

# Package ‘R2MLwiN’

January 30, 2024

**Version** 0.8-9

**Date** 2024-01-30

**Title** Running 'MLwiN' from Within R

**Depends** R (>= 3.5.0),

**Imports** methods, stats, stats4, Matrix, foreign (>= 0.8-46),  
R2WinBUGS, digest (>= 0.6.5), texreg, foreach, parallel,  
doParallel, coda (>= 0.16-1), lattice, memisc, broom, tibble

**Suggests** doBy, car, lmtest, mitools, reshape

**Description** An R command interface to the 'MLwiN' multilevel  
modelling software package.

**License** GPL (>= 2)

**URL** <http://www.bristol.ac.uk/cmm/software/r2mlwin/>,<https://r-forge.r-project.org/projects/r2mlwin/>,<http://www.bristol.ac.uk/cmm/software/mlwin/>

**SystemRequirements** MLwiN, (OpenBUGS, WinBUGS optional)

**Encoding** UTF-8

**RoxygenNote** 7.3.0

**LazyData** true

**NeedsCompilation** no

**Author** Zhengzheng Zhang [aut, cre],  
Chris Charlton [aut],  
Richard Parker [aut],  
George Leckie [aut],  
William Browne [aut]

**Maintainer** Zhengzheng Zhang <zhengzheng236@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-01-30 03:20:16 UTC

**R topics documented:**

alevchem	4
augment.mlwinfitIGLS	5
augment.mlwinfitMCMC	6
bang	6
bang1	8
BD	10
bes83	11
caterpillar	13
caterpillarR	14
coef.mlwinfitIGLS-method	15
coef.mlwinfitMCMC-method	16
coef.mlwinfitIGLS	16
coef.mlwinfitMCMC	17
cvd	17
deviance.mlwinfitIGLS	19
df.residual.mlwinfitIGLS	19
df2matrix	20
diag1	21
double2singlePrecision	22
extract.mlwinfitIGLS-method	23
extract.mlwinfitMCMC-method	23
fitted.mlwinfitIGLS	24
fitted.mlwinfitMCMC	25
formula.mlwinfitIGLS	25
formula.mlwinfitMCMC	26
Formula.translate	26
Formula.translate.compat	28
fysio	30
gcsecomp1	32
gcsemv1	33
getSummary.mlwinfitIGLS	34
getSummary.mlwinfitMCMC	35
glance.mlwinfitIGLS	35
glance.mlwinfitMCMC	36
height	36
hungary1	38
jspmix1	39
lips1	40
lmdp	43
logLik.mlwinfitIGLS-method	44
logLik.mlwinfitIGLS	45
matrix2df	45
MCSE	46
mlwin2bugs	47
mlwinfitIGLS-class	49
mlwinfitMCMC-class	50

mmec	52
nobs.mlwinfitIGLS	54
nobs.mlwinfitMCMC	54
predCurves	55
predict.mlwinfitIGLS	56
predict.mlwinfitMCMC	57
predLines	58
print.mlwinfitIGLS	59
print.mlwinfitMCMC	60
prior2macro	61
rats	62
reading1	63
residuals.mlwinfitIGLS	65
residuals.mlwinfitMCMC	65
runMLwiN	66
show.mlwinfitIGLS-method	76
show.mlwinfitMCMC-method	76
show.mlwinfitIGLS	77
show.mlwinfitMCMC	77
sixway	78
summary.mlwinfitIGLS-method	79
summary.mlwinfitMCMC-method	80
summary.mlwinfitIGLS	80
summary.mlwinfitMCMC	81
tidy.mlwinfitIGLS	81
tidy.mlwinfitMCMC	82
trajectories	82
tutorial	83
Untoggle	85
update.mlwinfitIGLS-method	86
update.mlwinfitMCMC-method	87
update.mlwinfitIGLS	88
update.mlwinfitMCMC	89
vcov.mlwinfitIGLS-method	89
vcov.mlwinfitMCMC-method	90
vcov.mlwinfitIGLS	90
vcov.mlwinfitMCMC	91
wage1	91
write.IGLS	93
write.MCMC	97
ws2foreign	105
xc	106
xc1	107
[,mlwinfitIGLS,ANY,ANY,ANY-method	109
[,mlwinfitMCMC,ANY,ANY,ANY-method	109

---

alevchem

*Chemistry A-level results from one exam board*

---

## Description

Chemistry A-level results from one exam board; subset from Yang & Woodhouse, 2001. See also Rasbash et al. (2012) and Browne (2012).

## Usage

alevchem

## Format

A data frame with 2166 observations on the following 8 variables:

**lea** Local Education Authority ID.

**estab** Establishment (institution) ID.

**pupil** Pupil ID.

**a\_point** A-level point score (an ordered factor with levels: F, E, D, C, B, A).

**gcse\_tot** Total GCSE point score.

**gcse\_no** Number of GCSEs taken.

**cons** Constant of ones

**gender** Pupil's gender (a factor with levels: male, female).

## Details

The alevchem dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009).

## Source

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) *A User's Guide to MLwiN Version 2.26*. Centre for Multilevel Modelling, University of Bristol.

Yang, M., Woodhouse, G. (2001) Progress from GCSE to A and AS level: institutional and gender differences, and trends over time. *British Educational Research Journal* 27: 245-267.

**Examples**

```
## Not run:

data(alevchem, package = "R2MLwiN")

alevchem$gcseav <- alevchem$gcse_tot/alevchem$gcse_no - 6

# Note: Establishment codes on their own do not uniquely identify schools.
# Schools are instead uniquely identified by LEA code, establishment ID
# combination. Thus, here we generated a unique school ID.

alevchem$school <- as.numeric(factor(paste0(alevchem$lea, alevchem$estab)))

(mymodel <- runMLwiN(logit(a_point, cons, 6) ~ 1 + gcseav[1:5] + I(gcseav^2)[1:5] +
  gender[1:5] + (1[1:5] + gcseav[1:5] | school),
  D = "Ordered Multinomial", estoptions = list(EstM = 1), data = alevchem))

## End(Not run)
```

---

`augment.mlwinfitIGLS` *Augment data frame with information derived from the model fit (broom package).*

---

**Description**

Augment data frame with information derived from the model fit (broom package).

**Usage**

```
## S3 method for class 'mlwinfitIGLS'
augment(x, data = x@frame, newdata = NULL, type.predict, type.residuals, ...)
```

**Arguments**

<code>x</code>	An <code>mlwinfitIGLS</code> -class model.
<code>data</code>	original data onto which columns should be added
<code>newdata</code>	new data to predict on, optional
<code>type.predict</code>	Type of prediction to compute
<code>type.residuals</code>	Type of residuals to compute
<code>...</code>	Other arguments.

**See Also**

[augment](#)

---

`augment.mlwinfitMCMC` *Augment data frame with information derived from the model fit (broom package).*

---

### Description

Augment data frame with information derived from the model fit (broom package).

### Usage

```
## S3 method for class 'mlwinfitMCMC'
augment(x, data = x@data, newdata = NULL, type.predict, type.residuals, ...)
```

### Arguments

<code>x</code>	An <code>mlwinfitMCMC-class</code> model.
<code>data</code>	original data onto which columns should be added
<code>newdata</code>	new data to predict on, optional
<code>type.predict</code>	Type of prediction to compute
<code>type.residuals</code>	Type of residuals to compute
<code>...</code>	Other arguments.

### See Also

[augment](#)

---

`bang` *Sub-sample from the 1989 Bangladesh Fertility Survey (see Huq & Cleveland, 1990)*

---

### Description

A subset of data from the 1989 Bangladesh Fertility Survey, consisting of 2867 women across 61 districts.

### Usage

`bang`

**Format**

A data frame with 2867 observations on the following 12 variables:

**woman** Identifying code for each woman (level 1 unit).

**district** Identifying code for each district (level 2 unit).

**use** Contraceptive use status at time of survey; a factor with levels Not\_using and Using.

**use4** Contraceptive use status and method (a factor with levels: Sterilization, Modern\_reversible\_method, Traditional\_method, Not\_using\_contraception).

**lc** Number of living children at time of survey; a factor with ordered levels None, One\_child, Two\_children, Three\_plus.

**age** Age of woman at time of survey (in years), centred on sample mean of 30 years.

**urban** Type of region of residence; levels are Rural and Urban.

**educ** Woman's level of education (a factor with ordered levels None, Lower\_primary, Upper\_primary, Secondary\_and\_above).

**hindu** Woman's religion; levels are Muslim and Hindu.

**d\_lit** Proportion of women in district who are literate.

**d\_pray** Proportion of Muslim women in district who pray every day (a measure of religiosity).

**cons** Constant of ones.

**Details**

The bang dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), and is a subset of data from the 1989 Bangladesh Fertility Survey (Huq and Cleland, 1990) used by Rasbash et al. (2012) as an example when fitting logistic models for binary and binomial responses. The full sample was analysed in Amin et al. (1997).

**Source**

Amin, S., Diamond, I., Steele, F. (1997) Contraception and religiosity in Bangladesh. In: G. W. Jones, J. C. Caldwell, R. M. Douglas, R. M. D'Souza (eds) *The Continuing Demographic Transition*, 268–289. Oxford: Oxford University Press.

Huq, N. M., Cleland, J. (1990) *Bangladesh fertility survey, 1989*. Dhaka: National Institute of Population Research and Training (NIPORT).

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) *A User's Guide to MLwiN Version 2.26*. Centre for Multilevel Modelling, University of Bristol.

**See Also**

See m1mRev package for an alternative format of the same dataset, with fewer variables.

**Examples**

```

## Not run:

data(bang, package = "R2MLwiN")

bang$use4 <- relevel(bang$use4, 4)

# Change contrasts if wish to avoid warning indicating that, by default,
# specified contrasts for ordered predictors will be ignored by runMLwiN
# (they will be fitted as "contr.treatment" regardless of this setting). To
# enable specified contrasts, set allowcontrast to TRUE (this will be the
# default in future package releases).
my_contrasts <- options("contrasts")$contrasts
options(contrasts = c(ordered = "contr.treatment",
                      ordered = "contr.treatment"))

# As an alternative to changing contrasts, can instead use C() to specify
# contrasts for ordered predictors in formula object, e.g.:

# F1 <- log(use4, cons) ~ 1 + C(lc, "contr.treatment") + (1 | district)

# (mymodel <- runMLwiN(Formula = F1,
#                       D = "Unordered Multinomial",
#                       estoptions = list(EstM = 1, nonlinear = c(1, 2)),
#                       data = bang,
#                       allowcontrast = TRUE))

F1 <- log(use4, cons) ~ 1 + lc + (1 | district)

(mymodel <- runMLwiN(Formula = F1,
                    D = "Unordered Multinomial",
                    estoptions = list(EstM = 1, nonlinear = c(1, 2)),
                    data = bang))

# Change contrasts back to pre-existing:
options(contrasts = my_contrasts)

## End(Not run)

```

---

bang1

---

*Sub-sample from the 1989 Bangladesh Fertility Survey*


---

**Description**

A subset of data from the 1989 Bangladesh Fertility Survey, consisting of 1934 women across 60 districts.



**Usage**

bang1

**Format**

A data frame with 1934 observations on the following 11 variables:

**woman** Identifying code for each woman (level 1 unit).

**district** Identifying code for each district (level 2 unit).

**use** Contraceptive use status at time of survey; a factor with levels Not\_using and Using.

**lc** Number of living children at time of survey; an ordered factor with levels None, One\_child, Two\_children, Three\_plus.

**age** Age of woman at time of survey (in years), centred on sample mean of 30 years.

**urban** Type of region of residence; a factor with levels Rural and Urban.

**educ** Woman's level of education; an ordered factor with levels None, Lower\_primary, Upper\_primary, Secondary\_and\_above.

**hindu** Woman's religion; a factor with levels Muslim and Hindu.

**d\_illit** Proportion of women in district who are literate.

**d\_pray** Proportion of Muslim women in district who pray every day (a measure of religiosity).

**cons** A column of ones. If included as an explanatory variable in a regression model (e.g. in MLwiN), its coefficient is the intercept.

**Details**

The bang1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), and is a subset of data from the 1989 Bangladesh Fertility Survey (Huq and Cleland, 1990) used by Browne (2012) as an example when fitting logistic models for binary and binomial responses. The full sample was analysed in Amin et al. (1997).

**Source**

Amin, S., Diamond, I., Steele, F. (1997) Contraception and religiosity in Bangladesh. In: G. W. Jones, J. C. Caldwell, R. M. Douglas, R. M. D'Souza (eds) *The Continuing Demographic Transition*, 268–289. Oxford: Oxford University Press.

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Huq, N. M., Cleland, J. (1990) *Bangladesh fertility survey, 1989*. Dhaka: National Institute of Population Research and Training (NIPORT).

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

**See Also**

See m1mRev package for an alternative format of the same dataset, with fewer variables.

## Examples

```
## Not run:

data(bang1, package = "R2MLwiN")

bang1$denomb <- 1

# Change contrasts if wish to avoid warning indicating that, by default,
# specified contrasts for ordered predictors will be ignored by runMLwiN
# (they will be fitted as "contr.treatment" regardless of this setting). To
# enable specified contrasts, set allowcontrast to TRUE (this will be the
# default in future package releases).
my_contrasts <- options("contrasts")$contrasts
options(contrasts = c(ordered = "contr.treatment",
                      ordered = "contr.treatment"))

# As an alternative to changing contrasts, can instead use C() to specify
# contrasts for ordered predictors in formula object, e.g.:

# F1 <- logit(use, denomb) ~ 1 + age + C(lc, "contr.treatment") + urban +
#   (1 + urban | district)

# (mymodel <- runMLwiN(Formula = F1,
#                       D = "Binomial",
#                       estoptions = list(EstM = 1),
#                       data = bang1,
#                       allowcontrast = TRUE))

F1 <- logit(use, denomb) ~ 1 + age + lc + urban + (1 + urban | district)

(mymodel <- runMLwiN(Formula = F1,
                    D = "Binomial",
                    estoptions = list(EstM = 1),
                    data = bang1))

# Change contrasts back to pre-existing:
options(contrasts = my_contrasts)

## End(Not run)
```

---

 BD

*Calculates Brooks-Draper diagnostic*


---

## Description

An internal function, for use in [sixway](#), which calculates the Brooks-Draper diagnostic, based on an unpublished paper by David Draper. It estimates the length of a Markov chain required to produce a

mean estimate to  $k$  significant figures with a given accuracy ( $\alpha$ ). See Browne (2012) for further details.

### Usage

```
BD(est, var, rho, k = 2, alpha = 0.05)
```

### Arguments

<code>est</code>	Numeric scalar for the mean of the distribution
<code>var</code>	Numeric scalar for the variance of the distribution
<code>rho</code>	The first lag (i.e. after zero) of the auto-correlation function (ACF) diagnostic
<code>k</code>	Integer scalar corresponding to the number of significant figures (defaults to 2)
<code>alpha</code>	Numeric scalar indicating the desired accuracy (defaults to 0.05)

### Value

The Brooks-Draper diagnostic statistic is returned.

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

### References

Browne, W.J. (2012) MCMC Estimation in MLwiN, v2.26. Centre for Multilevel Modelling, University of Bristol.

### See Also

[sixway](#)

---

bes83

*Subsample from British Election Study, '83.*

---

### Description

Subsample from British Election Study, consisting of 800 voters across 110 areas.

### Usage

```
bes83
```

## Format

A data frame with 800 observations on the following 10 variables:

**voter** Voter identifier.

**area** Identifier for voters' constituencies.

**defence** Score on a 21 point scale of attitudes towards nuclear weapons with low scores indicating disapproval of Britain possessing them. This variable is centred about its mean.

**unemp** Score on a 21 point scale of attitudes towards unemployment with low scores indicating strong opposition and higher scores indicating a preference for greater unemployment if it results in lower inflation. This variable is centred about its mean.

**taxes** Score on a 21 point scale of attitudes towards tax cuts with low scores indicating a preference for higher taxes to pay for more government spending. This variable is centred about its mean.

**privat** Score on a 21 point scale of attitudes towards privatization of public services with low scores indicating opposition. This variable is centred about its mean.

**votecons** If respondent voted Conservative; a factor with levels Other and Voted\_Conservative.

**cons** This variable is constant (= 1) for all voters.

**denom** This variable is constant (= 1) for all voters.

## Details

The bes83 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009). See Heath et al (1996), and also Rasbash et al (2012) and Browne (2012).

## Source

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Heath, A., Yang, M., Goldstein, H. (1996). Multilevel analysis of the changing relationship between class and party in Britain 1964-1992. *Quality and Quantity*, 30:389-404.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) *A User's Guide to MLwiN Version 2.26*. Centre for Multilevel Modelling, University of Bristol.

## Examples

```
## Not run:
```

```
data(bes83, package = "R2MLwiN")
```

```
(mymodel <- runMLwiN(logit(votecons, cons) ~ 1 + defence + unemp + taxes + privat + (1 | area),
  D = "Binomial", estoptions = list(EstM = 1), data = bes83))
```

```
## End(Not run)
```

---

caterpillar	<i>Draws a caterpillar plot (in MLwiN style).</i>
-------------	---

---

### Description

A convenient wrapper for the `plot` function with the addition of error bars, e.g. to create caterpillar plots.

### Usage

```
caterpillar(  
  y,  
  x,  
  qtlow,  
  qtup,  
  xlab = "",  
  ylab = "",  
  xlim = NULL,  
  ylim = NULL,  
  main = ""  
)
```

### Arguments

<code>y</code>	A numerical vector specifying the y coordinates (e.g. the means or medians); see <a href="#">plot.default</a> .
<code>x</code>	A numerical vector specifying the x coordinates; see <a href="#">plot.default</a> .
<code>qtlow</code>	A numerical vector (e.g. of lower-quantiles) to be used to plot lower error bars.
<code>qtup</code>	A numerical vector (e.g. of upper-quantiles) to be used to upper plot error bars.
<code>xlab</code>	A label for the x axis. This is empty by default; see <a href="#">plot.default</a> .
<code>ylab</code>	A label for the y axis. This is empty by default; see <a href="#">plot.default</a> .
<code>xlim</code>	The x limits ( <code>x1</code> , <code>x2</code> ) of the plot. Note that <code>x1 &gt; x2</code> is allowed and leads to a 'reversed axis'. The default value, <code>NULL</code> , indicates that the range of the finite values to be plotted should be used; see <a href="#">plot.default</a> .
<code>ylim</code>	The y limits of the plot; see <a href="#">plot.default</a> .
<code>main</code>	A main title for the plot; see <a href="#">plot.default</a> .

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol, U.K.

### See Also

[caterpillarR](#)

## Examples

```
## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved where R2MLwiN defaults to:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

# Example using tutorial dataset
data(tutorial, package = 'R2MLwiN')
(mymodel <- runMLwiN(normexam ~ 1 + (1 | school) + (1 | student),
                    estoptions = list(resi.store = TRUE),
                    data = tutorial))

# For each school, calculate the CIs...
residuals <- mymodel$residual$lev_2_resi_est_Intercept
residualsCI <- 1.96 * sqrt(mymodel$residual$lev_2_resi_var_Intercept)
residualsRank <- rank(residuals)
rankno <- order(residualsRank)

caterpillar(y = residuals[rankno], x = 1:65, qtlow = (residuals - residualsCI)[rankno],
            qtup = (residuals + residualsCI)[rankno], xlab = 'Rank', ylab = 'Intercept')

## End(Not run)
```

---

caterpillarR

*Draws caterpillar plots of the residuals from a `mlwinfitIGLS`-class or `mlwinfitMCMC`-class object, at a chosen level of the multilevel model.*

---

## Description

Uses [qqmath](#) in the **lattice** package to draw Quantile-Quantile plots of the residuals at a chosen level of a multilevel model against a theoretical distribution.

## Usage

```
caterpillarR(resi, lev = 2)
```

## Arguments

resi	A <code>mlwinfitIGLS</code> -class or <code>mlwinfitMCMC</code> -class object (model must be fitted specifying <code>resi.store = TRUE</code> and including 'variances' or 'sampling' in the list of <code>resioptions</code> (both in <code>estoptions</code> ) to not return an error).
lev	An integer scalar specifying the level of a multilevel model for which to produce a plot for.

**Value**

See [qqmath](#).

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**See Also**

[caterpillar](#), [qqmath](#)

**Examples**

```
## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved where R2MLwiN defaults to:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

# Example using tutorial dataset
data(tutorial, package = 'R2MLwiN')
mymodel <- runMLwiN(normexam ~ 1 + (1 | school) + (1 | student),
                   estoptions = list(resi.store = TRUE),
                   data = tutorial)

# Caterpillar plot
caterpillarR(mymodel['residual'], lev = 2)

## End(Not run)
```

---

coef,mlwinfitIGLS-method

*Extract the coefficient vector from "mlwinfitIGLS" objects*

---

**Description**

Extract the coefficient vector from "mlwinfitIGLS" objects

**Usage**

```
## S4 method for signature 'mlwinfitIGLS'
coef(object, ...)
```

**Arguments**

object            An [mlwinfitIGLS-class](#) object  
 ...                Other arguments

**See Also**

[coef-methods](#)

coef,mlwinfitMCMC-method

*Extract the coefficient vector from "mlwinfitMCMC" objects*

**Description**

Extract the coefficient vector from "mlwinfitMCMC" objects

**Usage**

```
## S4 method for signature 'mlwinfitMCMC'
coef(object, ...)
```

**Arguments**

object            An [mlwinfitMCMC-class](#) object  
 ...                Other arguments

**See Also**

[coef-methods](#)

coef.mlwinfitIGLS

*Extract the coefficient vector from "mlwinfitIGLS" objects*

**Description**

Extract the coefficient vector from "mlwinfitIGLS" objects

**Usage**

```
## S3 method for class 'mlwinfitIGLS'
coef(object, ...)
```

**Arguments**

object            An [mlwinfitIGLS-class](#) object  
 ...                Other arguments



**See Also**[coef](#)

---

coef.mlwinfitMCMC	<i>Extract the coefficient vector from "mlwinfitMCMC" objects</i>
-------------------	---

---

**Description**

Extract the coefficient vector from "mlwinfitMCMC" objects

**Usage**

```
## S3 method for class 'mlwinfitMCMC'  
coef(object, ...)
```

**Arguments**

object	An <code>mlwinfitMCMC</code> -class object
...	Other arguments

**See Also**[coef](#)

---

cvd	<i>Data from the 1998 Scottish Health Survey on cardiovascular disease status of 8804 respondents</i>
-----	---

---

**Description**

Data from the 1998 Scottish Health Survey, with 8804 respondents aged between 18 and 64. The outcome, `cvddef`, is a self-report of a doctor-diagnosed cardiovascular disease (CVD) condition (angina, diabetes, hypertension, acute myocardial infarction, etc.). This is a binary response, whether (1) or not (0) respondents have CVD condition.

**Usage**

```
cvd
```

## Format

A data frame with 8804 observations on the following 9 variables:

**age** Age.

**sex** Gender (factor with levels: male, female).

**sc** Social class (factor with levels: 12 (social class 1 and 2), 3 (social class 3), 45 (social class 4 and 5)).

**cvddef** Self-reported cardiovascular disease (0 = does not have condition, 1 = has condition)

**carstair** Carstairs score.

**smoke** Smoking frequency (factor with levels: lite (<10 a day), mod (10-19 a day), hvy (20+ a day), ex (ex-smoker), nevr (never smoked)).

**id** Respondent identifier.

**area** Postcode sector

## Details

The cvd dataset is one of the example datasets analysed in Leyland and Groenewegen (2020), and provided with the multilevel-modelling software package MLwiN (Charlton et al., 2024), as cvd\_data.

## Source

Charlton, C., Rasbash, J., Browne, W.J., Healy, M. and Cameron, B. (2024) *MLwiN Version 3.09* Centre for Multilevel Modelling, University of Bristol.

Leyland A.H. (2005) Socioeconomic gradients in the prevalence of cardiovascular disease in Scotland: the roles of composition and context. *J Epidemiol Community Health* 59:799–803

Leyland, A.H., Groenewegen, P.P. (2020). Untangling Context and Composition. In: *Multilevel Modelling for Public Health and Health Services Research*. Springer, Cham. doi:10.1007/9783-030348014\_13

## Examples

```
## Not run:

data(cvd, package = "R2MLwiN")

# Example taken from Leyland and Groenewegen (2020)

F1 <- logit(cvddef) ~ 1 + I(age^3) + I(age^3):I(log(age)) +
  sex + sex:I(age^3) + sex:I(age^3):I(log(age)) +
  (1 | area)

(mod_MQL1 <- runMLwiN(Formula = F1,
                      D = "Binomial",
                      data = cvd))
```

```
(mod_PQL2 <- runMLwiN(Formula = F1,
  D = "Binomial",
  data = cvd,
  estoptions = list(
    nonlinear = c(N = 1, M = 2),
    startval = list(FP.b = mod_MQL1@FP,
      FP.v = mod_MQL1@FP.cov,
      RP.b = mod_MQL1@RP,
      RP.v = mod_MQL1@RP.cov))))

## End(Not run)
```

---

deviance.mlwinfitIGLS *Returns the deviance from "mlwinfitIGLS" objects.*

---

### Description

Returns the deviance from "mlwinfitIGLS" objects.

### Usage

```
## S3 method for class 'mlwinfitIGLS'
deviance(object, ...)
```

### Arguments

object	An <a href="#">mlwinfitIGLS-class</a> object
...	Other arguments

### See Also

[deviance](#)

---

df.residual.mlwinfitIGLS

*Returns the residual degrees-of-freedom extracted from "mlwinfit-IGLS" objects.*

---

### Description

Returns the residual degrees-of-freedom extracted from "mlwinfitIGLS" objects.

### Usage

```
## S3 method for class 'mlwinfitIGLS'
df.residual(object, ...)
```

**Arguments**

object            An `mlwinfitIGLS-class` object.  
 ...              Other arguments

**See Also**

[nobs](#), [coef](#)

---

df2matrix	<i>Translates a data.frame, formatted for use in multiple membership modelling in MLwiN, to a matrix.</i>
-----------	---

---

**Description**

Translates a `data.frame`, in a form usable by MLwiN for multiple membership models, into a `matrix`. The `data.frame` needs to contain (a) columns with membership IDs (e.g. first row of which might be 2, 3, 5, 6, 0, 0) and (b) columns containing weights (e.g. first row of which might be 0.25, 0.25, 0.25, 0.25, 0, 0; in this example the first row of resulting matrix would be 0, 1, 1, 0, 1, 1).

**Usage**

```
df2matrix(data, idcols, weightcols)
```

**Arguments**

data            A `data.frame` object.  
 idcols         String vector of the identifier column names.  
 weightcols     String vector of the weight column names.

**Value**

An adjacency matrix as returned by `sparseMatrix`.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol, U.K.

**See Also**

[matrix2df](#)

---

diag1

*Examination dataset*

---

### Description

Examination data for 907 students within 18 schools.

### Usage

diag1

### Format

A data frame with 907 observations on the following 9 variables:

**school** School identifier.

**sex** Pupil gender.

**vrq** Verbal Reasoning quotient.

**ilea** O-level/CSE examination results.

**type** School type: a factor with levels Comprehensive and Grammar.

**pupil** Pupil identifier.

**cons** Constant (=1).

**n\_ilea** O-level/CSE examination results (normal scores).

**n\_vrq** Verbal Reasoning quotient (normal scores).

### Details

The diag1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), originally analysed in Aitkin & Longford (1986), and described further in Rasbash et al. (2012).

### Source

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

Rasbash, J., Steele, F., Browne, W.J., Goldstein, H. (2012) *A User's Guide to MLwiN v2.26*. University of Bristol: Centre for Multilevel Modelling.

Aitkin, M. & Longford, N. (1986). Statistical modelling in school effectiveness studies (with discussion). *Journal of the Royal Statistical Society, Series A*, 149:1-43.

## Examples

```
## Not run:

data(diag1, package = "R2MLwiN")

(mymodel <- runMLwiN(n_ilea ~ 1 + n_vrq + (1 + n_vrq | school) + (1 | pupil),
  estoptions = list(resi.store = TRUE, resioptions = c("standardised",
    "leverage", "influence", "deletion")), data = diag1))

## End(Not run)
```

---

double2singlePrecision

*Converts numerical values from double precision to single precision.*

---

## Description

This function converts numeric column(s) of a data frame object, matrix or vector from double precision to single precision, e.g. to avoid a warning from MLwiN which currently only stores data in single precision.

## Usage

```
double2singlePrecision(x)
```

## Arguments

x                    A [data.frame](#) object, matrix or vector to be converted. Column(s) of these objects will be ignored during conversion if they are not numeric.

## Value

An object of numerical values in single precision will be returned.

## Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

---

extract,mlwinfitIGLS-method

*Extract coefficients and GOF measures from a statistical object (texreg package).*

---

### Description

Extract coefficients and GOF measures from a statistical object (texreg package).

### Usage

```
## S4 method for signature 'mlwinfitIGLS'  
extract(  
  model,  
  include.nobs = TRUE,  
  include.loglik = TRUE,  
  include.deviance = TRUE,  
  ...  
)
```

### Arguments

model	An <a href="#">mlwinfitIGLS-class</a> model.
include.nobs	should the number of observations be reported?
include.loglik	should the log-likelihood be reported?
include.deviance	should the deviance be reported?
...	Other arguments.

### See Also

[extract](#)

---

extract,mlwinfitMCMC-method

*Extract coefficients and GOF measures from a statistical object.*

---

### Description

Extract coefficients and GOF measures from a statistical object.

**Usage**

```
## S4 method for signature 'mlwinfitMCMC'
extract(
  model,
  include.nobs = TRUE,
  include.dbar = TRUE,
  include.dthetabar = TRUE,
  include.pd = TRUE,
  include.dic = TRUE,
  ...
)
```

**Arguments**

model	An <a href="#">mlwinfitMCMC-class</a> model.
include.nobs	should the number of observations be reported?
include.dbar	should the Dbar be reported?
include.dthetabar	should the D(thetabar) be reported?
include.pd	should the pD be reported?
include.dic	should the DIC be reported?
...	Other arguments.

**See Also**

[extract](#)

---

fitted.mlwinfitIGLS    *Returns the fitted values from "mlwinfitIGLS" objects.*

---

**Description**

Returns the fitted values from "mlwinfitIGLS" objects.

**Usage**

```
## S3 method for class 'mlwinfitIGLS'
fitted(object, ...)
```

**Arguments**

object	An <a href="#">mlwinfitIGLS-class</a> object.
...	Other arguments.

**See Also**

[fitted.values](#)



---

fitted.mlwinfitMCMC *Returns the fitted values from "mlwinfitMCMC" objects.*

---

### Description

Returns the fitted values from "mlwinfitMCMC" objects.

### Usage

```
## S3 method for class 'mlwinfitMCMC'  
fitted(object, ...)
```

### Arguments

object	An <a href="#">mlwinfitMCMC-class</a> object.
...	Other arguments

### See Also

[fitted.values](#)

---

formula.mlwinfitIGLS *"mlwinfitIGLS" model formula*

---

### Description

"mlwinfitIGLS" model formula

### Usage

```
## S3 method for class 'mlwinfitIGLS'  
formula(x, env = parent.frame(), ...)
```

### Arguments

x	See <a href="#">formula</a>
env	See <a href="#">formula</a>
...	Other arguments; see <a href="#">formula</a>

---

```
formula.mlwinfitMCMC  "mlwinfitMCMC" model formula
```

---

**Description**

"mlwinfitMCMC" model formula

**Usage**

```
## S3 method for class 'mlwinfitMCMC'
formula(x, env = parent.frame(), ...)
```

**Arguments**

x	See <a href="#">formula</a>
env	See <a href="#">formula</a>
...	Other arguments; see <a href="#">formula</a>

---

```
Formula.translate  An internal function to translate an R formula into an R list object.
```

---

**Description**

A model formula, as a formula object written in R-type syntax, is translated into an R list object.

**Usage**

```
Formula.translate(Formula, D = "Normal", indata)
```

**Arguments**

Formula	A <a href="#">formula</a> object specifying a multilevel model. See <a href="#">formula</a> for notes on general usage, and <a href="#">runMLwiN</a> for further details.
D	A character string/vector specifying the type of distribution to be modelled, which can include 'Normal' (the default), 'Binomial', 'Poisson', 'Negbinom', 'Unordered Multinomial', 'Ordered Multinomial', 'Multivariate Normal', or 'Mixed'. In the case of the latter, 'Mixed' precedes the response types which also need to be listed in D, e.g. <code>c('Mixed', 'Normal', 'Binomial')</code> ; these need to be listed in the same order to which they are referred to in the Formula object.
indata	A data.frame object containing the data to be modelled. Optional (can <a href="#">attach</a> as an alternative) but recommended.

**Value**

Outputs an R list object, which is then used as the input for `write.IGLS` or `write.MCMC`.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**See Also**

`runMLwiN`, `write.IGLS`, `write.MCMC`; for function allowing back-compatibility with Formula syntax used in older versions of **R2MLwiN** (<0.8.0) see `Formula.translate.compat`.

**Examples**

```
## Not run:
# NB: See demo(package = 'R2MLwiN') for a wider range of examples.
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

# Two-level random intercept model with student (level 1) nested within
# school (level 2) and standlrt added to the fixed part.
# Importantly, the ordering of school and student reflects their hierarchy,
# with the highest level (school) specified first.
# E.g. see demo(UserGuide04)
data(tutorial, package = 'R2MLwiN')
(mymodel1 <- runMLwiN(normexam ~ 1 + standlrt + (1 | school) + (1 | student),
  data = tutorial))

# Adding a random slope
(mymodel2 <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | school)
  + (1 | student), data = tutorial))

# Exploring complex level 1 variation
# E.g. see demo(UserGuide07)
(mymodel3 <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | school)
  + (1 + standlrt | student), data = tutorial))

# Logit link with cons specified as denominator
# Note level 1 ID not explicitly specified
# E.g. see demo(UserGuide09)
data(bang, package = 'R2MLwiN')
(mymodel4 <- runMLwiN(logit(use, cons) ~ 1 + lc + age + (1 | district),
  D = 'Binomial', data = bang))

# Mixed response model
# Note using MCMC estimation (EstM = 1)
# Normal (english) and Bernoulli (behaviour) distributed responses
# probit link modelling behaviour with cons as denominator
```

```
# E.g. see demo(MCMCGuide19)
data(jspmix1, package = 'R2MLwiN')
(mymodel <- runMLwiN(c(english, probit(behaviour, cons)) ~
  1 + sex + ravens + fluent[1] + (1 | school) + (1[1] | id),
  D = c('Mixed', 'Normal', 'Binomial'),
  estoptions = list(EstM = 1,
  mcmcMeth = list(fixM = 1, residM = 1, Lev1VarM = 1)),
  data = jspmix1))

## End(Not run)
```

---

Formula.translate.compat

*An internal function, allowing back-compatibility, which translates a model formula from a formula object or character string into an R list object.*

---

## Description

Supports Formula syntax as used in earlier (<0.8-0) versions of **R2MLwiN**. A model formula, as a formula object (or a character string) is translated into an R list object. Called by `runMLwiN` if `oldsyntax = TRUE` (when user specifies `levID` not `NULL` in `runMLwiN` function call). For corresponding function supporting new syntax, see [Formula.translate](#).

## Usage

```
Formula.translate.compat(Formula, levID, D = "Normal", indata)
```

## Arguments

Formula	A formula object (or a character string) specifying a multilevel model. See Value for details.
levID	A character (vector) specifying the level ID(s).
D	A character string/vector specifying the distribution to be modelled, which can include 'Normal' (the default), 'Binomial', 'Poisson', 'Negbinom', 'Unordered Multinomial', 'Ordered Multinomial', 'Multivariate Normal', or 'Mixed'.
indata	A data.frame object containing the data to be modelled.

## Details

If Formula is a character string, then the following syntax applies:

- ~ A tilde is used to separate response variable(s) and explanatory variable(s).
- ( ) Round brackets are used to specify each random variable in the model together with its fixed/random part information.

- | Separates explanatory variable(s) (placed to the right of |) from the fixed/random part information (placed to the left of |) when placed within ().
- [] When placed immediately after an explanatory variable, indicates that the variable is categorical. The string in the [] represents the reference category; if empty, no reference category is used; See note.
- : Indicates an interaction term: i.e. the variables adjacent to :, and separated by it, are interacted with each other.
- 0 When placed to the left of | within () indicates that the variables to the right of | within the same () are to be added to the fixed part of the model.
- 1 When placed to the left of | within () indicates that the coefficients of the variables placed to the right of | within the same () are to be allowed to randomly vary at level 1 (and so on for 2 for level 2, 3 for level 3, etc.)
- 0s/0c When placed to the left of | within () indicates that separate (hence s) / common (hence c) coefficients for the variables to the right of | within the same () are to be added to the fixed part (hence 0) of multivariate normal, multinomial and mixed responses models.
- 2s/2c When placed to the left of | within () indicates that separate (hence s) / common (hence c) coefficients for the variables to the right of | within the same () are to be added to the random part of the model, and allowed to vary at level 2; applies to multivariate normal, multinomial and mixed responses models only.
- {} gives a vector of binary indicators specifying a common coefficient. 1 is to include the component at the corresponding positions; zero otherwise. These digits are separated by commas; applies to multivariate normal, multinomial and mixed responses models only.
- . Used for adding a separate coefficient for a particular component at a specific level; applies to multivariate normal, multinomial and mixed responses models only

If Formula is a formula object, 0s/0c, 2s/2c, .... and {} have to be replaced by `0s`/`0c`, `2s`/`2c`, .... and () respectively. Other syntax remains the same.

## Value

Outputs an R list object, which is then used as the input for `write.IGLS` and/or `write.MCMC`.

## Note

Note that some characters listed above have special meanings in the formula, so avoid using them when you name the random variable. Alphanumeric characters (i.e. [:a1num:]) are recommended for naming the random variable. They are also recommended for naming a reference category, inside []. Note: use [] notation only in the fixed part when there is no categorical variable in the random effects. If there is one in the random part, the categorical variable has to be converted into a set of binary variables (e.g., using `Untoggle`).

## Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**See Also**

[runMLwiN](#), [write.IGLS](#), [write.MCMC](#), [Formula.translate](#)

---

fysio	<i>Data on physiotherapy referrals from 100 general practices in the Netherlands, collected in 1987</i>
-------	---

---

**Description**

These data were collected in 1987 as part of a large national survey of general practice (Van der Velden 1999).

**Usage**

fysio

**Format**

A data frame with 16700 observations on the following 14 variables:

**gp<sub>id</sub>** GP identifier.

**pat<sub>id</sub>** Patient identifier.

**pat<sub>age</sub>** Patient age in years.

**page<sub>grp</sub>** Patient age group, with ages grouped into 7 categories (ordered factor with levels `page<35`, `35<=page<45`, `45<=page<55`, `55<=page<65`, `65<=page<75`, `75<=page<85`, `85<=page`).

**pat<sub>sex</sub>** Patient gender (factor with levels `female`, `male`).

**pat<sub>insur</sub>** Patient insurance indicator (factor with levels `privateins` (privately insured), `publicins` (publically insured)).

**pat<sub>edu</sub>** Patient education level (ordered factor with levels `none` (no formal education), `primary` (primary education), `secondary` (secondary and lower/middle vocational education), `higher` (higher vocational and university education)).

**diag** Primary diagnosis resulting from care episodes (factor with levels 1 (symptoms/complaints neck), 2 (symptoms/complaints back), 3 (myalgia/fibrositis), 4 (symptoms of multiple muscles), 5 (disabilities related to the locomotive system), 6 (impediments of the cervical spine), 7 (arthrosis cervical spine), 8 (lumbago), 9 (ischialgia), 10 (hernia nuclei pulposi), 11 (impediments of the shoulder), 12 (epicondylitis lateralis), 13 (tendinitis/synovitis)).

**gp<sub>exper</sub>** GP experience (number of years working as a GP divided by ten).

**gp<sub>workload</sub>** GP workload (number of contacts in the 3-month registration period divided by 1000).

**prac<sub>type</sub>** Practice type (factor with levels `solo`, `duo`, `group`, `healthcentre`).

**loc<sub>ation</sub>** Practice location (factor with levels `rural`, `suburban`, `urban`, `bigcity`).

**gp<sub>physifr</sub>** Indicator of whether the GP has physiotherapists in their social network (factor with levels `no`, `yes`).

**ref<sub>erral</sub>** Indicator of whether the patient was referred to a physiotherapist (factor with levels `no`, `yes`).

## Details

The `fysio` dataset is one of the example datasets analysed in Leyland and Groenewegen (2020), and provided with the multilevel-modelling software package MLwiN (Charlton et al., 2024).

## Source

Charlton, C., Rasbash, J., Browne, W.J., Healy, M. and Cameron, B. (2024) *MLwiN Version 3.08* Centre for Multilevel Modelling, University of Bristol.

Leyland, A.H., Groenewegen, P.P. (2020). Multilevel Logistic Regression Using MLwiN: Referrals to Physiotherapy. In: *Multilevel Modelling for Public Health and Health Services Research*. Springer, Cham. doi:10.1007/9783030348014\_12

Van der Velden, K. (1999). *General practice at work: its contribution to epidemiology and health policy*. NIVEL, PhD thesis Erasmus University, Utrecht

## Examples

```
## Not run:

data(fysio, package = "R2MLwiN")

# Example taken from Leyland and Groenewegen (2020)

# Change contrasts if wish to avoid warning indicating that, by default,
# specified contrasts for ordered predictors will be ignored by runMLwiN
# (they will be fitted as "contr.treatment" regardless of this setting). To
# enable specified contrasts, set allowcontrast to TRUE (this will be the
# default in future package releases).
my_contrasts <- options("contrasts")$contrasts
options(contrasts = c(ordered = "contr.treatment",
                      ordered = "contr.treatment"))

# As an alternative to changing contrasts, can instead use C() to specify
# contrasts for ordered predictors in formula object, e.g.:

# F1 <- logit(referral) ~ 1 + C(pagegrp, "contr.treatment") + patsex + diag +
#   C(patedu, "contr.treatment") + patinsur + gpexper + gpworkload +
#   practype + location + gpphysifr +
#   (1 | gpid)
#
# (mod_MQL1 <- runMLwiN(Formula = F1,
#                       D = "Binomial",
#                       data = fysio,
#                       allowcontrast = TRUE))

F1 <- logit(referral) ~ 1 + pagegrp + patsex + diag +
  patedu + patinsur + gpexper + gpworkload +
  practype + location + gpphysifr +
  (1 | gpid)

(mod_MQL1 <- runMLwiN(Formula = F1,
```

```

      D = "Binomial",
      data = fysio))

(mod_PQL2 <- runMLwiN(Formula = F1,
  estoptions = list(nonlinear = c(N = 1, M = 2),
    startval = list(FP.b = mod_MQL1@FP,
      FP.v = mod_MQL1@FP.cov,
      RP.b = mod_MQL1@RP,
      RP.v = mod_MQL1@RP.cov)),
  D = "Binomial",
  data = fysio))

# Change contrasts back to pre-existing:
options(contrasts = my_contrasts)

## End(Not run)

```

---

gcsecomp1

*Pupils' marks from GCSE exams (UK, 1989); complete cases only.*


---

## Description

Pupils' marks from GCSE exams, consisting of 1523 pupils across 73 schools. See Browne (2012) for further details.

## Usage

```
gcsecomp1
```

## Format

A data frame with 1523 observations on the following 6 variables:

**school** Identifying code for each school (level 2 unit).

**student** Identifying code for each pupil (level 1 unit).

**female** Gender of pupil: a factor with levels Male and Female.

**written** Exam score.

**csework** Coursework score.

**cons** Constant (=1).

## Details

The gcsecomp1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009).



**Source**

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

**Examples**

```
## Not run:

data(gcsecomp1, package="R2MLwiN")

(mymodel <- runMLwiN(c(written, csework) ~ 1 + female + (1 | school) + (1 | student),
  D = "Multivariate Normal", estoptions = list(EstM = 1), data = gcsecomp1))

## End(Not run)
```

---

gcsemv1

*Pupils' marks from GCSE exams (UK, 1989).*


---

**Description**

GCSE exam results, taken from 73 schools in England, consisting of 1905 pupils.

**Usage**

```
gcsemv1
```

**Format**

A data frame with 1905 observations on the following variables:

**school** School identification (level 2 unit).

**student** Student identification (level 1 unit).

**female** Gender: a factor with levels Female and Male.

**agemths** Age in months.

**written** Score on the written component.

**csework** Score on the coursework component.

**cons** Constant (= 1).

**Details**

The gcsemv1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009); for further details see Rasbash et al. (2012).

**Source**

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) *A User's Guide to MLwiN Version 2.26*. Centre for Multilevel Modelling, University of Bristol.

**See Also**

See `m1mRev` package for an alternative format of the same dataset, with fewer variables.

**Examples**

```
## Not run:

data(gcsemv1, package = "R2MLwiN")

(mymodel <- runMLwiN(c(written, csework) ~ 1 + female + (1 | school) + (1 | student),
  D = "Multivariate Normal", estoptions = list(EstM = 1), data = gcsemv1))

## End(Not run)
```

---

```
getSummary.mlwinfitIGLS
```

*Extract coefficients and GOF measures from a statistical object (memisc package).*

---

**Description**

Extract coefficients and GOF measures from a statistical object (memisc package).

**Usage**

```
## S3 method for class 'mlwinfitIGLS'
getSummary(obj, alpha = 0.05, ...)
```

**Arguments**

<code>obj</code>	An <code>mlwinfitIGLS</code> -class model.
<code>alpha</code>	level of the confidence intervals; their coverage should be $1-\alpha/2$
<code>...</code>	Other arguments.

**See Also**

[getSummary](#)

---

```
getSummary.mlwinfitMCMC
```

*Extract coefficients and GOF measures from a statistical object (memisc package).*

---

**Description**

Extract coefficients and GOF measures from a statistical object (memisc package).

**Usage**

```
## S3 method for class 'mlwinfitMCMC'  
getSummary(obj, alpha = 0.05, ...)
```

**Arguments**

obj	An <a href="#">mlwinfitMCMC-class</a> model.
alpha	level of the confidence intervals; their coverage should be 1-alpha/2
...	Other arguments.

**See Also**

[getSummary](#)

---

```
glance.mlwinfitIGLS
```

*Extract GOF measures from a statistical object (broom package).*

---

**Description**

Extract GOF measures from a statistical object (broom package).

**Usage**

```
## S3 method for class 'mlwinfitIGLS'  
glance(x, ...)
```

**Arguments**

x	An <a href="#">mlwinfitIGLS-class</a> model.
...	Other arguments.

**See Also**

[glance](#)

---

`glance.mlwinfitMCMC`     *Extract GOF measures from a statistical object (broom package).*

---

### Description

Extract GOF measures from a statistical object (broom package).

### Usage

```
## S3 method for class 'mlwinfitMCMC'  
glance(x, ...)
```

### Arguments

`x`                     An `mlwinfitMCMC`-class model.  
`...`                   Other arguments.

### See Also

[glance](#)

---

`height`                     *Height data.*

---

### Description

Height data for 100 adult males.

### Usage

```
height
```

### Format

A data frame with 100 observations on the following variable:

**height** Heights of 100 adult males measured in centimetres.

### Details

The height dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009); see Rasbash et al. (2012) for further details.

**Source**

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol. Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) *A User's Guide to MLwiN Version 2.26*. Centre for Multilevel Modelling, University of Bristol.

**Examples**

```
## Not run:
# from demo(UserGuide16)

data(height, package = "R2MLwiN")
summary(height)

hist(height$height)

1 - pnorm((200 - mean(height$height)) / sd(height$height))

heightsim1 <- function() {
  heightsim <- 175.35 + 10.002 * qnorm(runif(100))
  c(pmean = mean(heightsim), pvar = var(heightsim))
}

set.seed(1)

# Note: To obtain estimates as close as possible to the MLwiN manual,
# increase the number of reps to 10000.

simdata1 <- as.data.frame(t(replicate(1000, heightsim1())))
simdata1$iteration <- 1:nrow(simdata1)

plot(simdata1$iteration, simdata1$pmean, type = "l")

plot(density(simdata1$pmean))

quantile(simdata1$pmean, c(0.025, 0.975))

plot(simdata1$iteration, simdata1$pvar, type = "l")

plot(density(simdata1$pvar))

quantile(simdata1$pvar, c(0.025, 0.975))

heightsim2 <- function(variable) {
  samp <- sample(variable, replace = TRUE)
  c(npmean = mean(samp), npvar = var(samp))
}

simdata2 <- as.data.frame(t(replicate(1000, heightsim2(height$height))))
simdata2$iteration <- 1:nrow(simdata2)
```

```

plot(simdata2$iteration, simdata2$npmean, type = "l")
plot(density(simdata2$npmean))
quantile(simdata2$npmean, c(0.025, 0.975))
plot(simdata2$iteration, simdata2$npvar, type = "l")
plot(density(simdata2$npvar))
quantile(simdata2$npvar, c(0.025, 0.975))
## End(Not run)

```

---

hungary1	<i>Hungarian component of 2nd International Science Survey, '84; see Goldstein 2003</i>
----------	---

---

### Description

Hungarian component of 2nd International Science Survey, consisting of 2439 women across 99 districts.

### Usage

```
hungary1
```

### Format

A data frame with 2439 observations on the following 10 variables:

**school** Identifying code for each school (level 2 unit).  
**female** Gender indicator: a factor with levels Male and Female.  
**es\_core** Core Earth Sciences test result.  
**biol\_core** Core Biology test result.  
**biol\_r3** Optional Biology test result.  
**biol\_r4** Optional Biology test result.  
**phys\_core** Core Physics test result.  
**phys\_r2** Optional Physics test result.  
**cons** Constant(=1).  
**student** Identifying code for each student (level 1 unit).

### Details

The hungary1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009). Originally analysed in Goldstein (2003), further details can also be found in Browne (2012).

**Source**

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Goldstein, H.. (2003) *Multilevel Statistical Models*. Third Edition. London, Edward Arnold.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

**Examples**

```
## Not run:

data(hungary1, package = "R2MLwiN")

(mymodel <- runMLwiN(c(es_core, biol_core, biol_r3, biol_r4, phys_core, phys_r2) ~
  1 + female + (1 | school) + (1 | student),
  D = "Multivariate Normal", estoptions = list(EstM = 1), data = hungary1))

## End(Not run)
```

---

jspmix1

*Dataset of pupils' test scores, a subset of the Junior School Project.*


---

**Description**

An educational dataset of pupils' test scores, a subset of the Junior School Project (Mortimore et al., 1988).

**Usage**

```
jspmix1
```

**Format**

A data frame with 1119 observations on the following 8 variables:

**school** School identifying code.

**id** Pupil identifying code.

**sex** Sex of pupil; a factor with levels female and male.

**fluent** Fluency in English indicator, where 0 = beginner, 1 = intermediate, 2 = fully fluent; measured in Year 1.

**ravens** Test score, out of 40; measured in Year 1.

**english** Pupils' English test score, out of 100; measured in Year 3.

**behaviour** Pupils' behaviour score, where lowerquarter = pupil rated in bottom 25%, and upper otherwise; measured in Year 3.

**cons** A column of ones. If included as an explanatory variable in a regression model (e.g. in MLwiN), its coefficient is the intercept.

### Details

A subset of the Junior School Project (Mortimore et al., 1988), the `jspmix1` dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), and is used in Browne (2012) as an example of modelling mixed responses. It consists of test scores for 1119 pupils across 47 schools. Note that the behaviour variable originally had three categories, and the middle 50% and top 25% have been combined to produce a binary variable.)

### Source

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Mortimore, P., Sammons, P., Stoll, L., Lewis, D., Ecob, R. (1988) *School Matters*. Wells: Open Books.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

### Examples

```
## Not run:

data(jspmix1, package = "R2MLwiN")

jspmix1$denomb <- jspmix1$cons

(mymodel <- runMLwiN(c(english, probit(behaviour, denomb)) ~
  1 + sex + ravens + fluent[1] + (1 | school) + (1[1] | id),
  D = c("Mixed", "Normal", "Binomial"),
  estoptions = list(EstM = 1, mcmcMeth = list(fixM = 1, residM = 1, Lev1VarM = 1)),
  data = jspmix1))

## End(Not run)
```

---

lips1

*Lips data*

---

### Description

Observed counts of male lip cancer for the 56 regions of Scotland over the period 1975-1980.



**Usage**

lips1

**Format**

A data frame with 56 observations on the following 41 variables:

**area** Region ID.  
**cons** Constant (=1).  
**obs** Observed cases of lip cancer.  
**exp** Expected count.  
**perc\_aff** Percentage of the region who work in agriculture, fishing and forestry.  
**offs** Log of the expected count.  
**pcons** Constant (=1).  
**denom** Constant (=1).  
**neigh1** First neighbours.  
**neigh2** Second neighbours.  
**neigh3** Third neighbours.  
**neigh4** Fourth neighbours.  
**neigh5** Fifth neighbours.  
**neigh6** Sixth neighbours.  
**neigh7** Seventh neighbours.  
**neigh8** Eighth neighbours.  
**neigh9** Ninth neighbours.  
**neigh10** Tenth neighbours.  
**neigh11** Eleventh neighbours.  
**weight1** First neighbours' weights.  
**weight2** Second neighbours' weights.  
**weight3** Third neighbours' weights.  
**weight4** Fourth neighbours' weights.  
**weight5** Fifth neighbours' weights.  
**weight6** Sixth neighbours' weights.  
**weight7** Seventh neighbours' weights.  
**weight8** Eighth neighbours' weights.  
**weight9** Ninth neighbours' weights.  
**weight10** Tenth neighbours' weights.  
**weight11** Eleventh neighbours' weights.  
**wcar1** First neighbours' CAR weights.  
**wcar2** Second neighbours' CAR weights.

- wcar3** Third neighbours' CAR weights.
- wcar4** Fourth neighbours' CAR weights.
- wcar5** Fifth neighbours' CAR weights.
- wcar6** Sixth neighbours' CAR weights.
- wcar7** Seventh neighbours' CAR weights.
- wcar8** Eighth neighbours' CAR weights.
- wcar9** Ninth neighbours' CAR weights.
- wcar10** Tenth neighbours' CAR weights.
- wcar11** Eleventh neighbours' CAR weights.

### Details

The lips1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), and was analysed in Clayton & Kaldor (1987); see also Browne (2012) for more details.

### Source

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Clayton, D., Kaldor, J. (1987) Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics* 43: 671-681.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

### Examples

```
## Not run:  
  
data(lips1, package = "R2MLwiN")  
  
(mymodel <- runMLwiN(log(obs) ~ 1 + perc_aff + offset(offs) + (0 | neigh1) + (1 | area),  
  D = "Poisson", estoptions = list(EstM = 1), data = lips1))  
  
## End(Not run)
```

---

lmdp	<i>Details of deaths from all causes in England and Wales 1972-1992, taken from the local mortality datapack</i>
------	--

---

## Description

This dataset records the annual number of deaths in a given district in England and Wales, covering 14 years (1972-1992) across 403 districts. Note that there are only 5639 observations rather than the 5642 that might be expected (403 districts with an observation for each of 14 years); 3 data points were removed because extreme outlying values made them implausible.

## Usage

lmdp

## Format

A data frame with 5639 observations on the following 8 variables:

**county** County (within region) identifier. There are 54 counties, coded from 1 to 68.

**district** District (within county) identifier. There are 403 districts, coded 101 to 6820.

**region** Region identifier. There are 10 regions, coded 1 to 10.

**year** Year, 1979 (79) to 1992 (92).

**deaths** Number of deaths observed

**expected** Expected number of deaths, based on the 1992 national age- and sex- specific mortality rates.

**smr** Standardised mortality ratio (observed deaths / expected deaths \* 100).

**family** District classification into one of 6 groups as defined by the UK's Office for National Statistics (factor with levels Inner\_London (Inner London), Rural (Rural areas), Prospering (Prospering areas) Maturer (Maturer areas), Urban (Urban areas), Mining\_industrial (Mining and industrial areas)).

## Details

The lmdp dataset is one of the example datasets analysed in Leyland and Groenewegen (2020), and provided with the multilevel-modelling software package MLwiN (Charlton et al., 2024).

## Source

Charlton, C., Rasbash, J., Browne, W.J., Healy, M. and Cameron, B. (2024) *MLwiN Version 3.08* Centre for Multilevel Modelling, University of Bristol.

Leyland, A.H., Groenewegen, P.P. (2020). Multilevel Linear Regression Using MLwiN: Mortality in England and Wales, 1979-1992. In: *Multilevel Modelling for Public Health and Health Services Research*. Springer, Cham. doi:10.1007/9783030348014\_11

## Examples

```
## Not run:

data(lmdp, package = "R2MLwiN")

# Example taken from Leyland and Groenewegen (2020)

lmdp$ID <- seq(1:nrow(lmdp))

(mod_1 <- runMLwiN(smr ~ 1 + I(year - 79) +
                  (1 | district) + (1 | ID),
                  data = lmdp))

## End(Not run)
```

---

logLik,mlwinfitIGLS-method

*Returns the log-likelihood from "mlwinfitIGLS" objects.*

---

## Description

Returns the log-likelihood from "mlwinfitIGLS" objects.

## Usage

```
## S4 method for signature 'mlwinfitIGLS'
logLik(object, ...)
```

## Arguments

object	An <a href="#">mlwinfitIGLS-class</a> object.
...	Other arguments.

## See Also

[logLik-methods](#)

---

logLik.mlwinfitIGLS *Returns the log-likelihood from "mlwinfitIGLS" objects.*

---

### Description

Returns the log-likelihood from "mlwinfitIGLS" objects.

### Usage

```
## S3 method for class 'mlwinfitIGLS'
logLik(object, ...)
```

### Arguments

object            An `mlwinfitIGLS`-class object.  
 ...                Other arguments.

### See Also

[logLik](#)

---

matrix2df	<i>For multiple membership models, translates matrix into a data.frame formatted for MLwiN Translates a <a href="#">matrix</a> into a form usable by MLwiN for multiple membership models, namely a <a href="#">data.frame</a> with (a) columns containing membership IDs (if first row matrix is 0 1 1 0 1 1, then first row of generated ID vectors would be, say, 2, 3, 5, 6) and (b) columns containing weights (in this example, if standardise = TRUE, then first row of generated weight vectors would be, say, 0.25, 0.25, 0.25, 0.25, otherwise first row of generated weight vectors would be, say, 1, 1, 1, 1).</i>
-----------	--

---

### Description

For multiple membership models, translates matrix into a data.frame formatted for MLwiN

Translates a [matrix](#) into a form usable by MLwiN for multiple membership models, namely a [data.frame](#) with (a) columns containing membership IDs (if first row matrix is 0 1 1 0 1 1, then first row of generated ID vectors would be, say, 2, 3, 5, 6) and (b) columns containing weights (in this example, if standardise = TRUE, then first row of generated weight vectors would be, say, 0.25, 0.25, 0.25, 0.25, otherwise first row of generated weight vectors would be, say, 1, 1, 1, 1).

### Usage

```
matrix2df(mat, standardise = FALSE, idstub = "id", weightstub = "weight")
```

**Arguments**

<code>mat</code>	A matrix.
<code>standardise</code>	If TRUE, ensures the row sums to one; defaults to FALSE.
<code>idstub</code>	Prefix for columns containing IDs; defaults to <code>id</code> .
<code>weightstub</code>	Prefix for columns containing weights; defaults to <code>weight</code> .

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol, U.K.

**See Also**

[df2matrix](#)

---

MCSE

*Calculates the estimated Monte Carlo standard error (MCSE)*

---

**Description**

An internal function which calculates the estimated Monte Carlo standard error (MCSE) for the posterior estimate of the mean, for use in [sixway](#). As MCMC is a simulation-based approach this induces (Monte Carlo) uncertainty due to the random numbers it uses. This uncertainty reduces with more iterations, and is measured by the MCSE. See Browne (2012) for further details.

**Usage**

```
MCSE(chain, rho, ll = 0.5, ul = 20)
```

**Arguments**

<code>chain</code>	Vector or <a href="#">mcmc</a> object.
<code>rho</code>	ACF for first lag.
<code>ll</code>	Lower limit of x-axis, where value specified is multiplied by the length of the chain. Defaults to 0.5.
<code>ul</code>	Upper limit of x-axis, where value specified is multiplied by the length of the chain. Defaults to 20.

**Value**

The Monte Carlo standard error (MCSE) for the posterior estimate of the mean is returned.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

## References

Browne, W.J. (2012) MCMC Estimation in MLwiN, v2.26. Centre for Multilevel Modelling, University of Bristol.

## See Also

[sixway](#)

---

mlwin2bugs	<i>This function captures output files from MLwiN for estimation in WinBUGS/OpenBUGS.</i>
------------	---

---

## Description

This function allows R to call WinBUGS using the output files from MLwiN. This function uses functionalities in the [R2WinBUGS-package](#) package.

## Usage

```
mlwin2bugs(  
  D,  
  levID,  
  datafile,  
  initfiles,  
  modelfile,  
  bugEst,  
  fact,  
  addmore,  
  n.chains,  
  n.iter,  
  n.burnin,  
  n.thin,  
  debug = FALSE,  
  bugs.directory = bugs.directory,  
  bugsWorkingDir = tempdir(),  
  OpenBugs = FALSE,  
  cleanBugsWorkingDir = FALSE,  
  seed = NULL  
)
```

## Arguments

D	A vector specifying the type of distribution used in the model.
levID	A character (vector) specifying the level ID(s).
datafile	A file name where the BUGS data file will be saved in .txt format.
initfiles	A list of file names where the BUGS initial values will be saved in .txt format.

<code>modelfile</code>	A file name where the BUGS model will be saved in .txt format.
<code>bugEst</code>	A file name where the estimates from BUGS will be stored in .txt format.
<code>fact</code>	A list of objects used to specify factor analysis. See ‘Details’ below.
<code>addmore</code>	A vector of strings specifying additional coefficients to be monitored.
<code>n.chains</code>	The number of chains to be monitored.
<code>n.iter</code>	The number of iterations for each chain
<code>n.burnin</code>	The length of burn-in for each chain
<code>n.thin</code>	Thinning rate
<code>debug</code>	A logical value indicating whether (TRUE) or not (FALSE; the default) to close the BUGS window after completion of the model run
<code>bugs.directory</code>	The full path of location where WinBUGS is installed (ignored if <code>OpenBugs</code> is TRUE).
<code>bugsWorkingDir</code>	A directory where all the intermediate files are to be stored; defaults to <code>tempdir()</code> .
<code>OpenBugs</code>	If TRUE, OpenBUGS is used, if FALSE (the default) WinBUGS is used.
<code>cleanBugsWorkingDir</code>	If TRUE, the generated files will be removed from the <code>bugsWorkingDir</code> ; defaults to FALSE.
<code>seed</code>	An integer specifying the random seed.

### Details

A list of objects to specify factor analysis, as used in the argument `fact`:

- `nfact`: specifies the number of factors;
- `lev.fact`: Specifies the level/classification for the random part of the factor for each factor;
- `nfactcor`: specifies the number of correlated factors;
- `factcor`: a vector specifying the correlated factors: the first element corresponds to the first factor number, the second to the second factor number, the third element corresponds to the starting value for the covariance and the fourth element to whether this covariance is constrained (1) or not (0). If more than one pair of factors is correlated, then repeat this sequence for each pair.
- `loading`: a matrix specifying the starting values for the factor loadings and the starting value of the factor variance. Each row corresponds to a factor.
- `constr`: a matrix specifying indicators of whether the factor loadings and the factor variance are constrained (1) or not (0).

### Value

Returns an `mcmc` object.

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.



**See Also**

[runMLwiN,bugs](#)

---

mlwinfitIGLS-class      *An S4 class that stores the outputs of the fitted IGLS model.*

---

**Description**

An MLwiN model run via the IGLS estimation method is represented by an "mlwinfitIGLS" object

**Slots**

Nobs Computes the number of complete observations.

DataLength Total number of cases.

Hierarchy For each higher level of a multilevel model, returns the number of units at that level, together with the minimum, mean and maximum number of lower-level units nested within units of the current level.

D A vector specifying the type of distribution to be modelled, which can include 'Normal', 'Binomial', 'Poisson', 'Multinomial', 'Multivariate Normal', or 'Mixed'.

Formula A formula object (or a character string) specifying a multilevel model.

levID A character string (vector) of the specified level ID(s).

contrasts A list of contrast matrices, one for each factor in the model.

xlevels A list of levels for the factors in the model.

FP Displays the fixed part estimates.

RP Displays the random part estimates.

FP.cov Displays a covariance matrix of the fixed part estimates.

RP.cov Displays a covariance matrix of the random part estimates.

elapsed.time Calculates the CPU time used for fitting the model.

call The matched call.

LIKE The deviance statistic ( $-2 \cdot \log(\text{like})$ ).

Converged Boolean indicating whether the model has converged

Iterations Number of iterations that the model has run for

Meth If Meth = 0 estimation method is set to RIGLS. If Meth = 1 estimation method is set to IGLS.

residual If resi.store is TRUE, then the residual estimates at all levels are returned.

data The data.frame that was used to fit the model.

nonlinear A character vector specifying linearisation method used. The first element specifies marginal quasi-likelihood linearization (N = 0) or penalised quasi-likelihood linearization (N = 1); The second element specifies first (M = 1) or second (M = 2) order approximation.

version The MLwiN version used to fit the model

### An instance of the Class

An instance is created by calling function `runMLwiN`.

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

### See Also

`runMLwiN`

### Examples

```
## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

## Example: tutorial
data(tutorial, package = "R2MLwiN")

(mymodel <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | school) + (1 | student),
  data = tutorial))

##summary method
summary(mymodel)

##logLik method
logLik(mymodel)

## End(Not run)
```

---

mlwinfitMCMC-class      *An S4 class that stores the outputs of the fitted MCMC model.*

---

### Description

An MLwiN model run via the MCMC estimation method is represented by an "mlwinfitMCMC" object

**Slots**

- Nobs** Computes the number of complete observations.
- DataLength** Total number of cases.
- Hierarchy** For each higher level of a multilevel model, returns the number of units at that level, together with the minimum, mean and maximum number of lower-level units nested within units of the current level.
- burnin** An integer specifying length of the burn-in.
- nchains** An integer specifying number of MCMC chains run.
- iterations** An integer specifying the number of iterations after burn-in.
- D** A vector specifying the type of distribution to be modelled, which can include 'Normal', 'Binomial', 'Poisson', 'Multinomial', 'Multivariate Normal', or 'Mixed'.
- Formula** A formula object (or a character string) specifying a multilevel model.
- levID** A character string (vector) of the specified level ID(s).
- contrasts** A list of contrast matrices, one for each factor in the model.
- xlevels** A list of levels for the factors in the model.
- merr** A vector which sets-up measurement errors on predictor variables.
- fact** A list of objects specified for factor analysis, including `nfact`, `lev.fact`, `nfactor`, `factor`, `loading` and `constr`.
- xc** A list of objects specified for cross-classified and/or multiple membership models, including `class`, `N1`, `weight`, `id` and `car`.
- FP** Displays the fixed part estimates.
- RP** Displays the random part estimates.
- FP.cov** Displays a covariance matrix of the fixed part estimates.
- RP.cov** Displays a covariance matrix of the random part estimates.
- chains** Captures the MCMC chains from MLwiN for all parameters.
- elapsed.time** Calculates the CPU time used for fitting the model.
- BDIC** Bayesian Deviance Information Criterion (DIC)
- call** The matched call.
- LIKE** The deviance statistic ( $-2 \cdot \log(\text{like})$ ).
- fact.loadings** If `fact` is not empty, then the factor loadings are returned.
- fact.loadings.sd** If `fact` is not empty, then the factor loading standard deviations are returned.
- fact.cov** If `fact` is not empty, then factor covariances are returned.
- fact.cov.sd** If `fact` is not empty, then factor covariance standard deviations are returned.
- fact.chains** If `fact` is not empty, then the factor chains are returned.
- MIdata** If `dami[1]` is one then the mean complete response variable `y` is returned for each chain, if `dami[1]` is two then the SD is also included.
- imputations** If `dami[1]` is zero, then a list of completed datasets containing complete response variable `y` is returned.
- residual** If `resi.store` is TRUE, then the residual estimates at all levels are returned.

`resi.chains` If `resi.store.levs` is not empty, then the residual chains at these levels are returned.

`version` The MLwiN version used to fit the model

`data` The data.frame that was used to fit the model.

### An instance of the Class

An instance is created by calling function `runMLwiN`.

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

### See Also

`runMLwiN`

### Examples

```
## Not run:

library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

## Example: tutorial
data(tutorial, package = "R2MLwiN")

(mymodel <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | school) + (1 | student),
  estoptions = list(EstM = 1), data = tutorial))

##summary method
summary(mymodel)

##BDIC slot
mymodel@BDIC

## End(Not run)
```

---

mmmec

*EC data on UV radiation exposure & malignant melanoma.*

---

### Description

EC data on UV radiation exposure & malignant melanoma, consisting of 354 counties across 79 regions across 9 nations.

**Usage**

```
mmmec
```

**Format**

A data frame with 354 observations on the following variables:

**nation** Nation ID: a factor with levels corresponding to each country.

**region** Region (within-nation) ID.

**county** County (within-region) ID.

**obs** Number of male deaths due to malignant melanoma between 1971 and 1980.

**exp** Expected number of deaths - proportional to county population.

**cons** Constant (=1).

**uvbi** County-level measurement of UV B radiation, centered on the mean.

**Details**

The mmec dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009). Further information can be found in Langford et al. (1998) and Browne (2012).

**Source**

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Langford, I. H., Bentham, G., McDonald, A-L. (1998) Multi-level modelling of geographically aggregated health data: a case study on malignant melanoma mortality and UV exposure in the European Community. *Statistics in Medicine* 17: 41-57.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

**See Also**

See mlmRev package for an alternative format of the same dataset, with fewer variables.

**Examples**

```
## Not run:

data(mmmec, package = "R2MLwiN")

(mymodel3 <- runMLwiN(log(obs) ~ 1 + uvbi + offset(log(exp)) + (1 | nation) + (1 | region),
  D = "Poisson", estoptions = list(EstM = 1), data = mmmec))

## End(Not run)
```

---

nobs.mlwinfitIGLS      *Returns the number of used observations from "mlwinfitIGLS" objects.*

---

### Description

Returns the number of used observations from "mlwinfitIGLS" objects.

### Usage

```
## S3 method for class 'mlwinfitIGLS'
nobs(object, ...)
```

### Arguments

object      An [mlwinfitIGLS-class](#) object.  
 ...      Other arguments.

### See Also

[nobs](#)

---

nobs.mlwinfitMCMC      *Returns the number of used observations from "mlwinfitMCMC" objects.*

---

### Description

Returns the number of used observations from "mlwinfitMCMC" objects.

### Usage

```
## S3 method for class 'mlwinfitMCMC'
nobs(object, ...)
```

### Arguments

object      An [mlwinfitMCMC-class](#) object.  
 ...      Other arguments.

### See Also

[nobs](#)

---

predCurves	<i>Draws predicted curves (lines) using estimates from the fixed part of a fitted model.</i>
------------	--

---

### Description

This function draws predicted curves (lines) against an explanatory variable for each category of a categorical variable.

### Usage

```
predCurves(
  object,
  indata = NULL,
  xname,
  group = NULL,
  legend = TRUE,
  legend.space = "top",
  legend.ncol = 2,
  ...
)
```

### Arguments

object	Either an <a href="#">mlwinfitIGLS-class</a> or <a href="#">mlwinfitMCMC-class</a> object.
indata	A data.frame object containing the data. If not specified, data is extracted from the object.
xname	The name of variable to be plotted.
group	A character string or a sequence of length equivalent to rows of data to plot. group = NULL by default.
legend	A logical value indicating whether a legend for group is to be added.
legend.space	A character string specifies one of the four sides, which can be one of 'top', 'bottom', 'left' and 'right'. Default, legend.space = 'top'.
legend.ncol	An integer specifies a number of columns, possibly divided into blocks, each containing some rows. Default, legend.ncol = 2.
...	Other arguments to be passed to <a href="#">xyplot</a> .

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

### See Also

[predLines](#)

**Examples**

```

## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

## Read alevchem data
data(alevchem, package = "R2MLwiN")

alevchem$gcseav <- alevchem$gcse_tot/alevchem$gcse_no - 6
# Avoids warning when fitting factor as continuous response:
alevchem$a_point_num <- as.numeric(alevchem$a_point)

## Example: A-level Chemistry
(mymodel <- runMLwiN(a_point_num ~ 1 + gcseav + I(gcseav^2) + I(gcseav^3)
                    + gender + (1 | pupil), estoptions = list(EstM = 1, resi.store = TRUE),
                    data = alevchem))

predCurves(mymodel, xname = "gcseav", group = "genderfemale")

## End(Not run)

```

---

predict.mlwinfitIGLS *Returns the predicted data from "mlwinfitIGLS" objects.*

---

**Description**

Returns the predicted data from "mlwinfitIGLS" objects.

**Usage**

```

## S3 method for class 'mlwinfitIGLS'
predict(
  object,
  newdata = NULL,
  params = NULL,
  type = "link",
  se.fit = FALSE,
  terms = NULL,
  ...
)

```



**Arguments**

object	An <a href="#">mlwinfitIGLS-class</a> object.
newdata	data frame for which to evaluate predictions
params	a character vector specifying the parameters to use in evaluating predictions. If NULL, <code>names(object[["FP"]])</code> is used by default.
type	when this has the value "link" (default) the linear predictor is returned. When <code>type="terms"</code> each component of the linear predictor is returned seperately. When <code>type="response"</code> predictions on the scale of the response are returned.
se.fit	logical. When this is TRUE (not default) standard error estimates are returned for each prediction.
terms	if <code>type="terms"</code> , which terms (default is all terms), a character vector.
...	Other arguments.

**See Also**

[predict](#)

---

`predict.mlwinfitMCMC` Returns the predicted data from "mlwinfitMCMC" objects.

---

**Description**

Returns the predicted data from "mlwinfitMCMC" objects.

**Usage**

```
## S3 method for class 'mlwinfitMCMC'
predict(
  object,
  newdata = NULL,
  params = NULL,
  type = "link",
  se.fit = FALSE,
  terms = NULL,
  ...
)
```

**Arguments**

object	An <a href="#">mlwinfitMCMC-class</a> object.
newdata	data frame for which to evaluate predictions
params	a character vector specifying the parameters to use in evaluating predictions. If NULL, <code>names(object[["FP"]])</code> is used by default.

type	when this has the value "link" (default) the linear predictor is returned. When type="terms" each component of the linear predictor is returned separately. When type="response" predictions on the scale of the response are returned.
se.fit	logical. When this is TRUE (not default) standard error estimates are returned for each prediction.
terms	if type="terms", which terms (default is all terms), a character vector.
...	Other arguments

**See Also**

[predict](#)

---

predLines	<i>Draws predicted lines using a fitted model object</i>
-----------	--

---

**Description**

This function draws predicted lines against an explanatory variable for selected groups at a higher ( $\geq 2$ ) level. Note that it uses a lot of contiguous memory, and so we recommend running via 64-bit version R to mitigate against any potential problems.

**Usage**

```
predLines(
  object,
  indata = NULL,
  xname,
  lev = 2,
  selected = NULL,
  probs = c(0.025, 0.975),
  legend = TRUE,
  legend.space = "top",
  legend.ncol = 4,
  ...
)
```

**Arguments**

object	Either an <code>mlwinfitIGLS-class</code> or <code>mlwinfitMCMC-class</code> object.
indata	A <code>data.frame</code> object containing the data. If not specified, data is extracted from the object.
xname	The name of the variable to be plotted.
lev	A digit indicating the level (of the multilevel model) at which to plot.
selected	A vector specifying groups to selectively plot at the level specified in <code>lev</code> . If <code>selected = NULL</code> , then all groups at that level are included.

probs	A numeric vector of probabilities with values in $[0, 1]$ used to calculate the lower and upper quantiles from which the error bars are plotted. Currently, this is only available for an <code>mlwinfitMCMC-class</code> object.
legend	A logical value indicating whether a legend is to be added.
legend.space	A character string specifies one of the four sides, which can be one of 'top', 'bottom', 'left' and 'right'. Default, <code>legend.space = 'top'</code> .
legend.ncol	An integer specifies a number of columns, possibly divided into blocks, each containing some rows. Default, <code>legend.ncol = 2</code> .
...	Other arguments to be passed to <code>xyplot</code> .

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**See Also**

[predCurves](#)

**Examples**

```
## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

## Example: tutorial
data(tutorial, package = "R2MLwiN")
(mymodel <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | school) + (1 | student),
  estoptions = list(EstM = 1, resi.store.levs = 2), data = tutorial))

predLines(mymodel, xname = "standlrt", lev = 2, selected = c(30, 44, 53, 59),
  probs = c(0.025, 0.975))

## End(Not run)
```

---

print.mlwinfitIGLS      *Summarize "mlwinfitIGLS" objects*

---

**Description**

Summarize "mlwinfitIGLS" objects

**Usage**

```
## S3 method for class 'mlwinfitIGLS'
print(
  x,
  digits = max(3, getOption("digits") - 2),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

**Arguments**

x                    an [mlwinfitIGLS-class](#) object  
 digits              the number of significant digits to use when printing.  
 signif.stars       logical. If TRUE, 'significance stars' are printed for each coefficient.  
 ...                  other parameters

**See Also**

[print](#)

---

print.mlwinfitMCMC      *Summarize "mlwinfitMCMC" objects*

---

**Description**

Summarize "mlwinfitMCMC" objects

**Usage**

```
## S3 method for class 'mlwinfitMCMC'
print(
  x,
  digits = max(3, getOption("digits") - 2),
  signif.stars = getOption("show.signif.stars"),
  z.ratio = TRUE,
  ...
)
```

**Arguments**

x                    an [mlwinfitMCMC-class](#) object  
 digits              the number of significant digits to use when printing.  
 signif.stars       logical. If TRUE, 'significance stars' are printed for each coefficient.  
 z.ratio             logical. If TRUE, z-ratio values are displayed for each coefficient.  
 ...                  other parameters

**See Also**[print](#)


---

prior2macro	<i>Translates informative prior information into a concise MLwiN macro.</i>
-------------	---

---

**Description**

An internal function which takes an R list object containing informative prior information for a multilevel model and translates it into a concise vector object to be used in an MLwiN macro.

**Usage**

```
prior2macro(prior, D, fpart, nrand)
```

**Arguments**

prior	An R list object containing prior information for a multilevel model. See ‘Details’ below.
D	A character string specifying the type of distribution, which can be one of 'Normal', 'Binomial', 'Poisson', 'Negbinom', 'Multinomial', 'Multivariate Normal', or 'Mixed'
fpart	An R list containing the list of fixed part parameter labels.
nrand	An R list of lists, containing the number of random parameters at each level.

**Details**

The prior list can contain the following:

- `fixe`: For the fixed parameters, if proper normal priors are used for some parameters, a list of vectors of length two is provided, each of which specifies the mean and the standard deviation. If not given, default ('flat' or 'diffuse') priors are used for the parameters. The names used in the list should match those in the model output.
- `rp<level number>`: A list object specifying the Wishart or gamma prior for the covariance matrix or scalar variance at the levels specified, e.g. `rp1` for level 1, `rp2` for level 2, etc. Consists of: (1) `estimate` – a prior guess for the true value of the covariance matrix; (2) `size` – sample size for guess. Note that this is a weakly-informative prior and the default prior is used if missing.

**Value**

A long vector is returned in the format of MLwiN macro language. This includes all the specified prior parameters.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**See Also**

[runMLwiN](#)

---

rats

*Weights of 30 rats, measured weekly over 5 weeks.*

---

**Description**

Weights of 30 rats, measured weekly over 5 weeks.

**Usage**

rats

**Format**

A data frame with 30 observations on the following 7 variables:

**y8** Weight on day 8.

**y15** Weight on day 15.

**y22** Weight on day 22.

**y29** Weight on day 29.

**y36** Weight on day 36.

**cons** Constant(=1).

**rat** Rat ID

**Details**

The rats dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009); see Browne (2012) and Gelfand (1990) for further details.

**Source**

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Gelfand, A. E., Hills, S.E., Racine-Poon, A., Smith, A.F.M. (1990) Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association* 85: 972-985.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

**Examples**

```
## Not run:

data(rats, package = "R2MLwiN")

(mymodel <- runMLwiN(c(y8, y15, y22, y29, y36) ~ 1 + (1 | rat),
  D = "Multivariate Normal", estoptions = list(EstM = 1),
  data = rats))

## End(Not run)
```

---

reading1

*Students' reading attainment in inner London infant schools.*

---

**Description**

Reading score data for 407 pupils across 6 occasions.

**Usage**

```
reading1
```

**Format**

A data frame with 407 observations on the following 13 variables:

**id** Unique pupil identifying code.

**age1** Age at occasion 1.

**read1** Reading score at occasion 1.

**age2** Age at occasion 2.

**read2** Reading score at occasion 2.

**age3** Age at occasion 3.

**read3** Reading score at occasion 3.

**age4** Age at occasion 4.

**read4** Reading score at occasion 4.

**age5** Age at occasion 5.

**read5** Reading score at occasion 5.

**age6** Age at occasion 6.

**read6** Reading score at occasion 6.

## Details

The reading1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), and was analysed in Tizard et al. (1988); see also Rasbash et al. (2012) for further details.

## Source

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol. Rasbash, J., Steele, F., Browne, W.J., Goldstein, H. (2012) *A User's Guide to MLwiN v2.26*. University of Bristol: Centre for Multilevel Modelling. Tizard, B., Blatchford, P., Burke, J. & Farquhar, C. (1988). Young children at school in the inner city. Hove, Sussex: Lawrence Erlbaum.

## Examples

```
## Not run:
# from demo(UserGuide13)

data(reading1, package = "R2MLwiN")
summary(reading1)

reading1[reading1 == -10] <- NA

summary(reading1)

reading <- reshape(reading1, idvar = "student", timevar = "id",
  varying = c("read1", "age1", "read2", "age2", "read3", "age3",
    "read4", "age4", "read5", "age5", "read6", "age6"),
  sep = "", direction = "long")

reading <- reading[c("student", "id", "age", "read")]
reading <- reading[order(reading$student, reading$id), ]

colnames(reading) <- c("student", "occasion", "age", "reading")
rownames(reading) <- NULL

summary(reading)

head(reading, 5)

tab <- aggregate(reading ~ occasion, reading,
  function(x) c(N = length(x), mean = mean(x), sd = sd(x)))
tab <- rbind(tab, c(NA, NA))
tab$reading[7, ] <- c(length(na.omit(reading$reading)),
  mean(na.omit(reading$reading)),
  sd(na.omit(reading$reading)))
rownames(tab)[7] <- "Total"
tab

tab <- aggregate(age ~ occasion, reading,
```



```

      function(x) c(N = length(x), mean = mean(x), sd = sd(x)))
tab <- rbind(tab, c(NA, NA))
tab$age[7, ] <- c(length(na.omit(reading$age)),
                 mean(na.omit(reading$age)),
                 sd(na.omit(reading$age)))
rownames(tab)[7] <- "Total"
tab

(mymodel1 <- runMLwiN(reading ~ 1 + (1 | student) + (1 | occasion),
                    data = reading))

## End(Not run)

```

---

residuals.mlwinfitIGLS

*Returns the residual data from "mlwinfitIGLS" objects.*

---

### Description

Returns the residual data from "mlwinfitIGLS" objects.

### Usage

```

## S3 method for class 'mlwinfitIGLS'
residuals(object, ...)

```

### Arguments

object	An <a href="#">mlwinfitIGLS-class</a> object
...	Other arguments.

### See Also

[residuals](#)

---

residuals.mlwinfitMCMC

*Returns the residual data from "mlwinfitMCMC" objects.*

---

### Description

Returns the residual data from "mlwinfitMCMC" objects.

**Usage**

```
## S3 method for class 'mlwinfitMCMC'
residuals(object, ...)
```

**Arguments**

object            An `mlwinfitMCMC`-class object  
 ...              Other arguments.

**See Also**

[residuals](#)

---

runMLwiN

*Calls MLwiN from R.*

---

**Description**

This function executes MLwiN and then brings results back to R.

**Usage**

```
runMLwiN(
  Formula,
  levID = NULL,
  D = "Normal",
  data = NULL,
  estoptions = list(EstM = 0),
  BUGO = NULL,
  MLwiNPath = NULL,
  stdout = "",
  stderr = "",
  workdir = tempdir(),
  checkversion = TRUE,
  allowcontrast = FALSE,
  indata = NULL,
  saveworksheet = NULL
)
```

**Arguments**

Formula            A [formula](#) object specifying the model formula. See [Formula.translate](#) ([Formula.translate.compat](#) details back-compatible functionality for deprecated syntax used in versions of **R2MLwiN** prior to 0.8-0) and also ‘Details’ below.

levID              A character vector specifying the level ID(s). Deprecated syntax: by default this is NULL and level ID(s) are specified in the Formula object.

D	A character string/vector specifying the type of distribution to be modelled, which can include 'Normal' (the default), 'Binomial', 'Poisson', 'Negbinom', 'Unordered Multinomial', 'Ordered Multinomial', 'Multivariate Normal', or 'Mixed'. In the case of the latter, 'Mixed' precedes the response types which also need to be listed in D, e.g. <code>c('Mixed', 'Normal', 'Binomial')</code> ; these need to be listed in the same order to which they are referred to in the Formula object (see <a href="#">Formula.translate</a> , <a href="#">Formula.translate.compat</a> ). For (R)IGLS estimation (i.e. <code>EstM = 0</code> in <code>estoptions</code> ) 'Mixed' combinations can consist of 'Normal' and 'Binomial' or 'Normal' and 'Poisson'; for MCMC estimation (i.e. <code>EstM = 0</code> ), on the other hand, only a combination of 'Normal' and 'Binomial' is available.
data	A <code>data.frame</code> object containing the data to be modelled. Optional (but recommended): if empty, data taken from environment of <code>formula</code> .
estoptions	A list of options used for estimating the model. See 'Details' below.
BUGO	A vector specifying BUGS options. If non-null, then WinBUGS/OpenBUGS, in conjunction with MLwiN, are used for modelling. Non-null only applicable if <code>EstM = 1</code> . See 'Details', below.
MLwiNPath	A path to the MLwiN folder. By default, <code>MLwiNPath = NULL</code> and path set by <code>options('MLwiN_path')</code> , the default for which can be changed via <code>options(MLwiN_path = 'path/to/MLwiN vX.XX/')</code> .
stdout	See <a href="#">system2</a> ; '' by default (i.e. output to stdout sent to R console).
stderr	See <a href="#">system2</a> ; '' by default (i.e. output to stderr sent to R console).
workdir	A path to the folder where the outputted files are to be saved. If the folder specified does not exist, a new folder of that name is created; <code>workdir = tempdir()</code> by default.
checkversion	If TRUE (default), returns version number unless (a) version detected is unknown or newer than MLwiN version available when current version of R2MLwiN was released, in which case returns text to this effect, or (b) version detected > 1 year older than MLwiN version available when current version of R2MLwiN was released, in which case function call stopped and user invited to update via usual channels. Can disable via FALSE e.g. if slowing execution time down (for example in a simulation).
allowcontrast	If TRUE, factor variables will follow the R behaviour when creating contrast variables. If FALSE (default) factor variables will be converted into a series of zero/one dummies.
indata	A <code>data.frame</code> object containing the data to be modelled. Deprecated syntax: by default this is NULL and the <code>data.frame</code> is instead referenced via <code>data</code> .
saveworksheet	A file name (or list of file names if more than one chain is specified) used to store the MLwiN worksheet after the model has been estimated.

## Details

With regard to runMLwiN's Formula object, see [formula](#) for notes on general usage, noting the following differences:

- The intercept is not included by default (this is keeping with the manner in which models are specified in MLwiN). To include an intercept, then, one can specify e.g. `normexam ~ 1 + standlrt + (1 | student)` or, assuming `cons` is a constant of ones, `normexam ~ cons + standlrt + (cons | student)`. (Note also, as further detailed below, for normal response models the level 1 ID (`student` in this example) needs to be explicitly included in the random part of the model formula; this is not the case for discrete response models).
- The link function and denominator are included in the `Formula` object, e.g. fitting a logistic model in which the variable `denom` is specified as the denominator: `logit(resp, denom) ~ 1 + age + (1 | region)`.

Further details are as follows.

The random part of the model is specified in sets of parentheses arranged in descending order with respect to their hierarchy. E.g. in the case of a 3-level model, the variable containing the level 3 ID is specified first, then the variable containing the level 2 ID, etc. Note that the variable containing the level 1 ID also needs to be explicitly specified unless it is a discrete response model (in which case you should not specify it).

The table below summarises the options for the `Formula` argument in **R2MLwiN**. They assume an intercept is added (via `~ 1`; for alternative specifications see [formula](#)). `<link>` denotes the link function, `<y1>`, `<y2>`, etc. represent response variables, `<denom>` denotes the denominator, `<offs>` the offset (optional), `<L2>`, `<L1>`, etc. the variables containing the level 2 and level 1 identifying codes, and `<ref_cat>` represents the reference category of a categorical response variable (optional: if unspecified the lowest level of the factor is used as the reference category). Explanatory variables are specified as e.g. `<x1> + <x2>`. For 'Ordered Multinomial', 'Multivariate Normal' and 'Mixed' responses, [`<common>`] indicates a common coefficient (i.e. the same for each category) is to be fitted; here `<common>` takes the form of a numeric identifier indicating the responses for which a common coefficient is to be added (e.g. `[1:5]` to fit a common coefficient for categories 1 to 5 of a 6-point ordered variable, `[1]` to fit a common coefficient for the response variable specified first in the `Formula` object for a 'Mixed' response model, etc.) Otherwise a separate coefficient (i.e. one for each category) is added. For 'Mixed' response models, the `Formula` arguments need to be grouped in the order the distributions are listed in D.

\* denotes IGLS only in the table below.

### Distribution

'Normal'

'Poisson'

'Negbinom'

'Binomial'

'Unordered Multinomial'

'Ordered Multinomial'

'Multivariate Normal'

c('Mixed', 'Normal', 'Binomial')

c('Mixed', 'Normal', 'Poisson')\*

### Format of Formula object

`<y1> ~ 1 + <x1> + (1|<L2>) + (1|<L1>) + ...`

`<link>(<y1>) ~ 1 + offset(<offs>) + <x1> + (1|<L2>) + ...`

`<link>(<y1>) ~ 1 + offset(<offs>) + (1|<L2>) + ...`

`<link>(<y1>, <denom>) ~ 1 + <x1> + (1|<L2>) + ...`

`<link>(<y1>, <denom>, <ref_cat>) ~ 1 + <x1> + (1|<L2>) + ...`

`<link>(<y1>, <denom>, <ref_cat>) ~ 1 + <x1> + <x2>[<common>] + (1[<common>]|<L3>) + (1|<L2>)`

`c(<y1>, <y2>, ...) ~ 1 + <x1> + <x2>[<common>] + (1[<common>]|<L3>) + (1|<L2>)`

`c(<y1>, ..., <link>(<y2>, <denom>), ...) ~ 1 + <x1> + <x2>[<common>] + (1|<L2>)`

`c(<y1>, ..., <link>(<y2>, <offset>), ...) ~ 1 + <x1> + <x2>[<common>] + (1|<L2>)`

The argument `estoptions` is a list which can contain the following options used for estimating the model:

- `EstM`: specifies estimation method. When `EstM = 0` (default), estimation method is (R)IGLS,

otherwise `EstM = 1` specifies MCMC estimation.

- `resi.store`: a logical value indicating whether residuals are to be stored or not. Defaults to FALSE.
- `resioptions`: a string vector to specify the various residual options. The 'variance' option calculates the posterior variances instead of the posterior standard errors; the 'standardised', 'leverage', 'influence' and 'deletion' options calculate standardised, leverage, influence and deletion residuals respectively; the 'sampling' option calculates the sampling variance covariance matrix for the residuals; the 'norecode' option prevents residuals with values exceedingly close or equal to zero from being recoded to missing. When `EstM = 1` (i.e. MCMC estimation) 'variance' is default value, and the only other permissible value is 'standardised' (else function call stopped with appropriate error message). When `EstM = 0` (i.e. (R)IGLS estimation), 'variance' cannot be specified together with 'standardised', 'leverage' or 'deletion' (function call stopped with appropriate error message). Default is `resioptions = c('variance')`.
- `resi.store.levs`: an integer vector indicating the levels at which the residual chains are to be stored (NULL by default). Non-NULL values not valid when `EstM = 0` (i.e. (R)IGLS estimation), else if `EstM = 0` and `resi.store.levs` non-NULL, residual chains at specified levels are returned.
- `debugmode`: a logical value determining whether MLwiN is run in the background or not. The default value is FALSE: i.e. MLwiN is run in the background. If TRUE the MLwiN GUI is opened, and then pauses after the model has been set-up, allowing user to check starting values; pressing 'Resume macro' will then fit the model. Once fit, pressing 'Resume macro' once more will save the outputs to the `workdir` ready to be read by **R2MLwiN**. Users can instead opt to 'Abort macro' in which case the outputs are not saved to the `workdir`. This option currently works for 32 bit version of MLwiN only (automatically switches unless `MLwiNPath` or `options(MLwiNPath)` has been set directly to the executable).
- `x64`: a logical value indicating whether the 64 bit version of MLwiN is used (unless `MLwiNPath` or `options(MLwiNPath)` has been set directly to the executable). The default is determined by the characteristics of the operating system on which the script is executed. If FALSE, the 32 bit version is called, if TRUE 64 bit version is called.
- `clean.files`: specifies whether the generated files are removed from the `workdir` (TRUE, the default) or not (FALSE).
- `show.file`: a logical value indicating whether the output files (e.g. MLwiN macro file) are shown on the screen. Defaults to FALSE.
- `clre`: a matrix used to define which elements of the random effects matrix to remove (i.e. hold constant at zero). Removes from the random part at level <first row> the covariance matrix element(s) defined by the pair(s) of rows <second row> <third row>. Each column corresponds to a removed entry of the covariance matrix. See e.g. `demo(UserGuide07)` for an example.
- `notation`: specifies the model subscript notation to be used in the MLwiN equations window. 'class' means no multiple subscripts, whereas 'level' has multiple subscripts. If `notation = NULL`, defaults to 'level' if '`xc = NULL`' else defaults to 'class'.
- `mem.init`: sets and displays worksheet capacities for the current MLwiN session. A vector of length 5 corresponding to the following order: number of levels (defaults to 1 + the number of levels specified in the function call); worksheet size in thousands of cells (default is 6000); the number of columns (default is 2500); the number of explanatory variables (default is 10

+ number of explanatory variables calculated initially); the number of group labels (default is 20).

- `optimat`: instructs MLwiN to limit the maximum matrix size that can be allocated by the (R)IGLS algorithm. Specify `optimat = TRUE` if MLwiN gives the following error message 'Overflow allocating smatrix'. This error message arises if one or more higher-level units is/are extremely large (containing more than 800 lower-level units). In this situation runMLwiN's default behaviour is to instruct MLwiN to allocate a larger matrix size to the (R)IGLS algorithm than is currently possible. Specifying `optimat = TRUE` caps the maximum matrix size at 800 lower-level units, circumventing the MLwiN error message, and allowing most MLwiN functionality.
- `nonlinear`: a character vector specifying linearisation method for discrete response models estimated via IGLS (see Chapter 9 of Rasbash et al 2012, and Goldstein 2011). `N = 0` specifies marginal quasi-likelihood linearization (MQL), whilst `N = 1` specifies penalised quasi-likelihood linearization (PQL); `M = 1` specifies first order approximation, whilst `M = 2` specifies second order approximation. `nonlinear = c(N = 0, M = 1)` by default. First order marginal quasi-likelihood (MQL1) only option for single-level discrete response models. Pertains to discrete response models estimated via IGLS: i.e. when `EstM = 0` in `estoptions`, and for starting values when estimated via IGLS for MCMC (`EstM = 1`).
- `Meth`: specifies which maximum likelihood estimation method is to be used. If `Meth = 0` estimation method is set to RIGLS. If `Meth = 1` estimation method is set to IGLS (the default setting). Pertains to models estimated via (R)IGLS: i.e. when `EstM = 0` in `estoptions`, and for starting values when estimated via (R)IGLS for MCMC (`EstM = 1`).
- `merr`: a vector which sets-up measurement errors on predictor variables. The first element `N` defines the number of variables that have measurement errors. Then, for each variable with measurement error, a pair of inputs are required: the first of these is the explanatory variable name as a character string, and the second is the variance of the measurement error for this variable. See `demo(MCMCGuide14)` for an example.
- `fact`: a list of objects specified for factor analysis, including:
  - `nfact`: Specifies the number of factors
  - `lev.fact`: Specifies the level/classification for the random part of the factor for each factor.
  - `nfactcor`: Specifies the number of correlated factors
  - `factcor`: A vector specifying the correlated factors: the first element corresponds to the first factor number, the second to the second factor number, the third element corresponds to the starting value for the covariance and the fourth element to whether this covariance is constrained (1) or not (0). If more than one pair of factors is correlated, then repeat this sequence for each pair.
  - `loading`: A matrix specifying the starting values for the factor loadings and the starting value of the factor variance. Each row corresponds to a factor.
  - `constr`: A matrix specifying indicators of whether the factor loadings and the factor variance are constrained (1) or not (0).
- `weighting`: a deprecated option for specifying weights in IGLS estimation: see `fpsandwich` and `rpsandwich` for new method of doing so. `weighting` is a list of objects including `levels`, `weights`, `mode`, `FSDE` and `RSDE`; see [write.IGLS](#) for details.
- `centring`: deprecated method (only applicable when using old syntax pre-R2MLwiN v.0.8-0) specifying function by which explanatory variables are to be centred (users can instead

transform variables prior to runMLwiN call). If non-NULL, centring is used for the selected explanatory variables (`centring = NULL` by default). `centring` is a list of objects specifying the methods to be used to centre specific explanatory variables. E.g. `list(age = 1, ...)` specifies that the explanatory variable `age` is to be centred around its grand mean; `list(age = c(2, 'district'), ...)` specifies that `age` is to be centred around its group mean, where group defined by the variable `district`; and `list(age = c(3, 18), ...)` specifies that `age` is to be centred around the value 18.

- `xclass`: a deprecated option for specifying cross-classified and/or multiple membership models; see `xc` and `mm` for new method of doing so. `xclass` is a list of objects including `class`, `N1`, `weight`, `id` and `car`; see [write.MCMC](#) for details.
- `mcmcOptions`: a list of objects specifying MCMC options, including the following:
  - `orth`: If `orth = 1`, orthogonal fixed effect vectors are used; zero otherwise.
  - `hcen`: An integer specifying the level where we use hierarchical centering.
  - `smcm`: If `smcm = 1`, structured MCMC is used; zero otherwise.
  - `smvn`: If `smvn = 1`, the structured MVN framework is used; zero otherwise.
  - `paex`: A matrix of  $N \times 2$ ; in each row, if the second digit is 1, parameter expansion is used at level <the first digit>.
  - `mcco`: This command allows the user to have constrained settings for the lowest level variance matrix in a multivariate Normal model. If value is  $\emptyset$ , it estimates distinct variances for each residual error and distinct covariances for each residual error pair. Four other settings are currently available:

- 1 fits structured errors with a common correlation parameter and a common variance parameter;
- 2 fits AR1 errors with a common variance parameter;
- 3 fits structured errors with a common correlation parameter and independent variance parameters;
- 4 fits AR1 errors with independent variance parameters.

- `drop.data`: If TRUE (default) only the data involved in the model is passed to MLwiN, otherwise the entire dataset in `data` is passed.
- `drop.levels`: If TRUE (default) any unused levels are dropped from factors, otherwise the dataset is left unchanged.
- `fpsandwich`: specifies standard error type for fixed parameters. If `fpsandwich = TRUE`, robust or ‘sandwich’ standard errors based on raw residuals are used, if `fpsandwich = FALSE` (default) then standard, uncorrected, IGLS or RIGLS computation used.
- `rpsandwich`: specifies standard error type for random parameters. If `rpsandwich = TRUE`, robust or ‘sandwich’ standard errors based on raw residuals are used, if `rpsandwich = FALSE` (default) then standard, uncorrected, IGLS or RIGLS ‘plug in’ estimates used.
- `smat`: a matrix with two rows the levels at which a diagonal matrix is to be specified. The first row specifies the level. If the value of the second row is 1 then the random covariance matrix is set to be diagonal.
- `maxiter`: a numeric value specifying the maximum number of iterations, from the start, before (R)IGLS estimation halts. Pertains to models estimated via (R)IGLS: i.e. when `EstM = 0` in `estoptions`, and for starting values when estimated via (R)IGLS for MCMC (`EstM = 1`).

- `tol`: a numeric value specifying the convergence criterion. If value is `m`, estimation will be deemed to have converged when the relative change in the estimate for all parameters from one iteration to the next is less than  $10(-m)$ . Defaults to value of 2 for `m` if not otherwise specified. Pertains to models estimated via (R)IGLS: i.e. when `EstM = 0` in `estoptions`, and for starting values when estimated via (R)IGLS for MCMC (`EstM = 1`).
- `extra`: if TRUE, extra binomial, extra negative binomial, extra Poisson or extra multinomial distributions assumed, else FALSE. can only be specified for discrete response models (i.e. 'Binomial', 'Negbinom', 'Poisson', 'Multinomial') estimated via (R)IGLS (i.e. `EstM = 0`).
- `reset`: a vector specifying the action to be taken, at each level, if a variance parameter is estimated at a particular iteration to be negative during estimation. Values specified in ascending order of level hierarchy: if 0 a negative variance estimate is reset to zero and so are any associated covariances; if 1 a negative variance estimate is reset to zero but not the associated covariances; if 2 no resetting takes place. E.g. `reset = c(0, 1)` to assign value 0 to level 1 and value 1 to level 2 of two-level model.
- `constraints`: `fixed.ui` and `fixed.ci` are used to specify constraints on the fixed coefficients, and `random.ui` and `random.ci` to specify constraints on the random parameters. The syntax for specifying just fixed parameter constraints is `constraints = list(fixed.ui = <fixed matrix>, fixed.ci = <fixed values>)`, where `<fixed matrix>` is a matrix where each row represents one fixed part parameter, in the same order that they appear in the results table, each column represents one constraint, and the values in the matrix are multipliers for the parameters; and `<fixed values>` is a vector of values, one per constraint, to which the parameters multiplied by the multipliers in the corresponding column of `<fixed matrix>` should be equal. For example, if we have a model with formula  $y \sim 1 + x1 + x2 + x3 + x4 + (1|lev1ID)$ , then `constraints = list(fixed.ui = matrix(c(0, 1, -1, 0, 0, 0, 0, 0, 1, 2), nrow = 5), fixed.ci = c(0, 2))` specifies the constraints that the coefficient of `x1` equals the coefficient of `x2` and that the coefficient of `x3` plus twice the coefficient of `x4` equals 2. Random constraints are specified similarly, and fixed and random constraints may be applied simultaneously. Applies to `EstM = 0` (i.e. estimation via (R)IGLS) only.
- `xc`: indicates whether model is cross-classified (TRUE) or nested (FALSE). Ignored if `EstM = 0`, i.e. only applicable to models estimated via MCMC. Defaults to `xc = FALSE`, unless either `mm` or `car` are non-NULL, in which case `xc = TRUE`. Supersedes deprecated `xc.class`.
- `mm`: specifies the structure of a multiple membership model. Can be a list of variable names, a list of vectors, or a matrix (e.g. see [df2matrix](#)). In the case of the former, each element of the list corresponds to a level (classification) of the model, in descending order. If a level is not a multiple membership classification, then NA is specified. Otherwise, lists need to be assigned to `mmvar` and `weights`, with the former containing columns specifying the classification units, and the latter containing columns specifying the weights. Ignored if `EstM = 0`, i.e. only applicable to models estimated via MCMC. `mm = NULL` by default. Supersedes deprecated `xc.class`. E.g. (from `demo(MCMCGuide16)`) for  $\text{logearn} \sim 1 + \text{age}_{40} + \text{sex} + \text{parttime} + (1 | \text{company}) + (1 | \text{id})$ , if `company` is a multiple membership classification with the variables indicating the classifications in `company`, `company2`, `company3`, `company4` and their weights in `weight1`, `weight2`, `weight3` and `weight4` then `mm = list(list(mmvar = list('company', 'company2', 'company3', 'company4'), weights = list('weight1', 'weight2', 'weight3', 'weight4')), NA)` with the NA, listed last, corresponding to the level 1 identifier (`id`).
- `car`: specifies the structure of a conditional autoregressive (CAR) model. Can be a list of variable names, a list of vectors, or a matrix (e.g. see [df2matrix](#)). In the case of the former, each



element of the list corresponds to a level (classification) of the model, in descending order. If a level is not a spatial classification, then NA is specified. Otherwise, lists need to be assigned to `carvar` and `weights`, with the former containing columns specifying the spatial classification units, and the latter containing columns specifying the weights. See `demo(MCMCGuide17)` for examples. Ignored if `EstM = 0`, i.e. only applicable to models estimated via MCMC. `car = NULL` by default. Supersedes deprecated `xclass`. See `demo(MCMCGuide17)` for examples.

- `carcentre`: if CAR model (i.e. if `car` is non-NULL), `carcentre = TRUE` mean-centres all random effects at that level.
- `startval`: a list of numeric vectors specifying the starting values. If multiple chains requested (via `nchains`), then can be a list of such lists. `FP.b` corresponds to the estimates for the fixed part; `FP.v` specifies the variance/covariance estimates for the fixed part; `RP.b` specifies the variance estimates for the random part; `RP.v` corresponds to the variance/covariance matrix of the variance estimates for the random part. `startval = NULL` by default: i.e. when `EstM = 0` the OLS estimates are used, else if `EstM = 1` the estimates obtained from IGLS are used as the starting values for MCMC.
- `sort.force`: If `TRUE` will sort data based on hierarchy as determined by model formula; defaults to `FALSE`.
- `sort.ignore`: If `FALSE` will check data is sorted in a manner in keeping with the hierarchy implied by the model formula, and will return a warning if that is not the case.
- `rng.version`: An integer value specifying the random number generator version to be used by MLwiN. If 10 (the default) this will be the Mersenne Twister; If 0 this will be the 3-Seed Wichmann-Hill (default in MLwiN prior to version 3).
- `mcmcMeth`: list of objects specifying MCMC methodology and prior options, including the following (see `write.MCMC` for further details):
  - `iterations`: Number of main iterations post-burnin (i.e. monitoring chain length), defaults to 5000.
  - `burnin`: Length of burnin, defaults to 500.
  - `nchains`: Number of MCMC chains to run, defaults to 1.
  - `thinning`: Thinning factor, defaults to 1.
  - `seed`: MCMC random number seed, defaults to 1 when `nchains = 1`, and to `1:nchains` when multiple chains requested.
  - `priorParam`: A list specifying informative priors. This includes: `fixe` – for the fixed parameters, if proper normal priors are used for some parameters, a list of vectors of length two is provided, each of which specifies the mean and the standard deviation. If not given, default ('flat' or 'diffuse') priors are used for the parameters; `fixe.common` – for multivariate normal, multinomial and mixed response models, if common coefficients are added, use `fixe.common` rather than `fixe`; `fixe.sep` – if the common coefficients are added, use `fixe.sep` for the separate coefficients; `rp1` – a list object specifying the Wishart or gamma prior for the covariance matrix or scalar variance at level 1 (this consists of: (1) `estimate` – a prior guess for the true value of the covariance matrix; (2) `size` – sample size for guess. Note that this is a weakly-informative prior and the default prior is used if missing); `rp2` – a list object specifying the Wishart or gamma prior for the covariance matrix or scalar variance at level 2 (this consists of: (1) `estimate` – an estimate for the true value of the inverse of the covariance matrix; (2) `size` – the number of rows in the covariance matrix. Note that this is a weakly-informative prior and the default prior is used if missing).

- `scale`: Scale factor for proposal variances: this number will be multiplied by the estimated parameter variance (from IGLS/RIGLS) to give the proposal distribution variance. Defaults to 5.8.
- `refresh`: Number of iterations after which screen (in MLwiN GUI) is to be refreshed. Defaults to 50.
- `fixM`: Specifies the estimation method for the fixed effects: 1 for Gibbs sampling, 2 for univariate Metropolis-Hastings (MH) sampling and 3 for multivariate MH sampling. Defaults to 2 if Poisson, Multinomial, Binomial or Mixed model, else defaults to 1.
- `residM`: Specifies the estimation method for the random effects (residuals): 1 for Gibbs sampling, 2 for univariate Metropolis-Hastings (MH) sampling and 3 for multivariate MH sampling. Defaults to 2 if Poisson, Multinomial, Binomial or Mixed model, else defaults to 1.
- `Lev1VarM`: Specifies the estimation method for the level 1 variance: 1 for Gibbs sampling, 2 for univariate Metropolis-Hastings (MH) sampling and 3 for multivariate MH sampling. Defaults to 2 if Poisson, Multinomial, Binomial or Mixed model, else defaults to 1.
- `OtherVarM`: Specifies the estimation method for the higher level variance matrices: 1 for Gibbs sampling, 2 for univariate Metropolis-Hastings (MH) sampling and 3 for multivariate MH sampling. Defaults to 1.
- `adaption`: `adaption = 1` (the default) indicates adaptation is to be used, `adaption = 0` indicates it is not.
- `tol`: An integer specifying tolerance (as a percentage; defaults to 10) when `adaption = 1` (ignored if `adaption = 0`).
- `rate`: An integer specifying the acceptance rate (as a percentage; defaults to 50) when `adaption = 1` (ignored if `adaption = 0`).
- `priorcode`: A vector indicating which default priors are to be used for the variance parameters. It defaults to `c(gamma = 1)` in which case Gamma priors are used with MLwiN's defaults of Gamma a value (shape) = 0.001 and Gamma b value (scale) = 0.001, although alternative values for shape and scale can be specified in subsequent elements of the vector, e.g. `c(gamma = 1, shape = 0.5, scale = 0.2)`). Alternatively `c(uniform = 1)` specifies Uniform priors on the variance scale. To allow for back-compatibility with deprecated syntax used in versions of **R2MLwiN** prior to 0.8-2, if `priorcode` is instead specified as an integer, then 1 indicates that Gamma priors are used, whereas 0 indicates that Uniform priors are used. See the section on 'Priors' in the MLwiN help system for more details on the meaning of these priors.
- `startval`: Deprecated: starting values are now specified directly within `estoptions`.
- `lclo`: Toggles on/off the possible forms of complex level 1 variation when using MCMC. By default (`lclo = 0`) the level 1 variation is expressed as a function of the predictors. Else (`lclo = 1`) the log of the level 1 precision (1/variance) is expressed as a function of the predictors. Defaults to `lclo = 0`.
- `dami`: Outputs a complete (i.e. including non-missing responses) response variable `y`. If `dami = c(0, <iter1>, <iter2>, ...)` then the response variables returned will be the value of `y` at the iterations quoted (as integers `<iter1>`, `<iter2>`, etc.); these can be used for multiple imputation. If `dami = 1` the value of `y` will be the mean estimate from the iterations produced. `dami = 2` is as for `dami = 1` but with the standard errors of the estimate additionally being stored. `dami = NULL` by default.

The argument `BUGO` is a vector specifying BUGS options as follows:

- `n.chains`: specifies the number of chains used by BUGS.
- `debug`: determines whether BUGS stays open following completion of the model run; `debug = FALSE` by default.
- `seed`: sets the random number generator in BUGS.
- `bugs.directory`: specifies the path where WinBUGS has been installed (not required if `OpenBugs = TRUE`).
- `OpenBugs`: if `OpenBugs = TRUE`, OpenBUGS is used. Otherwise (i.e. `OpenBugs = FALSE`, the default) WinBUGS is used.

### Value

If BUGO is non-NULL then the output is an `mcmc.list` object.

If the IGLS algorithm is used (i.e., `EstM = 0`), then returns `mlwinfitIGLS-class` object; if MCMC estimation used (i.e., `EstM = 1`), then returns `mlwinfitMCMC-class` object.

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

### References

Goldstein, H. (2011) Multilevel Statistical Models. 4th Edition. London: John Wiley and Sons.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) A User's Guide to MLwiN Version 2.26. Centre for Multilevel Modelling, University of Bristol.

### See Also

`formula`, `Formula.translate`, `Formula.translate.compat`, `write.IGLS`, `write.MCMC`

### Examples

```
## The R2MLwiN package includes scripts to replicate all the analyses in
## Rasbash et al (2012) A User's Guide to MLwiN Version 2.26 and
## Browne, W.J. (2012) MCMC estimation in MLwiN Version 2.26.
## The MLwiN manuals are available online, see:
## http://www.bristol.ac.uk/cmm/software/mlwin/download/manuals.html

## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

## For a list of demo titles
demo(package = 'R2MLwiN')
```

```
## Take MCMCGuide03 as an example
## To view file
file.show(system.file('demo', 'MCMCGuide03.R', package='R2MLwiN'))

## To run the demo
demo(MCMCGuide03)

## End(Not run)
```

---

show,mlwinfitIGLS-method

*Show objects of class "mlwinfitIGLS"*

---

### Description

Show objects of class "mlwinfitIGLS"

### Usage

```
## S4 method for signature 'mlwinfitIGLS'
show(object)
```

### Arguments

object            an [mlwinfitIGLS-class](#) object

### See Also

[show-methods](#)

---

show,mlwinfitMCMC-method

*Show objects of class "mlwinfitMCMC"*

---

### Description

Show objects of class "mlwinfitMCMC"

### Usage

```
## S4 method for signature 'mlwinfitMCMC'
show(object)
```

### Arguments

object            an [mlwinfitIGLS-class](#) object

**See Also**[show-methods](#)

---

show.mlwinfitIGLS	<i>Summarize "mlwinfitIGLS" objects</i>
-------------------	---

---

**Description**

Summarize "mlwinfitIGLS" objects

**Usage**

```
## S3 method for class 'mlwinfitIGLS'  
show(object, ...)
```

**Arguments**

object	an <a href="#">mlwinfitIGLS-class</a> object
...	other parameters

**See Also**[show](#)

---

show.mlwinfitMCMC	<i>Summarize "mlwinfitMCMC" objects</i>
-------------------	---

---

**Description**

Summarize "mlwinfitMCMC" objects

**Usage**

```
## S3 method for class 'mlwinfitMCMC'  
show(object, ...)
```

**Arguments**

object	an <a href="#">mlwinfitMCMC-class</a> object
...	other parameters

**See Also**[show](#)

---

sixway	<i>Draws a sixway plot of MCMC diagnostics.</i>
--------	---

---

### Description

This function produces a variety of diagnostic plots and statistics for MCMC chains.

### Usage

```
sixway(chain, name = NULL, acf.maxlag = 100, pacf.maxlag = 10, ...)
```

### Arguments

chain	A numeric vector, <code>mcmc</code> object or <code>mcmc.list</code> object (in which case uses its <code>thin</code> argument, otherwise assumes <code>thinning = 1</code> ), storing the MCMC chain for a chosen parameter.
name	The parameter name. If <code>name = NULL</code> , the column name of <code>chain</code> will be used, unless that is also <code>NULL</code> in which case <code>'x'</code> is used.
acf.maxlag	Maximum lag at which to calculate the auto-correlation function. <code>acf.maxlag = 100</code> by default. See <a href="#">acf</a> .
pacf.maxlag	Maximum lag at which to calculate the partial auto-correlation function. <code>pacf.maxlag = 10</code> by default. See <a href="#">pacf</a> .
...	Other graphical parameters (see <a href="#">par</a> for details).

### Details

A variety of plots and statistics are displayed in an R graphic window, including the following:

- a trace plot of the plotted trajectory of an MCMC chain for a model parameter;
- a kernel density plot; kernel density estimates are computed using [density](#);
- a plotted autocorrelation function (uses [acf](#));
- a plotted partial autocorrelation function (uses [pacf](#));
- a plot of the estimated Monte Carlo standard error (MCSE) of the posterior estimate of the mean against the number of iterations. As MCMC is a simulation-based approach this induces (Monte Carlo) uncertainty due to the random numbers it uses. This uncertainty reduces with more iterations, and is measured by the MCSE, and so this graph details how long the chain needs to be run to achieve a specific MCSE;
- a box contains two contrasting accuracy diagnostics. The Raftery-Lewis diagnostic ([raftery.diag](#)) is a diagnostic based on a particular quantile of the distribution. The diagnostic  $N_{hat}$  is used to estimate the length of Markov chain required to estimate a particular quantile (e.g. the 2.5% and 97.5% quantiles) to a given accuracy. The Brooks-Draper diagnostic (BD) is a diagnostic based on the mean of the distribution. It is used to estimate the length of Markov chain required to produce a mean estimate to  $k(=2)$  significant figures with a given accuracy;
- a box of summary statistics including the posterior mean, sd, mode, quantiles and the effective sample size (ESS) of the chain.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**See Also**

[BD](#),[MCSE](#),[density](#),[acf](#),[pacf](#),[raftery.diag](#),[effectiveSize](#)

**Examples**

```
## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

## Example: tutorial
data(tutorial, package = "R2MLwiN")

(mymodel <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | school) + (1 | student),
  estoptions = list(EstM = 1, resi.store.levs = 2), data = tutorial))

sixway(mymodel@chains[, "FP_standlrt", drop = FALSE], "beta_1")

## End(Not run)
```

---

```
summary,mlwinfitIGLS-method
```

*Summarize "mlwinfitIGLS" objects*

---

**Description**

Summarize "mlwinfitIGLS" objects

**Usage**

```
## S4 method for signature 'mlwinfitIGLS'
summary(object, ...)
```

**Arguments**

object	an <a href="#">mlwinfitIGLS-class</a> object
...	other parameters

**See Also**

[summary-methods](#)

---

summary,mlwinfitMCMC-method  
*Summarize "mlwinfitMCMC" objects*

---

**Description**

Summarize "mlwinfitMCMC" objects

**Usage**

```
## S4 method for signature 'mlwinfitMCMC'  
summary(object, ...)
```

**Arguments**

object	an <a href="#">mlwinfitMCMC-class</a> object
...	other parameters

**See Also**

[summary-methods](#)

---

summary.mlwinfitIGLS *Summarize "mlwinfitIGLS" objects*

---

**Description**

Summarize "mlwinfitIGLS" objects

**Usage**

```
## S3 method for class 'mlwinfitIGLS'  
summary(object, ...)
```

**Arguments**

object	an <a href="#">mlwinfitIGLS-class</a> object
...	other parameters



---

summary.mlwinfitMCMC *Summarize "mlwinfitMCMC" objects*

---

**Description**

Summarize "mlwinfitMCMC" objects

**Usage**

```
## S3 method for class 'mlwinfitMCMC'
summary(object, ...)
```

**Arguments**

object	an <code>mlwinfitMCMC</code> -class object
...	other parameters

---

tidy.mlwinfitIGLS *Summarises information about the components of a model from a statistical object (broom package).*

---

**Description**

Summarises information about the components of a model from a statistical object (broom package).

**Usage**

```
## S3 method for class 'mlwinfitIGLS'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

**Arguments**

x	An <code>mlwinfitIGLS</code> -class model.
conf.int	should the confidence interval be included?
conf.level	confidence interval level
...	Other arguments.

**See Also**

[tidy](#)

---

<code>tidy.mlwinfitMCMC</code>	<i>Summarises information about the components of a model from a statistical object (broom package).</i>
--------------------------------	--

---

### Description

Summarises information about the components of a model from a statistical object (broom package).

### Usage

```
## S3 method for class 'mlwinfitMCMC'
tidy(x, conf.int = FALSE, conf.level = 0.95, ...)
```

### Arguments

<code>x</code>	An <code>mlwinfitMCMC</code> -class model.
<code>conf.int</code>	should the confidence interval be included?
<code>conf.level</code>	confidence interval level
<code>...</code>	Other arguments.

### See Also

[tidy](#)

---

<code>trajectories</code>	<i>Plots MCMC chain trajectories</i>
---------------------------	--------------------------------------

---

### Description

This function draws trajectories of MCMC chains.

### Usage

```
trajectories(object, Range = c(1, 5000), selected = NULL)
```

### Arguments

<code>object</code>	An <code>mlwinfitMCMC</code> -class, <code>mcmc</code> or <code>mcmc.list</code> object, or other object that can be converted to an <code>mcmc</code> object.
<code>Range</code>	An integer vector of length two specifying the first and last iterations of the chains.
<code>selected</code>	A character vector specifying the selected chains to be plotted.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**See Also**

[sixway](#)

**Examples**

```
## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')

## Example: tutorial
data(tutorial, package = "R2MLwiN")

(mymodel <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | student),
                    estoptions = list(EstM = 1), data = tutorial))

trajectories(mymodel, Range = c(4501, 5000))

## End(Not run)
```

---

tutorial

*Exam results for six inner London Education Authorities*

---

**Description**

A subset of data from a much larger dataset of examination results from six inner London Education Authorities (school boards).

**Usage**

```
tutorial
```

**Format**

A data frame with 4059 observations on the following 10 variables:

**school** Numeric school identifier.

**student** Numeric student identifier.

**normexam** Students' exam score at age 16, normalised to have approximately a standard Normal distribution.

**cons** A column of ones. If included as an explanatory variable in a regression model (e.g. in MLwiN), its coefficient is the intercept.

**standlrt** Students' score at age 11 on the London Reading Test (LRT), standardised using Z-scores.

**sex** Sex of pupil; a factor with levels boy, girl.

**schgend** Schools' gender; a factor with levels corresponding to mixed school (mixedsch), boys' school (boysch), and girls' school (girlsSch).

**avslrt** Average LRT score in school.

**schav** Average LRT score in school, coded into 3 categories: low = bottom 25%, mid = middle 50%, high = top 25%.

**vrband** Students' score in test of verbal reasoning at age 11, a factor with 3 levels: vb1 = top 25%, vb2 = middle 50%, vb3 = bottom 25%.

## Details

The tutorial dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), and is a subset of data from a much larger dataset of examination results from six inner London Education Authorities (school boards). The original analysis (Goldstein et al., 1993) sought to establish whether some secondary schools had better student exam performance at 16 than others, after taking account of variations in the characteristics of students when they started secondary school; i.e., the analysis investigated the extent to which schools 'added value' (with regard to exam performance), and then examined what factors might be associated with any such differences. See also Rasbash et al. (2012) and Browne (2012).

## Source

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Goldstein, H., Rasbash, J., Yang, M., Woodhouse, G., Pan, H., Nuttall, D., Thomas, S. (1993) A multilevel analysis of school examination results. *Oxford Review of Education*, **19**, 425–433.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) *A User's Guide to MLwiN Version 2.26*. Centre for Multilevel Modelling, University of Bristol.

## See Also

See `mLmRev` package for an alternative format of the same dataset.

## Examples

```
## Not run:
```

```
data(tutorial, package = "R2MLwiN")
```

```
# Fit 2-level variance components model, using IGLS (default estimation method)
(VarCompModel <- runMLwiN(normexam ~ 1 + (1 | school) + (1 | student), data = tutorial))
```

```

# print variance partition coefficient (VPC)
print(VPC <- coef(VarCompModel)[["RP2_var_Intercept"]] /
      (coef(VarCompModel)[["RP1_var_Intercept"]] +
       coef(VarCompModel)[["RP2_var_Intercept"]]))

# Fit same model using MCMC
(VarCompMCMC <- runMLwiN(normexam ~ 1 + (1 | school) + (1 | student),
  estoptions = list(EstM = 1), data = tutorial))

# return diagnostics for VPC
VPC_MCMC <- VarCompMCMC@chains[, "RP2_var_Intercept"] /
  (VarCompMCMC@chains[, "RP1_var_Intercept"] +
   VarCompMCMC@chains[, "RP2_var_Intercept"])
sixway(VPC_MCMC, name = "VPC")

# Adding predictor, allowing its coefficient to vary across groups (i.e. random slopes)
(standlrtRS_MCMC <- runMLwiN(normexam ~ 1 + standlrt + (1 + standlrt | school) + (1 | student),
  estoptions = list(EstM = 1), data = tutorial))

# Example modelling complex level 1 variance
# fit log of precision at level 1 as a function of predictors
(standlrtC1V_MCMC <- runMLwiN(normexam ~
  1 + standlrt + (school | 1 + standlrt) + (1 + standlrt | student),
  estoptions = list(EstM = 1, mcmcMeth = list(lc1o = 1)),
  data = tutorial))

## End(Not run)

```

---

Untoggle

*Converts a categorical variable into several separate binary variables*


---

### Description

This function converts a vector (factor) of categorical character strings (integers) into several separate vectors of binary indicators to enable back-compatibility with versions of **R2MLwiN** prior to 0.8-0.

### Usage

```
Untoggle(categrv, name = NULL)
```

### Arguments

categrv	A vector (factor) of categorical character strings (integers).
name	A character string specifying a name of a prefix to be appended the categories when generating dummy variables. If NULL, v__ is used as a prefix.

**Value**

A matrix containing the generated dummy variables.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**Examples**

```
## Not run:
library(R2MLwiN)
# NOTE: Assumes MLwiN path is C:/Program Files (x86)/MLwiN v2.30/
# ...so please change relevant line if different
# if using R2MLwiN via WINE, the path may look like
# options(MLwiN_path='/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN v2.30/')

# Example: tutorial
data(tutorial)
names(tutorial)
tutorial = cbind(tutorial, Untoggle(tutorial$school, 'school'))
names(tutorial)

## End(Not run)
```

---

update,mlwinfitIGLS-method

*Update "mlwinfitIGLS" objects*

---

**Description**

Update "mlwinfitIGLS" objects

**Usage**

```
## S4 method for signature 'mlwinfitIGLS'
update(
  object,
  Formula.,
  levID.,
  estoptions.,
  ...,
  keep.order = TRUE,
  evaluate = TRUE
)
```

**Arguments**

object	a valid mlwinfitIGLS class object with an R function call component named call, the expression used to create itself.
Formula.	changes to the formula. This is a two sided formula where "." is substituted for existing components in the Formula component of object\$call.
levID.	changes to the specifications of level ID(s).
estoptions.	changes to the specifications of a list of options used for estimating the model.
...	additional arguments to the call, or arguments with changed values.
keep.order	a logical value indicating whether the terms should keep their positions.
evaluate	if TRUE (the default) the new call is evaluated; otherwise the call is returned as an unevaluated expression.

**Value**

either a new updated mlwinfitIGLS class object, or else an unevaluated expression for creating such an object.

**See Also**

[update-methods](#)

---

update,mlwinfitMCMC-method

*Update "mlwinfitMCMC" objects*

---

**Description**

Update "mlwinfitMCMC" objects

**Usage**

```
## S4 method for signature 'mlwinfitMCMC'
update(
  object,
  Formula.,
  levID.,
  estoptions.,
  ...,
  keep.order = TRUE,
  evaluate = TRUE
)
```

**Arguments**

object	a valid mlwinfitMCMC class object with an R function call component named call, the expression used to create itself.
Formula.	changes to the formula. This is a two sided formula where "." is substituted for existing components in the Formula component of object\$call.
levID.	changes to the specifications of level ID(s).
estoptions.	changes to the specifications of a list of options used for estimating the model.
...	additional arguments to the call, or arguments with changed values.
keep.order	a logical value indicating whether the terms should keep their positions.
evaluate	if TRUE (the default) the new call is evaluated; otherwise the call is returned as an unevaluated expression.

**Value**

either a new updated mlwinfitMCMC class object, or else an unevaluated expression for creating such an object.

**See Also**

[update-methods](#)

---

update.mlwinfitIGLS    *Update "mlwinfitIGLS" objects*

---

**Description**

Update "mlwinfitIGLS" objects

**Usage**

```
## S3 method for class 'mlwinfitIGLS'
update(object, ...)
```

**Arguments**

object	a valid mlwinfitIGLS class object with an R function call component named call, the expression used to create itself.
...	additional arguments to the call, or arguments with changed values.

**Value**

either a new updated mlwinfitIGLS class object, or else an unevaluated expression for creating such an object.

**See Also**

[update](#)



---

update.mlwinfitMCMC    *Update "mlwinfitMCMC" objects*

---

**Description**

Update "mlwinfitMCMC" objects

**Usage**

```
## S3 method for class 'mlwinfitMCMC'
update(object, ...)
```

**Arguments**

object	a valid mlwinfitMCMC class object with an R function call component named call, the expression used to create itself.
...	additional arguments to the call, or arguments with changed values.

**Value**

either a new updated mlwinfitMCMC class object, or else an unevaluated expression for creating such an object.

**See Also**

[update](#)

---

vcov.mlwinfitIGLS-method

*Extract the approximate variance-covariance matrix from "mlwinfit-IGLS" objects*

---

**Description**

Extract the approximate variance-covariance matrix from "mlwinfitIGLS" objects

**Usage**

```
## S4 method for signature 'mlwinfitIGLS'
vcov(object, ...)
```

**Arguments**

object	An <a href="#">mlwinfitIGLS-class</a> object
...	Other arguments

**See Also**[vcov-methods](#)

---

`vcov.mlwinfitMCMC-method`*Extract the approximate variance-covariance matrix from "mlwinfitMCMC" objects*

---

**Description**

Extract the approximate variance-covariance matrix from "mlwinfitMCMC" objects

**Usage**

```
## S4 method for signature 'mlwinfitMCMC'  
vcov(object, ...)
```

**Arguments**

<code>object</code>	An <a href="#">mlwinfitMCMC-class</a> object
<code>...</code>	Other arguments

**See Also**[vcov-methods](#)

---

`vcov.mlwinfitIGLS`*Extract the approximate variance-covariance matrix from "mlwinfitIGLS" objects*

---

**Description**

Extract the approximate variance-covariance matrix from "mlwinfitIGLS" objects

**Usage**

```
## S3 method for class 'mlwinfitIGLS'  
vcov(object, ...)
```

**Arguments**

<code>object</code>	An <a href="#">mlwinfitIGLS-class</a> object
<code>...</code>	Other arguments

**See Also**[vcov](#)

---

vcov.mlwinfitMCMC	<i>Extract the approximate variance-covariance matrix from "mlwinfitMCMC" objects</i>
-------------------	---

---

**Description**

Extract the approximate variance-covariance matrix from "mlwinfitMCMC" objects

**Usage**

```
## S3 method for class 'mlwinfitMCMC'
vcov(object, ...)
```

**Arguments**

object	An <code>mlwinfitMCMC</code> -class object
...	Other arguments

**See Also**

[vcov](#)

---

wage1	<i>Simulated dataset of office workers' salary and other employment details.</i>
-------	--

---

**Description**

A simulated dataset of office workers' salary (and associated information) in which workers exhibit multiple membership of companies worked for over past year.

**Usage**

```
wage1
```

**Format**

A data frame with 3022 observations on the following 21 variables:

**id** Unique office worker identifying code.

**company** Identifying code for company worked for over the last 12 months.

**company2** If worked for >1 company over the last 12 months, identifying code for second company.

**company3** If worked for >2 companies over the last 12 months, identifying code for third company.

- company4** If worked for >3 companies over the last 12 months, identifying code for fourth company.
- age** Age of worker.
- parttime** Part or full-time, a factor with levels Fulltime and Parttime.
- sex** Sex of worker, a factor with levels male and female.
- cons** A column of ones. If included as an explanatory variable in a regression model (e.g. in MLwiN), its coefficient is the intercept.
- earnings** Workers' earnings over the last financial year.
- logearn** Workers' (natural) log-transformed earnings over the last financial year.
- numjobs** The number of companies worked for over the last 12 months.
- weight1** Proportion of time worked for employer listed in company.
- weight2** Proportion of time worked for employer listed in company2.
- weight3** Proportion of time worked for employer listed in company3.
- weight4** Proportion of time worked for employer listed in company4.
- ew1** Alternative (equal) weighting for company (1/numjobs).
- ew2** Alternative (equal) weighting for company2 (if numjobs >1 then 1/numjobs, else 0).
- ew3** Alternative (equal) weighting for company3 (if numjobs >2 then 1/numjobs, else 0).
- ew4** Alternative (equal) weighting for company4 (if numjobs >3 then 1/numjobs, else 0).
- age\_40** Age of worker, centered on 40 years.

## Details

The simulated wage1 dataset is one of the sample datasets provided with the multilevel modelling software package MLwiN (Rasbash et al., 2009), and described in Browne (2012). It consists of salary and associated information for office workers, and is used by Browne (2012) as an example of modelling a multiple membership structure. The dataset exhibits multiple membership in that workers are clustered across the companies employing them over the past year, but some have worked for more than one company during that time.)

## Source

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

## Examples

```
## Not run:

data(wage1, package = "R2MLwiN")

(mymodel <- runMLwiN(logearn ~ 1 + age_40 + numjobs + (1 | company) + (1 | id),
```

```

estoptions = list(EstM = 1,
mm = list(list(mmvar = list("company", "company2", "company3", "company4"),
weights = list("weight1", "weight2", "weight3", "weight4")), NA)),
data = wage1))

```

```
## End(Not run)
```

---

write.IGLS	<i>Writes MLwiN macros to fit models using the iterative generalized least squares (IGLS) algorithm.</i>
------------	--

---

### Description

write.IGLS is an internal function which creates an MLwiN macro file to fit a multilevel model using IGLS.

### Usage

```

write.IGLS(
  indata,
  dtafile,
  oldsyntax = FALSE,
  resp,
  levID,
  expl,
  rp,
  D = "Normal",
  nonlinear = c(0, 1),
  categ = NULL,
  notation = NULL,
  nonfp = NA,
  clre = NULL,
  Meth = 1,
  extra = FALSE,
  reset,
  rcon = NULL,
  fcon = NULL,
  maxiter = 20,
  convtol = 2,
  mem.init = "default",
  optimat = FALSE,
  weighting = NULL,
  fpsandwich = FALSE,
  rpsandwich = FALSE,
  macrofile,
  IGLSfile,

```

```

  resi.file,
  resi.store = FALSE,
  resioptions,
  debugmode = FALSE,
  startval = NULL,
  namemap = sapply(colnames(indata), as.character),
  saveworksheet = NULL,
  rng.version = 10
)

```

### Arguments

<code>indata</code>	A data.frame object containing the data to be modelled.
<code>dtacfile</code>	The name of the temporary file used to send the data to MLwiN, which is in Stata format (i.e. with extension .dta).
<code>oldsyntax</code>	Specified as FALSE if new syntax has been used in Formula object, else specified as TRUE (to enable back-compatibility).
<code>resp</code>	A character string (vector) of the response variable name(s).
<code>levID</code>	A character string (vector) of the specified level ID name(s). The ID(s) should be sorted in the descending order of levels (e.g. <code>levID = c('level2', 'level1')</code> where 'level2' is the higher level).
<code>expl</code>	A character string (vector) of explanatory (predictor) variable name(s).
<code>rp</code>	A character string (vector) of random part of random variable name(s).
<code>D</code>	A character string/vector specifying the type of distribution to be modelled, which can include 'Normal' (the default), 'Binomial', 'Poisson', 'Negbinom', 'Unordered Multinomial', 'Ordered Multinomial', 'Multivariate Normal', or 'Mixed'. In the case of the latter, 'Mixed' precedes the response types which also need to be listed in D, e.g. <code>c('Mixed', 'Normal', 'Binomial')</code> ; these need to be listed in the same order to which they are referred to in the Formula object (see <a href="#">runMLwiN</a> , <a href="#">Formula.translate</a> , <a href="#">Formula.translate.compat</a> . 'Mixed' combinations can consist of 'Normal' and 'Binomial' or 'Normal' and 'Poisson'.
<code>nonlinear</code>	A character vector specifying linearisation method for discrete response models (see Chapter 9 of Rasbash et al 2012, and Goldstein 2011). $N = 0$ specifies marginal quasi-likelihood linearization (MQL), whilst $N = 1$ specifies penalised quasi-likelihood linearization (PQL); $M = 1$ specifies first order approximation, whilst $M = 2$ specifies second order approximation. <code>nonlinear = c(N = 0, M = 1)</code> by default. First order marginal quasi-likelihood (MQL1) only option for single-level discrete response models.
<code>categ</code>	Specifies categorical variable(s) as a matrix. Each column corresponds to a categorical variable; the first row specifies the name(s) of variable(s); the second row specifies the name(s) of reference group(s), NA(s) if no reference group; the third row states the number of categories for each variable. Supports back-compatibility with <b>R2MLwiN</b> versions <0.8-0.
<code>notation</code>	Specifies the model subscript notation to be used in the MLwiN equations window. 'class' means no multiple subscripts, whereas 'level' has multiple subscripts.

nonfp	Removes the fixed part of random variable(s). NA if no variable to be removed.
clre	A matrix used to define which elements of the random effects matrix to remove (i.e. hold constant at zero). Removes from the random part at level <first row> the covariance matrix element(s) defined by the pair(s) of rows <second row> <third row>. Each column corresponds to a removed entry of the covariance matrix. See e.g. <code>demo(UserGuide07)</code> for an example.
Meth	Specifies which maximum likelihood estimation method to be used. If <code>Meth = 0</code> estimation method is set to RIGLS. If <code>Meth = 1</code> estimation method is set to IGLS (the default setting).
extra	If TRUE, extra binomial, extra negative binomial, extra Poisson or extra multinomial distributions assumed, else FALSE.
reset	A vector of length( <code>levID</code> ) specifying the action to be taken, at each level, if a variance parameter is estimated at a particular iteration to be negative during estimation. Values specified in ascending order of level hierarchy: if 0 a negative variance estimate is reset to zero and so are any associated covariances; if 1 a negative variance estimate is reset to zero but not the associated covariances; if 2 no resetting takes place. E.g. <code>reset = c(0, 1)</code> to assign value 0 to level 1 and value 1 to level 2 of two-level model.
rcon	Matrix specifying constraints on the random parameters as specified in <code>random.ui</code> and <code>random.ci</code> in the <code>constraints</code> option within <code>estoptions</code> (see <code>runMLwiN</code> ). <code>random.ci</code> is appended to the bottom row of <code>random.ui</code> .
fcon	Matrix specifying constraints on the fixed coefficients as specified in <code>fixed.ui</code> and <code>fixed.ci</code> in the <code>constraints</code> option within <code>estoptions</code> (see <code>runMLwiN</code> ). <code>fixed.ci</code> is appended to the bottom row of <code>fixed.ui</code> .
maxiter	Numeric value specifying the maximum number of iterations, from the start, before estimation halts.
convtol	Numeric value specifying the convergence criterion, as specified in the <code>tol</code> option within <code>estoptions</code> (see <code>runMLwiN</code> ). If value of <code>convtol</code> is <code>m</code> , estimation will be deemed to have converged when the relative change in the estimate for all parameters from one iteration to the next is less than $10(-m)$ . Defaults to value of 2 for <code>m</code> if not otherwise specified.
mem.init	If calling <code>write.IGLS</code> directly, if wish to use defaults, value needs to be specified as 'default', else specify a vector of length 5 corresponding to the following order: number of levels; worksheet size in thousands of cells; the number of columns; the number of explanatory variables; the number of group labels.
optimat	This option instructs MLwiN to limit the maximum matrix size that can be allocated by the (R)IGLS algorithm. Specify <code>optimat = TRUE</code> if MLwiN gives the following error message 'Overflow allocating smatrix'. This error message arises if one more higher-level units is extremely large (contains more than 800 lower-level units). In this situation <b>R2MLwiN</b> 's default behaviour is to instruct MLwiN to allocate a larger matrix size to the (R)IGLS algorithm than is currently possible. Specifying <code>optimat = TRUE</code> caps the maximum matrix size at 800 lower-level units, circumventing the MLwiN error message, and allowing most MLwiN functionality.

weighting	A list of two items, one of which is a list called <code>weightvar</code> the length of which corresponds to the number of levels in the model, in descending order from highest level first. The other is an option <code>standardised</code> which is TRUE or FALSE.
fpsandwich	Specifies standard error type for fixed parameters. If <code>fpsandwich = TRUE</code> , robust or 'sandwich' standard errors based on raw residuals are used, if <code>fpsandwich = FALSE</code> (default) then standard, uncorrected, IGLS or RIGLS computation used.
rpsandwich	Specifies standard error type for random parameters. If <code>rpsandwich = TRUE</code> , robust or 'sandwich' standard errors based on raw residuals are used, if <code>rpsandwich = FALSE</code> (default) then standard, uncorrected, IGLS or RIGLS 'plug in' estimates used.
macrofile	A file name where the MLwiN macro file will be saved. The default location is in the temporary folder.
IGLSfile	A file name where the parameter estimates will be saved. The default location is in the temporary folder.
resifile	A file name where the residuals will be saved. The default location is in the temporary folder.
resi.store	A logical value to indicate if the residuals are to be stored (TRUE) or not (FALSE).
resioptions	A string vector to specify the various residual options. The 'variances' option calculates the posterior variances instead of the posterior standard errors; the 'standardised', 'leverage', 'influence' and 'deletion' options calculate standardised, leverage, influence and deletion residuals respectively; the 'sampling' option calculates the sampling variance covariance matrix for the residuals; the 'norecode' option prevents residuals with values exceedingly close or equal to zero from being recoded to missing; the <code>reflate</code> option returns unshrunk residuals. Note that the default option is <code>resioptions = c('variance')</code> ; 'variance' cannot be used together with the other options to calculate standardised, leverage, influence and deletion residuals.
debugmode	A logical value determining whether MLwiN is run in the background or not. The default value is FALSE: i.e. MLwiN is run in the background. If TRUE the MLwiN GUI is opened, and then pauses after the model has been set-up, allowing user to check starting values; pressing 'Resume macro' will then fit the model. Once fit, pressing 'Resume macro' once more will save the outputs to the <code>workdir</code> ready to be read by <b>R2MLwiN</b> . Users can instead opt to 'Abort macro' in which case the outputs are not saved to the <code>workdir</code> . This option currently works for 32 bit version of MLwiN only (automatically switches unless <code>MLwiNPath</code> or <code>options(MLwiNPath)</code> has been set directly to the executable).
startval	A list of numeric vectors specifying the starting values. <code>FP.b</code> corresponds to the estimates for the fixed part; <code>FP.v</code> specifies the variance/covariance estimates for the fixed part; <code>RP.b</code> specifies the variance estimates for the random part; <code>RP.v</code> corresponds to the variance/covariance matrix of the variance estimates for the random part.
namemap	A mapping of column names to DTA friendly shorter names
saveworksheet	A file name used to store the MLwiN worksheet after the model has been estimated.
rng.version	An integer indicating which random number generator should be used by MLwiN.



**Value**

Outputs a modified version of namemap containing newly generated short names.

**Author(s)**

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

**References**

Goldstein, H. (2011) Multilevel Statistical Models. 4th Edition. London: John Wiley and Sons.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) A User's Guide to MLwiN Version 2.26. Centre for Multilevel Modelling, University of Bristol.

**See Also**

[write.MCMC](#)

---

write.MCMC

*Writes MLwiN macros to fit models using Markov chain Monte Carlo (MCMC) methods*

---

**Description**

write.MCMC is an internal function which creates an MLwiN macro file to fit models using MCMC.

**Usage**

```
write.MCMC(
  indata,
  dtafile,
  oldsyntax = FALSE,
  resp,
  levID,
  expl,
  rp,
  D = "Normal",
  nonlinear = c(0, 1),
  categ = NULL,
  notation = NULL,
  nonfp = NULL,
  clre = NULL,
  Meth = 1,
  merr = NULL,
  carcentre = FALSE,
  maxiter = 20,
  convtol = 2,
```

```
seed = 1,
iterations = 5000,
burnin = 500,
scale = 5.8,
thinning = 1,
priorParam = "default",
refresh = 50,
fixM = 1,
residM = 1,
Lev1VarM = 1,
OtherVarM = 1,
adaption = 1,
priorcode = c(gamma = 1),
rate = 50,
tol = 10,
lclo = 0,
mcmcOptions,
fact = NULL,
xc = NULL,
mm = NULL,
car = NULL,
BUGO = NULL,
mem.init = "default",
optimat = FALSE,
modelfile,
initfile,
datafile,
macrofile,
IGLSfile,
MCMCfile,
chainfile,
MIfile,
resifile,
resi.store = FALSE,
resioptions,
resichains,
FACTchainfile,
resi.store.levs = NULL,
debugmode = FALSE,
startval = NULL,
dami = NULL,
namemap = sapply(colnames(indata), as.character),
saveworksheet = NULL,
rng.version = 10
)
```

### Arguments

`indata`            A data.frame object containing the data to be modelled.

dtafile	The file name of the dataset to be imported into MLwiN, which is in Stata format (i.e. with extension .dta).
oldsyntax	Specified as FALSE if new syntax has been used in Formula object, else specified as TRUE (to enable backcompatibility).
resp	A character string (vector) of response variable(s).
levID	A character string (vector) of the specified level ID(s). The ID(s) should be sorted in the descending order of levels (e.g. levID = c('level2', 'level1')) where 'level2' is the higher level).
expl	A character string (vector) of explanatory (predictor) variable(s).
rp	A character string (vector) of random part of random variable(s).
D	A character string/vector specifying the type of distribution to be modelled, which can include 'Normal' (the default), 'Binomial', 'Poisson', 'Unordered Multinomial', 'Ordered Multinomial', 'Multivariate Normal', or 'Mixed'. In the case of the latter, 'Mixed' precedes the response types which also need to be listed in D, e.g. c('Mixed', 'Normal', 'Binomial'); these need to be listed in the same order to which they are referred to in the Formula object (see <a href="#">runMLwiN</a> , <a href="#">Formula.translate</a> , <a href="#">Formula.translate.compat</a> . 'Mixed' combinations can only consist of 'Normal' and 'Binomial' for MCMC estimation.
nonlinear	A character vector specifying linearisation method for IGLS starting values for discrete response models (see Chapter 9 of Rasbash et al 2012, and Goldstein 2011). N = 0 specifies marginal quasi-likelihood linearization (MQL), whilst N = 1 specifies penalised quasi-likelihood linearization (PQL); M = 1 specifies first order approximation, whilst M = 2 specifies second order approximation. nonlinear = c(N = 0, M = 1) by default. First order marginal quasi-likelihood (MQL1) only option for single-level discrete response models.
categ	Specifies categorical variable(s) as a matrix. Each column corresponds to a categorical variable; the first row specifies the name(s) of variable(s); the second row specifies the name(s) of reference group(s), NA(s) if no reference group; the third row states the number of categories for each variable.
notation	Specifies the model subscript notation to be used in the MLwiN equations window. 'class' means no multiple subscripts, whereas 'level' has multiple subscripts.
nonfp	Removes the fixed part of random variable(s). NA if no variable is removed.
clre	A matrix used to estimate some, but not all, of the variances and covariances for a set of coefficients at a particular level. Remove from the random part at level <first row> the covariance matrix element(s) defined by the pair(s) of rows <second row> <third row>. Each row corresponds to a removed entry of the covariance matrix.
Meth	Specifies the maximum likelihood estimation method to be used when generating starting values via (R)IGLS. If Meth = 0 estimation method is set to RIGLS. If Meth = 1 estimation method is set to IGLS (the default setting). If Meth is absent, alternate between IGLS and RIGLS.
merr	A vector which sets-up measurement errors on predictor variables. The first element N defines the number of variables that have measurement errors. Then, for each variable with measurement error, a pair of inputs is required: the first

of which is the explanatory variable name as a character string, and the second of which is the variance of the measurement error for this variable. See `demo(MCMCGuide14)` for an example.

<code>carcentre</code>	If CAR model (i.e. if <code>car</code> is non-NULL), <code>carcentre = TRUE</code> mean-centres all random effects at that level.
<code>maxiter</code>	When generating starting values via (R)IGLS, a numeric value specifying the total number of iterations, from the start, before IGLS estimation halts (if <code>startval = NULL</code> ).
<code>convtol</code>	When generating starting values via (R)IGLS, a numeric value specifying the IGLS convergence criterion, as specified in the <code>tol</code> option within <code>estoptions</code> , where <code>startval = NULL</code> (see <code>runMLwiN</code> ). If value of <code>convtol</code> is <code>m</code> , estimation will be deemed to have converged when the relative change in the estimate for any parameter from one iteration to the next is less than $10(-m)$ . Defaults to value of 2 for <code>m</code> if not otherwise specified.
<code>seed</code>	An integer specifying the random seed in MLwiN.
<code>iterations</code>	An integer specifying the number of iterations after burn-in.
<code>burnin</code>	An integer specifying length of the burn-in.
<code>scale</code>	An integer specifying the scale factor of proposed variances; this number will be multiplied by the estimated parameter variance (from IGLS/RIGLS) to give the proposal distribution variance.
<code>thinning</code>	An integer specifying the frequency with which successive values in the Markov chain are stored. By default <code>thinning = 1</code> .
<code>priorParam</code>	A vector specifying the informative priors used, as output from <code>prior2macro</code> .
<code>refresh</code>	An integer specifying how frequently the parameter estimates are refreshed on the screen during iterations; only applies if <code>debugmode = TRUE</code> in <code>estoptions</code> : see <code>runMLwiN</code> .
<code>fixM</code>	Specifies the fixed effect method: 1 for Gibbs Sampling, 2 for univariate MH Sampling and 3 for multivariate MH Sampling.
<code>residM</code>	Specifies the residual method: 1 for Gibbs Sampling, 2 for univariate MH Sampling and 3 for multivariate MH Sampling.
<code>Lev1VarM</code>	Specifies the level 1 variance method: 1 for Gibbs Sampling, 2 for univariate MH Sampling and 3 for multivariate MH Sampling.
<code>OtherVarM</code>	Specifies the variance method for other levels: 1 for Gibbs Sampling, 2 for univariate MH Sampling and 3 for multivariate MH Sampling.
<code>adaption</code>	<code>adaption = 1</code> indicates adaptation is to be used; $\emptyset$ otherwise.
<code>priorcode</code>	A vector indicating which default priors are to be used for the variance parameters. It defaults to <code>c(gamma = 1)</code> in which case Gamma priors are used with MLwiN's defaults of Gamma a value ( <code>shape</code> ) = 0.001 and Gamma b value ( <code>scale</code> ) = 0.001, although alternative values for <code>shape</code> and <code>scale</code> can be specified in subsequent elements of the vector, e.g. <code>c(gamma = 1, shape = 0.5, scale = 0.2)</code> ). Alternatively <code>c(uniform = 1)</code> specifies Uniform priors on the variance scale. To allow for back-compatibility with deprecated syntax used in versions of <b>R2MLwiN</b> prior to 0.8-2, if <code>priorcode</code> is instead specified as an integer, then 1 indicates that Gamma priors are used, whereas $\emptyset$ indicates that Uniform priors

are used. See the section on 'Priors' in the MLwiN help system for more details on the meaning of these priors.

rate	An integer specifying the acceptance rate (as a percentage); this command is ignored if <code>adaptation = 0</code> .
tol	An integer specifying tolerance (as a percentage) for the acceptance rate.
lclo	This command toggles on/off the possible forms of complex level 1 variation when using MCMC. <code>lclo = 0</code> expresses the level 1 variation as a function of the predictors, whereas <code>lclo = 1</code> expresses the log of the level 1 precision (1/variance) as a function of the predictors.
mcmcOptions	A list of other MCMC options used. See 'Details' below.
fact	A list of objects specified for factor analysis. See 'Details' below.
xc	Indicates whether model is cross-classified (TRUE) or nested (FALSE). <code>xc = NULL</code> by default (corresponding to FALSE), unless either <code>mm</code> or <code>car</code> are not null, in which case <code>xc = TRUE</code> .
mm	Specifies the structure of a multiple membership model. Can be a list of variable names, a list of vectors, or a matrix (e.g. see <a href="#">df2matrix</a> ). In the case of the former, each element of the list corresponds to a level (classification) of the model, in descending order. If a level is not a multiple membership classification, then NA is specified. Otherwise, lists need to be assigned to <code>mmvar</code> and <code>weights</code> , with the former containing columns specifying the classification units, and the latter containing columns specifying the weights. Ignored if <code>EstM = 0</code> , i.e. only applicable to models estimated via MCMC. <code>mm = NULL</code> by default. Supersedes deprecated <code>xclass</code> . E.g. (from <code>demo(MCMCGuide16)</code> ) for <code>logearn ~ 1 + age_40 + sex + parttime + (company 1) + (id 1)</code> , if <code>company</code> is a multiple membership classification with the variables indicating the classifications in <code>company</code> , <code>company2</code> , <code>company3</code> , <code>company4</code> and their weights in <code>weight1</code> , <code>weight2</code> , <code>weight3</code> and <code>weight4</code> then <code>mm = list(list(mmvar = list('company', 'company2', 'company3', 'company4'), weights = list('weight1', 'weight2', 'weight3', 'weight4')), NA)</code> with the NA, listed last, corresponding to the level 1 identifier ( <code>id</code> ).
car	A list specifying structure of a conditional autoregressive (CAR) model. Each element of the list corresponds to a level (classification) of the model, in descending order. If a level is not a spatial classification, then NA is specified. Otherwise, lists need to be assigned to <code>carvar</code> and <code>weights</code> , with the former containing columns specifying the spatial classification units, and the latter containing columns specifying the weights. See <code>demo(MCMCGuide17)</code> for examples. <code>car = NULL</code> by default.
BUGO	If non-NULL uses BUGS for MCMC estimation using files specified in <code>modelfile</code> , <code>initfile</code> and <code>datafile</code> .
mem.init	A vector which sets and displays worksheet capacities for the current MLwiN session according to the value(s) specified. By default, the number of levels is <code>nlev+1</code> ; worksheet size in thousands of cells is 6000; the number of columns is 2500; the number of explanatory variables is <code>num_vars+10</code> ; the number of group labels is 20. <code>nlev</code> is the number of levels specified by <code>levID</code> , and <code>num_vars</code> is approximately the number of explanatory variables calculated initially.

optimat	This option instructs MLwiN to limit the maximum matrix size that can be allocated by the (R)IGLS algorithm. Specify <code>optimat = TRUE</code> if MLwiN gives the following error message 'Overflow allocating smatrix'. This error message arises if one more higher-level units is extremely large (contains more than 800 lower-level units). In this situation runmlwin's default behaviour is to instruct MLwiN to allocate a larger matrix size to the (R)IGLS algorithm than is currently possible. Specifying <code>optimat = TRUE</code> caps the maximum matrix size at 800 lower-level units, circumventing the MLwiN error message, and allowing most MLwiN functionality.
modelfile	A file name where the BUGS model will be saved in .txt format.
initfile	A file name where the BUGS initial values will be saved in .txt format.
datafile	A file name where the BUGS data will be saved in .txt format.
macrofile	A file name where the MLwiN macro file will be saved.
IGLSfile	A file name where the IGLS estimates will be saved.
MCMCfile	A file name where the MCMC estimates will be saved.
chainfile	A file name where the MCMC chains will be saved.
MIfile	A file name where the missing values will be saved.
resifile	A file name where the residual estimates will be saved.
resi.store	A logical value to indicate if residuals are to be stored (TRUE) or not (FALSE).
resioptions	A string vector to specify the various residual options. The 'variance' option calculates the posterior variances instead of the posterior standard errors; the 'standardised' option calculates standardised residuals.
resichains	A file name where the residual chains will be saved.
FACTchainfile	A file name where the factor chains will be saved.
resi.store.levs	An integer vector indicating the levels at which the residual chains are to be stored.
debugmode	A logical value determining whether MLwiN is run in the background or not. The default value is FALSE: i.e. MLwiN is run in the background. If TRUE the MLwiN GUI is opened, and then pauses after the model has been set-up, allowing user to check starting values; pressing 'Resume macro' will then fit the model. Once fit, pressing 'Resume macro' once more will save the outputs to the workdir ready to be read by <b>R2MLwiN</b> . Users can instead opt to 'Abort macro' in which case the outputs are not saved to the workdir. This option currently works for 32 bit version of MLwiN only (automatically switches unless <code>MLwiNPath</code> or <code>options(MLwiNPath)</code> has been set directly to the executable).
startval	A list of numeric vectors specifying the starting values when using MCMC. <code>FP.b</code> corresponds to the estimates for the fixed part; <code>FP.v</code> specifies the variance/covariance estimates for the fixed part; <code>RP.b</code> specifies the variance estimates for the random part; <code>RP.v</code> corresponds to the variance/covariance matrix of the variance estimates for the random part.
dami	This command outputs a complete (i.e. including non-missing responses) response variable <code>y</code> . If <code>dami = c(0, &lt;iter1&gt;, &lt;iter2&gt;, ...)</code> then the response

variables returned will be the value of  $y$  at the iterations quoted (as integers  $\langle \text{iter1} \rangle$ ,  $\langle \text{iter2} \rangle$ , etc.); these can be used for multiple imputation. If  $\text{dami} = 1$  the value of  $y$  will be the mean estimate from the iterations produced.  $\text{dami} = 2$  is as for  $\text{dami} = 1$  but with the standard errors of the estimate additionally being stored.

<code>namemap</code>	A mapping of column names to DTA friendly shorter names
<code>saveworksheet</code>	A list of file names (one for each chain) used to store the MLwiN worksheet after the model has been estimated.
<code>rng.version</code>	An integer indicating which random number generator should be used by MLwiN.

### Details

A list of other MCMC options as used in the argument `mcmcOptions`:

- `orth`: If `orth = 1`, orthogonal fixed effect vectors are used; zero otherwise.
- `hcen`: An integer specifying the level where we use hierarchical centering.
- `smcm`: If `smcm = 1`, structured MCMC is used; zero otherwise.
- `smvn`: If `smvn = 1`, the structured MVN framework is used; zero otherwise.
- `paex`: A matrix of  $N \times 2$ ; in each row, if the second digit is 1, parameter expansion is used at level  $\langle \text{the first digit} \rangle$ .
- `mcco`: This command allows the user to have constrained settings for the lowest level variance matrix in a multivariate Normal model. If value is  $\emptyset$ , it estimates distinct variances for each residual error and distinct covariances for each residual error pair. Four other settings are currently available:
  - 1 fits structured errors with a common correlation parameter and a common variance parameter;
  - 2 fits AR1 errors with a common variance parameter;
  - 3 fits structured errors with a common correlation parameter and independent variance parameters;
  - 4 fits AR1 errors with independent variance parameters.

A list of objects specified for cross-classified and/or multiple membership models, as used in the argument `xclass`:

- `class`: An integer (vector) of the specified class(es).
- `N1`: This defines a multiple membership across  $N1$  units at level `class`.  $N1 > 1$  if there is multiple membership.
- `weight`: If there is multiple membership then the column number `weight`, which is the length of the dataset, will contain the first set of weights for the multiple membership. Note that there should be  $N1$  weight columns and they should be sequential in the worksheet starting from `weight`.
- `id`: If the response is multivariate then the column number `id` must be input and this contains the first set of identifiers for the classification. Note that for a  $p$ -variate model each lowest level unit contains  $p$  records and the identifiers (sequence numbers) for each response variate

need to be extracted into `id` and following columns. There should be `N1` of these identifier columns and they should be sequential starting from `id` in the multivariate case.

- `car`: `car = TRUE` indicates the spatial CAR model; `FALSE` otherwise. `car = FALSE` if ignored.

A list of objects specified for factor analysis, as used in the argument `fact`:

- `nfact`: Specifies the number of factors
- `lev.fact`: Specifies the level/classification for the random part of the factor for each factor.
- `nfactcor`: Specifies the number of correlated factors
- `factcor`: a vector specifying the correlated factors: the first element corresponds to the first factor number, the second to the second factor number, the third element corresponds to the starting value for the covariance and the fourth element to whether this covariance is constrained (1) or not (0). If more than one pair of factors is correlated, then repeat this sequence for each pair.
- `loading`: A matrix specifying the starting values for the factor loadings and the starting value of the factor variance. Each row corresponds to a factor.
- `constr`: A matrix specifying indicators of whether the factor loadings and the factor variance are constrained (1) or not (0).

### Value

Outputs a modified version of `namemap` containing newly generated short names.

### Note

Note that for `FixM`, `residM`, `Lev1VarM` and

`OtherVarM`, not all combinations of methods are available for all sets of parameters and all models.

### Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

### References

Goldstein, H. (2011) *Multilevel Statistical Models*. 4th Edition. London: John Wiley and Sons.

Rasbash, J., Steele, F., Browne, W.J. and Goldstein, H. (2012) *A User's Guide to MLwiN Version 2.26*. Centre for Multilevel Modelling, University of Bristol.

### See Also

[write.IGLS](#)



---

 ws2foreign

*Converts data from one format to another within MLwiN*


---

## Description

This function converts data from one format to another within MLwiN, via MLwiN macro language. The foreign package allows R to read data files for most of these formats.

## Usage

```
ws2foreign(wsfile, foreignfile, MLwiNPath = NULL, x64 = NULL)
```

## Arguments

wsfile	A file name specifying the data file (with a specific extension) to be converted.
foreignfile	A file name specifying the data file (with a specific extension) after conversion.
MLwiNPath	A path to the MLwiN folder. By default, MLwiNPath = NULL and path set by options('MLwiN_path'), the default for which can be changed via options(MLwiN_path = 'path/to/MLwiN vX.XX/').
x64	A logical value indicating whether the 64 bit version of MLwiN is used, the default for this is determined by the version of R used. If FALSE, the 32 bit version is called.

## Details

MLwiN supports conversion between MLwiN (\*.wsz, \*.ws), Minitab (\*.mtw), SAS (\*.xpt), SPSS (\*.sav), and Stata (\*.dta) files.

The converted data file (with a specific extension) will be saved to the file specified by foreignfile.

## Author(s)

Zhang, Z., Charlton, C.M.J., Parker, R.M.A., Leckie, G., and Browne, W.J. (2016) Centre for Multilevel Modelling, University of Bristol.

## See Also

[read.dta](#)

## Examples

```
## Not run:
library(R2MLwiN)
# NOTE: if MLwiN not saved in location R2MLwiN defaults to, specify path via:
# options(MLwiN_path = 'path/to/MLwiN vX.XX/')
# If using R2MLwiN via WINE, the path may look like this:
# options(MLwiN_path = '/home/USERNAME/.wine/drive_c/Program Files (x86)/MLwiN vX.XX/')
```

```

wsfile = normalizePath(file.path(getOption("MLwiN_path"), "samples/tutorial.ws"), winslash = "/")
## the tutorial.dta file will be saved in the temporary folder
inputfile = normalizePath(file.path(tempdir(), "tutorial.dta"), winslash = "/", mustWork = FALSE)
ws2foreign(wsfile, foreignfile = inputfile)
library(foreign)
indata = read.dta(inputfile)
str(indata)
unlink(inputfile)

## End(Not run)

```

xc

*Examination scores of 16-year olds in Fife, Scotland.***Description**

A dataset of examination scores of 16-year olds in Fife, Scotland, in which the secondary school the pupil attended is cross-classified by the primary school the pupil attended.

**Usage**

xc

**Format**

A data frame with 3435 observations on the following 11 variables:

**vrq** A verbal reasoning score resulting from tests pupils took when they entered secondary school.

**attain** Attainment score of pupils at age sixteen.

**pid** Primary school identifying code.

**sex** Pupils' gender: a factor with levels Male and Female.

**sc** Pupils' social class (scaled from low to high).

**sid** Secondary school identifying code.

**fed** Fathers' education.

**choice** Choice number of secondary school attended (where 1 is first choice, etc.)

**med** Mothers' education.

**cons** A column of ones. If included as an explanatory variable in a regression model (e.g. in MLwiN), its coefficient is the intercept.

**pupil** Pupil identifying code.

**Details**

The xc dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), analysed by Paterson (1991). The data are cross-classified in that not all children who attended the same primary school subsequently entered the same secondary school. See also Rasbash et al. (2012).

## Source

Paterson, L. (1991) Socio economic status and educational attainment: a multidimensional and multilevel study. *Evaluation and Research in Education*, 5, 97-121. Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol. Rasbash, J., Steele, F., Browne, W.J., Goldstein, H. (2012) *A User's Guide to MLwiN v2.26*. University of Bristol: Centre for Multilevel Modelling.

## See Also

See `mlmRev` package for an alternative format of the same dataset.

## Examples

```
## Not run:

data(xc, package = "R2MLwiN")

(mymodel <- runMLwiN(attain ~ 1 + (1 | sid) + (1 | pid) + (1 | pupil),
  estoptions = list(xc = TRUE, EstM = 1), data = xc))

## End(Not run)
```

---

 xc1

*Examination scores of 16-year olds in Fife, Scotland.*

---

## Description

A dataset of examination scores of 16-year olds in Fife, Scotland, in which the secondary school the pupil attended is cross-classified by the primary school the pupil attended.

## Usage

```
xc1
```

## Format

A data frame with 3435 observations on the following 11 variables:

**vrq** A verbal reasoning score resulting from tests pupils took when they entered secondary school.

**attain** Attainment score of pupils at age sixteen.

**pid** Primary school identifying code.

**sex** Pupils' gender: a factor with levels Male and Female.

**sc** Pupils' social class (scaled from low to high).

**sid** Secondary school identifying code.

**fed** Fathers' education.

**choice** Choice number of secondary school attended (where 1 is first choice, etc.)

**med** Mothers' education.

**cons** A column of ones. If included as an explanatory variable in a regression model (e.g. in MLwiN), its coefficient is the intercept.

**pupil** Pupil identifying code.

## Details

The xc1 dataset is one of the sample datasets provided with the multilevel-modelling software package MLwiN (Rasbash et al., 2009), analysed by Paterson (1991). The data are cross-classified in that not all children who attended the same primary school subsequently entered the same secondary school. See also Browne (2012).

## Source

Browne, W. J. (2012) *MCMC Estimation in MLwiN Version 2.26*. University of Bristol: Centre for Multilevel Modelling.

Paterson, L. (1991) Socio economic status and educational attainment: a multidimensional and multilevel study. *Evaluation and Research in Education*, 5, 97-121.

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. (2009) *MLwiN Version 2.1*. Centre for Multilevel Modelling, University of Bristol.

## See Also

See mlmRev package for an alternative format of the same dataset.

## Examples

```
## Not run:

data(xc1, package = "R2MLwiN")

(mymodel <- runMLwiN(attain ~ 1 + (1 | sid) + (1 | pid) + (1 | pupil),
  estoptions = list(xc = TRUE, EstM = 1), data = xc1))

## End(Not run)
```

---

```
[,mlwinfitIGLS,ANY,ANY,ANY-method
```

*Extract or Replace parts of "mlwinfitIGLS" objects*

---

### Description

Extract or Replace parts of "mlwinfitIGLS" objects

### Usage

```
## S4 method for signature 'mlwinfitIGLS,ANY,ANY,ANY'
x[i, j, drop]
```

```
## S4 replacement method for signature 'mlwinfitIGLS,ANY,ANY,ANY'
x[i, j] <- value
```

```
## S4 method for signature 'mlwinfitIGLS'
x[[i, j, drop]]
```

```
## S4 replacement method for signature 'mlwinfitIGLS'
x[[i, j]] <- value
```

### Arguments

x	data frame
i, j	elements to extract or replace. For [, and [[, these are character.
drop	not used.
value	a suitable replacement value.

---

```
[,mlwinfitMCMC,ANY,ANY,ANY-method
```

*Extract or Replace parts of "mlwinfitMCMC" objects*

---

### Description

Extract or Replace parts of "mlwinfitMCMC" objects

### Usage

```
## S4 method for signature 'mlwinfitMCMC,ANY,ANY,ANY'
x[i, j, drop]
```

```
## S4 replacement method for signature 'mlwinfitMCMC,ANY,ANY,ANY'
x[i, j] <- value
```

```
## S4 method for signature 'mlwinfitMCMC'  
x[[i, j, drop]]  
  
## S4 replacement method for signature 'mlwinfitMCMC'  
x[[i, j]] <- value
```

**Arguments**

x	data frame
i, j	elements to extract or replace. For [ and [[, these are character.
drop	not used.
value	a suitable replacement value.

# Index

## \* datasets

- alevchem, [4](#)
  - bang, [6](#)
  - bang1, [8](#)
  - bes83, [11](#)
  - cvd, [17](#)
  - diag1, [21](#)
  - fysio, [30](#)
  - gcsecomp1, [32](#)
  - gcsemv1, [33](#)
  - height, [36](#)
  - hungary1, [38](#)
  - jspmix1, [39](#)
  - lips1, [40](#)
  - lmdp, [43](#)
  - mmec, [52](#)
  - rats, [62](#)
  - reading1, [63](#)
  - tutorial, [83](#)
  - wage1, [91](#)
  - xc, [106](#)
  - xc1, [107](#)
- [,mlwinfitIGLS,ANY,ANY,ANY-method, [109](#)
- [,mlwinfitMCMC,ANY,ANY,ANY-method, [109](#)
- [[<- ,mlwinfitIGLS,ANY,ANY,ANY-method  
([,mlwinfitIGLS,ANY,ANY,ANY-method),  
[109](#)
- [[<- ,mlwinfitMCMC,ANY,ANY,ANY-method  
([,mlwinfitMCMC,ANY,ANY,ANY-method),  
[109](#)
- [[,mlwinfitIGLS-method  
([,mlwinfitIGLS,ANY,ANY,ANY-method),  
[109](#)
- [[,mlwinfitMCMC-method  
([,mlwinfitMCMC,ANY,ANY,ANY-method),  
[109](#)
- [[<- ,mlwinfitIGLS-method  
([,mlwinfitIGLS,ANY,ANY,ANY-method),  
[109](#)
- [[<- ,mlwinfitMCMC-method  
([,mlwinfitMCMC,ANY,ANY,ANY-method),  
[109](#)
- acf, [78](#), [79](#)
- alevchem, [4](#)
- attach, [26](#)
- augment, [5](#), [6](#)
- augment.mlwinfitIGLS, [5](#)
- augment.mlwinfitMCMC, [6](#)
- bang, [6](#)
- bang1, [8](#)
- BD, [10](#), [78](#), [79](#)
- bes83, [11](#)
- bugs, [49](#)
- caterpillar, [13](#), [15](#)
- caterpillarR, [13](#), [14](#)
- coef, [17](#), [20](#)
- coef.mlwinfitIGLS-method, [15](#)
- coef.mlwinfitMCMC-method, [16](#)
- coef.mlwinfitIGLS, [16](#)
- coef.mlwinfitMCMC, [17](#)
- cvd, [17](#)
- data.frame, [20](#), [22](#), [45](#)
- density, [78](#), [79](#)
- deviance, [19](#)
- deviance.mlwinfitIGLS, [19](#)
- df.residual.mlwinfitIGLS, [19](#)
- df2matrix, [20](#), [46](#), [72](#), [101](#)
- diag1, [21](#)
- double2singlePrecision, [22](#)
- effectiveSize, [79](#)
- extract, [23](#), [24](#)
- extract.mlwinfitIGLS-method, [23](#)
- extract.mlwinfitMCMC-method, [23](#)
- fitted.mlwinfitIGLS, [24](#)

- fitted.mlwinfitMCMC, 25
- fitted.values, 24, 25
- formula, 25, 26, 66–68, 75
- formula.mlwinfitIGLS, 25
- formula.mlwinfitMCMC, 26
- Formula.translate, 26, 28, 30, 66, 67, 75, 94, 99
- Formula.translate.compat, 27, 28, 66, 67, 75, 94, 99
- fysio, 30
- gcsecomp1, 32
- gcseml, 33
- getSummary, 34, 35
- getSummary.mlwinfitIGLS, 34
- getSummary.mlwinfitMCMC, 35
- glance, 35, 36
- glance.mlwinfitIGLS, 35
- glance.mlwinfitMCMC, 36
- height, 36
- hungary1, 38
- jspmix1, 39
- lips1, 40
- lmdp, 43
- logLik, 45
- logLik.mlwinfitIGLS-method, 44
- logLik.mlwinfitIGLS, 45
- matrix, 20, 45
- matrix2df, 20, 45
- mcmc, 46, 48, 78, 82
- mcmc.list, 75, 78, 82
- MCSE, 46, 78, 79
- mlwin2bugs, 47
- mlwinfitIGLS-class, 49
- mlwinfitMCMC-class, 50
- mmec, 52
- nobs, 20, 54
- nobs.mlwinfitIGLS, 54
- nobs.mlwinfitMCMC, 54
- pacf, 78, 79
- par, 78
- plot.default, 13
- predCurves, 55, 59
- predict, 57, 58
- predict.mlwinfitIGLS, 56
- predict.mlwinfitMCMC, 57
- predLines, 55, 58
- print, 60, 61
- print.mlwinfitIGLS, 59
- print.mlwinfitMCMC, 60
- prior2macro, 61, 100
- qqmath, 14, 15
- raftery.diag, 78, 79
- rats, 62
- read.dta, 105
- reading1, 63
- residuals, 65, 66
- residuals.mlwinfitIGLS, 65
- residuals.mlwinfitMCMC, 65
- runMLwiN, 26–28, 30, 49, 50, 52, 62, 66, 94, 95, 99, 100
- show, 77
- show.mlwinfitIGLS-method, 76
- show.mlwinfitMCMC-method, 76
- show.mlwinfitIGLS, 77
- show.mlwinfitMCMC, 77
- sixway, 10, 11, 46, 47, 78, 83
- sparseMatrix, 20
- summary.mlwinfitIGLS-method, 79
- summary.mlwinfitMCMC-method, 80
- summary.mlwinfitIGLS, 80
- summary.mlwinfitMCMC, 81
- system2, 67
- tidy, 81, 82
- tidy.mlwinfitIGLS, 81
- tidy.mlwinfitMCMC, 82
- trajectories, 82
- tutorial, 83
- Untoggle, 29, 85
- update, 88, 89
- update.mlwinfitIGLS-method, 86
- update.mlwinfitMCMC-method, 87
- update.mlwinfitIGLS, 88
- update.mlwinfitMCMC, 89
- vcov, 90, 91
- vcov.mlwinfitIGLS-method, 89
- vcov.mlwinfitMCMC-method, 90
- vcov.mlwinfitIGLS, 90



vcov.mlwinfitMCMC, [91](#)

wage1, [91](#)

write.IGLS, [27](#), [29](#), [30](#), [70](#), [75](#), [93](#), [104](#)

write.MCMC, [27](#), [29](#), [30](#), [71](#), [73](#), [75](#), [97](#), [97](#)

ws2foreign, [105](#)

xc, [106](#)

xc1, [107](#)

xyplot, [55](#), [59](#)