Package ‘Replication’

October 12, 2022

Type Package

Title Test Replications by Means of the Prior Predictive p-Value

Version 0.1.2

Description Allows for the computation of a prior predictive p-value to test replication of relevant features of original studies. Relevant features are captured in informative hypotheses. The package also allows for the computation of power. The statistical underpinnings are described in Zondervan-Zwijnenburg (2019) <doi:10.31234/osf.io/uvh5s>.

License GPL (>= 3)

Depends lavaan(>= 0.6-3), blavaan, mice

Imports quadprog, graphics, MASS, runjags(>= 2.0.4-2), rjags

Encoding UTF-8

LazyData true

NeedsCompilation no

Author M.A.J. Zondervan-Zwijnenburg [aut, cre],
Herbert Hoijtink [ths],
Caspar J van Lissa [ctb],
Consortium Individual Development [fnd] (NWO grant number 024.001.003)

Maintainer M.A.J. Zondervan-Zwijnenburg <`m.a.j.zwijnenburg@uu.nl`>

Repository CRAN

Date/Publication 2020-04-09 12:10:02 UTC

R topics documented:

`llratio.f` *Likelihood ratio statistic for constrained versus unconstrained model*

Description

The function first uses solve.QP to find the best fitting estimates under the imposed constraints. Subsequently, it calculates the likelihood ratio for the constrained versus unconstrained model.

Usage

```
llratio.f(BKcov, Q, R, r = NULL, E = 0L)
```

Arguments

`BKcov`	BKcov is the variance-covariance matrix of the model parameters.
`Q`	Q is a vector with the (unconstrained) model estimates.
`R`	A matrix defining the constraints under which we want to minimize the quadratic function.
`r`	vector holding the minimimum sum for each row in R.
`E`	Numeric. The first E constraints are treated as equality constraints, all further as inequality constraints. Default value = 0.

Value

`llratio`	scalar, the value of the resulting likelihood ratio.

Author(s)

M. A. J. Zondervan-Zwijnenburg

See Also

`solve.QP`

Examples

```
data <- data.frame(y=ChickWeight$weight,g=ChickWeight$Diet)
aggregate(data$y,by=list(data$g),mean)
reg <- lm(data$y~data$g)
BKcov <- vcov(reg)
Q <- reg$coefficients

#b1=102, b3-b2>20, b3-b4>10, b4>30
R <- rbind(c(1,0,0,0),c(0,-1,1,0),c(0,0,1,-1),c(0,0,0,1))
r = c(102,20,10,30)
E = 1

llratio.f(BKcov=BKcov, Q=Q, R=R, r=r, E = E)
```

`llratio.imp` *Log likelihood ratio and p-value for data imputed with mice*

Description

Computes the log likelihood ratio and p-value for data that is imputed with mice

Usage

```
llratio.imp(step2step3,imp,model,effectsize=FALSE,s.i,
             sample.cov = NULL, sample.mean = NULL, sample.nobs = NULL,
         group = NULL, cluster = NULL, constraints = "", WLS.V = NULL, NACOV = NULL,
             bayes=FALSE,dp=NULL,nchains=2)
```

Arguments

`step2step3`	The likelihood ratio values for simulated data as obtained with the ppc.step2step3 function.
`imp`	A mids object created with the R-package mice
`model`	The lavaan model that is to be applied to the data
`effectsize`	Logic; if TRUE, the constraints concern effectsizes.
`s.i`	A vector of length p holding with indices for the (pooled) standard deviation parameters with which the effect sizes should be computed
`sample.cov`	Numeric matrix. A sample variance-covariance matrix. The rownames and/or colnames must contain the observed variable names. For a multiple group analysis, a list with a variance-covariance matrix for each group. Note that if maximum likelihood estimation is used and likelihood="normal", the user provided covariance matrix is internally rescaled by multiplying it with a factor (N-1)/N, to ensure that the covariance matrix has been divided by N. This can be turned off by setting the sample.cov.rescale argument to FALSE.
`sample.mean`	A sample mean vector. For a multiple group analysis, a list with a mean vector for each group.
`sample.nobs`	Number of observations if the full data frame is missing and only sample moments are given. For a multiple group analysis, a list or a vector with the number of observations for each group.
`group`	A variable name in the data frame defining the groups in a multiple group analysis.
`cluster`	The cluster variable for multilevel data (beta!).
`constraints`	Additional (in)equality constraints not yet included in the model syntax. See model.syntax for more information. Note that the replication hypothesis should not be specified here!

`WLS.V`	A user provided weight matrix to be used by estimator "WLS"; if the estimator is "DWLS", only the diagonal of this matrix will be used. For a multiple group analysis, a list with a weight matrix for each group. The elements of the weight matrix should be in the following order (if all data is continuous): first the means (if a meanstructure is involved), then the lower triangular elements of the covariance matrix including the diagonal, ordered column by column. In the categorical case: first the thresholds (including the means for continuous variables), then the slopes (if any), the variances of continuous variables (if any), and finally the lower triangular elements of the correlation/covariance matrix excluding the diagonal, ordered column by column.
`NACOV`	A user provided matrix containing the elements of (N times) the asymptotic variance-covariance matrix of the sample statistics. For a multiple group analysis, a list with an asymptotic variance-covariance matrix for each group. See the WLS.V argument for information about the order of the elements.
`bayes`	Logic; if TRUE, a Bayesian estimator is used.
`dp`	blavaan default prior distributions on different types of parameters, typically the result of a call to dpriors(). See the dpriors() help file for more information.
`nchains`	A scalar indicating the number of chains to be used in the Bayesian analysis. Default value = 2.

Value

`pT`	The average parameter table
`llratio.i`	The likelihood ratio values for each of the imputed datasets
`pvals`	The prior-predictive p-values for each of the imputed datasets

Author(s)

M. A. J. Zondervan-Zwijnenburg

See Also

`llratio.f`

Examples

```
#the following example can be used, but takes >10 seconds

#create data
rnorm2 <- function(n,mean,sd) { mean+sd*scale(rnorm(n)) }

set.seed(9)
#step 1 input
#create/load data
n.o=30 #sample size original data
y.o <- data.frame(y=rnorm2(n.o,0,1),x=rnorm2(n.o,3,1))
n.r=80 #sample size new data
y.r <- data.frame(y=rnorm2(n.r,0.5,1),x=rnorm2(n.r,3,1))
```

```
y.r$y[runif(5,1,n.r)] <- NA #random missing data

#blavaan model
model <- '
y ~ x     #regression
y ~1      #intercept not default in lavaan (but is in blavaan)
'

step1.reg <- ppc.step1(y.o=y.o,model=model,n.r=n.r)

#H0: #reg > est, int = est
pT <- step1.reg$pT #parameter table
int.id <- which(pT$lhs=="y"&pT$op=="~1"&pT$rhs=="") #identify B0
reg.id <- which(pT$lhs=="y"&pT$op=="~"&pT$rhs=="x") #identify B1
hyp <- cbind(pT[c(int.id,reg.id),"plabel"],c("<",">"),c(pT[c(int.id,reg.id),"est"]))
print(hyp)
H0 <- paste(hyp[,1],hyp[,2],hyp[,3],collapse="&")

step23.reg <- ppc.step2step3(step1=step1.reg,y.r=NULL,model=model,H0)

y.r$y[runif(5,1,n.r)] <- NA #random missing data
imp <- mice(y.r,maxit=10,m=10)
llratio.imp(step2step3=step23.reg,imp=imp,model=model)
```

---

posterior.step1 *Posterior Distribution*

Description

Samples from the posterior distribution of the data by means of blavaan. These samples form the basis for the predictive distribution in the prior predictive check.

Usage

```
posterior.step1(y.o, model,
    sample.cov = NULL, sample.mean = NULL, sample.nobs = NULL,
    group = NULL, constraints = "", WLS.V = NULL, NACOV = NULL,
   nchains=2, nadapt, nburnin, nsample, dp = NULL, convergence = "manual",target,
    imp=imp)
```

Arguments

`y.o`	A data frame containing the original data for the replication test.
`model`	The (b)lavaan model that is to be fitted to the data.
`sample.cov`	Numeric matrix. A sample variance-covariance matrix. The rownames and/or colnames must contain the observed variable names. For a multiple group analysis, a list with a variance-covariance matrix for each group. Note that if maximum likelihood estimation is used and likelihood="normal", the user provided

covariance matrix is internally rescaled by multiplying it with a factor (N-1)/N, to ensure that the covariance matrix has been divided by N. This can be turned off by setting the sample.cov.rescale argument to FALSE.

`sample.mean`	A sample mean vector. For a multiple group analysis, a list with a mean vector for each group.
`sample.nobs`	Number of observations if the full data frame is missing and only sample moments are given. For a multiple group analysis, a list or a vector with the number of observations for each group.
`group`	A variable name in the data frame defining the groups in a multiple group analysis.
`constraints`	Additional (in)equality constraints not yet included in the model syntax. See model.syntax for more information. Note that the replication hypothesis should not be specified here!
`WLS.V`	A user provided weight matrix to be used by estimator "WLS"; if the estimator is "DWLS", only the diagonal of this matrix will be used. For a multiple group analysis, a list with a weight matrix for each group. The elements of the weight matrix should be in the following order (if all data is continuous): first the means (if a meanstructure is involved), then the lower triangular elements of the covariance matrix including the diagonal, ordered column by column. In the categorical case: first the thresholds (including the means for continuous variables), then the slopes (if any), the variances of continuous variables (if any), and finally the lower triangular elements of the correlation/covariance matrix excluding the diagonal, ordered column by column.
`NACOV`	A user provided matrix containing the elements of (N times) the asymptotic variance-covariance matrix of the sample statistics. For a multiple group analysis, a list with an asymptotic variance-covariance matrix for each group. See the WLS.V argument for information about the order of the elements.
`nchains`	A scalar indicating the number of chains to be used in the Bayesian analysis. Default value = 2.
`nadapt`	The number of blavaan adaptive iterations to use at the start of the simulation. Default value = 1,000
`nburnin`	A scalar indicating the number of burnin iterations to be used in the Bayesian analysis.
`nsample`	A scalar indicating the number of samples to be taken from the posterior after burnin.
`convergence`	Default = "manual". If "auto", parameters will be sampled until convergence is achieved (via autorun.jags). In this case, the arguments burnin and sample are passed to autorun.jags as startburnin and startsample, respectively. Otherwise, parameters are sampled as specified by the user (or by the run.jags defaults).
`target`	Desired MCMC package ("jags" is default, but "stan" also available).
`dp`	blavaan default prior distributions on different types of parameters, typically the result of a call to dpriors(). See the dpriors() help file for more information.
`imp`	A multiply imputed dataset stored in an object of class mids as generated by mice. If there is no imputed data, specify imp = NULL.

Value

`post`	A matrix with samples from the posterior.
`pT`	A data.frame containing the parameter table for the fitted model as given by parTable lavaan
`free.i`	A vector with indices for the freely estimated parameters as given in pT.

See Also

`blavaan`, `lavaan`

`ppc.plot` *Plot ppc.step2step3 output*

Description

Creates a plot using the output of ppc.step2step3.

Usage

```
ppc.plot(llratio.s, llratio.r)
```

Arguments

`llratio.s`	A vector with llratio values for the predicted data.
`llratio.r`	The llratio value for the new data

Value

Returns a plot histogram of llratio values for the predicted data, and indicates the value of llratio for the new data with a red line. If llratio is equal to 0 for at least 10% of the llratio for the predicted data, the function will create a seperate thick black line for this group of values.

Author(s)

Zondervan-Zwijnenburg, M.A.J.

Examples

```


#the following example can be used, but may take >10 seconds

#create data
rnorm2 <- function(n,mean,sd) { mean+sd*scale(rnorm(n)) }

# simple regression -------------------------------------------------------
```

```
set.seed(9)
#step 1 input
#create/load data
n.o=30 #sample size original data
y.o <- data.frame(y=rnorm2(n.o,0,1),x=rnorm2(n.o,3,1))
#y.o <- correlate(as.matrix(y.o), corm=.70); y.o <- data.frame(y=y.o[,1],x=y.o[,2])
n.r=50 #sample size new data
y.r <- data.frame(y=rnorm2(n.r,0.5,1),x=rnorm2(n.r,3,1))

#blavaan model
model <- '
y ~ x     #regression
y ~1      #intercept not default in lavaan (but is in blavaan)
'

#Warning: This is a minimal example;
step1.reg <- ppc.step1(y.o=y.o,model=model,nchains=2,n.r=50)

print(step1.reg$pT)
#H0: #reg > est, int = est       =       B1>0.302 & B0= -0.878
pT <- step1.reg$pT #parameter table
int.id <- which(pT$lhs=="y"&pT$op=="~1"&pT$rhs=="") #identify B0
reg.id <- which(pT$lhs=="y"&pT$op=="~"&pT$rhs=="x") #identify B1
hyp <- cbind(pT[c(int.id,reg.id),"plabel"],c("=",">"),c(pT[c(int.id,reg.id),"est"]))
print(hyp)
H0 <- paste(hyp[,1],hyp[,2],hyp[,3],collapse="&")

step23.reg <- ppc.step2step3(step1=step1.reg,y.r=y.r,model=model,H0)

ppc.plot(step23.reg$llratio.s,step23.reg$llratio.r)
```

`ppc.step1` *Prior predictive check step 1*

Description

Samples from the posterior distribution of the data by means of blavaan and simulates data y.s using lavaan simulateData. The data y.s are based on samples from the posterior and represent samples from the predictive distribution.

Usage

```
ppc.step1(y.o, model,
    sample.cov = NULL, sample.mean = NULL, sample.nobs = NULL,
    group = NULL, n.groups, constraints = "", WLS.V = NULL, NACOV = NULL,
    nchains = 2, nadapt = 1000, nburnin=5000, nsample=5000,
    dp = NULL, convergence= "manual",target="jags",
    imp = NULL, n.r, nsim=5000, post, pT, free.i)
```

Arguments

`y.o`	A data frame containing the original data for the replication test.
`model`	The (b)lavaan model that is to be fitted to the data.
`sample.cov`	Numeric matrix. A sample variance-covariance matrix. The rownames and/or colnames must contain the observed variable names. For a multiple group analysis, a list with a variance-covariance matrix for each group. Note that if maximum likelihood estimation is used and likelihood="normal", the user provided covariance matrix is internally rescaled by multiplying it with a factor (N-1)/N, to ensure that the covariance matrix has been divided by N. This can be turned off by setting the sample.cov.rescale argument to FALSE.
`sample.mean`	A sample mean vector. For a multiple group analysis, a list with a mean vector for each group.
`sample.nobs`	Number of observations if the full data frame is missing and only sample moments are given. For a multiple group analysis, a list or a vector with the number of observations for each group.
`group`	A variable name in the data frame defining the groups in a multiple group analysis.
`n.groups`	If applicable, the number of groups.
`constraints`	Additional (in)equality constraints not yet included in the model syntax. See model.syntax for more information. Note that the replication hypothesis should not be specified here!
`WLS.V`	A user provided weight matrix to be used by estimator "WLS"; if the estimator is "DWLS", only the diagonal of this matrix will be used. For a multiple group analysis, a list with a weight matrix for each group. The elements of the weight matrix should be in the following order (if all data is continuous): first the means (if a meanstructure is involved), then the lower triangular elements of the covariance matrix including the diagonal, ordered column by column. In the categorical case: first the thresholds (including the means for continuous variables), then the slopes (if any), the variances of continuous variables (if any), and finally the lower triangular elements of the correlation/covariance matrix excluding the diagonal, ordered column by column.
`NACOV`	A user provided matrix containing the elements of (N times) the asymptotic variance-covariance matrix of the sample statistics. For a multiple group analysis, a list with an asymptotic variance-covariance matrix for each group. See the WLS.V argument for information about the order of the elements.
`nchains`	A scalar indicating the number of chains to be used in the Bayesian analysis. Default value = 2.
`nadapt`	The number of blavaan adaptive iterations to use at the start of the simulation. Default value = 1,000
`nburnin`	A scalar indicating the number of burnin iterations to be used in the Bayesian analysis. Default value = 5,000
`nsample`	A scalar indicating the number of samples to be taken from the posterior after burnin. Default value = 5,000

`dp`	blavaan default prior distributions on different types of parameters, typically the result of a call to dpriors(). See the dpriors() help file for more information.
`convergence`	Default = "manual". If "auto", parameters will be sampled until convergence is achieved (via autorun.jags). In this case, the arguments burnin and sample are passed to autorun.jags as startburnin and startsample, respectively. Otherwise, parameters are sampled as specified by the user (or by the run.jags defaults).
`target`	Desired MCMC package ("jags" is default, but "stan" also available).
`imp`	Default = NULL. If imputed data is present, the imputed object of class mids as generated by mice can be included here.
`n.r`	Sample size for the new data (y.r).
`nsim`	Number of datasets y.s to be created. Default value = 5,000.
`post`	Do not use this argument. Matrix with samples from the posterior distribution for the original data y.o
`pT`	Do not use this argument. The blavaan parameter table.
`free.i`	Do not use this argument. A vector with indices for the freely estimated parameters as given in pT.

Value

`pT`	A data.frame containing the parameter table for the fitted model as given by parTable lavaan
`y.s`	A list containing all simulated data y.s.

Author(s)

M. A. J. Zondervan-Zwijnenburg

See Also

`bsem`

Examples

```
#the following example can be used, but takes >10 seconds

#step 1 input
data <- data.frame(y=ChickWeight$weight,x=ChickWeight$Time)

model <- '
y ~ x     #regression
y ~1      #intercept not default in lavaan (but is in blavaan)
'

reg.step1 <- ppc.step1(y.o=data, model=model,n.r=50)
```

`ppc.step2step3` *Prior predictive check step 2 and 3*

Description

Calculates an approximate likelihood ratio (D) for new data (y.r) and predicted data (y.s) according to the proposed constraints, and generates a prior predictive p-value.

Usage

```
ppc.step2step3(step1, y.r, model = model, H0, s.i, H0check=TRUE, y.o,
      ordered = NULL, sample.cov = NULL, sample.mean = NULL, sample.nobs = NULL,
      group = NULL, cluster = NULL, constraints = "", WLS.V = NULL, NACOV = NULL,
        bayes = FALSE, dp = NULL, convergence = "manual", nchains = 2)
```

Arguments

`step1`	An object containing the output of ppc.step1.
`y.r`	A data.frame with the new data. If y.r = NA, the approximate likelihood ratio will only be computed for the predicted data y.s.
`model`	The (b)lavaan model that is to be fitted to the data.
`H0`	The replication hypothesis within quotes "" with the lavaan plabels as parameter names and parts separated with &. For more information on hypothesis specification, see the details section below.
`s.i`	A vector of length p with indices for the (pooled) standard deviation parameters with which the effect sizes should be computed. Default = NULL.
`H0check`	Logic. If TRUE, the function will check whether H0 is in line with the original data (which it should be) before performing the ppc. The ppc will return an error if H0 does not pass the check.
`y.o`	A data.frame with the original data, required to run the H0check.
`ordered`	Character vector. Only used if the data is in a data.frame. Treat these variables as ordered (ordinal) variables, if they are endogenous in the model. Importantly, all other variables will be treated as numeric (unless they are declared as ordered in the original data.frame
`sample.cov`	Numeric matrix. A sample variance-covariance matrix. The rownames and/or colnames must contain the observed variable names. For a multiple group analysis, a list with a variance-covariance matrix for each group. Note that if maximum likelihood estimation is used and likelihood="normal", the user provided covariance matrix is internally rescaled by multiplying it with a factor (N-1)/N, to ensure that the covariance matrix has been divided by N. This can be turned off by setting the sample.cov.rescale argument to FALSE.
`sample.mean`	A sample mean vector. For a multiple group analysis, a list with a mean vector for each group.

`sample.nobs`	Number of observations if the full data frame is missing and only sample moments are given. For a multiple group analysis, a list or a vector with the number of observations for each group.
`group`	A variable name in the data frame defining the groups in a multiple group analysis.
`cluster`	The cluster variable for multilevel data (beta!).
`constraints`	Additional (in)equality constraints not yet included in the model syntax. See model.syntax for more information. Note that the replication hypothesis should not be specified here!
`WLS.V`	A user provided weight matrix to be used by estimator "WLS"; if the estimator is "DWLS", only the diagonal of this matrix will be used. For a multiple group analysis, a list with a weight matrix for each group. The elements of the weight matrix should be in the following order (if all data is continuous): first the means (if a meanstructure is involved), then the lower triangular elements of the covariance matrix including the diagonal, ordered column by column. In the categorical case: first the thresholds (including the means for continuous variables), then the slopes (if any), the variances of continuous variables (if any), and finally the lower triangular elements of the correlation/covariance matrix excluding the diagonal, ordered column by column.
`NACOV`	A user provided matrix containing the elements of (N times) the asymptotic variance-covariance matrix of the sample statistics. For a multiple group analysis, a list with an asymptotic variance-covariance matrix for each group. See the WLS.V argument for information about the order of the elements.
`bayes`	Logic; if TRUE, a Bayesian estimator is used.
`dp`	If bayes = TRUE, blavaan default prior distributions on different types of parameters, typically the result of a call to dpriors(). See the dpriors() help file for more information.
`convergence`	If bayes = TRUE, default convergence setting = "manual". If "auto", parameters will be sampled until convergence is achieved (via autorun.jags). In this case, the arguments burnin and sample are passed to autorun.jags as startburnin and startsample, respectively. Otherwise, parameters are sampled as specified by the user (or by the run.jags defaults).
`nchains`	If bayes = TRUE, A scalar indicating the number of chains to be used in the Bayesian analysis. Default value = 2.

Details

The specification of ‘H0‘ in ‘ppc.step2step3‘:

‘H0‘ is a character string that specifies which informative hypothesis has to be evaluated. A simple example is `H0 <- ".p1. > .p2. > .p3. & .p1. = 2"` which specifies a hypothesis using three estimates with names ".p1.", ".p2.", and ".p3.", respectively.

The hypothesis specified has to adhere to the following rules:

- When using `ppc.step2step3`, the ‘plabels‘ of the `blavaan` output resulting from `ppc.step1` have to be used to indicate which parameters are involved in the informative hypothesis H0. For example ‘.p1.‘ and ‘.p2.‘ can be the labels of the parameters of interest.

- Linear combinations of parameters must be specified adhering to the following rules: a) Each parameter name is used at most once. b) Each parameter name may or may not be premultiplied with a number. c) A constant may be added or subtracted from each parameter name. Examples are: `"3 *.p1.+ 5"`; `".p1.+ 2 * .p2.+ 3 * .p3.- 2"` and `".p1.- .p2."`.
- (Linear combinations of) parameters can be constrained using <, >, and =. For example, `".p1.> 0"` or `".p1.> .p2.= 0"` or `"2 *.p1.< .p2.+ .p3.> 5"`.
- The ampersand `&` can be used to combine different parts of a hypothesis. For example, `".p1.> .p2.& .p2.> .p3."` which is equivalent to `".p1.> .p2.> .p3."` or `".p1.> 0 & .p2.> 0 & .p3.> 0"`.
- Sets of (linear combinations of) parameters subjected to the same constraints can be specified using (). For example, `".p1.> (.p2.,.p3.)"` which is equivalent to `".p1.> .p2.&.p1.> .p3."`.
- Hypotheses have to be possible. A hypothesis is impossible if estimates in agreement with the hypothesis do not exist. For example: values for `.p1.` in agreement with `".p1.= 0 & .p1. > 2"` do not exist. It is the responsibility of the user to ensure that the hypotheses specified are possible. If not, `ppc.step2step`‘ will return an error message: `Error in solve.QP(Dmat, dvec = dvec, t(R), r, meq = E, factorized = FALSE) : constraints are inconsistent, no solution!`.

Value

Generates a histogram of llratio.s in which llratio.r is indicated with a vertical line. The proportion of llratio.s at the right of this line constitutes the prior predictive p-value.

`llratio.r`	The likelihood ratio for the new dataset.
`p-value`	The prior predictive p-value.
`llratio.s`	The likelihood ratio’s for each of the datasets y.s.
`H0 matrices`	Matrices R, r, and E that constitute the matrix form of H0: R*theta>r or R*theta=r for equality constraints. The value in E specifies the number of equality constraints in the hypothesis.
`pT.s`	Parameter table columns for y.s indicating which parameter received which id and which label in the lavaan analysis of the predicted data. If the id’s and labels differ from those of the blavaan analysis used in step1, the function will generate a warning for the user to check whether this difference affects parameters in H0.

Author(s)

M. A. J. Zondervan-Zwijnenburg

Examples

```
#the following example can be used, but may take >10 seconds

#create data
rnorm2 <- function(n,mean,sd) { mean+sd*scale(rnorm(n)) }
```

```
# simple regression --------------------------------------------------------

set.seed(9)
#step 1 input
#create/load data
n.o=30 #sample size original data
y.o <- data.frame(y=rnorm2(n.o,0,1),x=rnorm2(n.o,3,1))
#y.o <- correlate(as.matrix(y.o), corm=.70); y.o <- data.frame(y=y.o[,1],x=y.o[,2])
n.r=50 #sample size new data
y.r <- data.frame(y=rnorm2(n.r,0.5,1),x=rnorm2(n.r,3,1))

#blavaan model
model <- '
y ~ x     #regression
y ~1
'

#Warning: This is a minimal example;
step1.reg <- ppc.step1(y.o=y.o,model=model,nchains=2,n.r=50)

print(step1.reg$pT)
#H0: # int < estb reg > est      =       B0< -1.05 & B1>0.35
pT <- step1.reg$pT #parameter table
int.id <- which(pT$lhs=="y"&pT$op=="~1"&pT$rhs=="") #identify B0
reg.id <- which(pT$lhs=="y"&pT$op=="~"&pT$rhs=="x") #identify B1
hyp <- cbind(pT[c(int.id,reg.id),"plabel"],c("<",">"),c(pT[c(int.id,reg.id),"est"]))
print(hyp)
H0 <- paste(hyp[,1],hyp[,2],hyp[,3],collapse="&")

step23.reg <- ppc.step2step3(step1=step1.reg,y.r=y.r,model=model,H0)
```

`sim.step1` *Simulate Data for the Predictive Distribution*

Description

Simulates data y.s using lavaan simulateData. The data y.s are based on samples from the posterior and represent samples from the predictive distribution.

Usage

```
sim.step1(n.r, nsim, post, pT, free.i, group=NULL, n.groups)
```

Arguments

`n.r`	Sample size for the new data (y.r)
`nsim`	Number of datasets y.s to be created.
`post`	Matrix with samples from the posterior distribution for the original data y.o

`pT`	The blavaan parameter table.
`free.i`	A vector with indices for the freely estimated parameters as given in pT.
`group`	If applicable, the name of the grouping variable.
`n.groups`	If applicable, the number of groups.

Value

`y.s`	A list containing all y.s.

Author(s)

M. A. J. Zondervan-Zwijnenburg

See Also

`simulateData`

Index