identical path estimates in each group. See Examples for more details.

Reflective/Formative Indicators in the gesca Model Syntax:

In gesca 1.0, the relationships between indicators and latent variables can be either reflective or formative and this can be defined by the operator `(1)` or `(0)`, respectively. The user can input these operators in the model syntax lines that contain the `=~` and/or the `=:` operators. If `(1)` or `(0)` argument is omitted, by default, the `gesca.run` function will treat all the relationships defined by `=~` and/or `=:` as reflective. For example, the model syntax `Y =~ X1 + X2` or `Y(1) =~ X1 + X2` means that `X1` and `X2` are reflective. Similarly, when the second-order latent variable has effects on its first-order latent variables, the relationship can be defined as `H1 =: Y1+Y2` or `H1(1) =: Y1+Y2`. On the other hand, if the first-order latent variables are considered formative, the model syntax is expressed as `Y(0) =~ X1 + X2`, which involves no loadings for `X1` and `X2`.

Missing Value Imputation:

To use the missing value identifier of the gesca.run function, missing values in the data frame should be represented by numeric codes (e.g. -9999, 99, -9) and specified by the missingvalue option. This can be used only when the moption argument is used. If the data have missing value codes but the user omits the missingvalue argument, the gesca.run function will treat the missing values as regular data values (except for the value of -9999 by default).

Value

An object of class gesca.run, for which a summary method and four functions in gesca 1.0 (effectmeasures, fitmeasures, latentmeasures, and qualmeasures) are available.

References

Hwang, H., & Takane, Y. (2014). Generalized structured component analysis: A Component-Based Approach to Structural Equation Modeling (pp.13-126). Boca Raton, FL: Chapman & Hall/CRC Press.

Examples

```
library(gesca)
data(gesca.rick2) # Organizational identification example of Bagozzi
                  # has the grouping variable "gender"

# Model specification
myModel <- "
# Measurement model
OP =~ cei1 + cei2 + cei3
OI =~ ma1 + ma2 + ma3
AC_J =~ orgcmt1 + orgcmt2 + orgcmt3
AC_L =~ orgcmt5 + orgcmt6 + orgcmt8

# Structural model
OI ~ OP
AC_J ~ OI
AC_L ~ OI
"

# In case the user wants to apply a single-group analysis ignoring the group information:
rick2.GSCA <- gesca.run(myModel, gesca.rick2)

# Run a multiple-group GSCA with the grouping variable gender:
GSCA.group <- gesca.run(myModel, gesca.rick2, group.name = "gender")
summary(GSCA.group)
GSCA.group.nbt0 <- gesca.run(myModel, gesca.rick2, group.name ="gender", nbt = 0)
summary(GSCA.group.nbt0) # no bootstrapped std errors
effectmeasures(GSCA.group.nbt0)

# Models with the imposition of constraints on parameters across groups
# (1) equality constraints
equl.cstr <- gesca.run(myModel, gesca.rick2, group.name ="gender",
                        group.equal = c("loadings", "paths"))
```

```
# (2) identical loadings but different path coefficients
equl.loadings <- gesca.run(myModel, gesca.rick2, group.name ="gender",
                            group.equal = c("loadings"))

# (3) More constraints across groups with some formative indicators
myModel.con <- "
OP =~ cei1 + cei2 + cei3
OI(0) =~ ma1 + ma2 + ma3
AC_J =~ c(g,g)*orgcmt1 + c(h,h)*orgcmt2 + c(i,i)*orgcmt3
AC_L(0) =~ orgcmt5 + orgcmt6 + orgcmt8

OI ~ 0.37*OP
AC_J ~ OI
AC_L ~ OI
"
Const.Form <- gesca.run(myModel.con, gesca.rick2, group.name ="gender", nbt = 50)
summary(Const.Form)


# Second-order latent variable modeling
myModel5 <- "
# Measurement model
OP =~ cei1 + cei2 + cei3 + cei4 + cei5 + cei6 + cei7 + cei8
OI =~ ma1 + ma2 + ma3 + ma4 + ma5 + ma6
AC_J =~ orgcmt1 + orgcmt2 + orgcmt3 + orgcmt7
AC_L =~ orgcmt5 + orgcmt6 + orgcmt8

# Second-order latents
AC =: AC_J + AC_L

# Structural model
OI ~ OP
AC ~ OI
"

second.GSCA <- gesca.run(myModel5, gesca.rick2, group.name = "gender")
summary(second.GSCA)
second.GSCA.nbt0 <- gesca.run(myModel5, gesca.rick2, group.name = "gender", nbt=0)
summary(second.GSCA.nbt0)
```

`latentmeasures` *Means, Variances, and Correlations of Latent Variables*

Description

The means and variances of latent variables and the correlations among the latent variables. In gesca 1.0, the individual scores of latent variables are calculated based on Fornell's (1992) approach.

Usage

```
latentmeasures(object)
```

Arguments

`object`	An object of class. This can be created via the `gesca.run` function.

Value

Numeric vectors of means and variances, and correlation matrices.

References

Fornell, C. (1992). A national customer satisfaction barometer, the Swedish experience. *Journal of Marketing*, *56*, 6-21.

Hwang, H., & Takane, Y. (2014). Generalized structured component analysis: A Component-Based Approach to Structural Equation Modeling (p.26). Boca Raton, FL: Chapman & Hall/CRC Press.

See Also

`gesca.run`

Examples

```
library(gesca)
data(gesca.rick2) # Organizational identification example of Bagozzi

# Model specification
myModel <- "
# Measurement model
OP =~ cei1 + cei2 + cei3
OI =~ ma1 + ma2 + ma3
AC_J =~ orgcmt1 + orgcmt2 + orgcmt3
AC_L =~ orgcmt5 + orgcmt6 + orgcmt8

# Structural model
OI ~ OP
AC_J ~ OI
AC_L ~ OI
"

# Run a multiple-group GSCA with the grouping variable gender:
GSCA.group <- gesca.run(myModel, gesca.rick2, group.name = "gender", nbt=10)
# Note: bootstrap size is set to 10 for quick execution.
# For your actual analysis, make sure to use an adequate bootstrap sample size
# (e.g., n = 100 or 500) to obtain reliable results.
latentmeasures(GSCA.group)
```

qualmeasures *Reliability and Validity Measures*

Description

A variety of reliability and validity measures.

Usage

```
qualmeasures(object)
```

Arguments

object	An object of class. This can be created via the gesca.run function.

Details

Cronbach's alpha and Dillon-Goldstein's rho (or the composite reliability) can be used for checking internal consistency of indicators for each latent variable. The average variance extracted (AVE) can be used to examine the convergent validity of a latent variable. The number of eigenvalues greater than one per block of indicators can be used to check uni-dimensionality of the indicators.

Value

Numeric vectors of the reliability and validity measures.

See Also

gesca.run

Examples

```
library(gesca)
data(gesca.rick2) # Organizational identification example of Bagozzi

# Model specification
myModel <- "
# Measurement model
OP =~ cei1 + cei2 + cei3
OI =~ ma1 + ma2 + ma3
AC_J =~ orgcmt1 + orgcmt2 + orgcmt3
AC_L =~ orgcmt5 + orgcmt6 + orgcmt8

# Structural model
OI ~ OP
AC_J ~ OI
AC_L ~ OI
"
```

```
# Run a multiple-group GSCA with the grouping variable gender:
GSCA.group <- gesca.run(myModel, gesca.rick2, group.name = "gender", nbt=50)
qualmeasures(GSCA.group)
```

---

summary.gesca — *A Summary and Parameter Estimates of a Fitted Model*

Description

A summary of a fitted model such as model fit measures and parameter estimates obtained from GSCA.

Usage

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

Arguments

`object`	This function prints out a nice summary of the fitted model by the gesca.run function, including (1) information on convergence of the alternating least-squares (ALS) algorithm, (2) the number of observations (per group), parameters, and bootstrap samples, (3) six model fit measures, (4) GSCA parameter estimates such as weights, loadings, and path coefficients, and (5) R-squared values of each endogenous latent variable.
`...`	Further arguments passed to or from other methods.

Value

Object values of general information and parameter estimates of a fitted model via the gesca.run function.

References

Hwang, H., & Takane, Y. (2014). Generalized structured component analysis: A Component-Based Approach to Structural Equation Modeling (p.26). Boca Raton, FL: Chapman & Hall/CRC Press.

See Also

gesca.run

Examples

```
library(gesca)
data(gesca.rick2) # Organizational identification example of Bagozzi

# Model specification
myModel <- "
# Measurement model
OP =~ cei1 + cei2 + cei3
OI =~ ma1 + ma2 + ma3
AC_J =~ orgcmt1 + orgcmt2 + orgcmt3
AC_L =~ orgcmt5 + orgcmt6 + orgcmt8

# Structural model
OI ~ OP
AC_J ~ OI
AC_L ~ OI
"

# Run a multiple-group GSCA with the grouping variable gender:
GSCA.group <- gesca.run(myModel, gesca.rick2, group.name = "gender", nbt=50)
summary(GSCA.group)
```

Index