Package ‘merror’

August 29, 2023

Version 3.0

Date 2023-09-01

Author Richard A. Bilonick <rabilonick@gmail.com>

Maintainer Richard A. Bilonick <rabilonick@gmail.com>

Title Accuracy and Precision of Measurements

Description N>=3 methods are used to measure each of n items.
The data are used to estimate simultaneously systematic error (bias)
and random error (imprecision). Observed measurements for each method
or device are assumed to be linear functions of the unknown true values
and the errors are assumed normally distributed. Pairwise calibration
curves and plots can be easily generated. Unlike the 'ncb.od' function,
the 'omx' function builds a one-factor measurement error model using 'OpenMx'
and allows missing values, uses full information maximum likelihood to
estimate parameters, and provides both likelihood-based and bootstrapped
confidence intervals for all parameters, in addition to Wald-type intervals.

License GPL (>= 2)

Imports graphics, grDevices, stats, utils, OpenMx

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-29 13:20:02 UTC

R topics documented:

`alpha.beta.sigma` *Build an alpha-beta-sigma Matrix for Use with the cplot Function*

Description

Creates a $3 \times N$ (no. of methods) `matrix` consisting of the estimated alphas, betas, and imprecision sigmas for use with the `cplot` function.

Usage

```
alpha.beta.sigma(x)
```

Arguments

`x`	A $k \times 3$ `data.frame` with parameter estimates in the second column where k is the number of methods $m \times 3$. The estimates should be arranged with the estimated $m - 1$ betas first, followed by the m residual variances, the variance of the true values, the $m - 1$ alphas, the mean of the true values. The `omx` function returns the fitted model in `fit` from which parameter estimates can be retrieved. See the examples below.

Details

This is primarily a helper function used by the `omx` function.

Value

A $3 \times N$ `matrix` consisting of alphas on the first row, betas on the second row, followed by raw imprecision sigmas.

See Also

cplot, omx.

Examples

```
## Not run:
library(OpenMx)
library(merror)
data(pm2.5)
pm <- pm2.5

# OpenMx does not like periods in data column names
names(pm) <- c('ms_conc_1','ws_conc_1','ms_conc_2','ws_conc_2','frm')

# Fit model with FRM sampler as reference
omxfit <- omx(data=pm[,c(5,1:4)],bs.q=c(0.025,0.5,0.975),reps=100)

# Extract the estimates
alpha.beta.sigma(summary(omxfit$fit)$parameters[,c(1,5,6)])

# Make a calibration plot
cplot(pm[,c(5,1:4)],1,2,alpha.beta.sigma=
  alpha.beta.sigma(summary(omxfit$fit)$parameters[,c(1,5,6)]))

# The easier way
cplot(pm[,c(5,1:4)],1,2,alpha.beta.sigma=omxfit$abs)

## End(Not run)
```

---

beta.bar *Compute the estimates of betas.*

Description

This function is used internally to compute the estimates of betas.

Usage

```
beta.bar(x)
```

Arguments

x	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts.

Details

See Jaech, p. 184.

Value

A vector of length N (no. of methods) containing the estimates of beta.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

cb.pd, ncb.od,lrt

Examples

```
data(pm2.5)
beta.bar(pm2.5) # estimate betas (accuracy parameter)
```

---

cb.pd *Compute accuracy estimates and maximum likelihood estimates of precision for the constant bias measurement error model using paired data.*

Description

Compute accuracy estimates and maximum likelihood estimates of precision for the constant bias measurement error model using paired data.

Usage

```
cb.pd(x, conf.level = 0.95, M = 40)
```

Arguments

x	n (no. of items) x N (no. of methods) matrix or data.frame containing the measurements. N must be >= 3 and n > N.
conf.level	Chosen onfidence level.
M	Maximum no.of iterations to reach convergence.

Details

Measurement Error Model:

x[i,k] = alpha[i] + beta[i]*mu[k] + epsilon[i,k]

where x[i,k] is the measurement by the ith method for the kth item, i = 1 to N, k = 1 to n, mu[k] is the true value for the kth item, epsilon[i,k] is the Normally distributed random error with variance sigma[i] squared for the ith method and the kth item, and alpha[i] and beta[i] are the accuracy parameters for the ith method.

The imprecision for the ith method is sigma[i]. If all alphas are zeroes and all betas are ones, there is no bias. If all betas equal 1, then there is a constant bias. Otherwise there is a nonconstant bias.

ME (method of moments estimator) and MLE are the same for N=3 instruments except for a factor of (n-1)/n: MLE = (n-1)/n * ME

Using paired differences forces Constant Bias model (beta[1] = beta[2] = ... = beta[N]). Also, the process variance CANNOT be estimated.

Value

`conf.level`	Confidence level used.
`sigma.table`	Table of accuracy and precision estimates and confidence intervals.
`n.items`	No. of items.
`N.methods`	No. of methods
`Grubbs.initial.sigma2`	N vector of initial imprecision estimates using Grubbs' method
`sigma2`	N vector of variances that measure the method imprecision.
`sigma2.se2`	N vector of squared standard errors of the estimated imprecisions (variances).
`alpha.cb`	N vector of estimated alphas for constant bias model.
`alpha.ncb`	N vector of estimated alphas for nonconstant bias model
`beta`	N vector of hypothesized betas for the constant bias model - all ones.
`df`	N vector of estimated degrees of freedom.
`chisq.low`	N vector of chi-square values for the lower tail (used to compute the ci upper bound).
`chisq.low`	N vector of chi-square values for the upper tail (used to compute the ci lower bound).
`lb`	N vector of lower bounds for confidence intervals
`ub`	N vector of upper bounds for confidence intervals

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

ncb.od, lrt

Examples

```
data(pm2.5)
cb.pd(pm2.5)
```

---

cplot   *Scatter plot of observations for a pair of devices with calibration curve.*

Description

Creates a scatter plot for any pair of observations in the data.frame and superimposes the calibration curve.

Usage

```
cplot(df, i, j, leg.loc="topleft", regress=FALSE, lw=1, t.size=1, alpha.beta.sigma=NULL)
```

Arguments

`df`	n (no. of items) x N (no. of methods) matrix or data.frame containing the measurements. N must be >= 3 and n > N.
`i`	Select column i for device i.
`j`	Select column j for device j not equal to i.
`leg.loc`	Location of the legend.
`regress`	If TRUE, add both naive regression lines (for comparison only).
`lw`	Line widths.
`t.size`	Text size.
`alpha.beta.sigma`	By default, `cplot` computes the bias (alpha, beta) and imprecision (sigma) estimates using ncb.od. You can override this by specifying a 3 x N matrix of values with alpha on the first row, beta on the second row, and sigma on the third row, in the same order as the methods.

Details

By default, `cplot` displays the corresponding calibration curve for devices i and j based on the parameter estimates for alpha, beta, and sigma computed using `ncb.od`. You can overide this calibration curve by providing argument alpha.beta.sigma with different estimates. Both naive regression lines (device i regressed on device j, and device j regressed on device i) by setting "regress=TRUE". Note, however, that the calibration curve will fall somewhere between these two regression lines, depending on the the ratio of the imprecision standard deviations (sigmas). (This may not hold if there are missing measurement data values given that ordinary regression requires deleting any item with one or more missing values.)

Value

Produces a scatter plot with the calibration curve and titles that includes the calibration equation and the scale-bias adjusted imprecision standard deviations.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

`merror.pairs`

Examples

```
library(merror)
data(pm2.5)

# Make various calibration plots for pm2.5 measurements
par(mfrow=c(2,2))
cplot(pm2.5,2,1)
cplot(pm2.5,3,1)
cplot(pm2.5,4,1)
# Add the naive regression lines JUST for comparison
cplot(pm2.5,5,1,regress=TRUE,t.size=0.9)

# This is redundant but illustrates using the
# argument alpha.beta.sigma
a <- ncb.od(pm2.5)$sigma.table$alpha.ncb[1:5]
b <- ncb.od(pm2.5)$sigma.table$beta[1:5]
s <- ncb.od(pm2.5)$sigma.table$sigma[1:5]

alpha.beta.sigma <- t(data.frame(a,b,s))

cplot(pm2.5,2,1,alpha.beta.sigma=alpha.beta.sigma)
cplot(pm2.5,2,1,alpha.beta.sigma=alpha.beta.sigma,regress=TRUE)
```

```
data(pm2.5)

## Not run:
# Use omx function to specify the data for alpha.beta.sigma
pm <- pm2.5

# omx uses OpenMx which does not like periods in data column names
names(pm) <- c('ms_conc_1','ws_conc_1','ms_conc_2','ws_conc_2','frm')

# Fit one-factor measurement error model with FRM sampler as reference
omxfit <- omx(data=pm[,c(5,1:4)],bs.q=c(0.025,0.5,0.975),reps=100)

# Make a calibration plot using the results from omx instead of the default ncb.od
cplot(pm[,c(5,1:4)],1,2,alpha.beta.sigma=omxfit$abs)

## End(Not run)
```

`errors.cb` *Extracts the estimated measurement errors assuming there is a constant bias and using the original data.*

Description

Extracts the estimated measurement errors assuming there is a constant bias and using the original data values.

Usage

```
errors.cb(x)
```

Arguments

`x`	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts. N must be >= 3 and n > N.

Details

Errors should have a zero mean and should be Normally distributed with constant variance for a given method.

Value

`errors.cb`	n x N matrix of estimated measurement errors.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley

.

See Also

`cb.pd`, `ncb.od`,`lrt`

Examples

```
data(pm2.5)
errors.cb(pm2.5)
```

`errors.nb` *Extracts the estimated measurement errors assuming there is no bias and using the original data.*

Description

Extracts the estimated measurement errors assuming there is no bias and using the original data values.

Usage

```
errors.nb(x)
```

Arguments

`x`	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts. N must be >= 3 and n > N.

Details

Errors should have a zero mean and should be Normally distributed with constant variance for a given method.

Value

`errors.nb`	n x N matrix of estimated errors.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

cb.pd, ncb.od,lrt

Examples

```
data(pm2.5)
errors.nb(pm2.5)
```

---

errors.ncb *Extracts the estimated measurement errors assuming there is a non-constant bias and using the original data values.*

Description

Extracts the estimated measurement errors assuming there is a nonconstant bias and using the original data values.

Usage

```
errors.ncb(x)
```

Arguments

x	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts. N must be >= 3 and n > N.

Details

Errors should have a zero mean and should be Normally distributed with constant variance for a given method.

Value

errors.ncb	n x N matrix of estimated errors.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

cb.pd, ncb.od,lrt

Examples

```
data(pm2.5)
errors.ncb(pm2.5)
```

---

lrt *Likelihood ratio test for all betas equalling one.*

Description

Likelihood ratio test statistic - H0: all betas = one.

Usage

```
lrt(x, M = 40)
```

Arguments

x	n (no. of items) x N (no. of methods) matrix or data.frame containing the measurements. N must be greater than 3 and n > N.
M	Maximum no. of iterations for convergence.

Details

See Jaech, pp. 204-205.

Value

n.items	No.of items.
N.methods	No. of methods.'
beta.bars	N vector of estimated betas.
lambda	Chi-square test statistic.
df	Degrees of freedom for the test (N-1).'
p.value	Empirical significance level for the observed test statistic.'

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

ncb.od,cb.pd,pm2.5

Examples

```
data(pm2.5)

lrt(pm2.5) # compare all 5 samplers (4 personal and 1 frm)

lrt(pm2.5[,1:4]) # compare only the personal samplers

stem(lrt(pm2.5)$beta.bars) # examine the estimated betas
```

---

merror.pairs *A modified "pairs" plot with all axes haveing the same range.*

Description

Creates all pairwise scatter plots.

Usage

```
merror.pairs(df,labels=names(df))
```

Arguments

df	n (no. of items) x N (no. of methods) matrix or data.frame containing the measurements. N must be >= 3 and n > N.
labels	Provide labels for each device down the diagnoal of the pairs plot.

Details

Creates all pairwise scatter plots with the same range for all axes and adds the diagonal line denote the "line of equality" or "no bias".).

Value

Produces a scatter plot with the calibration curve and titles that include the calibration equation and the scale-bias adjusted imprecision standard deviations.

Author(s)

Richard A. Bilonick

See Also

panel.merror

Examples

```
data(pm2.5)

# All pairwise plots after square root transformation to Normality
merror.pairs(sqrt(pm2.5))
```

---

mle *Compute maximum likelihood estimates of precision.*

Description

This is an internal function that computes the maximum likelihood estimates of precision for the constant bias model using paired data.

Usage

```
mle(v, r, ni)
```

Arguments

v	Variance-Covariance matrix for the n x N items by methods measurement data.
r	Initial estimates of imprecision, usually Grubbs.
ni	No. of items measured.

Value

An N vector containing the maximum likelihood estimates of precision.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

`mle.se2` *Compute squared standard errors for imprecision estimates for the constant bias model using paired data.*

Description

This is an internal function that computes squared standard errors for imprecision estimates of the constant bias model using paired data.

Usage

```
mle.se2(v, r, ni)
```

Arguments

`v`	Variance-Covariance matrix for the n x N items by methods measurement data.
`r`	Initial estimates of imprecision, usually Grubbs
`ni`	No. of items measured

Details

Computes the squared standard errors for the squared precisions. Before calling this function, compute the MLE's

Value

An N+1 symmetric H matrix. See p. 201 of Jaech, 1985, eq. 6.4.2.

Author(s)

Richard A. Bilonick

References

J. L. Jaech, Statistical Analysis of Measurement Errors, Wiley, New York: 1985.

`ncb.od` *Compute accuracy estimates and maximum likelihood estimates of precision for the nonconstant bias measurement error model using original data.*

Description

Compute accuracy estimates and maximum likelihood estimates of precision for the nonconstant bias measurement error model using original data.

Usage

```
ncb.od(x, beta = beta.bar(x), M = 40, conf.level = 0.95)
```

Arguments

`x`	n (no. of items) x N (no. of methods) `matrix` or `data.frame` containing the measurements. N must be >= 3. Missing values are not allowed.
`beta`	N vector of betas, either estimated by `beta.bar` function or hypothesized.
`M`	Maximum number of iterations for convergence.
`conf.level`	Chosen confidence level which must be greater than zero and less than one.

Details

Measurement Error Model:

x[i,k] = alpha[i] + beta[i]*mu[k] + epsilon[i,k]

where x[i,k] is the measurement by the ith method for the kth item, i = 1 to N, k = 1 to n, mu[k] is the true value for the kth item, epsilon[i,k] is the normally distributed random error with variance sigma[i] squared for the ith method and the kth item, and alpha[i] and beta[i] are the accuracy parameters for the ith method. The product of the betas is constrained to equal one (equivalently, the geometric average of the beta's is constrained to one). When the betas are all equal to one, the average of the alphas equals zero (equivalently, the sum of the alphas is constrained to zero).

The imprecision for the ith method is sigma[i]. If all alphas are zeroes and all betas are ones, there is no bias. If all betas equal 1, then there is a constant bias. If some of the betas differ from one there is a nonconstant bias. Note that the individual betas are not unique - only ratios of the betas are unique. If you divide all the betas by beta_i, then the betas represent the scale bias of the other devices/methods relative to device/method i. Also, when the betas differ from one, the sigmas are not directly comparable because the measurement scales (size of the units) differ. To make the sigmas comparable, divide them by their corresponding beta. This result is shown as bias.adj.sigma.

By using the original data values, the betas can be estimated and also the process variance, that is, the variance of the true values.

Technically, the alphas and betas describe the measurements in terms of the unknown true values (i.e., the unknown true values can be thought of as a latent variable). The "true values" are ALWAYS unknown (unless you have a real, highly accurate reference method/device). The real goal is to calibrate one device/method in terms of another. This is easily accomplished because each measurement is a function of the same unknow true values. By solving the measurement error model (in expectation) for mu and substituting, any two devices/methods i=1 and i=2 can be be related as:

E[x[1,k]] = alpha[1] - alpha[2]*beta[1]/beta[2] + beta[1]/beta[2]*E[x[2,k]]

or equivalently

E[x[2,k]] = alpha[2] - alpha[1]*beta[2]/beta[1] + beta[2]/beta[1]*E[x[1,k]].

Use `cplot` to display this calibration curve and the corresponding scale-bias adjusted imprecision standard deviations.

The `omx` function is to be preferred to `ncb.od`. `omx` can accomodate missing measurement data values and can provide both likelihood-based confidence intervals and bootstrapped intervals for all parameters and relevant functions of parameters.

Value

`conf.level`	Confidence level used.
`sigma.table`	Table of accuracy and precision estimates and confidence intervals.
`n.items`	No. of items.
`N.methods`	No. of methods
`sigma2`	N vector of variances that measure the method imprecision.
`alpha.cb`	N vector of estimated alphas for constant bias model.
`alpha.ncb`	N vector of estimated alphas for nonconstant bias model.
`beta`	N vector of estimated or hypothesized betas.
`df`	N vector of estimated degrees of freedom.
`lb`	N vector of lower bounds for confidence intervals.
`ub`	N vector of upper bounds for confidence intervals.
`bias.adj.sigma`	sigma adjusted for scale bias: sigma/beta.
`H`	N+1 symmetric H matrix (see p. 201, Jaech).
`errors.nb`	n x N matrix of estimated measurement errors for no bias model.
`errors.cb`	n x N matrix of estimated measurement errors for constant bias model.
`errors.ncb`	n x N matrix of estimated measurement errors for nonconstant bias model

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

`cb.pd`, `lrt`

Examples

```
library(merror)
data(pm2.5)
ncb.od(pm2.5)               # nonconstant bias model using original data values
ncb.od(pm2.5,beta=rep(1,5)) # constant bias model using original data values
```

---

omx   *Compute full information maximum likelihood estimates of accuracy and precision for the nonconstant bias measurement error model using 'OpenMx'.*

Description

Compute full information maximum likelihood (FIML) estimates of accuracy (bias) and precision (imprecision) for the nonconstant bias measurement error model. 'OpenMx' functions are used to construct and fit a one latent factor model. Likelihood-based confidence intervals and bootstrapped confidence interval are can be determined for all model parameters and relevant functions of the model parameters.

Usage

```
omx(data, rvEst=rep(1,ncol(data)), mubarEst=mean(data[,1]), interval=0.95,
  reps=500,bs.q=c(0.025,0.975), bs=TRUE)
```

Arguments

`data`	n (no. of items) $\times N$ (no. of methods) `matrix` or `data.frame` containing the measurements. N must be ≥ 3. Missing values are allowed.
`rvEst`	A `vector` of N residual variance starting values, one for each corresponding method.
`mubarEst`	A scalar starting value for estimating the the true mean value.
`interval`	Confidence level for likelihood-based confidence intervals. Should be a scalar value greater than 0 and less than 1.
`reps`	Number of bootstrap samples. Ignored if `bs=FALSE`.
`bs.q`	A `vector` of desired quantiles for bootstrapped samples. Default is `ci.q=c(0.025,0.975)`.
`bs`	A boolean indicating whether bootstapped samples are to be generated. Default is `TRUE`.

Details

Measurement Error Model:

$$x_{ik} = \alpha_i + \beta_i \mu_k + \epsilon_{ik}$$

where x_{ik} is the measurement by the ith of N methods for the kth of n items, $i = 1$ to $N \geq 3$, $k = 1$ to n, μ_k is the true value for the kth item, ϵ_{ik} is the normally distributed random error with variance σ_i^2 for the ith method and the kth item, and α_i and β_i are the accuracy parameters for the ith method. The beta for the first column of `data`) is set to one. The corresponding alpha is set to 0. These constraints or similar are required for model identification.

The imprecision for the ith method is σ_i. If all alphas are zeroes and all betas are ones, there is no bias. If all betas equal 1, then there is a constant bias. If some of the betas differ from one there is a nonconstant bias. Note that the individual betas are not unique - only ratios of the betas are unique. If you divide all the betas by β_i, then the betas represent the scale bias of the other devices/methods relative to device/method i. Also, when the betas differ from one, the sigmas are not directly comparable because the measurement scales (size of the units) differ. To make the sigmas comparable, divide them by their corresponding beta.

Technically, the alphas and betas describe the measurements in terms of the unknown true values (i.e., the unknown true values can be thought of as a latent variable). The "true values" are ALWAYS unknown (unless you have a real, highly accurate reference method/device). The real goal is to calibrate one device/method in terms of another. This is easily accomplished because each measurement is a linear function of the same unknown true values. For methods 1 and 2, the calibration curve is given by:

$$E[x_{1k}] = (\alpha_1 - \alpha_2 \beta_1/\beta_2) + (\beta_1/\beta_2)\, E[x_{2k}]$$

or equivalently

$$E[x_{2k}] = (\alpha_2 - \alpha_1 \beta_2/\beta_1) + (\beta_2/\beta_1)\, E[x_{1k}]$$

.

Use `cplot`, with the alpha.beta.sigma argument specified, to display this calibration curve, calibration equation, and the corresponding scale-bias adjusted imprecision standard deviations.

Note that likelihood confidence intervals and bootstrapped confidence intervals can be returned. Wald-type intervals based on the standard errors are alos available by using the `confint` function on the returned `fit` object. See examples.

Value

`fit`	'OpenMx' fit object containing the results (FIML parameter estimates, etc)
`ci`	Likelihood-based confidence intervals for all parameters and certain useful functions of parameters.
`boot`	Object created by 'mxBootstrap' 'OpenMx' function. Not returned if `bs=FALSE`.
`q.boot`	`data.frame` containing the standard error and quantile estimates based on bootstrapped samples. Not returned if `bs=FALSE`.

abs	A $3 \times N$ matrix of the estimated alphas, betas, and the raw imprecision standard deviations for each of the N methods. The results can be passed to the merror cplot to produce a calibration plot.
bsReps	Number of bootstrapped samples. Default is 500. Not returned if bs=FALSE.
model	The 'OpenMx' one-factor model.

Note

The following names are used to describe the estimates:

1) a1, a2, a3 and so forth denote the alphas (intercept).

2) b1, b2, b3 and so forth denote the betas (scale or slope).

3) ve1, ve2, ve3 and so forth denote the raw (uncorrected for scale) residual random error variances (imprecision variances).

4) se1 denotes the imprecison standard deviation for the reference method.

5) base2, base3 and so forth denote the scale bias-adjusted imprecision standard deviations on the scale of the reference method.

6) mubar is the estimated mean of the true values on the scale of the reference method.

7) sigma2 is the estimated variance of the true values on the scale of the reference method.

8) sigma is the estimated standard deviation of the true values on the scale of the reference method.

Author(s)

Richard A. Bilonick

See Also

ncb.od, alpha.beta.sigma

Examples

```
## Not run:
library(OpenMx)
library(merror)

data(pm2.5)

pm <- pm2.5

# OpenMx does not like periods in data column names
names(pm) <- c('ms_conc_1','ws_conc_1','ms_conc_2','ws_conc_2','frm')

# Fit model with FRM sampler as reference
omxfit <- omx(data=pm[,c(5,1:4)],bs.q=c(0.025,0.5,0.975),reps=100)

# Look at results
summary(omxfit$fit)$parameters[,c(1,5,6)] # Parameter estimates and standard errors
round(omxfit$ci[,1:3],3) # Likelihood-based intervals
```

```
# Estimated standard errors and quantiles based on bootstrapped samples
round(omxfit$q.boot,3)

# Wald-type intervals
#   - note variances not standard deviations and different ordering
confint(omxfit$fit)

omxfit$abs # Use with cplot

# Make a calibration plot using the results from omx instead of the default ncb.od
cplot(pm[,c(5,1:4)],1,2,alpha.beta.sigma=omxfit$abs)

## End(Not run)
```

---

panel.merror *Draw diagonal line (line of equality) on merror.pairs plots*

Description

This functionis used internally by the function merror.pairs.

Usage

```
panel.merror(x,y, ...)
```

Arguments

x	A vector of measurements for one device, of length n.
y	A vector of measurements for another device, of length n.
...	Additional arguments.

Value

Draws the diagonal line that represents the "line of equality", i.e., the "no bias model".

Author(s)

Richard A. Bilonick

See Also

merror.pairs

`pm2.5` *PM 2.5 Concentrations from SCAMP Collocated Samplers*

Description

Five filter-based samplers for measuring PM 2.5 concentrations were collocated and provided 77 complete sets of concentrations. This data was collected by the Stuebenville Comprehensive Air Monitoring Program (SCAMP) to check the accuracy and precision of the instruments.

Usage

```
data(pm2.5)
```

Format

A data frame with 77 sets of PM 2.5 concentrations (micrograms per cubic meter) from the following 5 samplers:

ms.conc.1 - personal sampler 1 - filter MS

ws.conc.1 - personal sampler 1 - filter WS

ms.conc.2 - personal sampler 2 - filter MS

ws.conc.2 - personal sampler 2 - filter WS

frm - Federal Reference Method sampler

Source

Stuebenville Comprehensive Air Monitoring Program (SCAMP)

Examples

```
data(pm2.5)
boxplot(pm2.5)
merror.pairs(pm2.5)

# estimates of accuracy and precision
#   for nonconstant bias model using
#   original data values
ncb.od(pm2.5)
```

`precision.grubbs.cb.pd` *Computes Grubbs' method of moments estimators of precision for the constant bias model using paired differences.*

Description

This is an internal function that computes Grubbs' method of moments estimators of precision for the constant bias model using paired differences

Usage

```
precision.grubbs.cb.pd(x)
```

Arguments

`x`	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts. N must be >= 3 and n > N.

Details

See Jaech 1985, Chapters 3 & 4, p. 144 in particular.

Value

Estimated squared imprecision estimates (variances).

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

`precision.grubbs.ncb.od`, `ncb.od`, `cb.pd`,`lrt`

`precision.grubbs.ncb.od`
Computes Grubbs' method of moments estimators of precision for the nonconstant bias model using original data values.

Description

This is an internal function that computes Grubbs' method of moments estimators of precision for the nonconstant bias model using original data values.

Usage

```
precision.grubbs.ncb.od(x, beta.bar.x = beta.bar(x))
```

Arguments

`x`	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts. N must be >= 3 and n > N.
`beta.bar.x`	Either estimates of beta or hypothesized values (one for each method in an N vector).

Details

See Jaech, p. 184.

Value

Grubbs' method of moments estimates of the squared imprecision (variances).

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

`precision.grubbs.cb.pd`, `ncb.od`, `cb.pd`,`lrt` `precision.mle.ncb.od` *Computes iterative approximation to mle precision estimates for non-constant bias model using original data.*

Description

This is an internal function that computes iterative approximation to mle precision estimates for nonconstant bias model using original data.

Usage

```
precision.mle.ncb.od(x, M = 20, beta.bars = beta.bar(x), jaech.errors = FALSE)
```

Arguments

`x`	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts. N must be >= 3 and n > N.
`M`	Maximum no. of iterations for convergence.
`beta.bars`	Estimates or hypothesized values for the betas.
`jaech.errors`	TRUE replicates the minor error in Jaech's Fortran code to allow comparison with his examples.

Details

Provides iterative approximation to MLE precision estimates for NonConstant Bias model using Original Data. See Jaech, p. 185-186.

Value

`sigma2`	Estimated squared imprecisions (variances) for methods.
`sigma.mu2`	Estimated process variance.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

`precision.grubbs.ncb.od`,`precision.grubbs.cb.pd` process.sd *Compute process standard deviation*

Description

This function computes the process standard deviation and is used internally by the function precision.grubbs.ncb.od.

Usage

```
process.sd(x)
```

Arguments

x	A matrix or numeric data.frame consisting of an n (no. of items) by N (no. of methods) matrix of measuremnts.

Details

The process standard deviation is the standard deviation of the true values uncontaminated by measurement error. See Jaech, p. 185.

Value

A scalar containing the method of moments estimate of the process standard deviation.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

See Also

precision.grubbs.ncb.od

Examples

```
data(pm2.5)
process.sd(pm2.5) # estimate of the sd of the "true values using the method of moments")
```

`process.var.mle` *Compute process variance.*

Description

This is an internal function to compute the process variance.

Usage

```
process.var.mle(sigma2, s, beta.bars, N, n)
```

Arguments

`sigma2`	Estimated imprecisions for each method in an N vector.
`s`	Variance-covariance N x N matrix.
`beta.bars`	Estimates or hypothesized values for the N betas.
`N`	No. of methods.
`n`	No. of items.

Details

See Jaech p. 186 equations 6.37 - 6.3.10.

Value

Estimated process variance.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

`process.var.mle.jaech.err`
Compute process variance but with minor error in Jaech Fortran code.

Description

This is an internal function to compute the process variance that replicates the minor error in Jaech's Fortran code. This allows comparing merror estimates to those shown in Jaech 1985.

Usage

```
process.var.mle.jaech.err(sigma2, s, beta.bars, N, n)
```

Arguments

`sigma2`	Estimated imprecisions for each method in an N vector.
`s`	Variance-covariance N x N matrix.
`beta.bars`	Estimates or hypothesized values for the N betas
`N`	No. of methods.
`n`	No. of items.

Details

See Jaech p. 186 equations 6.37 - 6.3.10. Jaech p. 288 line 2330 has s[i,j] instead of s[j,j]. Jaech p. 288 line 2410 omits "- 1/d2".

Value

Estimated process variance but replicating minor error in Jaech's Fortran code.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

`redshift` *Spectroscopic and Photometric Galaxy Redshift Measurements*

Description

The redshift observations were taken from DEEP 2 Galaxy Redshift Survey.

Usage

```
data(redshift)
```

Format

Redshift measurements are usually denoted by z.

A data frame with one spectroscopic redshift measurement and six different photometric measurements (by researcher) for 1432 galaxies:

z_spec Spectroscopic redshift

z_fink Photometric redshift - S. Finklestein

z_font Photometric redshift - A. Fontana

z_pfor Photometric redshift - J. Pforr

z_salv Photometric redshift - M. Salvator

z_wikl Photometric redshift - T. Wiklind

z_wuyt Photometric redshift - S. Wuyts

Details

Because the photometric methods depend on the same color information, a one-factor measurement error model incuding both the spectroscopic and photomentric measurements would not be a viable model because the photometric measurements would tend to be correlated. A two-factor model would be needed but would require at minimum replicated spectroscopic measurments.

Source

<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140013340.pdf>

References

Newman, Jeffrey A., Michael C. Cooper, Marc Davis, S. M. Faber, Alison L. Coil, Puragra Guhathakurta, David C. Koo et al. "The DEEP2 Galaxy Redshift Survey: Design, observations, data reduction, and redshifts." The Astrophysical Journal Supplement Series 208, no. 1 (2013): 5.

Examples

```
library(OpenMx)
library(merror)

data(redshift)
merror.pairs(redshift)

# estimates of accuracy and precision
#   parameters for a one-factor
#   measurement error model
head(redshift)
merror.pairs(redshift)

## Not run:
red <- omx(redshift[,-1],reps=200) # Drop the spectroscopic measurements

summary(red$fit)
red$ci
red$q.boot

cplot(redshift[,-1],1,2,alpha.beta.sigma=red$abs)

## End(Not run)
```

---

sigma_mle *Computes the ith iteration for computing the squared imprecision estimates.*

Description

This is an internal function that computes the ith iteration for computing the squared imprecision estimates.

Usage

```
sigma_mle(i, s, sigma2, sigma.mu2, beta.bars, N, n)
```

Arguments

`i`	Iteration i.
`s`	Variance-covariance N x N matrix.
`sigma2`	Estimated imprecisions for each method in an N vector
`sigma.mu2`	Estimated process varinace.
`beta.bars`	Estimates or hypothesized values for the N betas.
`N`	No. of methods.
`n`	No. of items.

Details

See Jaech p. 185-186 equations 6.3.1 - 6.3.6.

Value

Estimated squared imprecisions (variances) for the ith iteration.

Author(s)

Richard A. Bilonick

References

Jaech, J. L. (1985) *Statistical Analysis of Measurement Errors*. New York: Wiley.

Index