

# Package ‘metaSEM’

October 10, 2010

**Type** Package

**Title** Meta-Analysis using Structural Equation Modeling

**Version** 0.5-1

**Date** 2010-10-10

**Depends** R (>= 2.11.1), OpenMx

**Imports** Matrix

**Author** Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**Maintainer** Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**Description** The metaSEM package conducts univariate and multivariate meta-analyses using a structural equation modeling (SEM) approach via the OpenMx package. It also implements the two-stage SEM approach to conducting meta-analytic structural equation modeling on correlation/covariance matrices.

**License** GPL (>=2)

**LazyLoad** yes

**LazyData** yes

**URL** <http://courses.nus.edu.sg/course/psycwlm/Internet/metaSEM/>

## R topics documented:

metaSEM-package	2
as.mxMatrix	3
asyCov	4
bdiagMat	5
bdiagRep	6
Becker83	7
Berkey98	8
Cheung09	8
coef	9
Digman97	10
HedgesOlkin85	11
homoStat	12

Hox02 . . . . .	13
is.pd . . . . .	14
list2matrix . . . . .	14
matrix2bdiag . . . . .	15
meta . . . . .	16
print . . . . .	18
readData . . . . .	19
summary . . . . .	21
tssem1 . . . . .	22
vcov . . . . .	23
vec2symMat . . . . .	24
wls . . . . .	25
<b>Index</b>	<b>29</b>

---

metaSEM-package	<i>Meta-Analysis using Structural Equation Modeling</i>
-----------------	---

---

**Description**

The metaSEM package conducts univariate and multivariate meta-analyses using a structural equation modeling (SEM) approach via the OpenMx package. It also implements the two-stage SEM approach to conducting meta-analytic structural equation modeling on correlation/covariance matrices.

**Details**

Package:	metaSEM
Type:	Package
Version:	0.5-1
Date:	2010-10-10
License:	GPL (>=2)
LazyLoad:	yes

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>  
Maintainer: Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**References**

Cheung, M. W. L. (2008). A model for integrating fixed-, random-, and mixed-effects meta-analyses into structural equation modeling. *Psychological Methods*, **13**, 182-202.

Cheung, M. W. L. (2009). Constructing approximate confidence intervals for parameters with structural equation models. *Structural Equation Modeling*, **16**, 267-294.

Cheung, M. W. L. (2010). Modeling multivariate effect sizes with structural equation models. *Manuscript submitted for publication*.

Cheung, M. W. L., & Chan, W. (2004). Testing dependent correlation coefficients via structural equation modeling. *Organizational Research Methods*, **7**, 206-223.

Cheung, M. W. L., & Chan, W. (2005). Meta-analytic structural equation modeling: A two-stage approach. *Psychological Methods*, **10**, 40-64.

Cheung, M. W. L., & Chan, W. (2009). A two-stage approach to synthesizing covariance matrices in meta-analytic structural equation modeling. *Structural Equation Modeling*, **16**, 28-53.

---

as.mxMatrix

Convert a Matrix into MxMatrix-class

---

## Description

It converts a matrix into MxMatrix-class via mxMatrix.

## Usage

```
as.mxMatrix(x, ...)
```

## Arguments

x	A character or numeric matrix
...	Further arguments to be passed to <code>mxMatrix</code> . Please note that <code>type</code> , <code>nrow</code> , <code>ncol</code> , <code>values</code> , <code>free</code> and possibly <code>labels</code> will be created automatically. Thus, these arguments should not be avoided in ...

## Details

The dimensions are the same as x. If there are non-numeric values, it will be treated as a free parameter. If an "\*" is present, the numeric value on the left hand side will be treated as either a fixed value for a fixed parameter or a starting value for a free parameter. For example, "1" for a fixed parameter with "1" as the value and "5\*beta" for a free parameter with "5" as the starting value and "beta" as the label. If it is a matrix of numeric values, there is no free parameters in the output matrix.

## Value

A `MxMatrix-class` object with the same dimensions as x

## Author(s)

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

## See Also

`mxMatrix`

## Examples

```
# a and b are free parameters with starting values and labels
a1 <- matrix(c(1:4, "5*a", 6, "7*b", 8, 9), ncol=3, nrow=3)
a1 <- as.mxMatrix(a1, name="a1")

# Fixed parameters without any labels
a2 <- matrix(1:9, ncol=3, nrow=3)
a2 <- as.mxMatrix(a2, name="a2")

# Free parameters without starting values
a3 <- matrix(c(1:4, "a", 6, "b", 8, 9), ncol=3, nrow=3)
a3 <- as.mxMatrix(a3, name="a3")

# A free parameter without label
a4 <- matrix(c(1:4, "5*", 6, "7*b", 8, 9), ncol=3, nrow=3)
a4 <- as.mxMatrix(a4, name="a4")
```

asyCov

*Asymptotic Covariance Matrix of a Correlation/Covariance Matrix*

## Description

It estimates asymptotic covariance matrices of the correlation/covariance matrix by assuming multivariate normality.

## Usage

```
asyCov(x, n, cor.analysis = TRUE, dropNA = TRUE, as.matrix = TRUE,
       silent = TRUE, suppressWarnings = TRUE, ...)
```

## Arguments

<code>x</code>	A correlation/covariance matrix or a list of correlation/covariance matrices. NA on the variables or other values defined in <code>na.strings</code> will be removed before the analysis. Note that it only checks the diagonal elements of the matrices. If there are missing values, make sure that the diagonals are coded with NA or values defined in <code>na.string</code> .
<code>n</code>	Sample size or a vector of sample sizes
<code>cor.analysis</code>	Logical. The output is either a correlation or covariance matrix.
<code>dropNA</code>	Logical. If it is TRUE, the resultant dimensions will be reduced by dropping the missing variables. If it is FALSE, the resultant dimensions are the same as the input by keeping the missing variables.
<code>as.matrix</code>	Logical. If it is TRUE and <code>x</code> is a list of correlation/covariance matrices with the same dimensions, the asymptotic covariance matrices will be column vectorized and stacked together. If it is FALSE, the output will be a list of asymptotic covariance matrices. Note that if it is TRUE, <code>dropNA</code> will be FALSE automatically. This option is useful when passing the asymptotic covariance matrices to <a href="#">meta</a>
<code>silent</code>	Logical. Argument to be passed to <a href="#">mxRun</a>
<code>suppressWarnings</code>	Logical. If TRUE, warnings are suppressed. Argument to be passed to <a href="#">mxRun</a> .
<code>...</code>	Futher arguments to be passed to <a href="#">mxRun</a>

**Value**

An asymptotic covariance matrix of the vectorized correlation/covariance matrix or a list of these matrices. If `as.matrix=TRUE` and `x` is a list of matrices, the output is a stacked matrix.

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**References**

Cheung, M. W. L., & Chan, W. (2004). Testing dependent correlation coefficients via structural equation modeling. *Organizational Research Methods*, **7**, 206-223.

**Examples**

```
C1 <- matrix(c(1,0.5,0.4,0.5,1,0.2,0.4,0.2,1), ncol=3)
asyCov(C1, n=100)

# Data with missing values
C2 <- matrix(c(1,0.4,NA,0.4,1,NA,NA,NA,NA), ncol=3)
C3 <- matrix(c(1,0.2,0.2,1), ncol=2)

# Output is a list of asymptotic covariance matrices
asyCov(list(C1,C2,C3), n=c(100,50,50), as.matrix=FALSE)

# Output is a stacked matrix of asymptotic covariance matrices
asyCov(list(C1,C2), n=c(100,50), as.matrix=TRUE)
```

---

bdiagMat

---

*Create a Block Diagonal Matrix*


---

**Description**

It creates a block diagonal matrix from a list of numeric or character matrices.

**Usage**

```
bdiagMat(x)
```

**Arguments**

`x`                      A list of numeric or character matrices (or values)

**Value**

A numeric or character block diagonal matrix

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

## References

It was based on a function posted by Scott Chasalow at <http://www.math.yorku.ca/Who/Faculty/Monette/pub/stmp/0827.html>.

## See Also

[bdiagRep](#), [matrix2bdiag](#)

## Examples

```
# Block diagonal matrix of numerics
bdiagMat( list(matrix(1:4,nrow=2,ncol=2),
                 matrix(5:6,nrow=1,ncol=2)) )

# Block diagonal matrix of characters
bdiagMat( list(matrix(letters[1:4],nrow=2,ncol=2),
                 matrix(letters[5:6],nrow=1,ncol=2)) )
```

---

bdiagRep

---

*Create a Block Diagonal Matrix by Repeating the Input*


---

## Description

It creates a block diagonal matrix by repeating the input matrix several times.

## Usage

```
bdiagRep(x, times)
```

## Arguments

x	A numeric or character matrix (or values)
times	Number of times of x to be repeated

## Value

A numeric or character block diagonal matrix

## Author(s)

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

## See Also

[bdiagMat](#), [matrix2bdiag](#)

## Examples

```
# Block diagonal matrix of numerics
bdiagRep( matrix(1:4,nrow=2,ncol=2), 2 )

# Block diagonal matrix of characters
bdiagRep( matrix(letters[1:4],nrow=2,ncol=2), 2 )
```

---

Becker83*Studies on Sex Differences in Conformity Reported by Becker (1983)*

---

## Description

Studies on sex differences in conformity using the fictitious norm group paradigm reported by Becker (1983).

## Usage

```
data(Becker83)
```

## Details

The variables are:

**study** study number

**di** Standardized mean difference

**vi** Sampling variance of the effect size

**percentage** Percentage of male authors

**items** Number of items

## Source

Becker, B. J. (1983, April). Influence again: A comparison of methods for meta-analysis. *Paper presented at the annual meeting of the American Educational Research Association, Montreal.*

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Orlando, FL: Academic Press.

## References

Cheung, M. W. L. (2010). Fixed-effects meta-analyses as multiple-group structural equation models. *Structural Equation Modeling*, **17**, 481-509.

## Examples

```
data(Becker83)
## maybe str(Becker83) ; plot(Becker83) ...
```

---

Berkey98

*Five Published Trails from Berkey et al. (1998)*


---

### Description

Five published trials, reported by Berkey et al. (1998), comparing surgical and non-surgical treatments for medium-severity periodontal disease, one year after treatment.

### Usage

```
data(Berkey98)
```

### Details

The variables are:

**trial** Trial number

**pub\_year** Publication year

**no\_of\_patients** Number of patients

**PD** Patient improvements (mm) in *probing depth*

**AL** Patient improvements (mm) in *attachment level*

**var\_PD** Sampling variance of PD

**cov\_PD\_AL** Sampling covariance between PD and AD

**var\_AL** Sampling variance of AL

### Source

Berkey, C. S., Hoaglin, D. C., Antczak-Bouckoms, A., Mosteller, F. & Colditz, G. A. (1998). Meta-analysis of multiple outcomes by regression with random effects. *Statistics in Medicine*, **17**, 2537-2550.

### Examples

```
data(Berkey98)
## maybe str(Berkey98) ; plot(Berkey98) ...
```

---

Cheung09

*Data Set from TSSEM User's Guide Version 1.11 by Cheung (2009)*


---

### Description

Four studies were selected from the data set used by Cheung and Chan (2005; 2009). Some variables were randomly deleted to illustrate the analysis with missing data.

### Usage

```
data(Cheung09)
```



## Details

A list of data with the following structure:

**data** A list of 4 studies of correlation matrices

**n** A vector of sample sizes

## Source

Cheung, M. W. L. (2009). TSSEM: A LISREL syntax generator for two-stage structural equation modeling (Version 1.11) [Computer software]. Retrieved from <http://courses.nus.edu.sg/course/psycwlm/internet/tssem.zip>.

## References

Cheung, M. W. L., & Chan, W. (2005). Meta-analytic structural equation modeling: A two-stage approach. *Psychological Methods*, **10**, 40-64.

Cheung, M. W. L., & Chan, W. (2009). A two-stage approach to synthesizing covariance matrices in meta-analytic structural equation modeling. *Structural Equation Modeling*, **16**, 28-53.

## Examples

```
data(Cheung09)
## maybe str(Cheung09)
```

---

coef	<i>Extract Parameter Estimates from tssem1, wls and meta Objects</i>
------	--

---

## Description

It extracts the parameter estimates from either `tssem1`, `wls` or `meta` objects.

## Usage

```
## S3 method for class 'tssem1':
coef(object, ...)
## S3 method for class 'wls':
coef(object, ...)
## S3 method for class 'meta':
coef(object, ...)
```

## Arguments

<code>object</code>	An object returned from either class <code>tssem1</code> , class <code>wls</code> or class <code>meta</code>
<code>...</code>	Further arguments; currently none is used

## Value

Parameter estimates for both fixed- and random-effects (if any)

## Author(s)

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**See Also**

[tssem1](#), [wls](#), [meta](#)

**Examples**

```
## Random-effects meta-analysis
modell <- with( Hox02, meta(y=yi, v=vi) )
coef(modell)
```

---

Digman97

*Factor Correlation Matrices of Big Five Model from Digman (1997)*

---

**Description**

Fourteen studies of the factor correlation matrices of the Big Five Model of personality reported by Digman (1997).

**Usage**

```
data(Digman97)
```

**Details**

A list of data with the following structure:

**data** A list of 14 studies of correlation matrices. The variables are Extraversion (E), Agreeableness (A), Conscientiousness (C), Emotional Stability (ES) and Intellect (I)

**n** A vector of sample sizes

**cluster** Types of participants of the studies

**Source**

Digman, J.M. (1997). Higher-order factors of the Big Five. *Journal of Personality and Social Psychology*, **73**, 1246-1256.

**References**

Cheung, M. W. L., & Chan, W. (2005). Classifying correlation matrices into relatively homogeneous subgroups: A cluster analytic approach. *Educational and Psychological Measurement*, **65**, 954-979.

**Examples**

```
data(Digman97)
## maybe str(Digman97) ...
```

---

HedgesOlkin85

*Effects of Open Education Reported by Hedges and Olkin (1985)*


---

## Description

Effects of open education on attitude toward school and on reading achievement reported by Hedges and Olkin (1985).

## Usage

```
data(HedgesOlkin85)
```

## Details

The variables are:

**study** Study number

**d\_att** Standardized mean difference on *attitude*

**d\_ach** Standardized mean difference on *achievement*

**var\_att** Sampling variance of the effect size of *attitude*

**cov\_att\_ach** Sampling covariance between the effect sizes

**var\_ach** Sampling variance of the effect size of *achievement*

## Source

Hedges, L. V., & Olkin, I. (1985). *Statistical methods for meta-analysis*. Orlando, FL: Academic Press.

## References

Cheung, M. W. L. (2010). Fixed-effects meta-analyses as multiple-group structural equation models. *Structural Equation Modeling*, **17**, 481-509.

## Examples

```
data(HedgesOlkin85)
## maybe str(HedgesOlkin85) ; plot(HedgesOlkin85) ...
```

homoStat

*Test Statistic on Homogeneity of Effect Sizes***Description**

A test statistic on the homogeneity of univariate and multivariate effect sizes.

**Usage**

```
homoStat(y, v)
```

**Arguments**

<code>y</code>	A vector of effect size for univariate meta-analysis or a $k \times p$ matrix of effect sizes for multivariate meta-analysis where $k$ is the number of studies and $p$ is the number of effect sizes.
<code>v</code>	A vector of the sampling variance of the effect size for univariate meta-analysis or a $k \times p^*$ matrix of the sampling covariance matrix of the effect sizes for multivariate meta-analysis where $p^* = p(p + 1)/2$ . It is arranged by column major as used by <a href="#">vech</a> . It is assumed that there is no missing value on <code>v</code> . If there are missing values on <code>v</code> due to the missing values on <code>y</code> , the missing values on <code>v</code> will be removed automatically.

**Value**

A list of

<code>Q</code>	Q statistic on the null hypothesis of homogeneity of effect sizes. It has an approximate chi-square distribution under the null hypothesis. In fact, it is a badness-of-fit index.
<code>Q.dof</code>	Degrees of freedom of the Q statistic
<code>pval</code>	p value on the test of homogeneity of effect sizes

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**References**

- Becker, B. J. (1992). Using results from replicated studies to estimate linear models. *Journal of Educational Statistics*, **17**, 341-362.
- Cheung, M. W. L. (2010). Fixed-effects meta-analyses as multiple-group structural equation models. *Structural Equation Modeling*, **17**, 481-509.
- Cochran, W. G. (1954). The combination of estimates from different experiments. *Biometrics*, **10**, 101-129.

**See Also**

[meta](#)

**Examples**

```
with( Hox02, homoStat(yi, vi) )

with( HedgesOlkin85, homoStat(y=cbind(d_att, d_ach),
  v=cbind(var_att, cov_att_ach, var_ach)) )
```

Hox02

*Simulated Effect Sizes Reported by Hox (2002)***Description**

Twenty simulated studies on standardized mean difference and one continuous study characteristic reported by Hox (2002).

**Usage**

```
data(Hox02)
```

**Details**

The variables are:

**study** Study number

**yi** Effect size (standardized mean difference)

**vi** Sampling variance of the effect size

**weeks** Duration of the experimental intervention in terms of weeks

**Source**

Hox, J. J. (2002). *Multilevel analysis: Techniques and applications*. Mahwah, N.J.: Lawrence Erlbaum Associates.

**References**

Cheung, M. W. L. (2008). A model for integrating fixed-, random-, and mixed-effects meta-analyses into structural equation modeling. *Psychological Methods*, **13**, 182-202.

**Examples**

```
data(Hox02)
## maybe str(Hox02) ; plot(Hox02) ...
```

---

`is.pd`*Test Positive Definiteness of a List of Square Matrices*

---

**Description**

It tests the positive definiteness of a square matrix or a list of square matrices. It returns `FALSE` if the matrix is not symmetric. Variables with NA in the diagonals will be removed before the test.

**Usage**

```
is.pd(x, tol = 1e-06)
```

**Arguments**

<code>x</code>	A square matrix or a list of square matrices
<code>tol</code>	Relative tolerance of positiveness of smallest eigenvalue compared to largest eigenvalue. The matrix is considered positive definite if the ratio of the smallest eigenvalue to the largest eigenvalue is larger than <code>tol</code> . See <a href="#">nearPD</a>

**Value**

`TRUE` or `FALSE` or a list of it.

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**Examples**

```
A <- diag(1,3)
is.pd(A)
# TRUE

B <- matrix(c(1,2,2,1), ncol=2)
is.pd(B)
# FALSE

is.pd(list(A, B))
# TRUE FALSE
```

---

`list2matrix`*Convert a List of Symmetric Matrices into a Stacked Matrix*

---

**Description**

It converts a list of symmetric matrices into a stacked matrix. Dimensions of the symmetric matrices have to be the same. It tries to preserve the dimension names if possible. Dimension names will be created if there is no dimension names in the first symmetric matrices.

**Usage**

```
list2matrix(x, diag = FALSE)
```

**Arguments**

**x** A list of  $k$   $p \times p$  symmetric matrices.

**diag** Logical. If it is TRUE, `vech` is used to vectorize the matrices. If it is FALSE, `vechs` is used to vectorize the matrices.

**Value**

A  $k \times p^*$  stacked matrix where  $p^* = p(p-1)/2$  for `diag=FALSE` or  $p^* = p(p+1)/2$  for `diag=TRUE`.

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**Examples**

```
C1 <- matrix(c(1,0.5,0.4,0.5,1,0.2,0.4,0.2,1), ncol=3)
asyCov(C1, n=100)
C2 <- matrix(c(1,0.4,NA,0.4,1,NA,NA,NA,NA), ncol=3)
list2matrix(list(C1, C2))

dimnames(C1) <- list( c("x", "y", "z"), c("x", "y", "z") )
dimnames(C2) <- list( c("x", "y", "z"), c("x", "y", "z") )
list2matrix(list(C1, C2))
```

---

matrix2bdiag

---

*Convert a Matrix into a Block Diagonal Matrix*


---

**Description**

It converts a matrix into a block diagonal matrix.

**Usage**

```
matrix2bdiag(x, ...)
```

**Arguments**

**x** A  $k \times p$  matrix of numerics or characters.

**...** Further arguments to be passed to `vec2symMat`

**Details**

Each row of `x` is converted into a symmetric matrix via `vec2symMat`. Then the list of the symmetric matrices is converted into a block diagonal matrix via a function written by Scott Chasalow posted at <http://www.math.yorku.ca/Who/Faculty/Monette/pub/stmp/0827.html>.

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**See Also**

[vec2symMat](#)

**Examples**

```
matrix2bdiag( matrix(1:18, ncol=6, byrow=TRUE) )

matrix2bdiag( matrix(letters[1:18], ncol=6, byrow=TRUE) )
```

---

meta

*Univariate and Multivariate Meta-Analysis*

---

**Description**

It conducts univariate and multivariate meta-analysis with maximum likelihood estimation method. Mixed-effects meta-analysis can be conducted by including study characteristics as predictors. Equality constraints on intercepts, regression coefficients and variance components can be easily imposed.

**Usage**

```
meta(y, v, x, intercept.constraints, coeff.constraints, RE.constraints,
     RE.startvalues=0.1, RE.lbound = 1e-10, intervals.type = c("z", "LB"),
     suppressWarnings = TRUE, ...)
```

**Arguments**

- |                                    |   |
|------------------------------------|---|
| <code>y</code>                     | A vector of effect size for univariate meta-analysis or a $k \times p$ matrix of effect sizes for multivariate meta-analysis where $k$ is the number of studies and $p$ is the number of effect sizes.  |
| <code>v</code>                     | A vector of the sampling variance of the effect size for univariate meta-analysis or a $k \times p^*$ matrix of the sampling covariance matrix of the effect sizes for multivariate meta-analysis where $p^* = p(p + 1)/2$ . It is arranged by column major as used by <a href="#">vech</a> .   |
| <code>x</code>                     | A predictor or a $k \times m$ matrix of predictors where $m$ is the number of predictors.   |
| <code>intercept.constraints</code> | A $1 \times p$ matrix specifying whether the intercepts of the effect sizes are fixed or free. The default is that the intercepts are free. When there is no predictor, these intercepts are the same as the pooled effect sizes. The format of this matrix follows <a href="#">as.mxMatrix</a> . The parameter estimates will be constrained equally if the labels are the same. |
| <code>coeff.constraints</code>     | A $p \times m$ matrix specifying how the predictors predict the effect sizes. The default is that all $m$ predictors predict all $p$ effect sizes. The format of this matrix follows <a href="#">as.mxMatrix</a> . The parameter estimates will be constrained equally if the labels are the same.  |



<code>RE.constraints</code>	A $p \times p$ matrix specifying the variance components of the random effects. The default is that all covariance/variance components are free. The format of this matrix follows <code>as.mxMatrix</code> . The parameter estimates will be constrained equally if the labels are the same. If a zero matrix is specified, it becomes a fixed-effects meta-analysis.
<code>RE.startvalues</code>	A vector of $p$ starting values on the diagonals of the variance component of the random effects. If only one scalar is given, it will be duplicated across the diagonals. Starting values for the off-diagonals of the variance component are all 0. A $p \times p$ symmetric matrix of starting values is also acceptable.
<code>RE.lbound</code>	A vector of $p$ lower bounds on the diagonals of the variance component of the random effects. If only one scalar is given, it will be duplicated across the diagonals. Lower bounds for the off-diagonals of the variance component are set at NA. A $p \times p$ symmetric matrix of the lower bounds is also acceptable
<code>intervals.type</code>	Either <code>z</code> (default if missing) or <code>LB</code> . If it is <code>z</code> , it calculates 95% Wald confidence intervals (CIs) based on the <code>z</code> statistic. If it is <code>LB</code> , it calculates 95% likelihood-based CIs on the parameter estimates. Note that the <code>z</code> values and their associated <code>p</code> values are based on the <code>z</code> statistic. They are not related to the likelihood-based CIs.
<code>suppressWarnings</code>	Logical. If <code>TRUE</code> , warnings are suppressed. Argument to be passed to <code>mxRun</code> .
<code>...</code>	Further arguments to be passed to <code>mxRun</code>

## Value

An object of class `meta` with a list of

<code>call</code>	Object returned by <code>match.call</code>
<code>data</code>	A data matrix of <code>y</code> , <code>v</code> and <code>x</code>
<code>no.y</code>	No. of effect sizes
<code>no.x</code>	No. of predictors
<code>miss.x</code>	A vector indicating whether the predictors are missing. Studies will be removed before the analysis if they are <code>TRUE</code>
<code>meta.fit</code>	A fitted object returned from <code>mxRun</code>

## Author(s)

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

## References

- Cheung, M. W. L. (2008). A model for integrating fixed-, random-, and mixed-effects meta-analyses into structural equation modeling. *Psychological Methods*, **13**, 182-202.
- Cheung, M. W. L. (2009). Constructing approximate confidence intervals for parameters with structural equation models. *Structural Equation Modeling*, **16**, 267-294.
- Cheung, M. W. L. (2010). Modeling multivariate effect sizes with structural equation models. *Manuscript submitted for publication*.
- Hardy, R. J., & Thompson, S. G. (1996). A likelihood approach to meta-analysis with random effects. *Statistics in Medicine*, **15**, 619-629.

Neale, M. C., & Miller, M. B. (1997). The use of likelihood-based confidence intervals in genetic models. *Behavior Genetics*, **27**, 113-120.

## Examples

```
## Random-effects meta-analysis
attach(Hox02)
summary( meta(y=yi, v=vi) )

## Fixed-effects meta-analysis
summary( meta(y=yi, v=vi, RE.constraints=matrix(0, ncol=1, nrow=1)) )

## Mixed-effects meta-analysis with "weeks" as a predictor
## Use likelihood-based CI
summary( meta(y=yi, v=vi, x=weeks, intervals.type="LB") )
detach(Hox02)

## Multivariate meta-analysis
attach(Berkey98)
summary( meta(y=cbind(PD, AL), v=cbind(var_PD, cov_PD_AL, var_AL)) )

## Multivariate meta-analysis with "publication year-1979" as the predictor
summary( meta(y=cbind(PD, AL), v=cbind(var_PD, cov_PD_AL, var_AL),
  x=scale(pub_year, center=1979)) )

## Multivariate meta-analysis with an equality constraint on regression coefficients
summary( meta(y=cbind(PD, AL), v=cbind(var_PD, cov_PD_AL, var_AL),
  x=scale(pub_year, center=1979),
  coeff.constraints=matrix(c("0.3*Eq_slope", "0.3*Eq_slope"), nrow=2)) )
detach(Berkey98)
```

---

print

*Print Methods for tssem1, wls and meta Objects*

---

## Description

Print methods for either `tssem1`, `wls` or `meta` objects.

## Usage

```
## S3 method for class 'tssem1':
print(x, ...)
## S3 method for class 'wls':
print(x, ...)
## S3 method for class 'meta':
print(x, ...)
```

## Arguments

<code>x</code>	An object returned from either class <code>tssem1</code> , class <code>wls</code> or class <code>meta</code>
<code>...</code>	Further arguments to be passed to <code>summary.default</code>

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**See Also**

[tssem1](#), [wls](#), [meta](#)

---

readData

*Read External Correlation/Covariance Matrices*


---

**Description**

It reads full/lower triangle/stacked vectors of correlation/covariance data into a list of correlation/covariance matrices.

**Usage**

```
readFullMat(file, ...)
readStackVec(file, ...)
readLowTriMat(file, no.var, ...)
```

**Arguments**

file	File name of the data.
no.var	Number of variables in the data.
...	Further arguments to be passed to <a href="#">scan</a> for readLowTriMat and to <a href="#">read.table</a> for readFullMat and readStackVec.

**Value**

A list of correlation/covariance matrices.

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**Examples**

```
## Not run:
# Write two full correlation matrices into a file named "fullmat.dat".
# x2 is missing in the second matrix.
# The content of "fullmat.dat" is
#1.0 0.3 0.4
#0.3 1.0 0.5
#0.4 0.5 1.0
#1.0 NA 0.4
#NA NA NA
#0.4 NA 1.0
cat("1.0 0.3 0.4\n0.3 1.0 0.5\n0.4 0.5 1.0\n1.0 NA 0.4\nNA NA NA\n0.4 NA
1.0",
file="fullmat.dat", sep="")
```

```

# Read the correlation matrices
my.full1 <- readFullMat("fullmat.dat")

my.full1
#$`1`
#      x1  x2  x3
#x1 1.0 0.3 0.4
#x2 0.3 1.0 0.5
#x3 0.4 0.5 1.0
#
#$`2`
#      x1 x2  x3
#x1 1.0 NA 0.4
#x2 NA NA  NA
#x3 0.4 NA 1.0

# Write two lower triangle correlation matrices into a file named "lowertriangle.dat".
# x2 is missing in the second matrix.
# The content of "lowertriangle.dat" is
#1.0
#0.3 1.0
#0.4 0.5 1.0
#1.0
#NA NA
#0.4 NA 1.0
cat("1.0\n0.3 1.0\n0.4 0.5 1.0\n1.0\nNA NA\n0.4 NA 1.0",
    file="lowertriangle.dat", sep="")

# Read the lower triangle correlation matrices
my.lowertri <- readLowTriMat(file = "lowertriangle.dat", no.var = 3)

my.lowertri
#$`1`
#      x1  x2  x3
#x1 1.0 0.3 0.4
#x2 0.3 1.0 0.5
#x3 0.4 0.5 1.0
#
#$`2`
#      x1 x2  x3
#x1 1.0 NA 0.4
#x2 NA NA  NA
#x3 0.4 NA 1.0

# Write two vectors of correlation coefficients based on
# column major into a file named "stackvec.dat".
# x2 is missing in the second matrix.
# The content of "stackvec.dat" is
#1.0 0.3 0.4 1.0 0.5 1.0
#1.0 NA 0.4 NA NA 1.0
cat("1.0 0.3 0.4 1.0 0.5 1.0\n1.0 NA 0.4 NA NA 1.0",
    file="stackvec.dat", sep="")

my.vec <- readStackVec("stackvec.dat")

my.vec
#$`1`

```

```
#      x1  x2  x3
#x1 1.0 0.3 0.4
#x2 0.3 1.0 0.5
#x3 0.4 0.5 1.0
#
# $`2`
#      x1  x2  x3
#x1 1.0 NA 0.4
#x2 NA NA NA
#x3 0.4 NA 1.0

## End(Not run)
```

summary

*Summary Method for tssem1, wls and meta Objects***Description**

It summaries results for either `tssem1`, `wls` or `meta` objects.

**Usage**

```
## S3 method for class 'tssem1':
summary(object, ...)
## S3 method for class 'wls':
summary(object, ...)
## S3 method for class 'meta':
summary(object, ...)
## S3 method for class 'tssem1':
print.summary(x, ...)
## S3 method for class 'wls':
print.summary(x, ...)
## S3 method for class 'meta':
print.summary(x, ...)
```

**Arguments**

<code>object</code>	An object returned from either class <code>tssem1</code> , class <code>wls</code> or class <code>meta</code>
<code>x</code>	An object returned from either class <code>summary.tssem1</code> , class <code>summary.wls</code> or class <code>summary.meta</code>
<code>...</code>	Further arguments to be passed to <code>printCoefmat</code>

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**See Also**

`tssem1`, `wls`, `meta`

tssem1

*First Stage of the Two-Stage Structural Equation Modeling (TSSEM)***Description**

It conducts the first stage analysis of TSSEM by pooling correlation/covariance matrices with a fixed-effects model. The function expects that there is no missing data in the first group.

**Usage**

```
tssem1(my.df, n, start.values, cor.analysis = TRUE,
       suppressWarnings = TRUE, ...)
```

**Arguments**

<code>my.df</code>	A list of correlation/covariance matrices
<code>n</code>	A vector of sample sizes
<code>start.values</code>	A vector of starting values for the pooled correlation/covariance matrix based on column major. If it is missing, <code>.startValues</code> will be used to generate the starting values.
<code>cor.analysis</code>	Logical. The output is either a pooled correlation or a covariance matrix.
<code>suppressWarnings</code>	Logical. If TRUE, warnings are suppressed. Argument to be passed to <a href="#">mxRun</a> .
<code>...</code>	Further arguments to be passed to <a href="#">mxRun</a>

**Value**

<code>call</code>	The matched call
<code>data</code>	A list of correlation/covariance matrices from input
<code>pooledS</code>	The pooled correlation/covariance matrix
<code>acovS</code>	The asymptotic sampling covariance matrix of the pooled correlation/covariance matrix
<code>total.n</code>	Total sample size of all studies
<code>modelMinus2LL</code>	-2LogLikelihood of the model
<code>independentMinus2LL</code>	-2LogLikelihood of the independent model returned by <code>.minus2LL</code>
<code>saturatedMinus2LL</code>	-2LogLikelihood of the saturated model returned by <code>.minus2LL</code>
<code>tssem1.fit</code>	A fitted object returned from <a href="#">mxRun</a>

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

## References

Cheung, M. W. L., & Chan, W. (2005). Meta-analytic structural equation modeling: A two-stage approach. *Psychological Methods*, **10**, 40-64.

Cheung, M. W. L., & Chan, W. (2009). A two-stage approach to synthesizing covariance matrices in meta-analytic structural equation modeling. *Structural Equation Modeling*, **16**, 28-53.

## See Also

[wls](#)

## Examples

```
digman1 <- tssem1(Digman97$data, Digman97$n)
summary(digman1)
```

---

vcov

*Extract Variance-Covariance Matrix of Parameter Estimates from tssem1, wls and meta Objects*

---

## Description

It extracts the variance-covariance matrix of the parameter estimates from either `tssem1`, `wls` or `meta` objects.

## Usage

```
## S3 method for class 'tssem1':
vcov(object, ...)
## S3 method for class 'wls':
vcov(object, ...)
## S3 method for class 'meta':
vcov(object, ...)
```

## Arguments

<code>object</code>	An object returned from either class <code>tssem1</code> , class <code>wls</code> or class <code>meta</code>
<code>...</code>	Further arguments; currently none is used

## Value

A variance-covariance matrix of the parameter estimates for both fixed- and random-effects (if any)

## Author(s)

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

## See Also

[tssem1](#), [wls](#), [meta](#)

**Examples**

```
## Random-effects meta-analysis
modell <- with( Hox02, meta(y=yi, v=vi) )
vcov(modell)
```

vec2symMat

*Convert a Vector into a Symmetric Matrix***Description**

It converts a vector into a symmetric matrix.

**Usage**

```
vec2symMat(x, diag = TRUE)
```

**Arguments**

x	A vector of numerics or characters
diag	Logical. If it is TRUE (the default), the diagonals of the created matrix are replaced by elements of x; otherwise, the diagonals of the created matrix are replaced by "1".

**Value**

A symmetric square matrix based on column major

**Author(s)**

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

**See Also**

[matrix2bdiag](#)

**Examples**

```
vec2symMat(1:6)

vec2symMat(letters[1:6])

vec2symMat(1:6, diag=FALSE)
```



---

wls	<i>Conduct a Correlation/Covariance Structure Analysis with Weighted Least Squares</i>
-----	--

---

## Description

It fits a correlation or covariance structure with weighted least squares (WLS) where the inverse of the asymptotic covariance matrix is used as the weight matrix. `tssem2` conducts the second stage analysis of the two-stage structural equation modeling (TSSEM). `tssem2` is a wrapper of `wls`.

## Usage

```
wls(S, acovS, n, impliedS, matrices, cor.analysis = TRUE,
    intervals.type = c("z", "LB"), suppressWarnings = TRUE, ...)
tssem2(tssem1.obj, impliedS, matrices, intervals.type = c("z", "LB"),
       suppressWarnings = TRUE, ...)
```

## Arguments

<code>tssem1.obj</code>	An object returned from <code>tssem1</code>
<code>S</code>	A $p \times p$ sample correlation/covariance matrix where $p$ is the number of variables
<code>acovS</code>	A $p^* \times p^*$ asymptotic sampling covariance matrix of either <code>vechs</code> ( <code>S</code> ) or <code>vech</code> ( <code>S</code> ) where $p^* = p(p-1)/2$ for correlation matrix and $p^* = p(p+1)/2$ for covariance matrix
<code>n</code>	Sample size
<code>impliedS</code>	Model implied correlation/covariance matrix of an object of either <code>MxMatrix-class</code> or <code>MxAlgebra-class</code>
<code>matrices</code>	A list of matrices used to calculate <code>impliedS</code> . They are objects of either <code>MxMatrix-class</code> or <code>MxAlgebra-class</code>
<code>cor.analysis</code>	Logical. Analysis of correlation or covariance structure. If <code>cor.analysis=TRUE</code> , <code>vechs</code> is used to vectorize <code>S</code> ; otherwise, <code>vech</code> is used to vectorize <code>S</code> .
<code>intervals.type</code>	Either <code>z</code> (default if missing) or <code>LB</code> . If it is <code>z</code> , it calculates 95% Wald confidence intervals (CIs) based on the <code>z</code> statistic. If it is <code>LB</code> , it calculates 95% likelihood-based CIs on the parameter estimates. Please note that the <code>z</code> values and their associated <code>p</code> values are based on the <code>z</code> statistic. They are not related to the likelihood-based CIs.
<code>suppressWarnings</code>	Logical. If <code>TRUE</code> , warnings are suppressed. Argument to be passed to <code>mxRun</code> .
<code>...</code>	Futher arguments to be passed to <code>mxRun</code>

## Value

An object of class `wls` with a list of

<code>call</code>	The matched call
<code>noObservedStat</code>	Number of observed statistics
<code>n</code>	Sample size

```
indepModelChisq      Chi-square statistic of the independent model returned by .indepwlsChisq
indepModelDf         Degrees of freedom of the independent model returned by .indepwlsChisq
wls.fit              A fitted object returned from mxRun
```

### Author(s)

Mike W.-L. Cheung <mikewlcheung@nus.edu.sg>

### References

- Bentler, P.M., & Savalei, V. (2010). Analysis of correlation structures: current status and open problems. In Kolenikov, S., Thombs, L., & Steinley, D. (Eds.). *Recent Methodological Developments in Social Science Statistics* (pp. 1-36). Hoboken, NJ: Wiley.
- Cheung, M. W. L., & Chan, W. (2005). Meta-analytic structural equation modeling: A two-stage approach. *Psychological Methods*, **10**, 40-64.
- Cheung, M. W. L., & Chan, W. (2009). A two-stage approach to synthesizing covariance matrices in meta-analytic structural equation modeling. *Structural Equation Modeling*, **16**, 28-53.
- Joreskog, K. G., Sorbom, D., Du Toit, S., & Du Toit, M. (1999). *LISREL 8: New Statistical Features*. Chicago: Scientific Software International.

### See Also

[tssem1](#)

### Examples

```
#### Analysis of correlation structure
R1 <- matrix(c(1.00, 0.22, 0.24, 0.18,
               0.22, 1.00, 0.30, 0.22,
               0.24, 0.30, 1.00, 0.24,
               0.18, 0.22, 0.24, 1.00), ncol=4, nrow=4)

n <- 1000
acovR1 <- asyCov(R1, n)
## One-factor CFA model- P1: Factor variance; L1: Factor loadings
P1 <- mxMatrix("Full", ncol=1, nrow=1, value=1, free=FALSE, name="P1")
L1 <- mxMatrix("Full", ncol=1, nrow=4, free=TRUE, name="L1")
impliedR1 <- mxAlgebra(L1 %&% P1, name="impliedR1")
wls.fit1 <- wls(S=R1, acovS=acovR1, n=n, impliedS=impliedR1,
               matrices=c(P1, L1), cor.analysis=TRUE)
summary(wls.fit1)

#### Multiple regression analysis with RAM specification
## Variables in R2: y, x1, x2
R2 <- matrix(c(1.00, 0.22, 0.24,
               0.22, 1.00, 0.30,
               0.24, 0.30, 1.00,
               0.18, 0.22, 0.24), ncol=3, nrow=3)
acovR2 <- asyCov(R2, n)

## A2: Regression coefficients
#   y x1 x2
#y   F T T
```

```

#x1 F F F
#x2 F F F
A2 <- mxMatrix("Full", ncol=3, nrow=3, byrow=TRUE,
               free=c(FALSE, rep(TRUE, 2), rep(FALSE, 6)), name="A2")

## S2: Covariance matrix of free parameters
## Note that the diagonal elements are not involved in
## the analysis of correlation structure
#   y x1 x2
#y   F F F
#x1  F F F
#x2  F T F
S2 <- mxMatrix("Stand", ncol=3, nrow=3, free=c(FALSE, FALSE, TRUE), name="S2")

## Identity matrix
Id <- mxMatrix("Iden", ncol=3, nrow=3, name="Id")

## Model implied correlation matrix: (Id-A2)^-1 %*% S2 %*% ((Id-A2)^-1)'
impliedR2 <- mxAlgebra( solve(Id-A2) %*% S2, name="impliedR2")

wls.fit2 <- wls(S=R2, acovS=acovR2, n=n, impliedS=impliedR2,
               matrices=c(A2, S2, Id), cor.analysis=TRUE)
summary(wls.fit2)

#### Analysis of covariance structure
S3 <- matrix(c(1.50, 0.22, 0.24, 0.18,
              0.22, 1.60, 0.30, 0.22,
              0.24, 0.30, 1.80, 0.24,
              0.18, 0.22, 0.24, 1.30), ncol=4, nrow=4)

n <- 1000
acovS3 <- asyCov(S3, n, cor.analysis=FALSE)
P3 <- mxMatrix("Full", ncol=1, nrow=1, value=1, free=FALSE, name="P3")
L3 <- mxMatrix("Full", ncol=1, nrow=4, value=c(0.3, 0.4, 0.5, 0.4),
               free=TRUE, name="L3")
E3 <- mxMatrix("Diag", ncol=4, nrow=4, value=0.2, free=TRUE, name="E3")
impliedS3 <- mxAlgebra(L3 %*% P3 + E3, name="impliedS3")

## Use likelihood-based CI
wls.fit3 <- wls(S=S3, acovS=acovS3, n=n, impliedS=impliedS3,
               matrices=c(P3, L3, E3), cor.analysis=FALSE,
               intervals.type="LB")
summary(wls.fit3)

#### Example of tssem2
digman1 <- tssem1(Digman97$data, Digman97$n)

P4 <- mxMatrix("Stand", ncol=2, nrow=2, value=.2, free=TRUE, name="P4")
L4 <- mxMatrix("Full", ncol=2, nrow=5, value=c(0,.3,.3,.3,0,.3,0,0,0,.3),
               free=c(FALSE,TRUE,TRUE,TRUE,FALSE,TRUE,FALSE,FALSE,FALSE,TRUE), name="L4")
impliedR4 <- mxAlgebra(L4 %*% P4, name="impliedR4")

digman2 <- tssem2(digman1, impliedS=impliedR4, matrices=c(P4, L4))
summary(digman2)

```



# Index

## \*Topic **datasets**

Becker83, [6](#)  
Berkey98, [7](#)  
Cheung09, [8](#)  
Digman97, [9](#)  
HedgesOlkin85, [10](#)  
Hox02, [12](#)

## \*Topic **meta-analysis**

asyCov, [3](#)  
homoStat, [11](#)  
meta, [15](#)

## \*Topic **package**

metaSEM-package, [1](#)

## \*Topic **summary**

coef, [9](#)  
print, [17](#)  
summary, [20](#)  
vcov, [22](#)

## \*Topic **tssem**

tssem1, [21](#)  
wls, [24](#)

## \*Topic **utilities**

as.mxMatrix, [2](#)  
bdiagMat, [5](#)  
bdiagRep, [6](#)  
is.pd, [13](#)  
list2matrix, [14](#)  
matrix2bdiag, [14](#)  
readData, [18](#)  
vec2symMat, [23](#)

as.mxMatrix, [2](#), [16](#)

asyCov, [3](#)

bdiagMat, [5](#), [6](#)

bdiagRep, [5](#), [6](#)

Becker83, [6](#)

Berkey98, [7](#)

Cheung09, [8](#)

coef, [9](#)

Digman97, [9](#)

HedgesOlkin85, [10](#)

homoStat, [11](#)

Hox02, [12](#)

is.pd, [13](#)

list2matrix, [14](#)

match.call, [16](#)

matrix2bdiag, [5](#), [6](#), [14](#), [23](#)

meta, [4](#), [9](#), [12](#), [15](#), [18](#), [21–23](#)

metaSEM (*metaSEM-package*), [1](#)

metaSEM-package, [1](#)

MxAlgebra-class, [24](#)

mxMatrix, [2](#), [3](#)

MxMatrix-class, [3](#), [24](#)

mxRun, [4](#), [16](#), [21](#), [22](#), [24](#), [25](#)

nearPD, [13](#)

print, [17](#)

print.summary.meta (*summary*), [20](#)

print.summary.tssem1 (*summary*), [20](#)

print.summary.wls (*summary*), [20](#)

printCoefmat, [21](#)

read.table, [18](#)

readData, [18](#)

readFullMat (*readData*), [18](#)

readLowTriMat (*readData*), [18](#)

readStackVec (*readData*), [18](#)

scan, [18](#)

summary, [20](#)

summary.meta (*summary*), [20](#)

summary.tssem1 (*summary*), [20](#)

summary.wls (*summary*), [20](#)

tssem1, [9](#), [18](#), [21](#), [21–25](#)

tssem2 (*wls*), [24](#)

vcov, [22](#)

vec2symMat, [15](#), [23](#)

vech, [11](#), [14](#), [15](#), [24](#)

vechs, [14](#), [24](#)

wls, [9](#), [18](#), [21–23](#), [24](#)