Package ‘lavaan.survey’

December 22, 2016

Type Package
Title Complex Survey Structural Equation Modeling (SEM)
Version 1.1.3.1
Date 2016-12-22
Author Daniel Oberski
Maintainer Daniel Oberski <daniel.oberski@gmail.com>
Description Fit structural equation models (SEM) including factor analysis, multivariate regression models with latent variables and many other latent variable models while correcting estimates, standard errors, and chi-square-derived fit measures for a complex sampling design. Incorporate clustering, stratification, sampling weights, and finite population corrections into a SEM analysis. Wrapper around packages lavaan and survey.

Depends R(>= 2.14.0), lavaan(>= 0.5-20), survey(>= 3.30-3)
Imports MASS
License GPL (>= 2)
URL http://daob.nl/
Suggests testthat, mice, mitools
NeedsCompilation no
Repository CRAN
Date/Publication 2016-12-22 23:43:24

R topics documented:

lavaan.survey-package ........................................... 2
cardinale ..................................................... 3
ess.dk .......................................................... 4
ess4.gb ......................................................... 6
lavaan.survey .................................................. 7
liss ............................................................. 11
pisa.be.2003 ................................................... 13
pval.pFsum ..................................................... 18
Complex survey analysis of structural equation models (SEM)

Description

Takes a lavaan fit object and a complex survey design object as input and returns a structural equation modeling analysis based on the fit object, where the complex sampling design is taken into account.

The structural equation model parameter estimates and standard errors are design-based. See Satorra and Muthén (1995) for details on the procedure.

Details

Package: lavaan.survey
Type: Package
Version: 1.0
Date: 2013-11-06
License: GPL(>=2)

Author(s)

Author/maintainer: Daniel Oberski <daniel.oberski@gmail.com>

References


See Also

lavaansurvey
pval.pFs

Examples

# See lavaan.survey function.

cardinale Nutrient experiment on patches of algae in Californian streams.

Description

An experiment on patches of algae in Californian streams by Cardinale, Bennett, Nelson, and Gross (2009). The authors chose 20 streams in the Mono Lake and Owens River basins in the Sierra Nevada. In each stream, they placed 5 or 10 PVC elbows containing different levels of nutrients and a small patch of agar on which algae could grow. They then returned to the streams about 42 days later and measured 1) species diversity in the stream, 2) species diversity in each patch, 3) biomass of the algae, and 4) rate of oxygen production on each patch.

Usage

data(cardinale)

Format

A data frame with 127 observations of 7 variables.

PatchDiversity Number of species on agar.
Biomass Standing algal biomass.
O2Production Gross primary oxygen production.
logNutrient Experimentally manipulated nutrient supply rate (log_10 transformed).
logNutrient2 log(nutrient, base=10)^2.
StreamDiversity Number of species in stream.
Stream The stream in which the experiment was performed.

Author(s)

Daniel Oberski - http://daob.nl/ - <daniel.oberski@gmail.com>
Source

The results of the fit may be compared with Figure 5 in Cardinale et al. (2009, p. 1237).
Model and data from this example were obtained from Jarrett Byrnes’ GitHub: https://github.com/jebyrnes/Ecological-SEMs-in-lavaan
Note that I changed the variable names to be easier to understand, e.g. logNutrient was logN, PatchDiversity was SA, etc.

References


See Also

lavaan.survey

Examples

data(cardinale)
summary(cardinale)

model.card <- '
  PatchDiversity ~ logNutrient + logNutrient2 + StreamDiversity
  Biomass ~ PatchDiversity + logNutrient
  O2Production ~ logNutrient + Biomass
  logNutrient ~~ logNutrient2'

fit.card <- sem(model.card, data = cardinale, fixed.x = FALSE, estimator = "MLM")

des.card <- svydesign(ids = ~Stream, probs = ~1, data = cardinale)
fit.card.survey <- lavaan.survey(fit.card, des.card, estimator = "MLM")
pval.pFsum(fit.card.survey, survey.design = des.card)

Description

Consists of indices of political and social trust, political efficacy, and fear of crime. The interviewer number was also recorded.

The ESS round 4 in Denmark was a face-to-face probability survey with simple random sampling.
Usage
data(ess.dk)

Format
A data frame with 1546 observations of 6 variables.

idno  Respondent identifier
intnum  Interviewer identifier
socialTrust  Sum of three 0-10 social trust items.
efficacy  Sum of two 1-5 items measuring "internal political efficacy".
systemTrust  Sum of three 0-10 political trust items.
fearCrime  Sum of three 1-4 items measuring fear of crime.

Author(s)
Daniel Oberski - http://daob.org - <daniel.oberski@gmail.com>

Source
This dataset was retrieved from http://ess.nsd.uib.no/ess/round4/ and converted to an R dataset.

References

See Also
lavaan.survey

Examples
data(ess.dk)
head(ess.dk)

# A saturated model with reciprocal effects from Saris & Gallhofer (2007)
dk.model <- 
  socialTrust ~ 1 + systemTrust + fearCrime
  systemTrust ~ 1 + socialTrust + efficacy
  socialTrust ~~ systemTrust
"
lavaan.fit <- lavaan(dk.model, data=ess.dk, auto.var=TRUE, estimator="MLM")
summary(lavaan.fit)
# Create a survey design object with interviewer clustering
survey.design <- svydesign(ids=~intnum, prob=~1, data=ess.dk)

survey.fit <- lavaan.survey(lavaan.fit=lavaan.fit, survey.design=survey.design)
survey.fit

## British attitudes towards the welfare state.

### Description

European Social Survey (ESS) data from the 2008 (fourth) round in the United Kingdom. The data are from a questionnaire on "what the responsibilities of governments should or should not be". These were factor-analyzed by Roosma, Gelissen, and van Oorschot (2013). Also included are complex survey design variables.

### Usage

data(ess4.gb)

### Format

A data frame with 2273 observations of 13 variables.

- idno  Respondent identifier.
- psu    Primary sampling unit (PSU).
- dweight ESS design weights.
- stratval Stratification variable (UK regions).
- gvjbevn Job for everyone, governments’ responsibility (0-10).
- gvhltc Health care for the sick, governments’ responsibility (0-10).
- gvlvol Standard of living for the old, governments’ responsibility (0-10).
- gvlvue Standard of living for the unemployed, governments’ responsibility (0-10).
- gvcldcr Child care services for working parents, governments’ responsibility (0-10).
- gvpdlwk Paid leave from work to care for sick family, governments’ responsibility (0-10).
- sbprvpv Social benefits/services prevent widespread poverty (1-5).
- sbeqsoc Social benefits/services lead to a more equal society (1-5).
- sbcwkfm Social benefits/services make it easier to combine work and family (1-5).

### Author(s)

Daniel Oberski - http://daob.nl/ - <daniel.oberski@gmail.com>

### Source

This dataset was retrieved from http://www.europeansocialsurvey.org/data/download.html?r=4 and converted to an R dataset.
lavaan.survey

Complex survey analysis of structural equation models (SEM)

Description

Takes a lavaan fit object and a complex survey design object as input and returns a structural equation modeling analysis based on the fit object, where the complex sampling design is taken into account.

The structural equation model parameter estimates are "aggregated" (Skinner, Holt & Smith 1989), i.e. they consistently estimate parameters aggregated over any clusters and strata and no explicit modeling of the effects of clusters and strata is involved. Standard errors are design-based. See Satorra and Muthen (1995) and references below for details on the procedure.

References


See Also

lavaan.survey

Examples

data(ess4.gb)

# Two-factor model based on Roosma et al (2013).
model.cfa <-
"range =~ gvjevbn + gvhlthc + gvs1vol + gvs1vue + gvcldcr + gvpd1wk
  goals =~ sbprvpv + sbeqsoc + sbcwkfm"

# Fit the model using lavaan
fit.cfa.ml <- lavaan(model.cfa, data = ess4.gb, estimator = "MLM",
    meanstructure = TRUE, int.ov.free = TRUE, auto.var = TRUE,
    auto.fix.first = TRUE, auto.cov.lv.x = TRUE)
fit.cfa.ml

# Define the complex survey design for ESS 4 in the UK
des.gb <- svydesign(ids = ~psu, strata = ~stratval, weights = ~dweight,
    data = ess4.gb)

# Fit the two-factor model while taking the survey design into account.
fit.cfa.surv <- lavaan.survey(fit.cfa.ml, survey.design = des.gb)
fit.cfa.surv
Both the pseudo-maximum likelihood (PML) procedure popular in the SEM world (e.g. Asparouhov 2005; Stapleton 2006) and weighted least squares procedures similar to aggregate regression modeling with complex sampling (e.g. Fuller 2009, chapter 6) are implemented. It is possible to give a list of multiply imputed datasets to svydesign as data. lavaan.survey will then apply the standard Rubin (1987) formula to obtain point and variance estimates under multiple imputation. Some care is required with this procedure when survey weights are also involved, however (see Notes).

Usage

lavaan.survey(lavaan.fit, survey.design,
estimator=c("MLM", "MLMV", "MLMVS", "WLS", "DWLS", "ML"),
estimator.gamma=c("default","Yuan-Bentler"))

Arguments

lavaan.fit A lavaan object resulting from a lavaan call. Since this is the estimator that will be used in the complex sample estimates, for comparability it can be convenient to use the same estimator in the call generating the lavaan fit object as in the lavaan.survey call. By default this is "MLM".
survey.design An svydesign object resulting from a call to svydesign in the survey package. This allows for incorporation of clustering, stratification, unequal probability weights, finite population correction, and multiple imputation. See the survey documentation for more information.
estimator The estimator used determines how parameter estimates are obtained, how standard errors are calculated, and how the test statistic and all measures derived from it are adjusted. See lavaan. The default estimator is MLM. It is recommended to use one of the ML estimators.
estimator.gamma Whether to use the usual estimator of Gamma as given by svyvar (the variance-covariance matrix of the observed variances and covariances), or apply some kind of smoothing or adjustment. Currently the only other option is the Yuan-Bentler (1998) adjustment based on model residuals.

Details

The user specifies a complex sampling design with the survey package's svydesign function, and a structural equation model with lavaan. lavaan.survey follows these steps:

1. The covariance matrix of the observed variables (or matrices in the case of multiple group analysis) is estimated using the svyvar command from the survey package.

2. The asymptotic covariance matrix of the variances and covariances is obtained from the svyvar output (the "Gamma" matrix)
3. The last step depends on the estimation method chosen:

**MLM, MLMV, MLMVS** The lavaan model is re-fit using Maximum Likelihood with the covariance matrix as data. After normal-theory ML estimation, the standard errors (vcov matrix), likelihood ratio ("chi-square") statistic, and all derived fit indices and statistics are adjusted for the complex sampling design using the Gamma matrix. i.e. the Satorra-Bentler (SB) corrections are obtained ("MLM" estimation in lavaan terminology). This procedure is equivalent to "pseudo"-maximum likelihood (PML).

**WLS, DWLS** The lavaan model is re-fit using Weighted Least Squares with the covariance matrix as data, and the Moore-Penrose inverse of the Gamma matrix as estimation weights. If DWLS is chosen only the diagonal of the weight matrix is used.

**Value**

An object of class lavaan, where the estimates, standard errors, vcov matrix, chi-square statistic, and fit measures based on the chi-square take into account the complex survey design. Several methods are available for lavaan objects, including a summary method.

**Note**

1) Some care should be taken when applying multiple imputation with survey weights. The weights should be incorporated in the imputation, and even then the variance produced by the usual Rubin (1987) estimator may not be consistent (Kott 1995; Kim et al. 2006).

If multiple imputation is used to deal with unit nonresponse, calibration and/or propensity score weighting with jackknifing may be a more appropriate method. See the survey package.

2) Note that when using PML or WLS, the Gamma matrix need not be positive definite. Preliminary investigations suggest that it often is not. This may happen due to reduction of effective sample size from clustering, for instance. In itself this need not be a problem, depending on the restrictiveness of the model. In such cases lavaan.survey checks explicitly whether the covariance matrix of the parameter estimates is still positive definite and produces a warning otherwise.

3) Currently only structural equation models for continuous variables are implemented.

**Author(s)**

Daniel Oberski - [http://daob.nl/](http://daob.nl/) - <daniel.oberski@gmail.com>

**References**


See Also

pval.pFsum
cardinale.ess.dk ess4.gb liss pisa.be.2003
svydesign lavaan

Examples

```r
# A single group example

# European Social Survey Denmark data (SRS)
data(ess.dk)

# A saturated model with reciprocal effects from Saris & Gallhofer
dk.model <- "
  socialTrust ~ 1 + systemTrust + fearCrime
  systemTrust ~ 1 + socialTrust + efficacy
  socialTrust ~~ systemTrust
"
lavaan.fit <- lavaan(dk.model, data=ess.dk, auto.var=TRUE, estimator="MLM")
summary(lavaan.fit)

# Create a survey design object with interviewer clustering
survey.design <- svydesign(ids=-intnum, prob=-1, data=ess.dk)

survey.fit <- lavaan.survey(lavaan.fit=lavaan.fit, survey.design=survey.design)
summary(survey.fit)
```
### A multiple group example

```r
data(HolzingerSwineford1939)

# The Holzinger and Swineford (1939) example - some model with complex restrictions
HS.model <- ' visual =~ x1 + x2 + c(lam31, lam31)*x3
textual =~ x4 + x5 + c(lam62, lam62)*x6
speed =~ x7 + x8 + c(lam93, lam93)*x9
speed ~ textual
textual ~ visual'

# Fit multiple group per school
fit <- lavaan(HS.model, data=HolzingerSwineford1939, int.ov.free=TRUE, meanstructure=TRUE,
auto.var=TRUE, auto.fix.first=TRUE, group="school", 
auto.cov.lv.x=TRUE, estimator="MLM")
summary(fit, fit.measures=TRUE)

# Create fictional clusters in the HS data
set.seed(20120125)
HolzingerSwineford1939$clus <- sample(1:100, size=nrow(HolzingerSwineford1939), replace=TRUE)
survey.design <- svydesign(ids=~clus, prob=1, data=HolzingerSwineford1939)
summary(fit.survey <- lavaan.survey(fit, survey.design))
```

# For more examples, please see the Journal of Statistical Software Paper,
# the accompanying datasets cardinale essTNgb liss pisaNbeRPPS
# and my homepage http://daob.nl/

---

**liss**

*Internet use in a panel data set.*

---

**Description**

The longitudinal internet studies for the social sciences (LISS) panel is a web survey panel recruited by probability sampling of households. All household members participate in the survey. This dataset contains four waves of data (2008-2011) on panel participants’ internet usage.

**Usage**

```r
data(liss)
```

**Format**

A data frame with 7369 observations on the following 6 variables.

- `nohouse_encr`  Household identifier.
nomem_encr Person identifier.
cs08a247 Natural logarithm of number of hours internet usage at home per week in 2008.
cs09b247 Natural logarithm of number of hours internet usage at home per week in 2009.
cs10c247 Natural logarithm of number of hours internet usage at home per week in 2010.
cs11d247 Natural logarithm of number of hours internet usage at home per week in 2011.

Author(s)
Daniel Oberski - http://daob.nl/ - <daniel.oberski@gmail.com>

Source
Data were obtained from http://www.lissdata.nl/dataarchive/. See also http://www.lissdata.nl/dataarchive/data_variables/view/795.

References

Examples
data(liss)

# Estimating the reliability of internet usage with the "quasi-simplex"
# (a.k.a. Gaussian latent Markov) model.

# A quasi-simplex model for four time points, setting the error variance
# to be estimated equal over time.
model.liss <- "
cs08 =~ 1 * cs08a247
cs09 =~ 1 * cs09b247
cs10 =~ 1 * cs10c247
cs11 =~ 1 * cs11d247

cs09 ~ cs08
cs10 ~ cs09
cs11 ~ cs10

Cs08a247 ~~ vare * cs08a247
Cs09b247 ~~ vare * cs09b247
Cs10c247 ~~ vare * cs10c247
Cs11d247 ~~ vare * cs11d247

cs08 ~ varcs08 * cs08

reliab.ratio := varcs08 / (varcs08 + vare)
"
# Fit the model using listwise deletion
fit.liss <- lavaan(model.liss, auto.var = TRUE, meanstructure = TRUE, 
int.ov.free = TRUE, data = liss)

# Fit the model accounting for nesting of respondents within households
des.liss <- svydesign(ids = ~nohouse_encr, prob = ~1, data = liss)
fit.liss.surv <- lavaan.survey(fit.liss, des.liss)
fit.liss.surv

# Complex survey inference on the reliability of interest:
parameterEstimates(fit.liss.surv)[24,]

## To deal with missing data (including attrition), multiple imputation can be used.
## For example using the mice library (although any MI software is suitable)
## Uncomment below to run this time-intensive analysis
## NOT RUN:

# set.seed(20140221)
# library("mice")
# liss.imp <- mice(liss, m = 100, method = "norm", maxit = 100)

## Turn the mice object into a list() of imputed datasets
# liss.implist <- lapply(seq(liss.imp$m), function(im) complete(liss.imp, im))

## After obtaining the list of imputed datasets,
## use the mitools package to turn it into an imputation list
# library("mitools")
# liss.implist <- imputationList(liss.implist)

## Give the imputation list as data to a svydesign object
# des.liss.imp <- svydesign(ids = ~nohouse_encr, prob = ~1, data = liss.implist)

## lavaan.survey can be used as usual, using the
## svydesign object that has an imputation list as data
## Standard errors and chi-square tests will account for both the clustering and the
## imputation uncertainty applying Rubin's rules.
# fit.liss.surv.mi <- lavaan.survey(fit.liss, des.liss.imp)
# fit.liss.surv.mi

## After this analysis, we can again perform inference on the reliability of interest:
# parameterEstimates(fit.liss.surv.mi)[24,]
Description

Data from the OECD’s 2003 Programme for International Student Assessment (PISA) in Belgium. Also included are 80 so-called "replicate weights" calculated by Westat which must be accounted for using complex survey procedures.

Usage

data(pisa.be.2003)

Format

A data frame with 8796 observations of 102 variables.

PV1MATH1  "Plausible values" for child's overall math ability (imputation 1).
PV1MATH2  "Plausible values" for child's overall math ability (imputation 2).
PV1MATH3  "Plausible values" for child's overall math ability (imputation 3).
PV1MATH4  "Plausible values" for child's overall math ability (imputation 4).
ST31Q01  Feel confident doing task: "timetable" 1 (very) - 4 (not at all).
ST31Q02  Feel confident doing task: "discount" (1-4).
ST31Q03  Feel confident doing task: "area" (1-4).
ST31Q04  Feel confident doing task: "graphs" (1-4).
ST31Q05  Feel confident doing task: "linear" (1-4).
ST31Q06  Feel confident doing task: "distance" (1-4).
ST31Q07  Feel confident doing task: "quadratics" (1-4).
ST31Q08  Feel confident doing task: "rate" (1-4).
ST32Q02  "I am just not good at Mathematics" 1 (strongly agree) - 4 (strongly disagree).
ST32Q04  "I get good marks in Mathematics" (1-4).
ST32Q06  "I learn Mathematics quickly" (1-4).
ST32Q07  "I have always believed that Mathematics is one of my best subjects" (1-4).
ST32Q09  "In my Mathematics class, I understand even the most difficult work" (1-4).
ESCS  Index of Socio-Economic and Cultural Status.
male  Gender (1=Female, 2=male).
school.type  Difficulty level of secondary studies (1-3).
w_FSTUWT  Overall survey weight for students.
w_FSTR1  BRR replicate.
w_FSTR2  BRR replicate.
w_FSTR3  BRR replicate.
w_FSTR4  BRR replicate.
w_FSTR5  BRR replicate.
w_FSTR6  BRR replicate.
w_FSTR7  BRR replicate.
w_FSTR8  BRR replicate.
w_FSTR9  BRR replicate.
w_FSTR10 BRR replicate.
w_FSTR11 BRR replicate.
w_FSTR12 BRR replicate.
w_FSTR13 BRR replicate.
w_FSTR14 BRR replicate.
w_FSTR15 BRR replicate.
w_FSTR16 BRR replicate.
w_FSTR17 BRR replicate.
w_FSTR18 BRR replicate.
w_FSTR19 BRR replicate.
w_FSTR20 BRR replicate.
w_FSTR21 BRR replicate.
w_FSTR22 BRR replicate.
w_FSTR23 BRR replicate.
w_FSTR24 BRR replicate.
w_FSTR25 BRR replicate.
w_FSTR26 BRR replicate.
w_FSTR27 BRR replicate.
w_FSTR28 BRR replicate.
w_FSTR29 BRR replicate.
w_FSTR30 BRR replicate.
w_FSTR31 BRR replicate.
w_FSTR32 BRR replicate.
w_FSTR33 BRR replicate.
w_FSTR34 BRR replicate.
w_FSTR35 BRR replicate.
w_FSTR36 BRR replicate.
w_FSTR37 BRR replicate.
w_FSTR38 BRR replicate.
w_FSTR39 BRR replicate.
w_FSTR40 BRR replicate.
w_FSTR41 BRR replicate.
w_FSTR42 BRR replicate.
w_FSTR43 BRR replicate.
w_FSTR44  BRR replicate.
w_FSTR45  BRR replicate.
w_FSTR46  BRR replicate.
w_FSTR47  BRR replicate.
w_FSTR48  BRR replicate.
w_FSTR49  BRR replicate.
w_FSTR50  BRR replicate.
w_FSTR51  BRR replicate.
w_FSTR52  BRR replicate.
w_FSTR53  BRR replicate.
w_FSTR54  BRR replicate.
w_FSTR55  BRR replicate.
w_FSTR56  BRR replicate.
w_FSTR57  BRR replicate.
w_FSTR58  BRR replicate.
w_FSTR59  BRR replicate.
w_FSTR60  BRR replicate.
w_FSTR61  BRR replicate.
w_FSTR62  BRR replicate.
w_FSTR63  BRR replicate.
w_FSTR64  BRR replicate.
w_FSTR65  BRR replicate.
w_FSTR66  BRR replicate.
w_FSTR67  BRR replicate.
w_FSTR68  BRR replicate.
w_FSTR69  BRR replicate.
w_FSTR70  BRR replicate.
w_FSTR71  BRR replicate.
w_FSTR72  BRR replicate.
w_FSTR73  BRR replicate.
w_FSTR74  BRR replicate.
w_FSTR75  BRR replicate.
w_FSTR76  BRR replicate.
w_FSTR77  BRR replicate.
w_FSTR78  BRR replicate.
w_FSTR79  BRR replicate.
w_FSTR80  BRR replicate.

wVARSTRR  Randomized final variance stratum (1-80).
Author(s)
Daniel Oberski - http://daob.nl/ - <daniel.oberski@gmail.com>

Source
This dataset was retrieved from http://www.oecd.org/pisa/ and converted to an R dataset.

For more information regarding the variables and the model estimated below, please see Ferla et al. (2009) and/or the PISA manual (OECD, 2009).

References


See Also
lavaan.survey

Examples

data(pisa.be.2003)

# Simplified version of Ferla et al. (2009) model.
model.pisa <- 
  math ~ PV1MATH1 + PV1MATH2 + PV1MATH3 + PV1MATH4
  neg.efficacy ~ ST31Q01 + ST31Q02 + ST31Q03 + ST31Q04 +
                  ST31Q05 + ST31Q06 + ST31Q07 + ST31Q08
  neg.selfconcept ~ ST32Q02 + ST32Q04 + ST32Q06 + ST32Q07 + ST32Q09
  neg.selfconcept ~ neg.efficacy + ESCS + male
  neg.efficacy ~ neg.selfconcept + school.type + ESCS + male
  math ~ neg.selfconcept + neg.efficacy + school.type + ESCS + male

# Fit the model using lavaan
fit <- lavaan(model.pisa, data = pisa.be.2003, auto.var = TRUE, std.lv = TRUE,
              meanstructure = TRUE, int.ov.free = TRUE, estimator = "MLM")

## Not run due to CRAN policies,
## uncomment below to account for replicate weights:

## Define the survey design using the BRR replicate weights provided by PISA
## Note that these settings will work for _any_ analysis of PISA data...
#des.rep <- svrepdesign(ids = -1, weights = ~W_FSTUWT, data = pisa.be.2003,
#repweights = "W_FSTR[0-9]+", type = "Fay", rho = 0.5)
pval.pFsum

F test for model fit of complex survey structural equation models

Description

Uses the pfsum function from the survey package to obtain a p-value for the overall model fit of a lavaan fit object using an F reference distribution, where the denominator degrees of freedom is the design degrees of freedom, degf(survey.design).

Usage

pval.pFsum(lavaan.fit, survey.design, method = "saddlepoint")

Arguments

- lavaan.fit: A lavaan object resulting from a lavaan call.
  Since this is the estimator that will be used in the complex sample estimates, for comparability it can be convenient to use the same estimator in the call generating the lavaan fit object as in the lavaan.survey call. By default this is "MLM".

- survey.design: An svydesign object resulting from a call to svydesign in the survey package. This allows for incorporation of clustering, stratification, unequal probability weights, finite population correction, and multiple imputation. See the survey documentation for more information.

- method: The method by which the distribution of the overall model fit statistic is approximated. See Details on the pfsum help page.

Details

With a small number of primary sampling units (design degrees of freedom), the asymptotic chi-square approximation to the distribution of the test statistic may not be entirely accurate. In this case instead of a chi-square, an F-reference distribution using the design degrees of freedom may be used.

When degf is infinite, the p-value output by this function be equal the Satterthwaite ("MLMVS") p-value (see lavaan "test" options).

The eigenvalues of the U.Gamma matrix, which is used by lavaan to calculate Satorra-Bentler scaling corrections, will be the coefficients in the mixture of chi-squares distribution (Skinner, Holt & Smith, pp. 86-87).

An anonymous reviewer for the Journal of Statistical Software suggested that "in surveys with small numbers of primary sampling units this sort of correction has often improved the behavior of tests in other contexts."
pval.pFsum

Value

A p-value for the overall F test of model fit, adjusted for nonnormality and the complex sampling design.

Note

Thanks are due to an anonymous reviewer for the Journal of Statistical Software for suggesting this function, and to Yves Rosseel for adjusting the lavaan code to pass along the U.Gamma eigenvalues to the fit object (GitHub commit 225fab0).

Author(s)

Daniel Oberski - http://daob.org - <daniel.oberski@gmail.com>

References


See Also

cardinale lavaan.survey pFsum degf svydesign lavaan

Examples

# Load HolzingerSwineford1939 data
data("HolzingerSwineford1939")

# Create 43 fake clusters
HolzingerSwineford1939$fake.cluster <- rep(1:43, each=7)
# Create survey design object
des <- svydesign(ids=fake.cluster, probs=1, data=HolzingerSwineford1939)

# Show the design degrees of freedom: number of clusters - 1
degf(des) # 42

# A reduced factor model that has a larger p-value:
HS.model <- ' visual =~ x2 + x3
textual =~ x4 + x5 + x6'

# Fit the factor model without taking complex sampling into account
fit <- cfa(HS.model, data=HolzingerSwineford1939, estimator="MLMVS")

# Fit the factor model, taking the 43 clusters into account
fit.svy <- lavaan.survey(fit, survey.design=des, estimator="MLMVS")

# Calculate the F test p-value.
# Since degf is only 42, there is a difference with Satterthwaite chi-square
pval.pFsum(fit.svy, survey.design=des) # 0.0542468133
Index

*Topic datasets
  liss, 11

*Topic models
  lavaan.survey, 7
  pval.pFsum, 18

*Topic multivariate
  lavaan.survey, 7
  pval.pFsum, 18

*Topic package
  lavaan.survey-package, 2

*Topic regression
  lavaan.survey, 7
  pval.pFsum, 18

*Topic robust
  lavaan.survey, 7
  pval.pFsum, 18

*Topic survey
  lavaan.survey, 7
  pval.pFsum, 18

cardinale, 3, 3, 10, 19

cfa, 3

degf, 18, 19

ess.dk, 3, 4, 10

ess4.gb, 3, 6, 10

lavaan, 3, 8–10, 18, 19
lavaan.survey, 3–5, 7, 7, 17, 19
lavaan.survey-package, 2
liss, 3, 10, 11

pFsum, 18, 19
pisa.be.2003, 3, 10, 13
pval.pFsum, 3, 10, 18

sem, 3

svydesign, 3, 8, 10, 18, 19