Package 'LMMstar'

January 3, 2022

Type Package

Title Repeated Measurement Models for Discrete Times

Version 0.4.4

Date 2022-01-03

Description Companion R package for the course ``Statistical analysis of correlated and repeated measurements for health science researchers"
taught by the section of Biostatistics of the University of Copenhagen.
It implements linear mixed models where the model for the variance-covariance of the residuals is specified via patterns (compound symmetry, unstructured).
Statistical inference for mean, variance, and correlation parameters is performed based on the observed information and a Satterthwaite degrees of freedom.
Normalized residuals are provided to assess model misspecification.
Statistical inference can be performed for arbitrary linear combination(s) of model coefficients.
Predictions can be computed conditional to covariates only or also to outcome values.

License GPL-3

Encoding UTF-8

URL https://github.com/bozenne/LMMstar

BugReports https://github.com/bozenne/LMMstar/issues

Depends R (>= 3.5.0), nlme

Imports emmeans, ggplot2, lava, Matrix, multcomp, numDeriv, reshape2, sandwich

Suggests AICcmodavg, data.table, ggpubr, lattice, nlmeU, optimx, psych, Publish, qqtest, R.rsp, testthat

VignetteBuilder R.rsp

RoxygenNote 7.1.2

Collate '0-onload.R' 'LMMstar-package.R' 'LMMstar.options.R' 'anova.R' 'autoplot.R' 'baselineAdjustment.R' 'coef.R' 'confint.R' 'df.R' 'doc-data.R' 'dummy.coef.R' 'emmeans.R' 'estimate.R' 'fitted.R' 'formula.R' 'getCoef.R' 'getVarCov.R' 'iid.R' 'information.R' 'levels.R' 'lmm.R' 'logLik.R' 'model.frame.R' 'model.matrix.R' 'model.tables.R' 'moments.R' 'multcomp.R' 'nobs.R' 'plot.R'

'precompute.R' 'predict.R' 'print.R' 'reparametrize.R'
'residuals.R' 'sampleRem.R' 'score.R' 'structure-calc_Omega.R'
'structure-calc_d2Omega.R' 'structure-calc_dOmega.R'
'structure-initialization.R' 'structure-skeleton.R'
'structure.R' 'summarize.R' 'summary.R' 'terms.R' 'utils.R'
'vcov.R'

NeedsCompilation no

Author Brice Ozenne [aut, cre] (<https://orcid.org/0000-0001-9694-2956>),
Julie Forman [aut] (<https://orcid.org/0000-0001-7368-0869>)

Maintainer Brice Ozenne <brice.mh.ozenne@gmail.com>

Repository CRAN

Date/Publication 2022-01-03 11:10:05 UTC

R topics documented:

`LMMstar-package` *LMMstar package: repeated measurement models for discrete times*

Description

Companion R package for the course "Statistical analysis of correlated and repeated measurements for health science researchers" taught by the section of Biostatistics of the University of Copenhagen. It implements linear mixed models where the model for the variance-covariance of the residuals is specified via patterns (compound symmetry, unstructured). Statistical inference for mean, variance, and correlation parameters is performed based on the observed information and a Satterthwaite degrees of freedom. Normalized residuals are provided to assess model misspecification. Statistical inference can be performed for arbitrary linear combination(s) of model coefficients. Predictions can be computed conditional to covariates only or also to outcome values.

Currently only four types of model for the residual variance-covariance matrix are available:

- `"ID"`: Identity (no correlation, constant variance)
- `"IND"`: Independent (no correlation, time-specific variance)
- `"CS"`: compound symmetry (constant correlation, constant variable)
- `"UN"`: unstructured (time-specific correlation, time-specific variable)

It possible to stratify the last two structure with respect to a categorical variable.

The package is based on the `nlme::gls` function and the PROC MIXED from the SAS software. Adjustment for multiple comparisons is based on the multcomp package.

anova *Multivariate Wald Tests For Linear Mixed Model*

Description

Perform a Wald test testing simultaneously several null hypotheses corresponding to linear combinations of the model paramaters.

Usage

```
## S3 method for class 'lmm'
anova(
  object,
  effects = NULL,
  rhs = NULL,
  df = !is.null(object$df),
  ci = FALSE,
  transform.sigma = NULL,
  transform.k = NULL,
  transform.rho = NULL,
  transform.names = TRUE,
  ...
)

## S3 method for class 'anova_lmm'
confint(object, parm, level = 0.95, method = "single-step", ...)

## S3 method for class 'anova_lmm'
print(
  x,
  level = 0.95,
  method = "single-step",
  print.null = FALSE,
  columns = NULL,
  ...
)
```

Arguments

object	a lmm object. Only relevant for the anova function.
effects	[character] Should the Wald test be computed for all variables ("all"), or only variables relative to the mean ("mean" or "fixed"), or only variables relative to the variance structure ("variance"), or only variables relative to the correlation structure ("correlation"). Can also be use to specify linear combinations of coefficients, similarly to the linfct argument of the multcomp::glht function.
rhs	[numeric vector] the right hand side of the hypothesis. Only used when the argument effects is a matrix.

df	[logical] Should a F-distribution be used to model the distribution of the Wald statistic. Otherwise a chi-squared distribution is used.
ci	[logical] Should a confidence interval be output for each hypothesis?
transform.sigma, transform.k, transform.rho, transform.names	are passed to the vcov method. See details section in coef.lmm.
...	Not used. For compatibility with the generic method.
parm	Not used. For compatibility with the generic method.
level	[numeric, 0-1] nominal coverage of the confidence intervals.
method	[character] type of adjustment for multiple comparisons: one of "none", "bonferroni", "single-step". Not relevant for the global test (F-test or Chi-square test) - only relevant when testing each hypothesis and adjusting for multiplicity.
x	an anova_lmm object. Only relevant for print and confint functions.
print.null	[logical] should the null hypotheses be printed in the console?
columns	[character vector] Columns to be output. Can be any of "estimate", "se", "df", "lower", "upper", "p.value".

Details

By default confidence intervals and p-values are adjusted based on the distribution of the maximum-statistic. This is refered to as a single-step Dunnett multiple testing procedures in table II of Dmitrienko et al. (2013) and is performed using the multcomp package with the option `test = adjusted("single-step")`.

Value

A list of matrices containing the following columns:

- null: null hypothesis
- statistic: value of the test statistic
- df.num: degrees of freedom for the numerator (i.e. number of hypotheses)
- df.denom: degrees of freedom for the denominator (i.e. Satterthwaite approximation)
- p.value: p-value.

as well as an attribute contrast containing the contrast matrix encoding the linear combinations of coefficients (in columns) for each hypothesis (in rows).

References

Dmitrienko, A. and D'Agostino, R., Sr (2013), Traditional multiplicity adjustment methods in clinical trials. Statist. Med., 32: 5172-5218. https://doi.org/10.1002/sim.5990.

Examples

```
## simulate data in the long format
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

## fit Linear Mixed Model
eUN.lmm <- lmm(Y ~ X1 + X2 + X5, repetition = ~visit|id, structure = "UN", data = dL)

## chi-2 test
anova(eUN.lmm, df = FALSE)

## F-test
anova(eUN.lmm)
anova(eUN.lmm, effects = "all")
anova(eUN.lmm, effects = c("X1=0","X2+X5=10"), ci = TRUE)

if(require(multcomp)){
amod <- lmm(breaks ~ tension, data = warpbreaks)
e.glht <- glht(amod, linfct = mcp(tension = "Tukey"))
summary(e.glht, test = Chisqtest()) ## 0.000742

print(anova(amod, effect = mcp(tension = "Tukey"), df = FALSE), print.null = TRUE)

anova(amod, effect = mcp(tension = "Tukey"), ci = TRUE)
}
```

---

autoplot *Graphical Display For Linear Mixed Models*

Description

Graphical Display For Linear Mixed Models

Usage

```
## S3 method for class 'lmm'
autoplot(
  object,
  obs.alpha = 0,
  obs.size = c(2, 0.5),
  at = NULL,
  color = TRUE,
  ci = TRUE,
  ci.alpha = NA,
  plot = TRUE,
  mean.size = c(3, 1),
  size.text = 16,
  position.errorbar = "identity",
  ...
)
```

Arguments

object	a lmm object.
obs.alpha	[numeric, 0-1] When not NA, transparency parameter used to display the original data by cluster.
obs.size	[numeric vector of length 2] size of the point and line for the original data.
at	[data.frame] values for the covariates at which to evaluate the fitted values.
color	[character] name of the variable in the dataset used to color the curve.
ci	[logical] should confidence intervals be displayed?
ci.alpha	[numeric, 0-1] When not NA, transparency parameter used to display the confidence intervals.
plot	[logical] should the plot be displayed?
mean.size	[numeric vector of length 2] size of the point and line for the mean trajectory.
size.text	[numeric, >0] size of the font used to displayed text when using ggplot2.
position.errorbar	[character] relative position of the errorbars.
...	arguments passed to the predict method.

Value

A list with two elements

- data: data used to create the graphical display.
- plot: ggplot object.

---

baselineAdjustment *Perform Baseline Adjustment*

Description

Create a new variable based on a time variable and a group variable where groups are constrained to be equal at specific timepoints.

Usage

```
baselineAdjustment(object, variable, repetition, constrain, new.level = NULL)
```

Arguments

object	[data.frame] dataset
variable	[character] Column in the dataset to be constrained at specific timepoints.
repetition	[formula] Time and cluster structure, typically ~time|id. See examples below.
constrain	[vector] Levels of the time variable at which the variable is constained.
new.level	[character or numeric] Level used at the constraint. If NULL, then the first level of the variable argument is used.

Value

A vector of length the number of rows of the dataset.

Examples

```
data(ncgsL, package = "LMMstar")

## baseline adjustment 1
ncgsL$treat <- baselineAdjustment(ncgsL, variable = "group",
                 repetition= ~ visit|id, constrain = 1)
table(treat = ncgsL$treat, visit = ncgsL$visit, group = ncgsL$group)

e1.lmm <- suppressWarnings(lmm(cholest~visit*treat,
             data=ncgsL, repetition= ~ visit|id,
             structure = "CS"))


## baseline adjustment 2
ncgsL$treat2 <- baselineAdjustment(ncgsL, variable = "group", new.level = "none",
                 repetition= ~ visit|id, constrain = 1)
table(treat = ncgsL$treat2, visit = ncgsL$visit, group = ncgsL$group)

e2.lmm <- suppressWarnings(lmm(cholest~visit*treat2,
             data=ncgsL, repetition= ~ visit|id,
             structure = "CS"))
```

`blandAltmanL` *Data From The Bland Altman Study (Long Format)*

Description

Data From The Bland Altman Study where two methods to measure the peak expiratory flow rate (PEFR) where compared. This dataset is in the long format (i.e. one line per measurement).

- id Patient identifier.
- replicate Index of the measurement (first or second).
- method Device used to make the measurement (Wright peak flow meter or mini Wright peak flow meter).
- pefr Measurement (peak expiratory flow rate).

Usage

```
data(blandAltmanL)
```

References

Bland & Altman, Statistical methods for assessing agreement between two methods of clinical measurement, Lancet, 1986; i: 307-310.

`blandAltmanW` *Data From The Bland Altman Study (Wide Format)*

Description

Data From The Bland Altman Study where two methods to measure the peak expiratory flow rate (PEFR) where compared. This dataset is in the wide format (i.e. one line per patient).

- id Patient identifier
- wright1 First measurement made with a Wright peak flow meter.
- wright2 Second measurement made with a Wright peak flow meter.
- mini1 First measurement made with a mini Wright peak flow meter.
- mini2 Second measurement made with a mini Wright peak flow meter.

Usage

```
data(blandAltmanW)
```

References

Bland & Altman, Statistical methods for assessing agreement between two methods of clinical measurement, Lancet, 1986; i: 307-310.

`bloodpressureL` *Data From The Blood Pressure Study (Long Format)*

Description

Data from a cross-over trial comparing the impact of three formulations of a drug on the blood pressure. The study was conducted on 12 male volunteers randomly divided into tree groups and receiving each of the three formulations with a wash-out period of one week.

- id Patient identifier
- sequence sequence of treatment
- treatment Formulation of the treatment: A (50 mg tablet) B (100 mg tablet) C (sustained-release formulation capsule)
- period time period (in weeks)
- duration duration of the drug (in hours)

Usage

```
data(bloodpressureL)
```

References

TO ADD

`calciumL` *Data From The Calcium Supplements Study (Long Format)*

Description

Data from a randomized study including 112 girls at age 11 investigate the effect of a calcium supplement (n=55) vs. placebo (n=57) on bone mineral density over a 2 year follow-up. The clinical question is: does a calcium supplement help to increase bone gain in adolescent women? This dataset is in the long format (i.e. one line per measurement).

- girl Patient identifier
- grp Treatment group: calcium supplement (coded `C`) or placebo (coded `P`)
- visit Visit index
- bmd Bone mineral density (mg/cm3)
- time.obs Visit time (in years)
- time.num Scheduled visit time (numeric variable, in years)
- time.fac Scheduled visit time (factor variable)

Usage

```
data(calciumL)
```

References

TO ADD

calciumW *Data From The Calcium Supplements Study (Wide Format)*

Description

Data from a randomized study including 112 girls at age 11 investigate the effect of a calcium supplement (n=55) vs. placebo (n=57) on bone mineral density over a 2 year follow-up. The clinical question is: does a calcium supplement help to increase bone gain in adolescent women? This dataset is in the wide format (i.e. one line per patient).

- girl Patient identifier
- grp Treatment group: calcium supplement (coded `C`) or placebo (coded `P`)
- obstime1 Time after the start of the study at which the first visit took place (in years).
- obstime2 Time after the start of the study at which the second visit took place (in years).
- obstime3 Time after the start of the study at which the third visit took place (in years).
- obstime4 Time after the start of the study at which the fourth visit took place (in years).
- obstime5 Time after the start of the study at which the fifth visit took place (in years).
- bmd1 Bone mineral density measured at the first visit (in mg/cm3).
- bmd2 Bone mineral density measured at the second visit (in mg/cm3).
- bmd3 Bone mineral density measured at the third visit (in mg/cm3).
- bmd4 Bone mineral density measured at the fourth visit (in mg/cm3).
- bmd5 Bone mineral density measured at the fifth visit (in mg/cm3).

Usage

```
data(calciumW)
```

References

Vonesh and Chinchilli 1997. Linear and Nonlinear models for the analysis of repeated measurement (Table 5.4.1 on page 228). New York: Marcel Dekker.

ckdL *CKD long*

Description

TODO

- id Patient identifier
- allocation
- sex
- age
- visit
- time
- pwv
- aix
- dropout

Usage

```
data(ckdL)
```

References

TO ADD

ckdW *CKD wide*

Description

TODO

- id Patient identifier
- allocation
- sex
- age
- pwv0
- pwv12
- pwv24
- aix0
- aix12
- aix24
- dropout

Usage

```
data(ckdW)
```

References

TO ADD

---

coef *Extract Coefficients From a Linear Mixed Model*

Description

Extract coefficients from a linear mixed model.

Usage

```
## S3 method for class 'lmm'
coef(
  object,
  effects = NULL,
  strata = NULL,
  transform.sigma = "none",
  transform.k = "none",
  transform.rho = "none",
  transform.names = TRUE,
  ...
)
```

Arguments

object	a lmm object.
effects	[character] Should all coefficients be output ("all"), or only coefficients relative to the mean ("mean" or "fixed"), or only coefficients relative to the variance structure ("variance"), or only coefficients relative to the correlation structure ("correlation").
strata	[character vector] When not NULL, only output coefficient relative to specific levels of the variable used to stratify the mean and covariance structure.
transform.sigma	[character] Transformation used on the variance coefficient for the reference level. One of "none", "log", "square", "logsquare" - see details.
transform.k	[character] Transformation used on the variance coefficients relative to the other levels. One of "none", "log", "square", "logsquare", "sd", "logsd", "var", "logvar" - see details.
transform.rho	[character] Transformation used on the correlation coefficients. One of "none", "atanh", "cov" - see details.

`transform.names`
: [logical] Should the name of the coefficients be updated to reflect the transformation that has been used?

`...`
: Not used. For compatibility with the generic method.

Details

transform.sigma:

- `"none"` ouput residual standard error.
- `"log"` ouput log-transformed residual standard error.
- `"square"` ouput residual variance.
- `"logsquare"` ouput log-transformed residual variance.

transform.k:

- `"none"` ouput ratio between the residual standard error of the current level and the reference level.
- `"log"` ouput log-transformed ratio between the residual standard errors.
- `"square"` ouput ratio between the residual variances.
- `"logsquare"` ouput log-transformed ratio between the residual variances.
- `"sd"` ouput residual standard error of the current level.
- `"logsd"` ouput residual log-transformed standard error of the current level.
- `"var"` ouput residual variance of the current level.
- `"logvar"` ouput residual log-transformed variance of the current level.

transform.rho:

- `"none"` ouput correlation coefficient.
- `"atanh"` ouput correlation coefficient after tangent hyperbolic transformation.
- `"cov"` ouput covariance coefficient.

Value

A vector with the value of the model coefficients.

Examples

```
## simulate data in the long format
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

## fit linear mixed model
eUN.lmm <- lmm(Y ~ X1 + X2 + X5, repetition = ~visit|id, structure = "UN", data = dL, df = FALSE)
```

```
## output coefficients
coef(eUN.lmm)
coef(eUN.lmm, effects = "mean")
coef(eUN.lmm, transform.sigma = "none", transform.k = "none", transform.rho = "none")
```

---

confint *Statistical Inference for Linear Mixed Model*

Description

Compute confidence intervals (CIs) and p-values for the coefficients of a linear mixed model.

Usage

```
## S3 method for class 'lmm'
confint(
  object,
  parm = NULL,
  level = 0.95,
  effects = NULL,
  robust = FALSE,
  null = NULL,
  strata = NULL,
  columns = NULL,
  df = NULL,
  type.information = NULL,
  transform.sigma = NULL,
  transform.k = NULL,
  transform.rho = NULL,
  transform.names = TRUE,
  backtransform = NULL,
  ...
)
```

Arguments

object	a lmm object.
parm	Not used. For compatibility with the generic method.
level	[numeric,0-1] the confidence level of the confidence intervals.
effects	[character] Should the CIs/p-values for all coefficients be output ("all"), or only for mean coefficients ("mean" or "fixed"), or only for variance coefficients ("variance"), or only for correlation coefficients ("correlation").
robust	[logical] Should robust standard error (aka sandwich estimator) be output instead of the model-based standard errors. Not feasible for variance or correlation coefficients estimated by REML.
null	[numeric vector] the value of the null hypothesis relative to each coefficient.

strata	[character vector] When not NULL, only output coefficient relative to specific levels of the variable used to stratify the mean and covariance structure.
columns	[character vector] Columns to be output. Can be any of "estimate", "se", "statistic", "df", "null", "lower", "upper", "p.value".
df	[logical] Should a Student's t-distribution be used to model the distribution of the coefficient. Otherwise a normal distribution is used.
type.information, transform.sigma, transform.k, transform.rho, transform.names	are passed to the vcov method. See details section in coef.lmm.
backtransform	[logical] should the variance/covariance/correlation coefficient be backtransformed?
...	Not used. For compatibility with the generic method.

Value

A data.frame containing for each coefficient (in rows):

- column estimate: the estimate.
- column se: the standard error.
- column statistic: the test statistic.
- column df: the degree of freedom.
- column lower: the lower bound of the confidence interval.
- column upper: the upper bound of the confidence interval.
- column null: the null hypothesis.
- column p.value: the p-value relative to the null hypothesis.

See Also

the function anova to perform inference about linear combinations of coefficients and adjust for multiple comparisons.

Examples

```
## simulate data in the long format
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

## fit Linear Mixed Model
eUN.lmm <- lmm(Y ~ X1 + X2 + X5, repetition = ~visit|id, structure = "UN", data = dL)

## based on a Student's t-distribution with transformation
confint(eUN.lmm)
## based on a Student's t-distribution without transformation
confint(eUN.lmm, transform.sigma = "none", transform.k = "none", transform.rho = "none")
## based on a Normal distribution with transformation
confint(eUN.lmm, df = FALSE)
```

CS *Compound Symmetry Structure*

Description

Variance-covariance structure where the residuals have constant variance and correlation. Can be stratified on a categorical variable.

Usage

```
CS(formula, var.cluster, var.time, ...)
```

Arguments

formula	formula indicating on which variable to stratify the residual variance and correlation (left hand side) and variables influencing the residual variance (right hand side).
var.cluster	[character] cluster variable.
var.time	[character] time variable.
...	Not used. For compatibility with other structures.

Details

A typical formula would be ~1, indicating a variance constant over time and the same correlation between all pairs of times.

Value

An object of class CS that can be passed to the argument structure of the lmm function.

Examples

```
CS(~1, var.cluster = "id", var.time = "time")
CS(gender~1, var.cluster = "id", var.time = "time")
CS(list(~time,~1), var.cluster = "id", var.time = "time")
CS(list(gender~time,gender~1), var.cluster = "id", var.time = "time")
```

dummy.coef.lmm *Marginal Mean Values For Linear Mixed Model*

Description

Compute the marginal mean (via the emmeans package) for each combination of categorical covariates. When there is no numeric covariate, this outputs all the mean values fitted by the model.

Usage

```
## S3 method for class 'lmm'
dummy.coef(object, drop = TRUE, ...)
```

Arguments

object	a lmm object.
drop	[logical] should combinations of covariates that do no exist in the original dataset be removed?
...	arguments passed to emmeans.

Value

A data.frame containing the level for which the means have been computed (if more than one), the estimated mean (estimate), standard error (se), degree of freedom (df), and 95

estfun *Extract the Score Function for Multcomp*

Description

Extract the Score Function for Multcomp. For internal use.

Usage

```
## S3 method for class 'lmm'
estfun(x, ...)
```

Arguments

x	a lmm object.
...	Not used. For compatibility with the generic method.

Value

A matrix containing the score function for each model parameter (columns) relative to each cluster (rows).

Examples

```
## simulate data in the long format
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

## fit Linear Mixed Model
eUN.lmm <- lmm(Y ~ X1 + X2 + X5, repetition = ~visit|id, structure = "UN", data = dL, df = FALSE)

## test multiple linear hypotheses
if(require(multcomp)){
LMMstar.options(effects = c("mean"))
e.glht <- multcomp::glht(eUN.lmm)
e.glht$linfct
}
```

`fitted.lmm` *Predicted Mean Value For Linear Mixed Model*

Description

Predicted Mean Value For Linear Mixed Model

Usage

```
## S3 method for class 'lmm'
fitted(
  object,
  newdata = NULL,
  keep.newdata = FALSE,
  impute = FALSE,
  se.impute = FALSE,
  ...
)
```

Arguments

`object`	a `lmm` object.
`newdata`	[data.frame] the covariate values for each cluster.
`keep.newdata`	[logical] Should the argument `newdata` be output along side the predicted values?
`impute`	[logical] Should the missing data in the outcome be imputed based on covariates and other outcome values from the same cluster.
`se.impute`	[character] If `FALSE` the most likely value is imputed. Otherwise the imputed value is sampled from a normal distribution. The value of the argument determine which standard deviation is used: all uncertainty about the predicted value (`"total"`), only uncertainty related to the estimation of the model parameters

("estimate"), or only uncertainty related to the residual variance of the outcome ("residual"). Passed to predict.lmm.

... Not used. For compatibility with the generic method.

Value

When keep.newdata==FALSE:

- if impute=FALSE a vector of length the number of row of newdata containing the fitted values (i.e. based on the covariates only).
- if impute=TRUE a vector of length the number of missing values in the outcome of newdata containing the cluster-specific conditional means (i.e. based on the covariates and outcome measurements from the same cluster).

When keep.newdata==TRUE, a dataframe with an additional column containing the fitted values (i.e. based on the covariates only). If impute=TRUE, the missing value in the outcome column are replaced by the cluster-specific conditional means (i.e. based on the covariates and outcome measurements from the same cluster).

Examples

```
#### simulate data in the long format ####
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

#### fit Linear Mixed Model ####
eCS.lmm <- lmm(Y ~ X1 + X2 + X5, repetition = ~visit|id,
               structure = "CS", data = dL, df = FALSE)

## prediction
fitted(eCS.lmm)
fitted(eCS.lmm, newdata = data.frame(X1 = 1, X2 = 2, X5 = 3))
fitted(eCS.lmm, newdata = data.frame(X1 = 1, X2 = 2, X5 = 3), keep.newdata = TRUE)

#### fit Linear Mixed Model with missing data ####
dL2 <- dL
dL2[3,"Y"] <- NA
eCS2.lmm <- lmm(Y ~ X1 + X2 + X5, repetition = ~visit|id,
                structure = "CS", data = dL2, df = FALSE)

## most likely value to impute
fitted(eCS2.lmm, impute = TRUE)
head(fitted(eCS2.lmm, impute = TRUE, keep.newdata = TRUE))

## multiple imputation
dL2.imp1 <- data.frame(imp = "1",
    fitted(eCS2.lmm, impute = TRUE, se.impute = "total", keep.newdata = TRUE))
dL2.imp2 <- data.frame(imp = "2",
    fitted(eCS2.lmm, impute = TRUE, se.impute = "total", keep.newdata = TRUE))
head(dL2.imp1)
head(dL2.imp2)
```

gastricbypassL *Data From The Gastric Bypass Study (Long Format)*

Description

Data from the gastric bypass study where the bodyweight and serum glucagon (a gut hormone) were measured in 20 obese subjects prior and after gastric bypass surgery. This dataset is in the long format (i.e. one line per measurement).

- id Patient identifier
- visit The visit index.
- time The time at which the visit took place.
- weight Bodyweight (in kg) measured during the visit.
- glucagon Glucagon measured during the visit.

Usage

```
data(gastricbypassL)
```

References

The effect of Roux-en-Y gastric bypass surgery on the gut mucosal gene expression profile and circulating gut hormones. https://easddistribute.m-anage.com/from.storage?image=4iBH9mRQm1kfeEHULC2Cxovdly

gastricbypassW *Data From The Gastric Bypass Study (Wide Format)*

Description

Data from the gastric bypass study where the bodyweight and serum glucagon (a gut hormone) were measured in 20 obese subjects prior and after gastric bypass surgery. This dataset is in the wide format (i.e. one line per patient).

- id Patient identifier
- weight1 Bodyweight (in kg) 3 months before surgery.
- weight2 Bodyweight (in kg) 1 week before surgery.
- weight3 Bodyweight (in kg) 1 week after surgery.
- weight4 Bodyweight (in kg) 3 months after surgery.
- glucagonAUC1 Glucagon value 3 months before surgery.
- glucagonAUC2 Glucagon value 1 week before surgery.
- glucagonAUC3 Glucagon value 1 week after surgery.
- glucagonAUC4 Glucagon value 3 months after surgery.

Usage

```
data(gastricbypassW)
```

References

The effect of Roux-en-Y gastric bypass surgery on the gut mucosal gene expression profile and circulating gut hormones. https://easddistribute.m-anage.com/from.storage?image=4iBH9mRQm1kfeEHULC2Cxovdly

---

getCoef *Extract Model Coefficients With Confidence Intervals*

Description

Extract all model coefficients with confidence intervals.

Usage

```
getCoef(object, conf.level, effects, format, add.type, ...)
```

Arguments

object	a lm, gls, lme, or lmm object.
conf.level	[numeric 0-1] Confidence level of the confidence intervals.
effects	[character vector] Type of coefficient to be output. Can be coefficients relative to the expectation of the outcome ("mean" or "fixed") or to the variance-covariance structure of the residuals ("variance").
format	[character] How the output should be shaped. Can be "default", "estimate", "publish", or "SAS".
add.type	[logical] Should the type of parameter be added.
...	argument passed to the publish function (when format="publish").

Details

Argument **format**:
Setting the argument to "default" outputs a data.frame with columns type (mean or covariance), term (name of the coefficient), estimate, std.error, t.value, p.value, lower, upper.

Setting the argument to "publish" outputs a data.frame with columns Variable, Units Coefficients, CI, and p-value. Call the function publish from the publish package.

Setting the argument to "estimate" outputs a vector containing the estimated parameter values.

Argument **add.type**:
When TRUE, there can be 4 types of parameters in the output:

- "mean": coefficients relative to the conditional mean of the outcome given the covariates.
- "std.residual": (reference) residual standard deviation.
- "factor.std.residual": multiplicative factor to the residual standard deviation.
- "correlation": correlation coefficient between the residuals.
- "std.random": standard error of the random effects.

Value

A data.frame or a vector (see details section)

Examples

```
data(gastricbypassL, package = "LMMstar")
library(nlme)

#### linear model ####
## (wrong model as it does not account for repeated measurements)
e.lm <- lm(weight ~ time, data = gastricbypassL)

getCoef(e.lm)
getCoef(e.lm, format = "estimate")
getCoef(e.lm, effects = "variance")
getCoef(e.lm, effects = "variance", format = "estimate")
if(require(Publish)){
getCoef(e.lm, format = "publish")
}
getCoef(e.lm, format = "SAS")

#### gls model ####
e.gls <- gls(weight ~ time,
             correlation = corSymm(form =~as.numeric(visit)|id),
             weights = varIdent(form =~1|visit),
             data = gastricbypassL)
getCoef(e.gls)
getCoef(e.gls, effects = "variance")
getCoef(e.gls, effects = "variance", format = "estimate")
if(require(Publish)){
getCoef(e.gls, format = "publish")
}
getCoef(e.gls, format = "SAS")

#### lme model ####
e.lme <- lme(weight ~ time,
             random = ~1|id,
             weights = varIdent(form =~1|visit),
             data = gastricbypassL)
getCoef(e.lme)
getCoef(e.lme, effects = "variance")
getCoef(e.lme, effects = "variance", format = "estimate")
if(require(Publish)){
getCoef(e.lme, format = "publish")
}
getCoef(e.lme, format = "SAS")
```

---

getVarCov *Extract The Residuals Variance-Covariance Matrix From a Linear Mixed Model*

Description

Extract the unique set of residuals variance-covariance matrices or the one relative to specific clusters.

Usage

```
## S3 method for class 'lmm'
getVarCov(
  obj,
  individual = NULL,
  p = NULL,
  simplifies = TRUE,
  strata = NULL,
  ...
)
```

Arguments

obj	a lmm object.
individual	[character, data.frame, NULL] identifier of the cluster(s) for which to extract the residual variance-covariance matrix. For new clusters, a dataset containing the information (cluster, time, strata, ...) to be used to generate the residual variance-covariance matrices. When NULL, will output complete data covariance patterns.
p	[numeric vector] value of the model coefficients at which to evaluate the residual variance-covariance matrix. Only relevant if differs from the fitted values.
simplifies	[logical] When there is only one variance-covariance matrix, output a matrix instead of a list of matrices.
strata	[character vector] When not NULL and argument individual is not specified, only output the residual variance-covariance matrix relative to specific levels of the variable used to stratify the mean and covariance structure.
...	Not used. For compatibility with the generic method.

Value

A list where each element contains a residual variance-covariance matrix. Can also be directly a matrix when argument is simplifies=TRUE and there is a single residual variance-covariance matrix.

Examples

```
## simulate data in the long format
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

## fit Linear Mixed Model
eUN.lmm <- lmm(Y ~ X1 + X2 + X5, repetition = ~visit|id, structure = "UN", data = dL, df = FALSE)

## extract residuals variance covariance matrix
```

```
getVarCov(eUN.lmm) ## unique patterns
getVarCov(eUN.lmm, individual = c("1","5")) ## existing individuals
getVarCov(eUN.lmm, individual = dL[1:7,,drop=FALSE]) ## new individuals
```

ID *identity Structure*

Description

Variance-covariance structure where the residuals are independent and identically distribution. Can be stratified on a categorical variable.

Usage

```
ID(formula, var.cluster, var.time, ...)
```

Arguments

formula	formula indicating on which variable to stratify the residual variance (left hand side).
var.cluster	[character] cluster variable.
var.time	[character] time variable.
...	Not used. For compatibility with other structures.

Details

A typical formula would be ~1.

Value

An object of class IND that can be passed to the argument structure of the lmm function.

Examples

```
ID(NULL, var.cluster = "id", var.time = "time")
ID(~1, var.cluster = "id", var.time = "time")
ID(gender~1, var.cluster = "id", var.time = "time")
```

IND *Independence Structure*

Description

Variance-covariance structure where the residuals are independent. Can be stratified on a categorical variable.

Usage

```
IND(formula, var.cluster, var.time, add.time)
```

Arguments

formula	formula indicating variables influencing the residual variance, using either as a multiplicative factor (right hand side) or stratification (left hand side) to model their effect.
var.cluster	[character] cluster variable.
var.time	[character] time variable.
add.time	Should the default formula (i.e. when NULL) contain a time effect.

Details

A typical formula would be either ~1 indicating constant variance or ~time indicating a time dependent variance.

Value

An object of class IND that can be passed to the argument structure of the lmm function.

Examples

```
IND(NULL, var.cluster = "id", var.time = "time", add.time = TRUE)
IND(~1, var.cluster = "id", var.time = "time")
IND(gender~1, var.cluster = "id", var.time = "time")

IND(~time, var.cluster = "id", var.time = "time")
IND(gender~time, var.cluster = "id", var.time = "time")
IND(~time+gender, var.cluster = "id", var.time = "time")
```

information *Extract The Information From a Linear Mixed Model*

Description

Extract or compute the (expected) second derivative of the log-likelihood of a linear mixed model.

Usage

```
## S3 method for class 'lmm'
information(
  x,
  effects = NULL,
  data = NULL,
  p = NULL,
  indiv = FALSE,
  type.information = NULL,
  transform.sigma = NULL,
  transform.k = NULL,
  transform.rho = NULL,
  transform.names = TRUE,
  ...
)
```

Arguments

x	a lmm object.
effects	[character] Should the information relative to all coefficients be output ("all" or "fixed"), or only coefficients relative to the mean ("mean"), or only coefficients relative to the variance and correlation structure ("variance" or "correlation").
data	[data.frame] dataset relative to which the information should be computed. Only relevant if differs from the dataset used to fit the model.
p	[numeric vector] value of the model coefficients at which to evaluate the information. Only relevant if differs from the fitted values.
indiv	[logical] Should the contribution of each cluster to the information be output? Otherwise output the sum of all clusters of the derivatives.
type.information	[character] Should the expected information be computed (i.e. minus the expected second derivative) or the observed inforamtion (i.e. minus the second derivative).
transform.sigma	[character] Transformation used on the variance coefficient for the reference level. One of "none", "log", "square", "logsquare" - see details.
transform.k	[character] Transformation used on the variance coefficients relative to the other levels. One of "none", "log", "square", "logsquare", "sd", "logsd", "var", "logvar" - see details.

`transform.rho`	[character] Transformation used on the correlation coefficients. One of `"none"`, `"atanh"`, `"cov"` - see details.
`transform.names`	[logical] Should the name of the coefficients be updated to reflect the transformation that has been used?
`...`	Not used. For compatibility with the generic method.

Details

For details about the arguments **transform.sigma**, **transform.k**, **transform.rho**, see the documentation of the coef function.

Value

When argument indiv is `FALSE`, a matrix with the value of the infroamtion relative to each pair of coefficient (in rows and columns) and each cluster (in rows). When argument indiv is `TRUE`, a 3-dimensional array with the value of the information relative to each pair of coefficient (dimension 2 and 3) and each cluster (dimension 1).

`levels.lmm` *Contrasts and Reference Level*

Description

Contrasts and reference level used when modeling the mean in a linear mixed modek.

Usage

```
## S3 method for class 'lmm'
levels(x)
```

Arguments

`x`	an `lmm` object

Value

a list with two elements

- all: contrast matrix for each categorical or factor variable
- reference: reference level: one value for each categorical variable

`lmm` *Fit Linear Mixed Model*

Description

Fit a linear mixed model defined by a mean and a covariance structure. g

Usage

```
lmm(
  formula,
  repetition,
  structure,
  data,
  method.fit = NULL,
  df = NULL,
  type.information = NULL,
  trace = NULL,
  control = NULL
)
```

Arguments

`formula`	[formula] Specify the model for the mean. On the left hand side the outcome and on the right hand side the covariates affecting the mean value. E.g. Y ~ Gender + Gene.
`repetition`	[formula] Specify the model for the covariance. On the right hand side the time/repetition variable and the grouping variable, e.g. ~ timelid. On the left hand side, a possible stratification variable, e.g. group ~ timelid. In that case the mean structure should only be stratified on this variable using interactions.
`structure`	[character] type of covariance structure, either `"CS"` (compound symmetry) or `"UN"` (unstructured).
`data`	[data.frame] dataset (in the long format) containing the observations.
`method.fit`	[character] Should Restricted Maximum Likelihoood (`"REML"`) or Maximum Likelihoood (`"ML"`) be used to estimate the model parameters?
`df`	[logical] Should the degree of freedom be computed using a Satterthwaite approximation?
`type.information`	[character] Should the expected information be computed (i.e. minus the expected second derivative) or the observed inforamtion (i.e. minus the second derivative).
`trace`	[interger, >0] Show the progress of the execution of the function.
`control`	[list] Control values for the optimization method. The element `optimizer` indicates which optimizer to use and additional argument will be pass to the optimizer.

Details

Computation time the `lmm` has not been developped to be a fast function as, by default, it uses REML estimation with the observed information matrix and uses a Satterthwaite approximation to compute degrees of freedom (this require to compute the third derivative of the log-likelihood which is done by numerical differentiation). The computation time can be substantially reduced by using ML estimation with the expected information matrix and no calculation of degrees of freedom: arguments `method.fit="ML"`, `type.information="expected"`, `df=FALSE`. This will, however, lead to less accurate p-values and confidence intervals in small samples.

By default, the estimation of the model parameters will be made using the `nlme::gls` function. See argument optimizer in `LMMstar.options`

Value

an object of class `lmm` containing the estimated parameter values, the residuals, and relevant derivatives of the likelihood.

See Also

`summary.lmm` for a summary of the model fit.
`model.tables.lmm` for a data.frame containing estimates with their uncertainty.
`plot.lmm` for a graphical display of the model fit or diagnostic plots.
`levels.lmm` to display the reference level.
`anova.lmm` for testing linear combinations of coefficients (F-test, multiple Wald tests)
`getVarCov.lmm` for extracting estimated residual variance-covariance matrices.
`residuals.lmm` for extracting residuals or creating residual plots (e.g. qqplots).
`predict.lmm` for evaluating mean and variance of the outcome conditional on covariates or other outcome values.

Examples

```
#### 1- simulate data in the long format ####
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")
dL$X1 <- as.factor(dL$X1)
dL$X2 <- as.factor(dL$X2)

#### 2- fit Linear Mixed Model ####
eCS.lmm <- lmm(Y ~ X1 * X2 + X5, repetition = ~visit|id, structure = "CS", data = dL)

logLik(eCS.lmm) ## -670.9439
summary(eCS.lmm)


#### 3- estimates ####
## reference level
levels(eCS.lmm)$reference
## mean parameters
coef(eCS.lmm)
model.tables(eCS.lmm)
confint(eCS.lmm)
```

```
if(require(emmeans)){
  dummy.coef(eCS.lmm)
}

## all parameters
coef(eCS.lmm, effects = "all")
model.tables(eCS.lmm, effects = "all")
confint(eCS.lmm, effects = "all")

## variance-covariance structure
getVarCov(eCS.lmm)

#### 4- diagnostic plots ####
quantile(residuals(eCS.lmm))
quantile(residuals(eCS.lmm, type = "normalized"))

## Not run:
if(require(ggplot2)){
  ## investigate misspecification of the mean structure
  plot(eCS.lmm, type = "scatterplot")
  ## investigate misspecification of the variance structure
  plot(eCS.lmm, type = "scatterplot2")
  ## investigate misspecification of the correlation structure
  plot(eCS.lmm, type = "correlation")
  ## investigate misspecification of the residual distribution
  plot(eCS.lmm, type = "qqplot")
}

## End(Not run)

#### 5- statistical inference ####
anova(eCS.lmm) ## effect of each variable
anova(eCS.lmm, effects = "X11-X21=0") ## test specific coefficient
## test several hypothese with adjustment for multiple comparisons
anova(eCS.lmm, effects = c("X11=0","X21=0"), ci = TRUE)

#### 6- prediction ####
## conditional on covariates
newd <- dL[1:3,]
predict(eCS.lmm, newdata = newd, keep.newdata = TRUE)
## conditional on covariates and outcome
newd <- dL[1:3,]
newd$Y[3] <- NA
predict(eCS.lmm, newdata = newd, type = "dynamic", keep.newdata = TRUE)
```

LMMstar.options *Global options for LMMstar package*

Description

Update or select global options for the LMMstar package.

Usage

```
LMMstar.options(..., reinitialise = FALSE)
```

Arguments

...	options to be selected or updated
reinitialise	should all the global parameters be set to their default value

Details

The options are:

- backtransform.confint [logical]: should variance/covariance/correlation estimates be back-transformed when they are transformed on the log or atanh scale. Used by `confint`.
- columns.anova [character vector]: columns to ouput when using `anova` with argument `ci=TRUE`.
- columns.confint [character vector]: columns to ouput when using `confint`.
- columns.summary [character vector]: columns to ouput when displaying the model coefficients using `summary`.
- df [logical]: should approximate degrees of freedom be computed for Wald and F-tests. Used by `lmm`, `anova`, `predict`, and `confint`.
- drop.X [logical]: should columns causing non-identifiability of the model coefficients be dropped from the design matrix. Used by `lmm`.
- effects [character]: parameters relative to which estimates, score, information should be output.
- min.df [integer]: minimum possible degree of freedom. Used by `confint`.
- method.fit [character]: objective function when fitting the Linear Mixed Model (REML or ML). Used by `lmm`.
- method.numDeriv [character]: type used to approximate the third derivative of the log-likelihood (when computing the degrees of freedom). Can be `"simple"` or `"Richardson"`. See `numDeriv::jacobian` for more details. Used by `lmm`.
- optimizer [character]: method used to estimate the model parameters: can the `nlme::gls` (`"gls"`) or an algorithm combine fisher scoring for the variance parameters and generalized least squares for the mean parameters (`"FS"`).
- param.optimizer [numeric vector]: default option for the `FS` optimization routine: maximum number of gradient descent iterations (`n.iter`), maximum acceptable score value (`tol.score`), maximum acceptable change in parameter value (`tol.param`).
- precompute.moments [logical]: Should the cross terms between the residuals and design matrix be pre-computed. Useful when the number of subject is substantially larger than the number of mean paramters.
- trace [logical]: Should the progress of the execution of the `lmm` function be displayed?

- tranform.sigma, tranform.k, tranform.rho: transformation used to compute the confidence intervals/p-values for the variance and correlation parameters. See the detail section of the coef function for more information. Used by `lmm`, `anova` and `confint`.
- type.information [character]: Should the expected or observed information (`"expected"` or `"observed"`) be used to perform statistical inference? Used by `lmm`, `anova` and `confint`.

Value

A list containing the default options.

`LMMstar2emmeans` *Link to emmeans package*

Description

Link to emmeans package. Not meant for direct use.

Usage

```
## S3 method for class 'lmm'
recover_data(object, ...)

## S3 method for class 'lmm'
emm_basis(object, trms, xlev, grid, ...)
```

Arguments

`object`	a `lmm` object.
`...`	Not used. For compatibility with the generic method.
`trms`	see `emmeans::emm_basis` documentation
`xlev`	see `emmeans::emm_basis` documentation
`grid`	see `emmeans::emm_basis` documentation

Value

dataset or list used by the emmeans package.

`logLik` *Extract The Log-Likelihood From a Linear Mixed Model*

Description

Extract or compute the log-likelihood of a linear mixed model.

Usage

```
## S3 method for class 'lmm'
logLik(object, data = NULL, p = NULL, indiv = FALSE, ...)
```

Arguments

`object`	a `lmm` object.
`data`	[data.frame] dataset relative to which the log-likelihood should be computed. Only relevant if differs from the dataset used to fit the model.
`p`	[numeric vector] value of the model coefficients at which to evaluate the log-likelihood. Only relevant if differs from the fitted values.
`indiv`	[logical] Should the contribution of each cluster to the log-likelihood be output? Otherwise output the sum of all clusters of the derivatives.
`...`	Not used. For compatibility with the generic method.

Details

transform:

- 0 means no transformation i.e. ouput stanrdard error, ratio of standard errors, and correlations.
- 1 means log/atanh transformation i.e. ouput log(stanrdard error), log(ratio of standard errors), and atanh(correlations).
- 2 ouput variance coefficients and correlations.

indiv: only relevant when using maximum likelihood. Must be `FALSE` when using restricted maximum likelihood.

Value

A numeric value (total logLikelihood) or a vector of numeric values, one for each cluster (cluster specific logLikelihood).

model.tables *Statistical Inference for Linear Mixed Model*

Description

Export estimates, standard errors, degrees of freedom, confidence intervals (CIs) and p-values for the mean coefficients of a linear mixed model.

Usage

```
## S3 method for class 'lmm'
model.tables(x, ...)
```

Arguments

x	a lmm object.
...	arguments to be passed to the confint method. Should not contain the argument column.

Details

This function simply calls confint with a specific value for the argument column.

ncgsL *Data From National Cooperative Gallstone Study (Long Format)*

Description

Data from the National Cooperative Gallstone Study (NCGS), a randomized study where the level of serum cholesterol was measured at baseline and after intake of high-dose chenondiol (750mg/day) or placebo. This dataset is in the long format (i.e. one line per measurement).

- group Treatment group: highdose or placebo.
- id Patient identifier
- visit visit index.
- cholest cholesterol measurement.
- time time after the start of the study at which the measurement has been done (in month). Treatment is given at 0+.

Usage

```
data(ncgsL)
```

References

Grundy SM, Lan SP, Lachin J. The effects of chenodiol on biliary lipids and their association with gallstone dissolution in the National Cooperative Gallstone Study (NCGS). J Clin Invest. 1984 Apr;73(4):1156-66. doi: 10.1172/JCI111301.

`ncgsW` *Data From National Cooperative Gallstone Study (Wide Format)*

Description

Data from the National Cooperative Gallstone Study (NCGS), a randomized study where the level of serum cholesterol was measured at baseline and after intake of high-dose chenondiol (750mg/day) or placebo. This dataset is in the wide format (i.e. one line per patient).

- group Treatment group: highdose or placebo.
- id Patient identifier
- cholest1 cholesterol measurement at baseline (before treatment).
- cholest2 cholesterol measurement at 6 months (after treatment).
- cholest3 cholesterol measurement at 12 months (after treatment).
- cholest4 cholesterol measurement at 20 months (after treatment).
- cholest5 cholesterol measurement at 24 months (after treatment).

Usage

```
data(ncgsW)
```

References

Grundy SM, Lan SP, Lachin J. The effects of chenodiol on biliary lipids and their association with gallstone dissolution in the National Cooperative Gallstone Study (NCGS). J Clin Invest. 1984 Apr;73(4):1156-66. doi: 10.1172/JCI111301.

`plot` *Graphical Display For Linear Mixed Models*

Description

Display fitted values or residual plot for the mean, variance, and correlation structure. Can also display quantile-quantile plot relative to the normal distribution.

Usage

```
## S3 method for class 'lmm'
plot(
  x,
  type = "fit",
  type.residual = "normalized",
  by.time = TRUE,
  ci = TRUE,
  plot = TRUE,
  ci.alpha = 0.2,
  mean.size = c(3, 1),
  size.text = 16,
  ...
)
```

Arguments

x	a lmm object.
type	[character] the type of plot: "fit", "qqplot", "correlation", "scatterplot", "scatterplot2", "partial".
type.residual	[character] the type of residual to be used. Not relevant for type="fit". By default, normalized residuals are used except when requesting a partial residual plot.
by.time	[logical] should a separate plot be made at each repetition or a single plot over all repetitions be used? Only relevant for type="qqplot", type="scatterplot", and type="scatterplot2".
ci	[logical] should confidence intervals be displayed?
plot	[logical] should the plot be displayed?
ci.alpha	[numeric, 0-1] Transparency parameter used to display the confidence intervals.
mean.size	[numeric vector of length 2] size of the point and line for the mean trajectory.
size.text	[numeric, >0] size of the font used to displayed text when using ggplot2.
...	additional argument passed to residuals.lmm or autoplot.lmm.

Details

Call autoplot.lmm when codetype=="fit" and link(residuals.lmm) for the other types.

Value

A list with two elements

- data: data used to create the graphical display.
- plot: ggplot object.

potassiumRepeatedL *Data From The Potassium Intake Study (Long Format with intermediate measurements)*

Description

Data from the potassium intake study, a randomized placebo-controlled crossover study where the effect of potassium supplement (90 mmol/day) on the renin-angiostensin-aldosteron system (RAAS) was assessed. This dataset is in the long format (i.e. one line per measurement) and contains measurement over 6 timepoints for each time period.

- id Patient identifier
- sequence Treatment group to which the patient has been randomized.
- period Time period.
- treatment Treatment during the time period
- time Time within each period
- aldo ??

Usage

```
data(potassiumRepeatedL)
```

References

Dreier et al. Effect of increased potassium intake on the reninangiotensinaldosterone system and subcutaneous resistance arteries: a randomized crossover study, Nephrol Dial Transplant (2020) 110. doi: 10.1093/ndt/gfaa114

potassiumSingleL *Data From The Potassium Intake Study (Long Format)*

Description

Data from the potassium intake study, a randomized placebo-controlled crossover study where the effect of potassium supplement (90 mmol/day) on the renin-angiostensin-aldosteron system (RAAS) was assessed. This dataset is in the long format (i.e. one line per measurement).

- id Patient identifier
- sequence Treatment group to which the patient has been randomized.
- period Time period.
- treatment Treatment during the time period
- auc Area under the curve of ?? during the time period
- bsauc ??
- aldo ??

Usage

```
data(potassiumSingleL)
```

References

Dreier et al. Effect of increased potassium intake on the reninangiotensinaldosterone system and subcutaneous resistance arteries: a randomized crossover study, Nephrol Dial Transplant (2020) 110. doi: 10.1093/ndt/gfaa114

---

potassiumSingleW *Data From The Potassium Intake Study (Wide Format)*

Description

Data from the potassium intake study, a randomized placebo-controlled crossover study where the effect of potassium supplement (90 mmol/day) on the renin-angiostensin-aldosteron system (RAAS) was assessed. This dataset is in the wide format (i.e. one line per patient).

- id Patient identifier
- sequence Treatment group to which the patient has been randomized.
- treatment1 Treatment during the first time period.
- treatment2 Treatment during the second time period
- auc1 Area under the curve of ?? during the first time period
- auc2 Area under the curve of ?? during the second time period
- bsauc1 ??
- aldo1 ??
- aldo2 ??

Usage

```
data(potassiumSingleW)
```

References

Dreier et al. Effect of increased potassium intake on the reninangiotensinaldosterone system and subcutaneous resistance arteries: a randomized crossover study, Nephrol Dial Transplant (2020) 110. doi: 10.1093/ndt/gfaa114

predict.lmm *Predicted Mean Value With Uncertainty For Linear Mixed Model*

Description

Predicted mean value conditional on covariates or on covariates and other outcome values.

Usage

```
## S3 method for class 'lmm'
predict(
  object,
  newdata,
  se = "estimation",
  df = !is.null(object$df),
  type = "static",
  level = 0.95,
  keep.newdata = FALSE,
  se.fit,
  ...
)
```

Arguments

object	a lmm object.
newdata	[data.frame] the covariate values for each cluster.
se	[character] Type of uncertainty to be accounted for: estimation of the regression parameters ("estimation"), residual variance ("residual"), or both ("total"). Can also be NULL to not compute standard error, p-values, and confidence intervals.
df	[logical] Should a Student's t-distribution be used to model the distribution of the predicted mean. Otherwise a normal distribution is used.
type	[character] Should prediction be made conditional on the covariates only ("static") or also on outcome values at other timepoints ("dynamic"). Can also output the model term ("terms", similarly to stats::predict.lm.
level	[numeric,0-1] the confidence level of the confidence intervals.
keep.newdata	[logical] Should the argument newdata be output along side the predicted values?
se.fit	For internal use. When not missing mimic the output of predict.se. Overwrite argument se.
...	Not used. For compatibility with the generic method.

Details

Static prediction are made using the linear predictor $X\beta$ while dynamic prediction uses the conditional normal distribution of the missing outcome given the observed outcomes. So if outcome 1 is observed but not 2, prediction for outcome 2 is obtain by $X_2\beta + \sigma_{21}\sigma_{22}^{-1}(Y_1 - X_1\beta)$. In that case, the uncertainty is computed as the sum of the conditional variance $\sigma_{22} - \sigma_{21}\sigma_{22}^{-1}\sigma_{12}$ plus the uncertainty about the estimated conditional mean (obtained via delta method using numerical derivatives).

The model terms are computing by centering the design matrix around the mean value of the covariates used to fit the model. Then the centered design matrix is multiplied by the mean coefficients and columns assigned to the same variable (e.g. three level factor variable) are summed together.

Value

A data.frame with 5 columns:

- `estimate`: predicted mean.
- `se`: uncertainty about the predicted mean.
- `df`: degree of freedom
- `lower`: lower bound of the confidence interval of the predicted mean
- `upper`: upper bound of the confidence interval of the predicted mean

except when the argument `se.fit` is specified (see `predict.lm` for the output format).

Examples

```
## simulate data in the long format
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

## fit Linear Mixed Model
eUN.lmm <- lmm(Y ~ visit + X1 + X2 + X5,
               repetition = ~visit|id, structure = "UN", data = dL)

## prediction
newd <- data.frame(X1 = 1, X2 = 2, X5 = 3, visit = factor(1:3, levels = 1:3))
predict(eUN.lmm, newdata = newd)
predict(eUN.lmm, newdata = newd, keep.newdata = TRUE)
predict(eUN.lmm, newdata = newd, keep.newdata = TRUE, se = "total")

## dynamic prediction
newd.d1 <- cbind(newd, Y = c(NA,NA,NA))
predict(eUN.lmm, newdata = newd.d1, keep.newdata = TRUE, type = "dynamic")
newd.d2 <- cbind(newd, Y = c(6.61,NA,NA))
predict(eUN.lmm, newdata = newd.d2, keep.newdata = TRUE, type = "dynamic")
newd.d3 <- cbind(newd, Y = c(1,NA,NA))
predict(eUN.lmm, newdata = newd.d3, keep.newdata = TRUE, type = "dynamic")
newd.d4 <- cbind(newd, Y = c(1,1,NA))
predict(eUN.lmm, newdata = newd.d4, keep.newdata = TRUE, type = "dynamic")
```

`residuals` *Extract The Residuals From a Linear Mixed Model*

Description

Extract or compute the residuals of a linear mixed model.

Usage

```
## S3 method for class 'lmm'
residuals(
  object,
  type = "response",
  format = "long",
  data = NULL,
  p = NULL,
  keep.data = FALSE,
  var = NULL,
  plot = "none",
  engine.qqplot = "ggplot2",
  add.smooth = TRUE,
  digit.cor = 2,
  size.text = 16,
  scales = "free",
  ...
)
```

Arguments

`object`	a `lmm` object.
`type`	[character] type of residual to output: raw residuals (`"response"`), Pearson residuals (`"pearson"`), normalized residuals (`"normalized"`, scaled residual `"scaled"`), or partial residuals (`"partial"` or `"partial-ref"`). Can also be `"all"` to output all except partial residuals. See detail section.
`format`	[character] Should the residuals be output relative as a vector (`"long"`), or as a matrix with in row the clusters and in columns the outcomes (`"wide"`).
`data`	[data.frame] dataset relative to which the residuals should be computed. Only relevant if differs from the dataset used to fit the model.
`p`	[numeric vector] value of the model coefficients at which to evaluate the residuals. Only relevant if differs from the fitted values.
`keep.data`	[logical] Should the argument `data` be output along side the residuals? Only possible in the long format.
`var`	[character vector] name of the variable relative to which the partial residuals should be computed.

plot	[character] Should a qqplot ("qqplot"), or a heatmap of the correlation between residuals ("correlation", require wide format), or a plot of residuals along the fitted values ("scatterplot", require long format) be displayed?
engine.qqplot	[character] Should ggplot2 or qqtest be used to display quantile-quantile plots? Only used when argument plot is "qqplot".
add.smooth	[logical] should a local smoother be used to display the mean of the residual values across the fitted values. Only relevant for plot="scatterplot".
digit.cor	[integer, >0] Number of digit used to display the correlation coefficients? No correlation coefficient is displayed when set to 0. Only used when argument plot is "correlation".
size.text	[numeric, >0] Size of the font used to displayed text when using ggplot2.
scales	[character] Passed to ggplot2::facet_wrap.
...	Not used. For compatibility with the generic method.

Details

The argument type defines how the residuals are computed:

- "fitted": fitted value $X_{ij}\hat{\beta}$.
- "raw": observed outcome minus fitted value $\varepsilon = Y_{ij} - X_{ij}\hat{\beta}$.
- "pearson": each raw residual is divided by its modeled standard deviation $\varepsilon = \frac{Y_{ij}-X_{ij}\hat{\beta}}{\sqrt{\hat{\omega}_{ij}}}$.
- "studentized": same as "pearson" but excluding the contribution of the cluster in the modeled standard deviation $\varepsilon = \frac{Y_{ij}-X_{ij}\hat{\beta}}{\sqrt{\hat{\omega}_{ij}-\hat{q}_{ij}}}$.
- "normalized": raw residuals are multiplied, within clusters, by the inverse of the (lower) Cholesky factor of the modeled residual variance covariance matrix $\varepsilon = (Y_i - X_i\hat{\beta})\hat{C}^{-1}$.
- "normalized2": same as "normalized" but excluding the contribution of the cluster in the modeled residual variance covariance matrix $\varepsilon = (Y_i - X_i\hat{\beta})\hat{D}_i^{-1}$.
- "scaled": corresponds to the scaled residuals of PROC MIXED in SAS.
- "partial" or "partial-ref": the partial residual are computed as the raw residuals plus the effect of the covariates in argument var. "partial" uses $\hat{\beta}X + \hat{\varepsilon}$ where X is centered while "partial-ref" uses $\hat{\beta}X + \hat{\gamma}Z + \hat{\varepsilon}$ where the Z value are the same for all observations, i.e. uses a reference level.

where

- X the design matrix (default) or the design matrix restricted to the variable(s) in argument var (partial residuals).
- Y the outcome
- Z not defined (default) or the design matrix restricted to the variable(s) not in argument var (partial residuals).
- $\hat{\beta}$ the estimated mean coefficients relative to X
- $\hat{\gamma}$ the estimated mean coefficients relative to Z
- $\hat{\Omega}$ the modeled variance-covariance of the residuals and $\hat{\omega}$ its diagonal elements

- $\hat{C}$ the lower Cholesky factor of $\hat{\Omega}$, i.e. $\hat{C}\hat{C}^t = \hat{\Omega}$
- $\hat{Q}_i = X_i(X^t\hat{\Omega}X)^{-1}X_i^t$ a cluster specific correction factor, approximating the contribution of cluster i to $\hat{\Omega}$. Its diagonal elements are denoted $\hat{q}_i$.
- $\hat{D}_i$ the lower Cholesky factor of $\hat{\Omega} - \hat{Q}_i$

Value

When argument format is "long" and type.oobject is "lmm", a vector containing the value of the residual realtive to each observation. It is a matrix if the argument type contains several values. When argument format is "wide" and type.oobject is "lmm", a data.frame with the value of the residual relative to each cluster (in rows) at each timepoint (in columns).

Examples

```
## simulate data in the long format
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")

## fit Linear Model
e.lm <- lmm(Y ~ visit + X1 + X2 + X5, data = dL)
residuals(e.lm, type = "partial", var = "X1")
residuals(e.lm, type = "partial", var = "X1", keep.data = TRUE)

## fit Linear Mixed Model
eUN.lmm <- lmm(Y ~ visit + X1 + X2 + X5,
               repetition = ~visit|id, structure = "UN", data = dL)

## residuals
residuals(eUN.lmm, format = "long", type = c("normalized","pearson"))
residuals(eUN.lmm, format = "wide", plot = "correlation")
residuals(eUN.lmm, format = "wide", type = "normalized")
residuals(eUN.lmm, format = "wide", type = "scaled")

## residuals and predicted values
residuals(eUN.lmm, type = "all")
residuals(eUN.lmm, type = "all", keep.data = TRUE)
```

---

sampleRem *Sample Longuitudinal Data*

Description

Sample longuitudinal data with covariates

Usage

```
sampleRem(
  n,
  n.times,
  mu = 1:n.times,
  sigma = rep(1, n.times),
  lambda = rep(1, n.times),
  beta = c(2, 1, 0, 0, 0, 1, 1, 0, 0, 0),
  gamma = matrix(0, nrow = n.times, ncol = 10),
  format = "wide",
  latent = FALSE
)
```

Arguments

n	[integer,>0] sample size
n.times	[integer,>0] number of visits (i.e. measurements per individual).
mu	[numeric vector] expected measurement value at each visit (when all covariates are fixed to 0). Must have length n.times.
sigma	[numeric vector,>0] standard error of the measurements at each visit (when all covariates are fixed to 0). Must have length n.times.
lambda	[numeric vector] covariance between the measurement at each visit and the individual latent variable. Must have length n.times.
beta	[numeric vector of length 10] regression coefficient between the covariates and the latent variable.
gamma	[numeric matrix with n.times rows and 10 columns] regression coefficient specific to each timepoint (i.e. interaction with time).
format	[character] Return the data in the wide format ("wide") or long format ("long")
latent	[logical] Should the latent variable be output?

Details

The generative model is a latent variable model where each outcome (Y_j) load on the latent variable (η) with a coefficient lambda:

$$Y_j = \mu_j + \lambda_j * \eta + \sigma_j \epsilon_j$$

The latent variable is related to the covariates ($X_1, \ldots, X_(10)$):

$$\eta = \alpha + \beta_1 X_1 + ... + \beta_{10} X_{10} + \xi$$

ϵ_j and ξ are independent random variables with standard normal distribution.

Value

a data.frame

Examples

```
set.seed(10)
dW <- sampleRem(100, n.times = 3, format = "wide")
set.seed(10)
dL <- sampleRem(100, n.times = 3, format = "long")
```

---

score *Extract The Score From a Linear Mixed Model*

Description

Extract or compute the first derivative of the log-likelihood of a linear mixed model.

Usage

```
## S3 method for class 'lmm'
score(
  x,
  effects = "mean",
  data = NULL,
  p = NULL,
  indiv = FALSE,
  transform.sigma = NULL,
  transform.k = NULL,
  transform.rho = NULL,
  transform.names = TRUE,
  ...
)
```

Arguments

x	a lmm object.
effects	[character] Should the score relative to all coefficients be output ("all"), or only coefficients relative to the mean ("mean" or "fixed"), or only coefficients relative to the variance and correlation structure ("variance" or "correlation").
data	[data.frame] dataset relative to which the score should be computed. Only relevant if differs from the dataset used to fit the model.
p	[numeric vector] value of the model coefficients at which to evaluate the score. Only relevant if differs from the fitted values.
indiv	[logical] Should the contribution of each cluster to the score be output? Otherwise output the sum of all clusters of the derivatives.
transform.sigma	[character] Transformation used on the variance coefficient for the reference level. One of "none", "log", "square", "logsquare" - see details.

transform.k	[character] Transformation used on the variance coefficients relative to the other levels. One of "none", "log", "square", "logsquare", "sd", "logsd", "var", "logvar" - see details.
transform.rho	[character] Transformation used on the correlation coefficients. One of "none", "atanh", "cov" - see details.
transform.names	[logical] Should the name of the coefficients be updated to reflect the transformation that has been used?
...	Not used. For compatibility with the generic method.

Details

For details about the arguments **transform.sigma**, **transform.k**, **transform.rho**, see the documentation of the coef function.

Value

When argument indiv is FALSE, a vector with the value of the score relative to each coefficient. When argument indiv is TRUE, a matrix with the value of the score relative to each coefficient (in columns) and each cluster (in rows).

---

summarize *Compute summary statistics*

Description

Compute summary statistics (similar to the SAS macro procmean). This is essentially an interface to the stats::aggregate function.

Usage

```
summarize(
  formula,
  data,
  na.action = stats::na.pass,
  na.rm = FALSE,
 which = c("observed", "missing", "mean", "sd", "min", "median", "max", "correlation")
)
```

Arguments

formula	[formula] on the left hand side the outcome(s) and on the right hand side the grouping variables. E.g. Y1+Y2 ~ Gender + Gene will compute for each gender and gene the summary statistics for Y1 and for Y2. Passed to the stats::aggregate function.
data	[data.frame] dataset (in the wide format) containing the observations.

na.action	[function] a function which indicates what should happen when the data contain 'NA' values. Passed to the stats::aggregate function.
na.rm	[logical] Should the summary statistics be computed by omitting the missing values.
which	[character vector] name of the summary statistics to kept in the output. Can be any of, or a combination of: "observed" (number of observations with a measurement), "missing" (number of observations with a missing value), "mean", "sd", "min", "median", "max".

Value

a data frame containing summary statistics (in columns) for each outcome and value of the grouping variables (rows). It has an attribute "correlation" when it was possible to compute the correlation matrix for each outcome with respect to the grouping variable.

Examples

```
## simulate data in the wide format
set.seed(10)
d <- sampleRem(1e2, n.times = 3)

## add a missing value
d2 <- d
d2[1,"Y2"] <- NA

## run summarize
summarize(Y1+Y2 ~ 1, data = d)
summarize(Y1+Y2 ~ X1, data = d)

summarize(Y1+Y2 ~ X1, data = d2)
summarize(Y1+Y2 ~ X1, data = d2, na.rm = TRUE)

## long format
dL <- reshape2::melt(d, id.vars = c("id","X1"), measure.var = c("Y1","Y2","Y3"))
dL2 <- dL[-(4:5),]

summarize(value ~ variable + X1, data = dL)

## compute correlations
e.S <- summarize(value ~ variable + X1 | id, data = dL2, na.rm = TRUE)
e.S
attr(e.S, "correlation")
```

---

summary *Summary Output for a Linear Mixed Model*

Description

Summary output for a linear mixed model fitted with lmm. This is a modified version of the nlme::summary.gls function.

Usage

```
## S3 method for class 'lmm'
summary(
  object,
  digit = 3,
  level = 0.95,
  robust = FALSE,
  print = TRUE,
  columns = NULL,
  hide.fit = FALSE,
  hide.data = FALSE,
  hide.cor = FALSE,
  hide.var = TRUE,
  hide.sd = FALSE,
  hide.mean = FALSE,
  ...
)
```

Arguments

object	[lmm] output of the lmm function.
digit	[integer,>0] number of digit used to display numeric values.
level	[numeric,0-1] confidence level for the confidence intervals.
robust	[logical] Should robust standard error (aka sandwich estimator) be output instead of the model-based standard errors.
print	[logical] should the output be printed in the console.
columns	[character vector] Columns to be output for the fixed effects. Can be any of "estimate", "se", "statistic", "df", "null", "lower", "upper", "p.value".
hide.fit	[logical] should information about the model fit not be printed.
hide.data	[logical] should information about the dataset not be printed.
hide.cor	[logical] should information about the correlation structure not be printed.
hide.var	[logical] should information about the variance not be printed.
hide.sd	[logical] should information about the standard deviation not be printed.
hide.mean	[logical] should information about the mean structure not be printed.
...	not used. For compatibility with the generic function.

Value

A list containing elements displayed in the summary:

- correlation: the correlation structure.
- variance: the variance structure.
- sd: the variance structure expressed in term of standard deviations.
- mean: the mean structure.

swabsL *Data From The SWABS Study (Long Format)*

Description

Data from the swabs study, where the pneumococcus was studied in 18 families with different space available for the household. This dataset is in the long format (i.e. one line per measurement).

- crowding Space available in the household.
- family Family serial number
- name Type of family member.
- swabs number of times the swab measurement was positive.

Usage

```
data(swabsL)
```

References

TODO

swabsW *Data From The SWABS Study (Wide Format)*

Description

Data from the swabs study, where the pneumococcus was studied in 18 families with different space available for the household. This dataset is in the wide format (i.e. one line per patient).

- crowding Space available in the household.
- family Family serial number
- mother number of times the swab measurement was positive for the mother.
- father number of times the swab measurement was positive for the father.
- child1 number of times the swab measurement was positive for the first child.
- child2 number of times the swab measurement was positive for the second child.
- child3 number of times the swab measurement was positive for the third child.

Usage

```
data(swabsW)
```

References

Grundy SM, Lan SP, Lachin J. The effects of chenodiol on biliary lipids and their association with gallstone dissolution in the National Cooperative Gallstone Study (SWABS). J Clin Invest. 1984 Apr;73(4):1156-66. doi: 10.1172/JCI111301.

terms.lmm *Model Terms For Linear Mixed Models*

Description

Model terms for linear mixed models. Used by `multcomp::glht`.

Usage

```
## S3 method for class 'lmm'
terms(x, ...)
```

Arguments

`x`	a `lmm` object
`...`	not used, for compatibility with the generic method.

Value

An object of class `terms` giving a symbolic representation of the mean structure.

UN *Unstructured Structure*

Description

Variance-covariance structure where the residuals have time-specific variance and correlation. Can be stratified on a categorical variable.

Usage

```
UN(formula, var.cluster, var.time, add.time)
```

Arguments

`formula`	formula indicating on which variable to stratify the residual variance and correlation (left hand side) and variables influencing the residual variance (right hand side).
`var.cluster`	[character] cluster variable.
`var.time`	[character] time variable.
`add.time`	Should the default formula (i.e. when `NULL`) contain a time effect.

Details

A typical formula would be `~1`, indicating a time-specific variance parameter and a correlation parameter specific to each pair of times.

Value

An object of class UN that can be passed to the argument structure of the lmm function.

Examples

```
UN(NULL, var.cluster = "id", var.time = "time", add.time = TRUE)
UN(list(~gender,~time), var.cluster = "id", var.time = "time")
UN(gender~time, var.cluster = "id", var.time = "time")
UN(list(gender~time,gender~time), var.cluster = "id", var.time = "time")
```

---

vasscoresL *Data From The VAS Study (Long Format)*

Description

Data from the VAS Study, a randomized controlled clinial trial assessing the healing effect of topical zink sulfate on epidermal wound. The study includes 30 heatlhy volunteers with induced wounds on each buttock which where subsequently treated with a different treatment for each wound. Then the VAS-score (pain sensation on a 0-100mm visual analogue scale) was assessed after each treatment application and summarized by area under the curve. This dataset is in the long format (i.e. one line per measurement).

- id Patient identifier.
- group Treatment group to which the patient has been randomized.
- treat.num
- vas VAS-score relative to the wound.
- treatment Treatment used on the wound. A: active treatment (zink shower gel), B: placebo treatment (shower gel without zink), C: control treatment (demineralized water).

Usage

```
data(vasscoresL)
```

References

TODO

vasscoresW *Data From The VAS Study (Wide Format)*

Description

Data from the VAS Study, a randomized controlled clinial trial assessing the healing effect of topical zink sulfate on epidermal wound. The study includes 30 heatlhy volunteers with induced wounds on each buttock which where subsequently treated with a different treatment for each wound. Then the VAS-score (pain sensation on a 0-100mm visual analogue scale) was assessed after each treatment application and summarized by area under the curve. This dataset is in the wide format (i.e. one line per patient).

- id Patient identifier.
- group Treatment group to which the patient has been randomized.
- vasA VAS-score when using a zink shower gel.
- vasB VAS-score when using a placebo treatment (shower gel without zink).
- vasC VAS-score when using a control treatment with demineralized water.

Usage

```
data(vasscoresW)
```

References

TODO

vcov *Extract The Variance-Covariance Matrix From a Linear Mixed Model*

Description

Extract the variance-covariance matrix of the model coefficients of a linear mixed model.

Usage

```
## S3 method for class 'lmm'
vcov(
  object,
  effects = "mean",
  robust = FALSE,
  df = FALSE,
  strata = NULL,
  data = NULL,
  p = NULL,
```

```
  type.information = NULL,
  transform.sigma = NULL,
  transform.k = NULL,
  transform.rho = NULL,
  transform.names = TRUE,
  ...
)
```

Arguments

object	a lmm object.
effects	[character] Should the variance-covariance matrix for all coefficients be output ("all"), or only for coefficients relative to the mean ("mean" or "fixed"), or only for coefficients relative to the variance structure ("variance"), or only for coefficients relative to the correlation structure ("correlation").
robust	[logical] Should robust standard error (aka sandwich estimator) be output instead of the model-based standard errors. Not feasible for variance or correlation coefficients estimated by REML.
df	[logical] Should degree of freedom, computed using Satterthwaite approximation, for the model parameters be output.
strata	[character vector] When not NULL, only output the variance-covariance matrix for the estimated parameters relative to specific levels of the variable used to stratify the mean and covariance structure.
data	[data.frame] dataset relative to which the information should be computed. Only relevant if differs from the dataset used to fit the model.
p	[numeric vector] value of the model coefficients at which to evaluate the information. Only relevant if differs from the fitted values.
type.information	[character] Should the expected information be used (i.e. minus the expected second derivative) or the observed inforamtion (i.e. minus the second derivative).
transform.sigma	[character] Transformation used on the variance coefficient for the reference level. One of "none", "log", "square", "logsquare" - see details.
transform.k	[character] Transformation used on the variance coefficients relative to the other levels. One of "none", "log", "square", "logsquare", "sd", "logsd", "var", "logvar" - see details.
transform.rho	[character] Transformation used on the correlation coefficients. One of "none", "atanh", "cov" - see details.
transform.names	[logical] Should the name of the coefficients be updated to reflect the transformation that has been used?
...	Not used. For compatibility with the generic method.

Details

For details about the arguments **transform.sigma**, **transform.k**, **transform.rho**, see the documentation of the coef function.

Value

A matrix with an attribute `"df"` when argument df is set to `TRUE`.

`vitaminL` *Data From The Vitamin Study (Long Format)*

Description

Data from the vitamin Study, a randomized study where the growth of guinea pigs was monitored before and after intake of vitamin E/placebo. The weight of each guinea pig was recorded at the end of week 1, 3, 4, 5, 6, and 7. Vitamin E/placebo is given at the beginning of week 5. This dataset is in the long format (i.e. one line per measurement).

- group Treatment group: vitamin or placebo.
- animal Identifier
- weigth1 weight (in g) of the pig at the end of week 1 (before treatment).
- weigth3 weight (in g) of the pig at the end of week 3 (before treatment).
- weigth4 weight (in g) of the pig at the end of week 4 (before treatment).
- weigth5 weight (in g) of the pig at the end of week 5 (after treatment).
- weigth6 weight (in g) of the pig at the end of week 6 (after treatment).
- weigth7 weight (in g) of the pig at the end of week 7 (after treatment).

Usage

```
data(vitaminL)
```

References

Crowder and Hand (1990, p. 27) Analysis of Repeated Measures.

vitaminW *Data From The Vitamin Study (Wide Format)*

Description

Data from the vitamin Study, a randomized study where the growth of guinea pigs was monitored before and after intake of vitamin E/placebo. The weight of each guinea pig was recorded at the end of week 1, 3, 4, 5, 6, and 7. Vitamin E/placebo is given at the beginning of week 5. This dataset is in the wide format (i.e. one line per patient).

- group Treatment group: vitamin or placebo.
- animal Identifier
- weigth1 weight (in g) of the pig at the end of week 1 (before treatment).
- weigth3 weight (in g) of the pig at the end of week 3 (before treatment).
- weigth4 weight (in g) of the pig at the end of week 4 (before treatment).
- weigth5 weight (in g) of the pig at the end of week 5 (after treatment).
- weigth6 weight (in g) of the pig at the end of week 6 (after treatment).
- weigth7 weight (in g) of the pig at the end of week 7 (after treatment).

Usage

```
data(vitaminW)
```

References

TODO

Index