Package ‘brxx’

January 26, 2021

Type Package

Title Bayesian Test Reliability Estimation

Version 0.1.2

Author Joshua Ray Tanzer

Maintainer Joshua Ray Tanzer <jtanzer@lifespan.org>

Description When samples contain missing data, are small, or are suspected of bias, estimation of scale reliability may not be trustworthy. A recommended solution for this common problem has been Bayesian model estimation. Bayesian methods rely on user specified information from historical data or researcher intuition to more accurately estimate the parameters. This package provides a user friendly interface for estimating test reliability. Here, reliability is modeled as a beta distributed random variable with shape parameters alpha=true score variance and beta=error variance (Tanzer & Harlow, 2020) <doi:10.1080/00273171.2020.1854082>.

Depends MCMCpack, GPArotation, TeachingDemos, blavaan, blme

Imports MASS, rstan

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-26 17:00:06 UTC

R topics documented:

---

bcor *bcor: Bayesian Estimation of The Correlation Matrix*

Description

This function estimates coefficient omega internal consistency reliability.

Usage

```
bcor(data, iter, burn, seed, CI, S0, nu0, mu0)
```

Arguments

`data`	N by P data matrix.
`iter`	Number of iterations for the Gibbs sampler.
`burn`	Number of samples to burn in.
`seed`	Seed for the Gibbs sampler
`CI`	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).
`S0`	Prior variance covariance matrix.
`nu0`	Prior degrees of freedom for inverse Wishart prior distribution.
`mu0`	Prior means for each column.

Value

Returns median posterior estimates of the correlation matrix.

Examples

```
set.seed(999)
your_data=mvrnorm(n=15,mu=c(0,0),Sigma=matrix(c(4,3,3,9),nrow=2,ncol=2))
Mu0=c(0,0)
Sigma0=matrix(c(1,0.6,0.6,4),nrow=2,ncol=2)
Nu0=1
bcor(data=your_data,iter=5000,burn=2500,seed=999,CI=0.95,
    mu0=Mu0,S0=Sigma0,nu0=Nu0)
```

bcov *bcov: Bayesian Estimation of the Variance Covariance Matrix*

Description

This function estimates the variance covariance matrix for a

Usage

```
bcov(data, iter, burn, seed, CI, S0, nu0, mu0)
```

Arguments

`data`	N by P data matrix.
`iter`	Number of iterations for the Gibbs sampler.
`burn`	Number of samples to burn in.
`seed`	Seed for the Gibbs sampler
`CI`	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).
`S0`	Prior variance covariance matrix.
`nu0`	Prior degrees of freedom for inverse Wishart prior distribution.
`mu0`	Prior means for each column.

Value

Returns median posterior estimates of the variance covariance matrix.

Examples

```
## Not run:
set.seed(999)
your_data=mvrnorm(n=15,mu=c(0,0),Sigma=matrix(c(4,3,3,9),nrow=2,ncol=2))
Mu0=c(0,0)
Sigma0=matrix(c(1,0.6,0.6,4),nrow=2,ncol=2)
Nu0=3-1
bcov(data=your_data,iter=5000,burn=2500,seed=999,CI=0.95,
    mu0=Mu0,S0=Sigma0,nu0=Nu0)
## End(Not run)
```

bomega *bomega: Bayesian Estimation of Coefficient Omega*

Description

This function estimates coefficient omega internal consistency reliability.

Usage

```
bomega(K, mod, alpha, beta, CI)
```

Arguments

K	The number of test items.
mod	A measurement model estimated as a bsem object by blavaan.
alpha	Prior true score variance.
beta	Prior error variance.
CI	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).

Value

Returns estimated median and quantile based credible limits for omega.

Examples

```
your_data=data.frame(mvrnorm(n=20,mu=c(0,0,0,0,0),
Sigma=matrix(c(4,2,2,2,2,
              2,4,2,2,2,
              2,2,4,2,2,
              2,2,2,4,2,
              2,2,2,2,4),
            nrow=5, ncol=5)))
colnames(your_data)=c("x1","x2","x3","x4","x5")
mod='tau=~x1+x2+x3+x4+x5'
fit=bsem(mod,data=your_data)
bomega(K=5,mod=fit,alpha=3.51,beta=1.75,CI=0.95)
```

`bomega_general` *bomega_general: Bayesian Estimation of Coefficient Omega, General Form*

Description

This function estimates coefficient omega internal consistency reliability.

Usage

```
bomega_general(lambda, psi, alpha, beta, CI)
```

Arguments

`lambda`	vector of item loadings.
`psi`	vector of item variances.
`alpha`	Prior true score variance.
`beta`	Prior error variance.
`CI`	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).

Value

Returns estimated median and quantile based credible limits for omega.

Examples

```
lambda=c(0.7,0.5,0.6,0.7)
psi=c(0.2,0.4,0.3)
alpha=3.51
beta=1.75
bomega_general(lambda=lambda,psi=psi,alpha=alpha,beta=beta,CI=0.95)
```

`brxx_Cor` *brxx_Cor: Bayesian Estimation of Reliability from Correlation*

Description

This function estimates reliability from a correlation

Usage

```
brxx_Cor(x, y, alpha, beta, iter, burn, seed, CI, S0, nu0, mu0, items)
```

Arguments

x	First variable.
y	Second variable.
alpha	Prior true score variance (covariance between tests)
beta	Prior error variance (product of standard deviations minus covariance)
iter	Number of iterations for the Gibbs sampler.
burn	Number of samples to burn in.
seed	Seed for the Gibbs sampler
CI	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).
S0	Prior variance covariance matrix.
nu0	Prior degrees of freedom for inverse Wishart prior distribution.
mu0	Prior means for each column.
items	Number of test items.

Value

Returns median posterior estimates of the variance covariance matrix.

Examples

```
set.seed(999)
your_data=mvrnorm(n=15,mu=c(0,0),Sigma=matrix(c(4,5,5,9),nrow=2,ncol=2))
x=your_data[,1]
y=your_data[,2]
Mu0=c(0,0)
Sigma0=matrix(c(1,0.6,0.6,4),nrow=2,ncol=2)
Nu0=3-1
brxx_Cor(x=x,y=y,iter=5000,burn=2500,seed=999,CI=0.95,
mu0=Mu0,S0=Sigma0,nu0=Nu0,items=10)
```

---

brxx_Cor_general *brxx_Cor_general: Bayesian Estimation of Reliability from Correlation, General Form*

Description

This function estimates reliability from correlation given the correlation estimate.

Usage

```
brxx_Cor_general(cor, alpha, beta, CI, items)
```

Arguments

cor	Correlation estimate.
alpha	Prior true score variance.
beta	Prior error variance.
CI	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).
items	Number of test items.

Value

Returns estimated median and quantile based credible limits for reliability.

Examples

```
brxx_Cor_general(cor=0.85,alpha=3.51,beta=1.75,CI=0.95,items=10)
```

---

brxx_general — *brxx_general: Bayesian Estimation of Reliability from Variance Estimates*

Description

This function estimates reliability from given true and error variance estimates.

Usage

```
brxx_general(a, b, alpha, beta, CI, items)
```

Arguments

a	True score variance estimate.
b	Error variance estimate.
alpha	Prior true score variance.
beta	Prior error variance.
CI	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).
items	Number of test items.

Value

Returns estimated median and quantile based credible limits for reliability.

Examples

```
a=18.7
b=3.3
alpha=3.51
beta=1.75
brxx_general(a=a,b=b,alpha=alpha,beta=beta,CI=0.95,items=10)
```

---

brxx_ICC *brxx_ICC: Bayesian Estimation of Reliability from ICC*

Description

This function estimates reliability from intraclass correlation coefficient

Usage

```
brxx_ICC(mod, alpha, beta, CI, items)
```

Arguments

mod	A mixed effects model object estimated by blmer.
alpha	Prior true score variance (subject variance)
beta	Prior error variance (residual variance)
CI	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).
items	Number of test items.

Value

Returns estimated median and quantile based credible limits for ICC.

Examples

```
your_data_wide=mvrnorm(20,c(0,0),matrix(c(1,0.8,0.8,1),nrow=2,ncol=2))
your_data_long=c(as.vector(your_data_wide[,1]),as.vector(your_data_wide[,2]))
time=c(rep(0,20),rep(1,20))
id=c(rep(1:20,2))
mod=blmer(your_data_long~time+(1|id))
brxx_ICC(mod=mod,alpha=3.51,beta=1.75,CI=0.95,items=10)
```

brxx_ICC_general — *brxx_ICC_general: Bayesian Estimation of Reliability from ICC, General Form*

Description

This function estimates reliability from intraclass correlation given correlation.

Usage

```
brxx_ICC_general(WS, Resid, alpha, beta, CI, items)
```

Arguments

`WS`	Within subjects variance estimate.
`Resid`	Residual variance estimate.
`alpha`	Prior true score variance.
`beta`	Prior error variance.
`CI`	Credible interval quantile, as a decimal (ie, for 95 percent, 0.95).
`items`	Number of test items.

Value

Returns estimated median and quantile based credible limits for reliability.

Examples

```
WS=20.4
Resid=3.6
alpha=3.51
beta=1.75
brxx_ICC_general(WS=WS,Resid=Resid,alpha=alpha,beta=beta,CI=0.95,items=5)
```

prep — *prep: Prepare Data File for Bayesian Analysis*

Description

This function prepares data for analysis using Stan factor analysis code.

Usage

```
prep(data, nfactors, Prior)
```

Arguments

`data`	N by P data matrix.
`nfactors`	Number of factors to extract.
`Prior`	Prior loading matrix.

Value

Returns a formatted data file for use with Stan MCMC sampler.

Examples

```
set.seed(999)
your_data=data.frame(mvrnorm(n=20,mu=c(0,0,0,0,0),
                            Sigma=matrix(c(4,2,2,2,2,
                                           2,4,2,2,2,
                                           2,2,4,2,2,
                                           2,2,2,4,2,
                                           2,2,2,2,4),
                                         nrow=5, ncol=5)))
colnames(your_data)=c("x1","x2","x3","x4","x5")
your_data_miss=matrix(ncol=5,nrow=20)
for (i in 1:20){
for (p in 1:5){
   your_data_miss[i,p]=ifelse(runif(1,0,1)<0.2,NA,your_data[i,p])
 }
}
formatted_data=prep(your_data_miss,nfactors=3)
```

`process` *process: rotates and calulates reliability for Stan output*

Description

This function processes Stan loading matrix data.

Usage

```
process(Loading_Matrix, Format, Rotate)
```

Arguments

`Loading_Matrix`	S by P*Q matrix of loading samples.
`Format`	list formatted data file provided for Stan
`Rotate`	If Q>1, rotation (for options, see GPArotation package)

Value

Returns rotated loadings, uniqueness, communality, and reliability.

Examples

```
## Not run:
your_data_s=standardize(your_data)
formatted_data=prep(your_data_s,nfactors=3)
out=sampling(model, data=formatted_data, iter=5000, seed=999)
res=as.matrix(out)
unpacked=unpack(Samples=res,Format=formatted_data)

processed=process(Loading_Matrix=unpacked$Loading_Matrix,
                  Format=formatted_data,
                  Rotate="oblimin")
## End(Not run)
```

`scree` *scree: Scree Plot with Pairwise Complete Cases*

Description

This function provides a scree plot when data may be missing.

Usage

```
scree(data)
```

Arguments

`data`	N by P data matrix.

Value

Returns eigenvalues and scree plot.

Examples

```
set.seed(999)
your_data=data.frame(mvrnorm(n=20,mu=c(0,0,0,0,0),
                             Sigma=matrix(c(4,2,2,2,2,
                                            2,4,2,2,2,
                                            2,2,4,2,2,
                                            2,2,2,4,2,
                                            2,2,2,2,4),
                                          nrow=5, ncol=5)))
colnames(your_data)=c("x1","x2","x3","x4","x5")
```

```
your_data_miss=matrix(ncol=5,nrow=20)
for (i in 1:20){
 for (p in 1:5){
   your_data_miss[i,p]=ifelse(runif(1,0,1)<0.2,NA,your_data[i,p])
 }
}
scree(your_data_miss)
```

`standardize` *standardize: Standardization of Data Matrix*

Description

This function standardizes an N by P data matrix, as is strongly recommended before using any of the brxx reliability estimation functions

Usage

```
standardize(data)
```

Arguments

`data`	N by P data matrix.

Value

Returns an item level standardized data matrix.

Examples

```
set.seed(999)
your_data=data.frame(mvrnorm(n=20,mu=c(0,0,0,0,0,0,0),
                            Sigma=matrix(c(4,2,2,2,2,2,2,
                                           2,4,2,2,2,2,2,
                                           2,2,4,2,2,2,2,
                                           2,2,2,4,2,2,2,
                                           2,2,2,2,4,2,2,
                                           2,2,2,2,2,4,2,
                                           2,2,2,2,2,2,4),
                                         nrow=7, ncol=7)))
your_data_miss=matrix(ncol=5,nrow=20)
for (i in 1:20){
for (p in 1:5){
   your_data_miss[i,p]=ifelse(runif(1,0,1)<0.2,NA,your_data[i,p])
 }
}
standardize(your_data_miss)
```

summarize — *summarize: Summarize Stan output as median, SD, and HPD quantiles*

Description

This function converts raw MCMC sample data into matrix formatted summaries

Usage

```
summarize(Samples, nrow, ncol, CI)
```

Arguments

`Samples`	S by theta matrix of sampled parameter estimates.
`nrow`	Number of rows of target summary matrix
`ncol`	Number of columns of target summary matrix
`CI`	Creddible interval quantile, as a decimal (ie, for 95 percent, 0.95)

Value

Returns median, SD, and HPD CI limits

Examples

```
## Not run:
your_data_s=standardize(your_data)
formatted_data=prep(your_data_s,nfactors=3)
out=sampling(model, data=formatted_data, iter=5000, seed=999)
res=as.matrix(out)
unpacked=unpack(Samples=res,Format=formatted_data)
processed=process(Loading_Matrix=unpacked$Loading_Matrix,
                  Format=formatted_data,
                  Rotate="oblimin")

summarize(processed$Loadings,
         nrow=Formatted_data$P,
         ncol=Formatted_data$Q)$Table
summarize(processed$Communality,
         nrow=Formatted_data$P,
         ncol=1)$Table
summarize(processed$Uniqueness,
         nrow=Formatted_data$P,
         ncol=1)$Table
summarize(processed$G_Factor,
         nrow=Formatted_data$P,
         ncol=1)$Table
summarize(processed$Interfactor_Correlations,
```

```
         nrow=Formatted_data$Q,
         ncol=Formatted_data$Q)$Table
summarize(processed$Omega,
         nrow=1,
         ncol=1)$Table
summarize(unpacked$Tau_Matrix,
         nrow=Formatted_data$P,
         ncol=1)$Table
## End(Not run)
```

unpack *unpack: Unpack Stan output for factor analysis samples from Stan*

Description

This function unpacks raw Stan samples output.

Usage

```
unpack(Samples, Format)
```

Arguments

`Samples`	S by theta matrix of sample parameter estimates.
`Format`	list formatted data file provided for Stan

Value

Returns four matrices:

1). S by Q latent score matrix, x.

2). S by Q*P loading matrix, lambda.

3). S by P mean matrix, tau.

4). S by P loading variance matrix, alpha.

Examples

```
## Not run:
your_data_s=standardize(your_data)
formatted_data=prep(your_data_s,nfactors=3)
out=sampling(model, data=formatted_data, iter=5000, seed=999)
res=as.matrix(out)


unpacked=unpack(Samples=res,Format=formatted_data)
## End(Not run)
```

Index