Package: lcc (via r-universe)

August 23, 2024

Type Package

Title Longitudinal Concordance Correlation

Version 2.0.0

Author Thiago de Paula Oliveira [aut, cre] (<`https://orcid.org/0000-0002-4555-2584`>), Rafael de Andrade Moral [aut] (<`https://orcid.org/0000-0002-0875-3563`>), Silvio Sandoval Zocchi [ctb] (<`https://orcid.org/0000-0003-2535-908X`>), Clarice Garcia Borges Demetrio [ctb] (<`https://orcid.org/0000-0002-3609-178X`>), John Hinde [aut] (<`https://orcid.org/0000-0001-6507-5204`>)

Maintainer Thiago de Paula Oliveira

`<thiago.paula.oliveira@alumni.usp.br>`

Description Estimates the longitudinal concordance correlation to access the longitudinal agreement profile. The estimation approach implemented is variance components approach based on polynomial mixed effects regression model, as proposed by Oliveira, Hinde and Zocchi (2018) <doi:10.1007/s13253-018-0321-1>. In addition, non-parametric confidence intervals were implemented using percentile method or normal-approximation based on Fisher Z-transformation.

Date 2023-11-25

Depends R (>= 3.2.3), nlme (>= 3.1-124), ggplot2 (>= 2.2.1)

Imports hnp, parallel, doSNOW, doRNG, foreach

Suggests roxygen2 (>= 3.0.0), covr, testthat, MASS

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation yes

LazyData true

RoxygenNote 7.2.3

Repository https://prof-thiagooliveira.r-universe.dev

RemoteUrl https://github.com/prof-thiagooliveira/lcc

RemoteRef HEAD

RemoteSha a7745b41d287cef5e88241d8c6516c1504728eac

Contents

`AIC.lcc` *Akaike and Bayesian Information Criteria for an* `lcc` *Object*

Description

Calculates the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC) for a fitted longitudinal concordance correlation model represented by an `lcc` object.

Calculates the Bayesian Information Criterion (BIC) for a fitted longitudinal concordance correlation model represented by an `lcc` object. BIC is used for model selection, with lower values indicating a better model.

Usage

```
## S3 method for class 'lcc'
AIC(object, ..., k = 2)

## S3 method for class 'lcc'
BIC(object, ...)
```

Arguments

`object`	An object of class `lcc`, representing a fitted longitudinal concordance correlation function.
`...`	Optional arguments passed to the underlying `BIC` function from the `stats` package.
`k`	Numeric value used as a penalty coefficient for the number of parameters in the fitted model; the default `k = 2` corresponds to the classical AIC.

Details

The function computes AIC or BIC values as a measure of the relative quality of statistical models for a given set of data. Lower AIC or BIC values indicate a better model fit with fewer parameters. For more information, refer to the methods for `AIC` objects.

The function computes BIC as a measure of the trade-off between model fit and complexity. It is particularly useful for comparing models with different numbers of parameters. For more information, refer to the documentation for `BIC`.

See Also

`lcc`, `summary.lcc`, `coef.lcc`, `vcov.lcc`

`lcc`, `summary.lcc`, `coef.lcc`, `vcov.lcc`, `AIC.lcc`

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
AIC(fm1)

## End(Not run)

## Not run:
attach(simulated_hue)
fm6 <- lcc(data = simulated_hue, subject = "Fruit",
           resp = "Hue", method = "Method", time = "Time",
           qf = 2, qr = 1, components = TRUE,
           time_lcc = list(n=50, from=min(Time), to=max(Time)))
AIC(fm6)
BIC(fm6)

## End(Not run)
```

`anova.lcc` *Compare Likelihoods of Fitted Models from an `lcc` Object*

Description

Compares the fit of different longitudinal concordance correlation models (lcc objects). When comparing multiple models, the function returns a data frame with degrees of freedom, log-likelihood, AIC, and BIC for each model. For a single model, it returns F-values and P-values for fixed terms in the model.

Usage

```
## S3 method for class 'lcc'
anova(object, ..., test = TRUE, type = c("sequential", "marginal"),
  adjustSigma = TRUE, verbose = FALSE)
```

Arguments

`object`	An object inheriting from class `lcc` or `lme`.
`...`	Other optional fitted model objects inheriting from classes "lcc" or "lme".
`test`	Logical; if `TRUE`, performs likelihood ratio tests to compare models. Defaults to `TRUE`.
`type`	Character string specifying the type of sum of squares for F-tests. Options are "sequential" or "marginal". Defaults to "sequential".
`adjustSigma`	Logical; if `TRUE`, adjusts the residual standard error for maximum likelihood estimation. Defaults to `TRUE`.
`verbose`	Logical; if `TRUE`, prints additional model details. Defaults to `FALSE`.

Details

This function is an adaptation from `anova.lme`. It assesses whether the addition of terms significantly improves model fit.

See Also

`lcc`, `summary.lcc`

Examples

```
## Not run:
fm1.aov <- lcc(data = hue, subject = "Fruit", resp = "H_mean", method = "Method",
               time = "Time", qf = 2, qr = 1)
fm2.aov <- update(fm1.aov, qr = 2)
anova(fm1.aov, fm2.aov)

## End(Not run)
```

```
## Not run:
fm3.aov <- update(fm2.aov, REML = FALSE)
fm4.aov <- update(fm2.aov, REML = FALSE, qf = 3)
anova(fm3.aov, fm4.aov)

## End(Not run)

## Not run:
fm5.aov <- update(fm2.aov, var.class = varExp, weights.form = "time")
anova(fm1.aov, fm2.aov, fm5.aov)

## End(Not run)
```

`coef.lcc` *Extract Model Coefficients*

Description

The fixed effects estimated and corresponding random effects estimates are obtained at subject levels less or equal to i. The resulting estimates are returned as a data frame, with rows corresponding to subject levels and columns to coefficients.

Usage

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

Arguments

`object`	an object inheriting from class `lcc`, representing a fitted longitudinal concordance correlation function.
`...`	optional arguments passed to the `coef.lme` function.

Details

See methods for `nlme` objects to get more details.

Author(s)

Thiago de Paula Oliveira, `<thiago.paula.oliveira@alumni.usp.br>`

See Also

`lcc`, `summary.lcc`, `lccPlot`, `vcov.lcc`

Examples

```
## Not run:
fm1<-lcc(data = hue, subject = "Fruit", resp = "H_mean",
         method = "Method", time = "Time", qf = 2, qr = 2)
coef(fm1)

## End(Not run)
```

`fitted.lcc` *Extract Fitted Values from an* `lcc` *Object*

Description

Extracts and prints the fitted values from an object of class `lcc`, as returned by modeling functions. The function allows selection of different types of fitted values based on longitudinal data analysis.

Usage

```
## S3 method for class 'lcc'
fitted(object, type = "lcc", digits = NULL, ...)
```

Arguments

`object`	An object of class `lcc`, representing a fitted longitudinal concordance correlation model.
`type`	The type of fitted values to extract: "lcc" for longitudinal concordance correlation, "lpc" for longitudinal Pearson correlation, or "la" for longitudinal accuracy. Defaults to "lcc".
`digits`	Minimum number of significant digits to be printed. Default is `NULL`, which uses the default precision.
`...`	Additional arguments (currently not used).

Value

The function prints the fitted values and returns them as a data frame.

See Also

`lcc`, `summary.lcc`, `lccPlot`

Examples

```
data(hue)
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2,
           components = TRUE)
fitted(fm1)
fitted(fm1, type = "lpc")
fitted(fm1, type = "la")
```

`formatColumn` *Format Columns for Print*

Description

This internal helper function is used to format the columns of a data frame for printing, specifically for use within the ‘print.anova.lcc‘ function. It applies special formatting rules based on the column name, such as rounding and special handling of small p-values.

Usage

```
formatColumn(column, colName)
```

Arguments

`column`	A vector representing a column from a data frame.
`colName`	A string indicating the name of the column, which determines the formatting rules to be applied.

Details

The function specifically handles the following columns: - "p-value": Rounds the values to four decimal places, and represents values less than 0.0001 as "<.0001". - "AIC", "BIC", "logLik", "L.Ratio": Applies ‘zapsmall‘ for formatting. Other columns are returned without changes.

Value

A vector with the same length as ‘column‘, where each element has been formatted according to the column-specific rules.

Examples

```
data <- data.frame(
  pvalue = c(0.00005, 0.0234, 0.5),
  AIC = c(123.4567, 234.5678, 345.6789)
)
data$pvalue <- formatColumn(data$pvalue, "p-value")
data$AIC <- formatColumn(data$AIC, "AIC")
```

`getVarCov.lcc` *Extract Variance Components from a Fitted lcc Model*

Description

Retrieves the variance-covariance matrix of the specified component from a fitted `lcc` model object. The function can extract different types of variance-covariance matrices based on the specified component type.

Usage

```
## S3 method for class 'lcc'
getVarCov(obj, type = "random.effects", ...)
```

Arguments

`obj`	An object of class `lcc`, representing a fitted longitudinal concordance correlation model.
`type`	Specifies the type of variance-covariance matrix to extract. Options are `"random.effects"` for random-effects variance-covariance, `"conditional"` for conditional variance-covariance of the responses, and `"marginal"` for marginal variance-covariance of the responses. Default is `"random.effects"`.
`...`	Optional arguments passed to the underlying `getVarCov` function from the `nlme` package.

Details

This function is useful for detailed inspection of the variance components in different aspects of the model. For more information on the types of variance-covariance matrices and their interpretations, refer to the documentation of the `nlme` package.

See Also

`lcc`, `summary.lcc`, `coef.lcc`, `vcov.lcc`

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
getVarCov(fm1)

## End(Not run)
```

`hue` *Hue color data*

Description

An observational study conducted at the Vegetable Production Department at "Luiz de Queiroz" College of Agriculture/University of São Paulo in 2010/2011 to evaluate the peel color of 20 papaya fruits cv. Sunrise Solo over time. The color hue was measured on the equatorial region of each fruit using four points observed by the colorimeter and 1,000 points observed by the scanner. Thus, the circular mean hue was calculated for each fruit by each device at time t. The aim of the agreement study was to assess how well the colorimeter agreed with the scanner over time.

Usage

```
data(hue)
```

Format

A data frame with 554 observations on the mean hue variable. The format is:

`H_mean`	numeric; mean hue of papaya's peel
`Method`	a factor with levels `Colorimeter`, `Scanner`
`Time`	integer; time in days
`Fruit`	a factor with 20 levels; from `1` to `20` where each level is represented by one fruit.

Source

Oliveira, T.P.; Hinde, J.; Zocchi S.S. Longitudinal Concordance Correlation Function Based on Variance Components: An Application in Fruit Color Analysis. Journal of Agricultural, Biological, and Environmental Statistics, v. 23, n. 2, 233–254, 2018.

Oliveira, T.P.; Zocchi S.S.; Jacomino, A.P. Measuring color hue in 'Sunrise Solo' papaya using a flatbed scanner. *Rev. Bras. Frutic.*, v. 39, n. 2, e-911, 2017.

References

Oliveira, T.P.; Hinde, J.; Zocchi S.S. Longitudinal Concordance Correlation Function Based on Variance Components: An Application in Fruit Color Analysis. Journal of Agricultural, Biological, and Environmental Statistics, v. 23, n. 2, 233–254, 2018.

See Also

`lcc`.

Examples

```
data(hue)
summary(hue)
str(hue)
## Second degree polynomial model with random intercept, slope and
## quadratic term including an exponential variance function using
## time as covariate.
model<-lcc(data = hue, subject = "Fruit", resp = "H_mean",
          method = "Method", time = "Time", qf = 2, qr = 2,
          components = TRUE, time_lcc = list(from = min(hue$Time),
          to = max(hue$Time), n=40), var.class=varExp,
          weights.form="time")
summary(model, type="model")
summary(model, type="lcc")
## for discussion on the analysis of complete data set,
## see Oliveira et al. (2018)
```

`lcc` *Longitudinal Concordance Correlation (LCC) Estimated by Fixed Effects and Variance Components using a Polynomial Mixed-Effects Regression Model*

Description

The `lcc` function gives fitted values and non-parametric bootstrap confidence intervals for LCC, longitudinal Pearson correlation (LPC), and longitudinal accuracy (LA) statistics. These statistics can be estimated using different structures for the variance-covariance matrix for random effects and variance functions to model heteroscedasticity among the within-group errors using or not the time as a covariate.

Usage

```
lcc(data, resp, subject, method, time, interaction, qf,
    qr, covar, gs, pdmat, var.class, weights.form, time_lcc, ci,
    percentileMet, alpha, nboot, show.warnings, components,
    REML, lme.control,  numCore)
```

Arguments

`data`	an object of class `data.frame`.
`resp`	character string. Name of the response variable in the data set.
`subject`	character string. Name of the subject variable in the data set.
`method`	character string. Name of the method variable in the data set. The first level of method is used as the gold-standard method.
`time`	character string. Name of the time variable in the data set.

`interaction`	an option to estimate the interaction effect between `method` and `time`. If `TRUE`, the default, interaction effect is estimated. If `FALSE` only the main effects of time and method are estimated.
`qf`	an integer specifying the degree time polynomial trends, normally 1, 2 or 3. (Degree 0 is not allowed). Default is `qf=1`
`qr`	an integer specifying random effects terms to account for subject-to-subject variation. Note that `qr=0` specifies a random intercept (form ~ `1|subject`); `qr=1` specifies random intercept and slope (form ~ `time|subject`). If `qr=qf=q`, with $q \geq 1$, random effects at subject level are added to all terms of the time polynomial regression (form ~ `poly(time, q, raw = TRUE)|subject`). Default is `qr=0`.
`covar`	character vector. Name of the covariates to be included in the model as fixed effects. Default to `NULL`, never include.
`gs`	character string. Name of method level which represents the gold-standard. Default is the first level of method.
`pdmat`	standard classes of positive-definite matrix structures defined in the `pdClasses` function. The different positive-definite matrices structures available in the `lcc` function are `pdSymm`, the default, `pdLogChol`, `pdDiag`, `pdIdent`, `pdCompSymm`, and `pdNatural`.
`var.class`	standard classes of variance functions to model the variance structure of within-group errors using covariates, see `varClasses`. Default to `NULL`, correspond to homoscedastic within-group errors. Available standard classes: `varIdent`**:** allows different variances according to the levels of the stratification variable. `varExp`**:** exponential function of the variance covariate; see `varExp`.
`weights.form`	character string. An one-sided formula specifying a variance covariate and, optionally, a grouping factor for the variance parameters in the `var.class`. If `var.class=varIdent`, the option “method”, form `~1|method` or “time.ident”, form `~1|time`, must be used in the `weights.form` argument. If `var.class=varExp`, the option “time”, form `~time`, or “both”, form `~time|method`, must be used in the `weights.form` argument.
`time_lcc`	regular sequence for time variable merged with specific or experimental time values used for LCC, LPC, and LA predictions. Default is `NULL`. The list may contain the following components: `time`**:** a vector of specific or experimental time values of given length. The experimental time values are used as default. `from`**:** the starting (minimum) value of time variable. `to`**:** the end (maximum) value of time variable. `n`**:** an integer specifying the desired length of the sequence. Generally, `n` between 30 and 50 is adequate.
`ci`	an optional non-parametric boostrap confidence interval calculated for the LCC, LPC and LA statistics. If `TRUE` confidence intervals are calculated and printed in the output. Default is `FALSE`.

`percentileMet`	an optional method for calculating the non-parametric bootstrap intervals. If `FALSE`, the default, is the normal approximation method. If `TRUE`, the percentile method is used instead.
`alpha`	significance level. Default is 0.05.
`nboot`	an integer specifying the number of bootstrap samples. Default is 5,000.
`show.warnings`	an optional argument that shows the number of convergence errors in the bootstrap samples. If `TRUE` shows in which bootstrap sample the error occurred. If `FALSE`, the default, shows the total number of convergence errors.
`components`	an option to print LPC and LA statistics. If `TRUE` the estimates and confidence intervals for LPC and LA are printed in the output. If `FALSE`, the default, provides estimates and confidence interval only for the LCC statistic.
`REML`	if `TRUE`, the default, the model is fit by maximizing the restricted log-likelihood. If `FALSE` the log-likelihood is maximized.
`lme.control`	a list of control values for the estimation algorithm to replace the default values of the function `lmeControl` available in the `nlme` package. Defaults to an empty list. The returned list is used as the control argument for the `lme` function.
`numCore`	number of cores used in parallel during bootstrapping computation. Default is 1.

Value

an object of class lcc. The output is a list with the following components:

`model`	summary of the polynomial mixed-effects regression model.
`Summary.lcc`	fitted values for the LCC or LCC, LPC and LA (if `components=TRUE`); concordance correlation coefficient (CCC) between methods for each level of `time` as sampled values, and the CCC between mixed-effects model predicted values and observed values from data as goodness of fit (gof)
`data`	the input dataset.

Author(s)

Thiago de Paula Oliveira, `<thiago.paula.oliveira@alumni.usp.br>`, Rafael de Andrade Moral, John Hinde

References

Lin, L. A Concordance Correlation Coefficient to Evaluate Reproducibility. *Biometrics*, 45, n. 1, 255-268, 1989.

Oliveira, T.P.; Hinde, J.; Zocchi S.S. Longitudinal Concordance Correlation Function Based on Variance Components: An Application in Fruit Color Analysis. *Journal of Agricultural, Biological, and Environmental Statistics*, v. 23, n. 2, 233–254, 2018.

Oliveira, T.P.; Moral, R.A.; Zocchi, S.S.; Demetrio, C.G.B.; Hinde, J. lcc: an R packageto estimate the concordance correlation, Pearson correlation, and accuracy over time. *PeerJ*, 8:c9850, 2020. DOI:10.7717/peerj.9850

See Also

summary.lcc, fitted.lcc, print.lcc, lccPlot, plot.lcc, coef.lcc, ranef.lcc, vcov.lcc, getVarCov.lcc, residuals.lcc, AIC.lcc

Examples

```
data(hue)
## Second degree polynomial model with random intercept, slope and
## quadratic term
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
print(fm1)
summary(fm1)
summary(fm1, type="model")
lccPlot(fm1) +
 ylim(0,1) +
 geom_hline(yintercept = 1, linetype = "dashed") +
 scale_x_continuous(breaks = seq(1,max(hue$Time),2))

## Estimating longitudinal Pearson correlation and longitudinal
#accuracy
fm2 <- update(fm1, components = TRUE)
summary(fm2)
lccPlot(fm2) +
 ylim(0,1) +
 geom_hline(yintercept = 1, linetype = "dashed") +
 scale_x_continuous(breaks = seq(1,max(hue$Time),2)) +
 theme_bw()

## Not run:
## A grid of points as the Time variable for prediction
fm3 <- update(fm2, time_lcc = list(from = min(hue$Time),
           to = max(hue$Time), n=40))
summary(fm3)
lccPlot(fm3) +
 ylim(0,1) +
 geom_hline(yintercept = 1, linetype = "dashed") +
 scale_x_continuous(breaks = seq(1,max(hue$Time),2)) +
 theme_bw()

## End(Not run)

## Including an exponential variance function using time as a
#covariate.
fm4 <- update(fm2,time_lcc = list(from = min(hue$Time),
              to = max(hue$Time), n=30), var.class=varExp,
              weights.form="time")
summary(fm4,  type="model")
fitted(fm4)
fitted(fm4, type = "lpc")
fitted(fm4, type = "la")
lccPlot(fm4) +
```

```
 geom_hline(yintercept = 1, linetype = "dashed")
lccPlot(fm4, type = "lpc") +
 geom_hline(yintercept = 1, linetype = "dashed")
lccPlot(fm4, type = "la") +
 geom_hline(yintercept = 1, linetype = "dashed")

## Not run:
## Non-parametric confidence interval with 500 bootstrap samples
fm5 <- update(fm1, ci = TRUE, nboot = 500)
summary(fm5)
lccPlot(fm5) +
 geom_hline(yintercept = 1, linetype = "dashed")

## End(Not run)

## Considering three methods of color evaluation
## Not run:
data(simulated_hue)
attach(simulated_hue)
fm6 <- lcc(data = simulated_hue, subject = "Fruit",
           resp = "Hue", method = "Method", time = "Time",
           qf = 2, qr = 1, components = TRUE,
           time_lcc = list(n=50, from=min(Time), to=max(Time)))
summary(fm6)
lccPlot(fm6, scales = "free")
lccPlot(fm6, type="lpc", scales = "free")
lccPlot(fm6, type="la", scales = "free")
detach(simulated_hue)

## End(Not run)

## Including an additional covariate in the linear predictor
## (randomized block design)
## Not run:
data(simulated_hue_block)
attach(simulated_hue_block)
fm7 <- lcc(data = simulated_hue_block, subject = "Fruit",
           resp = "Hue", method = "Method",time = "Time",
           qf = 2, qr = 1, components = TRUE, covar = c("Block"),
           time_lcc = list(n=50, from=min(Time), to=max(Time)))
summary(fm7)
lccPlot(fm7, scales="free")
detach(simulated_hue_block)

## End(Not run)

## Testing interaction effect between time and method
fm8 <- update(fm1, interaction = FALSE)
anova(fm1, fm8)

## Not run:
## Using parallel computing with 3 cores, and a set.seed(123)
to verify model reproducibility.
```

```
set.seed(123)
fm9 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
              method = "Method", time = "Time", qf = 2, qr = 2,
              ci=TRUE, nboot = 30, numCore = 3)

# Repeating same model with same set seed.
set.seed(123)
fm10 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
              method = "Method", time = "Time", qf = 2, qr = 2,
              ci=TRUE, nboot = 30, numCore = 3)

## Verifying if both fitted values and confidence intervals
are identical
identical(fm9$Summary.lcc$fitted,fm10$Summary.lcc$fitted)

## End(Not run)
```

`lccPlot` *Plot Fitted Curves from an* `lcc` *Object*

Description

This function generates a plot of predictions versus the time covariate for an `lcc` object. Predicted values are connected by lines, while actual observations are denoted by circles. If `components=TRUE` was used in the `lcc` object, individual plots for each statistic (LCC, LPC, and LA) are produced on separate pages.

Usage

```
lccPlot(obj, type = "lcc", control = list(), ...)
```

Arguments

`obj`	An object inheriting from class "lcc", representing a fitted lcc model.
`type`	Character string specifying the type of plot to generate. • `"lcc"`: Produces the LCC plot. • `"lpc"`: Produces the LPC plot. Available only if `components` = `TRUE`. • `"la"`: Produces the LA plot. Available only if `components` = `TRUE`.
`control`	A list of graphical control values or character strings returned by the `plotControl` function. Defaults to an empty list. The list can contain components like shape, `colour`, `size`, `xlab`, `ylab`, `scale_y_continuous`, and `all.plot`.
`...`	Additional arguments passed to the `facet_wrap` function.

Value

An object of class `ggplot` or `viewport`, depending on the `all.plot` setting in `control`.

Author(s)

Thiago de Paula Oliveira, <thiago.paula.oliveira@alumni.usp.br>

See Also

`lcc`, `plotControl`

Examples

```
data(hue)
# Second degree polynomial model with random intercept, slope and quadratic term
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
          method = "Method", time = "Time", qf = 2, qr = 2, components = TRUE)
lccPlot(fm1, type = "lcc")
lccPlot(fm1, type = "lpc")
lccPlot(fm1, type = "la")

# Using ggplot2 themes
lccPlot(fm1, type = "lpc") + theme_bw() + labs(x = "Time (Days)", y = "LPC Value")

# Generating and saving plots
## Not run:
  ggsave("lccPlot.pdf", lccPlot(fm1, type = "lcc"))

## End(Not run)
```

`logLik.lcc` *Extract Log-Likelihood of an* `lcc` *Object*

Description

If `REML=TRUE`, the default, returns the restricted log-likelihood value of the linear mixed-effects model; else the log-likelihood value

Usage

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

Arguments

`object`	an object inheriting from class `lcc`, representing a fitted longitudinal concordance correlation function.
`...`	further arguments passed to `logLik`.
`REML`	an optional logical value. If `TRUE` the restricted log-likelihood is returned, else, if `FALSE`, the log-likelihood is returned.

Details

See methods for `nlme` objects to get more details.

Author(s)

Thiago de Paula Oliveira, <`thiago.paula.oliveira@alumni.usp.br`>

See Also

`lcc`, `summary.lcc`

Examples

```
## Not run:
fm1<-lcc(data = hue, subject = "Fruit", resp = "H_mean",
         method = "Method", time = "Time", qf = 2, qr = 2)
logLik(fm1)

## End(Not run)
```

`plot.lcc` *Diagnostic Plots for an* `lcc` *Object*

Description

Generates a series of diagnostic plots for evaluating the fit of a linear mixed-effects model represented by an `lcc` object. This function provides six types of plots, including residual plots, fitted value comparisons, and normal Q-Q plots. Users can select specific plots or display all by default.

Usage

```
## S3 method for class 'lcc'
plot(x, which = c(1L:6L),
     caption = list("Residuals vs Fitted",
                    "Residuals vs Time",
                    "Residuals by Subject",
                    "Observed values vs Fitted values",
                    "Normal Q-Q Plot (Conditional residuals)",
                    "Normal Q-Q Plot (Random effects)"),
     sub.caption = NULL, main = NULL,
     panel = if(add.smooth) panel.smooth else points,
     add.smooth = TRUE, ask = TRUE,
     id.n = 3, labels.id = names(residuals(x)),
     label.pos = c(4, 2), cex.id = 0.75, cex.caption = 1,
     cex.oma.man = 1.25, ...)
```

Arguments

`x`	An object of class `lcc`, representing a fitted longitudinal concordance correlation function.
`which`	A numeric vector specifying which plots to display. The valid range is c(1L:6L), corresponding to the plot types.
`caption`	Captions for the plots, provided as a vector or list of valid graphics annotations. Default captions are provided for each plot.
`sub.caption`	A common sub-title for all plots; defaults to `NULL`.
`main`	The main title for the plots, displayed above the captions.
`panel`	Panel function to be used for adding points to the plots. Defaults to `panel.smooth` if `add.smooth` is `TRUE`, otherwise `points`.
`add.smooth`	Logical; indicates whether a smoother should be added to most plots. Defaults to `TRUE`.
`ask`	Logical; if `TRUE`, prompts the user before displaying each plot in a multi-plot layout. Defaults to `TRUE`.
`id.n`	Number of extreme points to label in the first three plots.
`labels.id`	Labels for the extreme points, defaulting to observation numbers if `NULL`.
`label.pos`	Positioning of labels in the left and right halves of the graph, applicable for plots 1-3.
`cex.id`	Magnification factor for point labels.
`cex.caption`	Size of the plot captions.
`cex.oma.man`	Size of the overall margin annotation (applies only if `sub.caption` is above the figures in multi-plot layouts).
`...`	Additional graphical parameters passed to `par`.

Details

The Q-Q plots use normalized residuals. Standardized residuals are pre-multiplied by the inverse square-root factor of the estimated error correlation matrix, while random effects are adjusted using the estimated variances from matrix G. Simulation envelopes in Q-Q plots are generated using the `hnp` package.

The function is partly adapted from `plot.lm`.

Author(s)

Thiago de Paula Oliveira, `<thiago.paula.oliveira@alumni.usp.br>`

See Also

`lccPlot`, `lcc`, `mtext`, `text`, `plotmath`

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
plot(fm1)

## End(Not run)
```

`print.anova.lcc` *Print the Anova of an* `lcc` *Object*

Description

Method print for the `anova.lcc`.

Usage

```
## S3 method for class 'anova.lcc'
print(x, verbose, ...)
```

Arguments

`x`	an object inheriting from class `anova.lcc`, representing a fitted longitudinal concordance correlation function.
`verbose`	an optional logical value used to control the amount of printed output. If `TRUE`, the calling sequences for each fitted model object are printed with the rest of the output, being omitted if `verbose` = `FALSE`. Defaults to `FALSE`.
`...`	further arguments passed to `print`.

Details

Modified from `anova.lme`. For more details see methods for `nlme`.

Author(s)

Thiago de Paula Oliveira, `<thiago.paula.oliveira@alumni.usp.br>`

See Also

`summary.lcc`, `lccPlot`, `lcc`

Examples

```
## Not run:
## Second degree polynomial model with random intercept, slope and
## quadratic term
fm1<-lcc(data = hue, subject = "Fruit", resp = "H_mean",
         method = "Method", time = "Time", qf = 2, qr = 2)
print(anova(fm1))

## End(Not run)
```

`print.lcc` *Print Method for* `lcc` *Objects*

Description

Prints detailed information about the fitted longitudinal concordance correlation model contained in an `lcc` object.

Usage

```
## S3 method for class 'lcc'
print(x, digits = NULL, ...)
```

Arguments

`x`	An object of class `lcc`, representing a fitted longitudinal concordance correlation model.
`digits`	Minimum number of significant digits to be printed in values. Default is `NULL`, which uses the default precision.
`...`	Further arguments passed to `print`.

Value

The function is used for its side effect of printing and returns the input `lcc` object invisibly.

See Also

`lcc`, `summary.lcc`

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
          method = "Method", time = "Time", qf = 2, qr = 2)
print(fm1)

## End(Not run)
```

`print.summary.lcc` *Print Summary of an* `lcc` *Object*

Description

Provides a detailed summary of a fitted longitudinal concordance correlation model, including AIC, BIC, log-likelihood, and other relevant statistics. The function supports detailed output for different types of model fits.

Usage

```
## S3 method for class 'summary.lcc'
print(x, verbose = FALSE, digits = NULL, ...)
```

Arguments

`x`	An object of class `summary.lcc`, representing a summarized longitudinal concordance correlation function.
`verbose`	Logical value to control the amount of printed output for model details. Defaults to `FALSE`.
`digits`	Specifies the minimum number of significant digits to be printed in values. Default is `NULL`.
`...`	Further arguments passed to `print`.

See Also

`summary.lcc`, `lccPlot`, `lcc`

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
print(summary(fm1, type = "model"))

## End(Not run)
```

`ranef.lcc` *Extract Random Effects from an* `lcc` *Model*

Description

Extracts the estimated random effects from a fitted longitudinal concordance correlation model represented by an `lcc` object. The function returns a data frame with rows corresponding to different groups at a specified level and columns representing the random effects.

Usage

```
## S3 method for class 'lcc'
ranef(object, ...)
```

Arguments

`object`	An object inheriting from class `lcc`, representing a fitted longitudinal concordance correlation function.
`...`	Optional arguments passed to the `ranef.lme` function from the `nlme` package.

Details

This function is useful for examining the random effects associated with groups or subjects in the model. For a detailed explanation of these effects, see the documentation for `nlme` objects.

See Also

`lcc`, `coef.lcc`,

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
ranef(fm1)

## End(Not run)
```

`residuals.lcc` *Extract Residuals from a Fitted lcc Model*

Description

Extracts residuals from the fitted longitudinal concordance correlation model represented by an `lcc` object. Different types of residuals can be obtained based on the specified type.

Usage

```
## S3 method for class 'lcc'
residuals(object, type = "response", ...)
```

Arguments

`object`	An object of class `lcc`, representing a fitted longitudinal concordance correlation function.
`type`	A character string specifying the type of residuals to extract. Options are `"response"` for residuals obtained by subtracting the fitted values from the response (default), `"pearson"` for "response" residuals divided by the estimated within-group standard error, and `"normalized"` for normalized residuals. Partial matching is used, so only the first character of the type is necessary.
`...`	Optional arguments passed to the `residuals.lme` function from the `nlme` package.

Details

The function provides a convenient way to examine the differences between observed and predicted values in the model. Understanding these residuals can be crucial for model diagnostics and validation. For more information, refer to the methods for `nlme` objects.

See Also

`lcc`, `summary.lcc`, `coef.lcc`, `vcov.lcc`

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
residuals(fm1)

## End(Not run)
```

`simulated_hue` *Hue color simulated data*

Description

Simulated hue data set based on papaya's maturation over time considering three methods of measurement.

Usage

```
data(simulated_hue)
```

Format

A simulated data frame with 6,000 observations on the mean hue variable. The format is:

`Hue`	numeric; mean hue of papaya's peel
`Method`	a factor with levels labelled from `Method 1` to `Method 3`
`Time`	integer; time in days from 0 to 19
`Fruit`	a factor with 100 levels labelled from `1` to `100` where each level is represented by one fruit.

Details

A total of 100 fruits were observed over 20 days by three methods to evaluate the mean hue of fruit's peel. The aim of the agreement study was to assess how well the methods 2, and 3 agreed with method 1 over time.

See Also

`lcc`.

Examples

```
data(simulated_hue)
summary(simulated_hue)
str(simulated_hue)
```

`simulated_hue_block` *Hue color simulated data in a randomized block design*

Description

Simulated hue data set based on papaya's maturation over time considering four methods of measurement in a randomized block design.

Usage

```
data(simulated_hue_block)
```

Format

A simulated data frame with 24,000 observations on the mean hue variable. The format is:

`Hue`	numeric; mean hue of papaya's peel
`Block`	factor with levels labelled from `1` to `3`
`Method`	a factor with levels labelled from `Method 1`, to `Method 4`
`Time`	integer; time in days from 0 to 19
`Fruit`	a factor with 300 levels labelled from `1` to `300` where each level is represented by one fruit.

Details

A total of 100 fruits by block were observed over 20 days by four methods to evaluate the mean hue of fruit's peel. We considered three blocks in this simulation. The aim of the agreement study was to assess how well the methods 2, 3, and 4 agreed with method 1 over time.

See Also

`lcc`.

Examples

```
data(simulated_hue_block)
summary(simulated_hue_block)
str(simulated_hue_block)
```

`summary.lcc` *Summarize an `lcc` Object*

Description

Additional information about the fit of longitudinal concordance correlation, longitudinal Pearson correlation, and longitudinal accuracy represented by an object of class `lcc`. The returned object has a `print` method.

Usage

```
## S3 method for class 'lcc'
summary(object, type, adjustSigma, verbose, ...)
```

Arguments

`object`	an object inheriting from class `lcc`, representing a fitted longitudinal concordance correlation function.
`type`	an optional character string specifying the type of output to be returned. If `type="model"`, prints the summary of the polynomial mixed-effects regression model. If `type="lcc"`, prints the summary of the fitted and sampled values for LCC, LPC, and LA as well as the concordance correlation coefficient between fitted values from the model and observed values as goodness of fit (gof) measurement. Defaults to `type="model"`.
`adjustSigma`	an optional logical value used when `type = model`. If TRUE and the estimation method used to obtain object was maximum likelihood, the residual standard error is multiplied by `sqrt(nobs/(nobs - npar))`. See `summary.lme` for more information. Default is TRUE.
`verbose`	an optional logical value used to control the amount of output in the `print.summary.lme` method when `type = model` is used. Defaults to FALSE.
`...`	not used.

Value

an object inheriting from class `summary.lcc` including:

`fitted`	the fitted values extracted from the `lcc` object.
`gof`	the goodness of fit (gof) measurement is calculated using the concordance correlation coefficient between fitted and observed values. Value of 1 denote perfect concordance.
`AIC`	the Akaike Information Criterion corresponding to object.
`BIC`	the Bayesian Information Criterion corresponding to object.
`logLik`	If `REML=FALSE`, returns the log-likelihood value of the linear mixed-effects model; otherwise, the restricted log-likelihood is returned

Author(s)

Thiago de Paula Oliveira, `<thiago.paula.oliveira@alumni.usp.br>`

See Also

`AIC`, `BIC`, `print.summary.lcc`, `lcc`

Examples

```
## Second degree polynomial model with random intercept, slope and
## quadratic term
fm1<-lcc(data = hue, subject = "Fruit", resp = "H_mean",
         method = "Method", time = "Time", qf = 2, qr = 2)
summary(fm1, type="model")
summary(fm1, type="lcc")
```

`vcov.lcc` *Extract Variance-Covariance Matrix of the Fixed Effects for an lcc Object*

Description

Extracts the variance-covariance matrix of the fixed effects from a fitted `lcc` model object. This function provides insights into the variability and covariance structure of the fixed effects in the model.

Usage

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

Arguments

`object`	An object of class `lcc`, representing a fitted longitudinal concordance correlation model.
`...`	Optional arguments passed to the `vcov.lme` function from the `nlme` package.

Details

The function specifically retrieves the variance-covariance matrix associated with the fixed effects of the `lcc` object, which is useful for understanding the relationship between these effects. For more details on variance-covariance matrices, refer to the methods for `nlme` objects.

See Also

`summary.lcc`, `lccPlot`, `lcc`, `coef.lcc`

Examples

```
## Not run:
fm1 <- lcc(data = hue, subject = "Fruit", resp = "H_mean",
           method = "Method", time = "Time", qf = 2, qr = 2)
vcov(fm1)

## End(Not run)
```

Index