Package ‘semTools’

May 28, 2012

Type Package

Title Useful tools for structural equation modeling.

Version 0.1-1

Date 2012-05-28

Author Sunthud Pornprasertmanit <psunthud@ku.edu>, Patrick Miller <patr1ckm@ku.edu>, Alex Schoemann <schoemann@ku.edu>, Yves Rosseel <Yves.Rosseel@UGent.be>

Maintainer Sunthud Pornprasertmanit <psunthud@ku.edu>, Patrick Miller <patr1ckm@ku.edu>, Alex Schoemann <schoemann@ku.edu>

Depends R(>= 2.14), MASS, lavaan, methods

Suggests parallel, Amelia, mice

Description This package provide useful tools for structural equation modeling analysis.

License GPL (>= 2)

LazyLoad yes

URL https://github.com/simsem/semTools/wiki

R topics documented:

kurtosis *Finding excessive kurtosis*

Description

Finding excessive kurtosis (g2) of an object

Usage

```
kurtosis(object, population=FALSE)
```

Arguments

`object`	A vector used to find a excessive kurtosis
`population`	`TRUE` to compute the parameter formula. `FALSE` to compute the sample statistic formula.

Details

The excessive kurtosis computed is g2. The parameter excessive kurtosis γ_2 formula is

$$\gamma_2 = \frac{\mu_4}{\mu_2^2} - 3,$$

where μ_i denotes the i order central moment.

The excessive kurtosis formula for sample statistic g_2 is

$$g_2 = \frac{k_4}{k_2^2},$$

where k_i are the i order *k*-statistic.

The standard error of the excessive kurtosis is

$$Var(\hat{g}_2) = \frac{24}{N}$$

where N is the sample size.

Value

A value of an excessive kurtosis with a test statistic if the population is specified as `TRUE`

Author(s)

Sunthud Pornprasertmanit (University of Kansas; <psunthud@ku.edu>)

References

Weisstein, Eric W. (n.d.). *Kurtosis.* Retrived from MathWorld–A Wolfram Web Resource http://mathworld.wolfram.com/Kurtosis.html

Examples

```
kurtosis(1:5)
```

`measurementInvariance` *Measurement Invariance Tests*

Description

Testing measurement invariance across groups using a typical sequence of model comparison tests.

Usage

```
measurementInvariance(..., strict = FALSE, quiet = FALSE)
```

Arguments

`...`	The same arguments as for any lavaan model. See `cfa` for more information.
`strict`	If `TRUE`, the sequence requires 'strict' invariance. See details for more information.
`quiet`	If `TRUE`, a summary is printed out containing an overview of the different models that are fitted, together with some model comparison tests.

Details

If `strict = FALSE`, the following four models are tested in order:

1. Model 1: configural invariance. The same factor structure is imposed on all groups.
2. Model 2: weak invariance. The factor loadings are constrained to be equal across groups.
3. Model 3: strong invariance. The factor loadings and intercepts are constrained to be equal across groups.
4. Model 4: The factor loadings, intercepts and means are constrained to be equal across groups.

Each time a more restricted model is fitted, a chi-square difference test is reported, comparing the current model with the previous one, and comparing the current model to the baseline model (Model 1). In addition, the difference in cfi is also reported (delta.cfi).

If `strict = TRUE`, the following five models are tested in order:

1. Model 1: configural invariance. The same factor structure is imposed on all groups.
2. Model 2: weak invariance. The factor loadings are constrained to be equal across groups.
3. Model 3: strong invariance. The factor loadings and intercepts are constrained to be equal across groups.
4. Model 4: strict invariance. The factor loadings, intercepts and residual variances are constrained to be equal across groups.
5. Model 5: The factor loadings, intercepts, residual variances and means are constrained to be equal across groups.

Note that if the chi-square test statistic is scaled (eg. a Satorra-Bentler or Yuan-Bentler test statistic), a special version of the chi-square difference test is used as described in `http://www.statmodel.com/chidiff.shtml`

Value

Invisibly, all model fits in the sequence are returned as a list.

Author(s)

Yves Rosseel <Yves.Rosseel@UGent.be>

References

Vandenberg, R. J., and Lance, C. E. (2000). A review and synthesis of the measurement invariance literature: Suggestions, practices, and recommendations for organizational research. *Organizational Research Methods, 3,* 4-70.

Examples

```
HW.model <- ' visual =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed =~ x7 + x8 + x9 '

measurementInvariance(HW.model, data=HolzingerSwineford1939, group="school")
```

---

moreFitIndices *Calculate more fit indices*

Description

Calculate more fit indices that are not already provided in lavaan.

Usage

```
moreFitIndices(object, nPrior = 1)
```

Arguments

`object`	The lavaan model object provided after running the `cfa` or the `sem` functions.
`nPrior`	The sample size on which prior is based. This argument is used to compute BIC*.

Details

Normed Fit Index (nfi; West, Taylor, & Wu, 2012) is one of the relative fit indices which can be computed by

$$nfi = \frac{\chi^2_0 - \chi^2_k}{\chi^2_0},$$

where χ^2_k is the chi-square test statistic value of the target model, χ^2_0 is the chi-square test statistic value of the null model.

Incremental Fit Index (ifi; West, Taylor, & Wu, 2012) is one of the relative fit indices which can be computed by

$$ifi = \frac{\chi^2_0 - \chi^2_k}{\chi^2_0 - df_k},$$

where df_k is the degree of freedom when fitting the target model

Gamma Hat (gfi*; West, Taylor, & Wu, 2012) is one of the relative fit indices which can be computed by

$$gfi* = \frac{p}{p + 2 \times \frac{\chi^2_k - df_k}{N-1}},$$

where N is the sample size, p is the number of variables in the model.

Adjusted Gamma Hat (agfi*; West, Taylor, & Wu, 2012) is one of the relative fit indices which can be computed by

$$agfi* = \left(1 - \frac{p \times (p+1)}{2 \times df_k}\right) \times (1 - gfi*),$$

Corrected Akaike Information Criterion (AICc; Burnham & Anderson, 2003) is the corrected version of aic for small sample size:

$$aicc = f + \frac{2k(k+1)}{N-k-1},$$

where f is the minimized discrepancy function, which is the product of the log likelihood and -2, and k is the number of parameters in the target model.

Expected Value of Cross-Validation Index (ECVI; West, Taylor, & Wu, 2012) is the average discrepancy in the fitted covariance matrices between two samples of equal sample size across all possible combinations of two samples from the same population:

$$ecvi = f + \frac{2 \times k}{N},$$

Stochastic information criterion (sic; Preacher, 2006) is similar to aic or bic. This index will account for model complexity in the model's function form, in addition to the number of free parameters. sic can be computed by

$$sic = \frac{1}{2}\left(f - \log \det I(\hat{\theta})\right),$$

where $I(\hat{\theta})$ is the information matrix of the parameters.

Corrected Bayesian Information Criterion (BIC*; Kuha, 2004) is similar to bic but explicitly specifying the sample size on which the prior is based (N_{prior}).

$$bicc = f + k \log{(1 + N/N_{prior})},$$

Hannan-Quinn Information Criterion (hqc; Hannan & Quinn, 1979) is used for model selection similar to aic or bic.

$$hqc = f + 2k \log{(\log N)},$$

Value

1. nfi Normed Fit Index
2. ifi Incremental Fit Index
3. gfi* Gamma Hat
4. agfi* Adjusted Gamma Hat
5. aicc Corrected Akaike Information Criterion
6. ecvi Expected Value of Cross-Validation Index
7. sic Stochastic Information Criterion
8. bic* Bayesian Information Criterion with specifying the prior sample size
9. hqc Hannan-Quinn Information Criterion

Author(s)

Sunthud Pornprasertmanit <psunthud@ku.edu>, Aaron Boulton <aboulton@ku.edu>

References

Burnham, K., & Anderson, D. (2003). *Model selection and multimodel inference: A practical-theoretic approach.* New York, NY: Springer-Verlag.

Kuha, J. (2004). AIC and BIC: Comparisons of assumptions and performance. *Sociological Methods Research, 33,* 188-229.

Preacher, K. J. (2006). Quantifying parsimony in structural equation modeling. *Multivariate Behavioral Research, 43,* 227-259.

West, S. G., Taylor, A. B., & Wu, W. (2012). Model fit and model selection in structural equation modeling. In R. H. Hoyle (Ed.), *Handbook of Structural Equation Modeling.* New York: Guilford.

Examples

```
HS.model <- ' visual  =~ x1 + x2 + x3
              textual =~ x4 + x5 + x6
              speed   =~ x7 + x8 + x9 '

fit <- cfa(HS.model, data=HolzingerSwineford1939)
moreFitIndices(fit)
```

---

orthogonalize *Orthogonalize data for 2-way interaction in SEM*

Description

Orthogonalize indicators of a 2-way interaction between latent variables

Usage

```
orthogonalize(dat, xvars, zvars)
```

Arguments

`dat` Matrix or data frame of item level data.

`xvars` A vector of column numbers corresponding to indicators of the focal predictor (x).

`zvars` A vector of column numbers corresponding to indicators of the moderator (z).

Details

This functions will take a data frame or matrix and create orthogonalized product terms to compute latent variable interactions based on the method proposed by Little, Bovaird, & Widaman. The orthogonalized product terms can be entered into a SEM as indicators of a latent interaction variable. This function will compute all possible orthogonalized product terms (e.g., x has 3 indicators and z has 4 indicators, the function will return 3*4=12 new orthognalized product terms)

Value

1. data Original data with orthogonalized product terms appended.

Author(s)

Alexander M. Schoemann <schoemann@ku.edu>

References

Little, T. D., Bovaird, J. A., & Widaman, K. F. (2006). On the merits of orthogonalizing powered and product terms: Implications for modeling interactions among latent variables. *Structural Equation Modeling, 13* 497-519.

Examples

```
library(MASS)

n <- 500
means <- c(0,0)
covmat <- matrix(c(1, 0.3, 0.3, 1),nrow=2)

data <- mvrnorm(n,means,covmat)

x<-as.vector(data[,1])
z<-as.vector(data[,2])

y<-rnorm(n,0,1)+.4*x+.4*z+.2*x*z

x1<-rnorm(n,0.2,.2)+.7*x
x2<-rnorm(n,0.2,.2)+.7*x
x3<-rnorm(n,0.2,.2)+.7*x
z1<-rnorm(n,0.2,.2)+.7*z
z2<-rnorm(n,0.2,.2)+.7*z
z3<-rnorm(n,0.2,.2)+.7*z
y1<-rnorm(n,0.2,.2)+.7*y
y2<-rnorm(n,0.2,.2)+.7*y
y3<-rnorm(n,0.2,.2)+.7*y
```

```
dat<-data.frame(cbind(x1,x2,x3,z1,z2,z3,y1,y2,y3))

datOrth <-orthogonalize(dat,(1:3), (4:6))

#Fit model in Lavaan
library(lavaan)

syntax <- ’
x =~ x1 + x2 +x3
z =~ z1 + z2 + z3
xz =~ x1z1 + x1z2 + x1z3 + x2z1 + x2z2 + x2z3 + x3z1 + x3z2 + x3z3
y =~ y1 + y2 + y3
x ~~ z
x ~~ 0*xz
z ~~ 0*xz
y ~ x + z +xz
’

fit <- sem(model = syntax, data=datOrth, std.lv=TRUE)
summary(fit, fit.measures=TRUE)
```

runMI *Multiply impute and analyze data using lavaan*

Description

This function takes data with missing observations, multiple imputes the data, runs a SEM using lavaan and combines the results using Rubin’s rules.

Usage

```
runMI(data.mat,data.model, m, miPackage="Amelia", digits=3, seed=12345,
    std.lv = FALSE, estimator = "ML", group = NULL, group.equal = "", ...)
```

Arguments

`data.mat` Data frame with missing observations or a list of data frames where each data frame is one imputed data set (for imputed data generated outside of the function). If a list of data frames is supplied, then other options can be left at the default.

`data.model` lavaan syntax for the the model to be analyzed.

`m` Number of imputations wanted.

`miPackage` Package to be used for imputation. Currently runMI only uses Amelia or mice for imputation.

`digits` Number of digits to print in the results.

`seed` Random number seed to be used in imputations.

`std.lv` lavaan option. If TRUE, the metric of each latent variable is determined by fixing their variances to 1.0. If FALSE, the metric of each latent variable is determined by fixing the factor loading of the first indicator to 1.0.

`estimator` lavaan option. The estimator to be used. Can be one of the following: "ML" for maximum likelihood, "GLS" for generalized least squares, "WLS" for weighted least squares (sometimes called ADF estimation), "MLM" for maximum likelihood estimation with robust standard errors and a Satorra-Bentler scaled test statistic, "MLF" for maximum likelihood estimation with standard errors based on first-order derivatives and a conventional test statistic, "MLR" for maximum likelihood estimation with robust 'Huber-White' standard errors and a scaled test statistic which is asymptotically equivalent to the Yuan-Bentler T2-star test statistic. Note that the "MLM", "MLF" and "MLR" choices only affect the standard errors and the test statistic.

`group` lavaan option. A variable name in the data frame defining the groups in a multiple group analysis.

`group.equal` lavaan option. A vector of character strings. Only used in a multiple group analysis. Can be one or more of the following: "loadings", "intercepts", "means", "regressions", "residuals", "residual.covariances", "lv.variances" or "lv.covariances", specifying the pattern of equality constraints across multiple groups.

`...` Other arguments to be passed to the imputation package

Value

runMI returns a list with pooled fit indices, estimates, standard errors and fraction missing information.

`fit` Pooled fit information. The first set of fit information are simply averaged across imputations and are not trustworthy. The second set of fit information, is a pooled Chi-square statistic based on Li, Meng, Raghunathan, & Rubin (1991)

`parameters` Pooled parameter estimates and standard errors. Wald statistics and p values are computed from the pooled estimates and standard errors. Also contains two estimates of Fraction of Missing Information (FMI). The first estimate of FMI (FMI.1) is asymptotic FMI and the second estimate of FMI (FMI.2) is corrected for small numbers of imputation

Author(s)

Alexander M. Schoemann (University of Kansas; <`schoemann@ku.edu`>) Patrick Miller(University of Kansas; <`patr1ckm@ku.edu`>) Sunthud Pornprasertmanit (University of Kansas; <`psunthud@ku.edu`>) Mijke Rhemtulla (University of Kansas; <`mijke@ku.edu`>) Alexander Robitzsch (Federal Institute for Education Research, Innovation, and Development of the Austrian School System, Salzburg, Austria; <`a.robitzsch@bifie.at`>)

References

Li, K.H., Meng, X.-L., Raghunathan, T.E. and Rubin, D.B. (1991). Significance Levels From Repeated p-values with Multiply-Imputed Data. Statistica Sinica, 1, 65-92. Rubin, D.B. (1987) Multiple Imputation for Nonresponse in Surveys. J. Wiley & Sons, New York.

Examples

```
## No Example
```

`skew` *Finding skewness*

Description

Finding skewness (g1) of an object

Usage

```
skew(object, population=FALSE)
```

Arguments

`object`	A vector used to find a skewness
`population`	`TRUE` to compute the parameter formula. `FALSE` to compute the sample statistic formula.

Details

The skewness computed is g1. The parameter skewness γ_2 formula is

$$\gamma_2 = \frac{\mu_3}{\mu_2^{3/2}},$$

where μ_i denotes the i order central moment.

The excessive kurtosis formula for sample statistic g_2 is

$$g_2 = \frac{k_3}{k_2^2},$$

where k_i are the i order *k*-statistic.

The standard error of the skewness is

$$Var(\hat{g}_2) = \frac{6}{N}$$

where N is the sample size.

Value

A value of a skewness with a test statistic if the population is specified as `TRUE`

Author(s)

Sunthud Pornprasertmanit (University of Kansas; <`psunthud@ku.edu`>)

References

Weisstein, Eric W. (n.d.). *Skewness.* Retrived from MathWorld–A Wolfram Web Resource `http://mathworld.wolfram.com/Skewness.html`

Examples

```
skew(1:5)
```

Index