Package 'ClusROC'

July 21, 2025

Type Package

Title ROC Analysis in Three-Class Classification Problems for
Clustered Data

Date 2022-11-17

Version 1.0.2

Author Duc-Khanh To [aut, cre] (<https://orcid.org/
0000-0002-4641-0764>), with contributions from Gianfranco Adimari and Monica Chiogna

Maintainer Duc-Khanh To <toduc@stat.unipd.it>

Description Statistical methods for ROC surface analysis in three-class classification prob-
lems for clustered data and in presence of covariates. In particular, the package allows to ob-
tain covariate-specific point and interval estimation for:
(i) true class fractions (TCFs) at fixed pairs of thresholds;
(ii) the ROC surface;
(iii) the volume under ROC surface (VUS);
(iv) the optimal pairs of thresholds.
Methods considered in points (i), (ii) and (iv) are proposed and dis-
cussed in To et al. (2022) <doi:10.1177/09622802221089029>. Referring to point (iv), three dif-
ferent selection criteria are implemented: Generalized Youden Index (GYI), Closest to Perfec-
tion (CtP) and Maximum Volume (MV). Methods considered in point (iii) are proposed and dis-
cussed in Xiong et al. (2018) <doi:10.1177/0962280217742539>. Visualiza-
tion tools are also provided. We refer readers to the articles cited above for all details.

License GPL-3

Depends R (>= 3.5.0), stats, utils, graphics, nlme, Rcpp (>= 0.12.3)

Imports rgl, ellipse, numDeriv, ggplot2, ggpubr, foreach, iterators,
parallel, doParallel

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

LazyLoad yes

ByteCompile yes

RoxygenNote 7.2.2

NeedsCompilation yes

URL https://github.com/toduckhanh/ClusROC

BugReports https://github.com/toduckhanh/ClusROC/issues

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository CRAN

Date/Publication 2022-11-17 15:00:02 UTC

Contents

ClusROC-package   *ROC Analysis in Three-Class Classification Problems for Clustered Data*

Description

This package implements the techniques for ROC surface analysis, in cases of clustered data and in presence of covariates. In particular, the package allows to obtain covariate-specific point and interval estimation for: (i) true class fractions (TCFs) at fixed pairs of thresholds; (ii) the ROC surface; (iii) the volume under ROC surface (VUS); (iv) the optimal pairs of thresholds. Methods considered in points (i), (ii) and (iv) are proposed and discussed in To et al. (2022). Referring to point (iv), three different selection criteria are implemented: Generalized Youden Index (GYI), Closest to Perfection (CtP) and Maximum Volume (MV). Methods considered in point (iii) are proposed and discussed in Xiong et al. (2018). Visualization tools are also provided. We refer readers to the articles cited above for all details.

Details

Package:	ClusROC
Type:	Package
Version:	1.0-2
Date:	2022-10-10
License:	GPL 2	GPL 3
Lazy load:	yes

Major functions are `clus_lme`, `clus_roc_surface`, `clus_opt_thres3`, `clus_vus` and `clus_tcfs`.

Author(s)

Duc-Khanh To, with contributions from Gianfranco Adimari and Monica Chiogna

Maintainer: Duc-Khanh To <toduc@stat.unipd.it>

References

Bantis, L. E., Nakas, C. T., Reiser, B., Myall, D., and Dalrymple-Alford, J. C. (2017). Construction of joint confidence regions for the optimal true class fractions of Receiver Operating Characteristic (ROC) surfaces and manifolds. *Statistical methods in medical research*, **26**, 3, 1429-1442.

Gurka, M. J., Edwards, L. J. , Muller, K. E., and Kupper, L. L. (2006). Extending the Box-Cox transformation to the linear mixed model. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **169**, 2, 273-288.

Gurka, M. J. and Edwards, L. J. (2011). Estimating variance components and random effects using the box-cox transformation in the linear mixed model. *Communications in Statistics - Theory and Methods*, **40**, 3, 515-531.

Kauermann, G. and Carroll, R. J. (2001). A note on the efficiency of sandwich covariance matrix estimation. *Journal of the American Statistical Association*, **96**, 456, 1387-1396.

Liang, K. Y. and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 1, 13-22.

Mancl, L. A. and DeRouen, T. A. (2001). A covariance estimator for GEE with improved small-sample properties. *Biometrics*, **57**, 1, 126-134.

To, D-K., Adimari, G., Chiogna, M. and Risso, D. (2022). Receiver operating characteristic estimation and threshold selection criteria in three-class classification problems for clustered data. *Statistical Methods in Medical Research*, **7**, 31, 1325-1341.

Xiong, C., Luo, J., Chen L., Gao, F., Liu, J., Wang, G., Bateman, R. and Morris, J. C. (2018). Estimating diagnostic accuracy for clustered ordinal diagnostic groups in the three-class case – Application to the early diagnosis of Alzheimer disease. *Statistical Methods in Medical Research*, **27**, 3, 701-714.

---

ci_clus_vus   *Confidence Intervals for Covariate-specific VUS*

Description

Computes confidence intervals for covariate-specific VUS.

Usage

```
ci_clus_vus(x, ci_level = 0.95)
```

Arguments

x	an object of class "VUS", a result of [clus_vus](#) call.
ci_level	a confidence level to be used for constructing the confidence interval; default is 0.95.

Details

A confidence interval for covariate-specific VUS is given based on normal approximation. If the lower bound (or the upper bound) of the confidence interval is smaller than 0 (or greater than 1), it will be set as 0 (or 1). Also, logit and probit transformations are available if one wants guarantees that confidence limits are inside (0, 1).

Value

ci_clus_vus returns an object of class inheriting from "ci_VUS" class. An object of class "ci_VUS" is a list, containing at least the following components:

ci_vus_norm	the normal approximation-based confidence interval for covariate-specific VUS.
ci_vus_log	the confidence interval for covariate-specific VUS, after using logit-transformation.
ci_vus_prob	the confidence interval for covariate-specific VUS, after using probit-transformation.
ci_level	fixed confidence level.
newdata	value(s) of covariate(s).
n_p	total numbers of the regressors in the model.

See Also

[clus_vus](#)

---

clus_lme	*Linear Mixed-Effects Models for a continuous diagnostic test or a biomarker (or a classifier).*

Description

clus_lme fits the cluster-effect model for a continuous diagnostic test in a three-class setting as described in Xiong et al. (2018) and To et al. (2022).

Usage

```
clus_lme(
  fixed_formula,
  name_class,
  name_clust,
  data = sys.frame(sys.parent()),
  subset,
  na_action = na.fail,
  levl_class = NULL,
  ap_var = TRUE,
  boxcox = FALSE,
  interval_lambda = c(-2, 2),
  trace = TRUE,
  ...
)
```

Arguments

fixed_formula	a two-sided linear formula object, describing the fixed-effects part of the model for three classes, with the response on the left of ~ operator and the terms, separated by + operators, on the right. For example, Y ~ X1 + X2, Y ~ X1 + X2 + X1:X2 or log(Y) ~ X1 + X2 + I(X1^2).
name_class	name of variable indicating three classes (or three groups) in the data.
name_clust	name of variable indicating clusters in the data.
data	a data frame containing the variables in the model.
subset	an optional expression indicating the subset of the rows of data that should be used in the fit. This can be a logical vector, or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
na_action	a function that indicates what should happen when the data contain NAs. The default action (na.fail) causes clus_lme to print an error message and terminate if there are any incomplete observations.
levl_class	a vector (of strings) containing the ordered name chosen for the disease classes. The ordering is intended to be "increasing" with respect to the disease severity. If levl_class = NULL (default), the elements of the vector will be automatically

determined from data, by considering the order of the means of the test values for each disease class (diagnostic group).

ap_var   a logical value. Default = TRUE. If set to TRUE, the estimated covariance matrix for all estimated parameters in the model will be obtained (by using the sandwich formula).

boxcox   a logical value. Default = FALSE. If set to TRUE, a Box-Cox transformation will be applied to the model.

interval_lambda
                     a vector containing the end-points of the interval for searching the Box-Cox parameter, lambda. Default = (-2, 2).

trace   a logical value. Default = TRUE. If set to TRUE, the information about the check for the monotonic ordering of test values will be provided.

...   additional arguments for [lme](), such as control, contrasts.

Details

This function fits a linear mixed-effect model for a continuous diagnostic test in a three-class setting in order to account for the cluster and covariates effects on the test result. See Xiong et al. (2018) and To et al. (2022) for more details.

- Estimation is done by using [lme]() with the restricted maximum log-likelihood (REML) method.
- Box-Cox transformation for the model can be used when the distributions of test results are skewed (Gurka et al. 2006). The estimation procedure is described in To et al. (2022). The Box-Cox parameter λ is estimated by a grid search on the interval (-2, 2), as discussed in Gurka and Edwards (2011).
- The estimated variance-covariance matrix for the estimated parameters are obtained by sandwich formula (see, Liang and Zeger, 1986; Kauermann and Carroll, 2001; Mancl and DeRouen, 2001) as discussed in To et al. (2022).

Value

clus_lme returns an object of class "clus_lme" class, i.e., a list containing at least the following components:

call   the matched call.

est_para   a vector containing the estimated parameters.

se_para   a vector containing the standard errors.

vcov_sand   the estimated covariance matrix for all estimated parameters.

residual   a list of residuals.

fitted   a list of fitted values.

randf   a vector of estimated random effects for each cluster level.

n_coef   total number of coefficients included in the model.

n_p   total numbers of regressors in the model.

icc   an estimate of intra-class correlation - ICC

terms	the [terms](#) object used.
boxcox	logical value indicating whether the Box-Cox transformation was applied or not.
data	data frame is used to fitting model.

Generic functions such as `print` and `plot` are also used to show results of the fit.

References

Gurka, M. J., Edwards, L. J. , Muller, K. E., and Kupper, L. L. (2006) "Extending the Box-Cox transformation to the linear mixed model". *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **169**, 2, 273-288.

Gurka, M. J. and Edwards, L. J. (2011) "Estimating variance components and random effects using the box-cox transformation in the linear mixed model". *Communications in Statistics - Theory and Methods*, **40**, 3, 515-531.

Kauermann, G. and Carroll, R. J. (2001) "A note on the efficiency of sandwich covariance matrix estimation". *Journal of the American Statistical Association*, **96**, 456, 1387-1396.

Liang, K. Y. and Zeger, S. L. (1986) "Longitudinal data analysis using generalized linear models". *Biometrika*, **73**, 1, 13-22.

Mancl, L. A. and DeRouen, T. A. (2001) "A covariance estimator for GEE with improved small-sample properties". *Biometrics*, **57**, 1, 126-134.

To, D-K., Adimari, G., Chiogna, M. and Risso, D. (2022) "Receiver operating characteristic estimation and threshold selection criteria in three-class classification problems for clustered data". *Statistical Methods in Medical Research*, **7**, 31, 1325-1341.

Xiong, C., Luo, J., Chen L., Gao, F., Liu, J., Wang, G., Bateman, R. and Morris, J. C. (2018) "Estimating diagnostic accuracy for clustered ordinal diagnostic groups in the three-class case – Application to the early diagnosis of Alzheimer disease". *Statistical Methods in Medical Research*, **27**, 3, 701-714.

Examples

```
## Example 1:
data(data_3class)
head(data_3class)
## A model with two covariate
out1 <- clus_lme(fixed_formula = Y ~ X1 + X2, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)
print(out1)
plot(out1)


## Example 2: Box-Cox transformation
data(data_3class_bcx)
out2 <- clus_lme(fixed_formula = Y ~ X, name_class = "D",
                 name_clust = "id_Clus", data = data_3class_bcx,
                 boxcox = TRUE)
print(out2)
plot(out2)
```

---

clus_opt_thres3	*Estimation of the covariate-specific optimal pair of thresholds for clustered data.*

Description

`clus_opt_thres3` estimates covariate-specific optimal pair of thresholds of a continuous diagnostic test in a clustered design, with three classes of diseases.

Usage

```
clus_opt_thres3(
  method = c("GYI", "CtP", "MV"),
  out_clus_lme,
  newdata,
  ap_var = TRUE,
  control = list()
)
```

Arguments

method	the method to be used. See 'Details'.
out_clus_lme	an object of class "clus_lme", i.e., a result of [clus_lme](#) call.
newdata	a data frame (containing specific value(s) of covariate(s)) in which to look for variables with which to estimate covariate-specific optimal pair of thresholds. In absence of covariate, no values have to be specified.
ap_var	logical value. If set to `TRUE`, the variance-covariance matrix of (estimated) covariate-specific optimal thresholds is estimated.
control	a list of control parameters. See 'Details'.

Details

This function implements estimation methods discussed in To et al. (2022) for covariate-specific optimal pair of thresholds in a clustered design with three ordinal groups. The estimators are based on the results from [clus_lme](#) function, which fits the linear mixed-effect model by using REML approach.

Before performing estimation, a check for the monotone ordering assumption is performed. This means that, for the fixed values of covariates, three predicted mean values for test results in three diagnostic groups are compared. If the assumption is not meet, the covariate-specific optimal pair of thresholds at the values of covariates are not estimated.

The estimation procedure uses three criteria. Method `"GYI"` is Generalized Youden Index, which maximizes the sum of three covariate-specific True Class Fractions - TCFs. Method `"CtP"` is based on Closest to Pefection approach. By using this method, the optimal pair of thresholds is obtained by minimizing the distance, in the unit cube, between a generic point on the covariate-specific ROC surface and the top corner (1, 1, 1). Method `"MV"` is based on Maximum Volume approach,

which searches for thresholds that maximize the volume of a box under the covariate-specific ROC surface. The user can select more than one method. This function allows to estimate covariate-specific optimal pair of thresholds at multiple points for covariates.

The asymptotic variance-covariance matrix of the (estimated) covariate-specific optimal thresholds is estimated by using the Delta method under the normal assumption. If the Box-Cox transformation is applied to the linear mixed-effect model, a nonparametric bootstrap procedure for clustered data will be used to obtain the estimated asymptotic covariance matrix (see To et al. 2022, for more details).

The `control` argument is a list that can supply any of the following components:

method_optim Optimization method to be used. There are three options: `"L-BFGS-B"`, `"BFGS"` and `"Nelder-Mead"`. Default is `"L-BFGS-B"`.

start Starting values in the optimization procedure. If it is `NULL`, a starting point will be automatically obtained.

maxit The maximum number of iterations. Default is 200.

lower, upper Possible bounds on the threshold range, for the optimization based on "L-BFGS-B" method. Defaults are `-Inf` and `Inf`.

n_boot Number of bootstrap replicates for estimating the covariance matrix (when Box-Cox transformation is applied). Default is 250.

parallel A logical value. If set to `TRUE`, a parallel computing is employed in the bootstrap resampling process.

ncpus Number of processes to be used in parallel computing. Default is 2.

Value

`clus_opt_thres3` returns an object of `"clus_opt_thres3"` class, which is a list containing at least the following components:

call	the matched call.
method	the methods used to obtain the estimated optimal pair of threholds.
thres3	a vector or matrix containing the estimated optimal thresholds.
thres3_se	a vector or matrix containing the estimated standard errors.
vcov_thres3	a matrix or list of matrices containing the estimated variance-covariance matrices.
tcfs	a vector or matrix containing the estimated TCFs at the optimal thresholds.
mess_order	a diagnostic message from checking the monontone ordering.
newdata	value(s) of covariate(s).
n_p	total number of regressors in the model.

Generic functions such as `print` and `plot` are also used to show the results.

References

To, D-K., Adimari, G., Chiogna, M. and Risso, D. (2022) "Receiver operating characteristic estimation and threshold selection criteria in three-class classification problems for clustered data". *Statistical Methods in Medical Research*, **7**, 31, 1325-1341.

Examples

```
data(data_3class)
## One covariate
out1 <- clus_lme(fixed_formula = Y ~ X1, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### Estimate covariate-specific optimal thresholds at multiple values of one covariate,
### with 3 methods
out_thres_1 <- clus_opt_thres3(method = c("GYI", "MV", "CtP"),
                               out_clus_lme = out1,
                               newdata = data.frame(X1 = 1), ap_var = TRUE)
print(out_thres_1)
plot(out_thres_1)

## Two covariates
out2 <- clus_lme(fixed_formula = Y ~ X1 + X2, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### Estimate covariate-specific optimal thresholds at one point, with 3 methods
out_thres_2 <- clus_opt_thres3(method = c("GYI", "MV", "CtP"),
                               out_clus_lme = out2,
                               newdata = data.frame(X1 = 1, X2 = 0),
                               ap_var = TRUE)
print(out_thres_2)
plot(out_thres_2)

### Estimate covariate-specific optimal thresholds at three points, with 3 methods
out_thres_3 <- clus_opt_thres3(method = c("GYI", "MV", "CtP"),
                               out_clus_lme = out2,
                               newdata = data.frame(X1 = c(-0.5, 0.5, 0.5),
                                                    X2 = c(0, 0, 1)),
                               ap_var = TRUE)
print(out_thres_3)
plot(out_thres_3, colors = c("forestgreen", "blue"))
```

| clus_roc_surface | *Plot an estimated covariate-specific ROC surface for clustered data.* |

Description

clus_roc_surface estimates and makes a 3D plot of a covariate-specific ROC surface for a continuous diagnostic test, in a clustered design, with three ordinal groups.

Usage

```
clus_roc_surface(
  out_clus_lme,
  newdata,
```

```
    step_tcf = 0.01,
    main = NULL,
    file_name = NULL,
    ellips = FALSE,
    thresholds = NULL,
    ci_level = ifelse(ellips, 0.95, NULL)
)
```

Arguments

out_clus_lme	an object of class "clus_lme", a result of `clus_lme` call.
newdata	a data frame with 1 row (containing specific value(s) of covariate(s)) in which to look for variables with which to estimate covariate-specific ROC. In absence of covariate, no values have to be specified.
step_tcf	number: increment to be used in the grid for $p1 = tcf1$ and $p3 = tcf3$.
main	the main title for plot.
file_name	File name to create on disk.
ellips	a logical value. If set to TRUE, the function adds an ellipsoidal confidence region for TCFs (True Class Fractions), at a specified pair of values for the thresholds, to the plot of estimated covariate-specific ROC surface.
thresholds	a specified pair of thresholds, used to construct the ellipsoidal confidence region for TCFs.
ci_level	a confidence level to be used for constructing the ellipsoidal confidence region; default is 0.95.

Details

This function implements a method in To et al. (2022) for estimating covariate-specific ROC surface of a continuous diagnostic test in a clustered design, with three ordinal groups. The estimator is based on the results from `clus_lme` with REML approach.

Before performing estimation, a check for the monotone ordering assumption is performed. This means that, for the fixed values of covariates, three predicted mean values for test results in three diagnostic groups are compared. If the assumption is not meet, the covariate-specific ROC surface at the values of covariates is not estimated.

The ellipsoidal confidence region for TCFs at a given pair of thresholds, if required, is constructed by using normal approximation and is plotted in the ROC surface space. The confidence level (default) is 0.95. Note that, if the Box-Cox transformation is applied for the linear mixed-effect model, the pair of thresholds must be input in the original scale. If the constructed confidence region for TCFs is outside the unit cube, a probit transformation will be automatically applied to obtain an appropriate confidence region, which is inside the unit cube (see Bantis et. al., 2017).

Value

`clus_roc_surface` returns a 3D `rgl` plot of the estimated covariate-specific ROC surface.

References

Bantis, L. E., Nakas, C. T., Reiser, B., Myall, D., and Dalrymple-Alford, J. C. (2017). "Construction of joint confidence regions for the optimal true class fractions of Receiver Operating Characteristic (ROC) surfaces and manifolds". *Statistical methods in medical research*, **26**, 3, 1429-1442.

To, D-K., Adimari, G., Chiogna, M. and Risso, D. (2022) "Receiver operating characteristic estimation and threshold selection criteria in three-class classification problems for clustered data". *Statistical Methods in Medical Research*, **7**, 31, 1325-1341.

Examples

```
data(data_3class)
## One covariate
out1 <- clus_lme(fixed_formula = Y ~ X1, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### plot only covariate-specific ROC surface
clus_roc_surface(out_clus_lme = out1, newdata = data.frame(X1 = 1))

### plot covariate-specific ROC surface and a 95% ellipsoidal confidence region for TCFs
clus_roc_surface(out_clus_lme = out1, newdata = data.frame(X1 = 1),
                 ellips = TRUE, thresholds = c(0.9, 3.95))

## Two covariates
out2 <- clus_lme(fixed_formula = Y ~ X1 + X2, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### plot only covariate-specific ROC surface
clus_roc_surface(out_clus_lme = out2, newdata = data.frame(X1 = 1, X2 = 1))

### plot covariate-specific ROC surface and a 95% ellipsoidal confidence region for TCFs
clus_roc_surface(out_clus_lme = out2, newdata = data.frame(X1 = 1, X2 = 1),
                 ellips = TRUE, thresholds = c(0.9, 3.95))
```

---

clus_tcfs   *Estimation of the covariate-specific TCFs for clustered data.*

Description

clus_tcfs estimates covariate-specific True Class Fractions (TCFs), at a specified pair of thresholds, of a continuous diagnostic test in a clustered design with three ordinal groups. This function allows to estimate covariate-specific TCFs at multiple points for covariates.

Usage

```
clus_tcfs(out_clus_lme, newdata, thresholds, ap_var = FALSE)
```

Arguments

out_clus_lme	an object of class "clus_lme", a result of clus_lme call.
newdata	a data frame (containing specific value(s) of covariate(s)) in which to look for variables with which to estimate covariate-specific TCFs. In absence of covariate, no values have to be specified.
thresholds	a specified pair of thresholds.
ap_var	logical value. If set to TRUE, the variance-covariance matrix of estimated covariate-specific TCFs is estimated.

Details

This function implements a method in To et al. (2022) for estimating covariate-specific TCFs at a specified pair of thresholds of a continuous diagnostic test in a clustered design with three ordinal groups. The estimator is based on results from clus_lme, which uses the REML approach. The asymptotic variance-covariance matrix of the estimated covariate-specific TCFs is estimated through the Delta method. Note that, if the Box-Cox transformation is applied for the linear mixed-effect model, the pair of thresholds must be input in the original scale.

Before performing estimation, a check for the monotone ordering assumption is performed. This means that, for the fixed values of covariates, three predicted mean values for test results in three diagnostic groups are compared. If the assumption is not meet, the covariate-specific TCFs at the values of covariates are not estimated.

Value

TCFs returns an object of class "TCFs", which is a list containing at least the following components:

call	the matched call.
tcfs_est	a vector or matrix containing the estimated TCFs.
tcf_vcov	a matrix or list of matrices containing the estimated variance-covariance matrices.
thresholds	specified pair of thresholds.
mess_order	a diagnostic message from checking the monontone ordering.
newdata	value(s) of covariate(s).
n_p	total number of regressors in the model.

Generic functions such as print is also used to show the results.

References

To, D-K., Adimari, G., Chiogna, M. and Risso, D. (2022) "Receiver operating characteristic estimation and threshold selection criteria in three-class classification problems for clustered data". *Statistical Methods in Medical Research*, **7**, 31, 1325-1341.

Examples

```
data(data_3class)
## One covariate
out1 <- clus_lme(fixed_formula = Y ~ X1, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### Estimate TCFs at one single value of X1, (t1, t2) = (1, 4)
out_tcfs_1 <- clus_tcfs(out_clus_lme = out1, newdata = data.frame(X1 = 1),
                        thresholds = c(1, 4), ap_var = TRUE)
print(out_tcfs_1)

## Two covariates
out2 <- clus_lme(fixed_formula = Y ~ X1 + X2, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### Estimate covariate-specific TCFs at point (X1, X2) = (1, 0), and (t1, t2) = (1, 4)
out_tcfs_2 <- clus_tcfs(out_clus_lme = out2,
                        newdata = data.frame(X1 = 1, X2 = 0),
                        thresholds = c(1, 4), ap_var = TRUE)
print(out_tcfs_2)

### Estimate covariate-specific TCFs at three points and (t1, t2) = (1, 4)
out_tcfs_3 <- clus_tcfs(out_clus_lme = out2,
                        newdata = data.frame(X1 = c(-0.5, 0.5, 0.5),
                                             X2 = c(0, 0, 1)),
                        thresholds = c(1, 4), ap_var = TRUE)
print(out_tcfs_3)
```

---

clus_vus   *Estimation of the covariate-specific VUS for clustered data.*

Description

This function estimates the covariate-specific VUS of a continuous diagnostic test in the setting of clustered data as described in Xiong et al. (2018). This function allows to estimate covariate-specific VUS at multiple points for covariates.

Usage

```
clus_vus(out_clus_lme, newdata, ap_var = TRUE, subdivisions = 1000, ...)
```

Arguments

out_clus_lme   an object of class "clus_lme", a result of [clus_lme](#) call.

newdata   a data frame (containing specific value(s) of covariate(s)) in which to look for variables with which to estimate covariate-specific VUS. In absence of covariate, no values have to be specified.

ap_var	logical value. If set to TRUE (default), the standard error for (estimated) covariate-specific VUS are estimated.
subdivisions	the maximum number of subintervals used to approximate integral. Default is 1000.
...	additional arguments to be passed to integrate.

Details

This function implements a method in Xiong et al. (2018) for estimating covariate-specific VUS of a continuous diagnostic test in a clustered design with three ordinal groups. The estimator is based on results from clus_lme, which uses the REML approach. The standard error of the estimated covariate-specific VUS is approximated through the Delta method.

Before performing estimation, a check for the monotone ordering assumption is performed. This means that, for the fixed values of covariates, three predicted mean values for test results in three diagnostic groups are compared. If the assumption is not meet, the covariate-specific VUS at the values of covariates are not estimated. In addition, this function also performs the statistical test, $H_0 : VUS = 1/6$ versus an alternative of interest.

Value

clus_vus returns an object of class "VUS" which is a list containing at least the following components:

call	the matched call.
vus_est	a vector containing the estimated covariate-specific VUS.
vus_se	a vector containing the standard errors.
mess_order	a diagnostic message from checking the monontone ordering.
newdata	value(s) of covariate(s).
n_p	total number of regressors in the model.

Generic functions such as print is also used to show the results.

References

Xiong, C., Luo, J., Chen L., Gao, F., Liu, J., Wang, G., Bateman, R. and Morris, J. C. (2018) "Estimating diagnostic accuracy for clustered ordinal diagnostic groups in the three-class case – Application to the early diagnosis of Alzheimer disease". *Statistical Methods in Medical Research*, **27**, 3, 701-714.

Examples

```
data(data_3class)
## One covariate
out1 <- clus_lme(fixed_formula = Y ~ X1, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### Estimate covariate-specific VUS at one value of one covariate
out_vus1 <- clus_vus(out1, newdata = data.frame(X1 = 0.5))
```

```
ci_clus_vus(out_vus1, ci_level = 0.95)

### Estimate covariate-specific VUS at multiple values of one covariate
out_vus2 <- clus_vus(out1, newdata = data.frame(X1 = c(-0.5, 0, 0.5)))
ci_clus_vus(out_vus2, ci_level = 0.95)

## Two covariates
out2 <- clus_lme(fixed_formula = Y ~ X1 + X2, name_class = "D",
                 name_clust = "id_Clus", data = data_3class)

### Estimate covariate-specific VUS at one point
out_vus3 <- clus_vus(out2, newdata = data.frame(X1 = 1.5, X2 = 1))
ci_clus_vus(out_vus3, ci_level = 0.95)

### Estimate covariate-specific VUS at three points
out_vus4 <- clus_vus(out2, newdata = data.frame(X1 = c(-0.5, 0.5, 0.5),
                                                X2 = c(0, 0, 1)))
ci_clus_vus(out_vus4, ci_level = 0.95)
```

---

data_3class　　　　　　　　　*A simulated data*

Description

A simulated data example with 30 clusters.

Usage

```
data(data_3class)
```

Format

A data frame with 225 observations (from 30 clusters).

id_Clus the id number of cluster.

Y a vector containing test results.

D a factor with 3 levels for the disease status, 1, 2, 3. The levels correspond to benign disease, early stage and late stage.

X1 a continuous covariate.

X2 a binary covariate.

---

data_3class_bcx *A simulated data*

Description

A simulated data example with 60 clusters. This dataset is used in a example of analysis with Box-Cox transformation.

Usage

```
data(data_3class_bcx)
```

Format

A data frame with 582 observations (from 60 clusters).

id_Clus the id number of cluster.

Y a vector containing test results.

D a factor with 3 levels for disease status, 1, 2, 3. The levels correspond to benign disease, early stage and late stage.

X a continuous covariate.

EnergyEthiopia *A subset of energy choice data in 4 cities of Ethiopia*

Description

A subset of energy choice data used in Alem et al. (2016). The authors are used the full dataset to investigate the determinants of household cooking fuel choice and energy transition in urban Ethiopia. A full data is publicly available at doi:10.1016/j.eneco.2016.06.025.

Usage

```
data(EnergyEthiopia)
```

Format

A data frame with 2088 observations from 1123 households (or clusters) in the capital Addis Ababa and 9 variables:

uqid the id of household (which yield 1123 clusters).

energy2 a factor with 3 levels (types) of cooking energy state at each time (2000, 2004, 2009), i.e., 1 (clean fuel only - electricity, gas and kerosene), 2 (a mix of clean and biomass), 3 (biomass fuel only - firewood, charcoal, dung and crop residues).

hhs household size.

hhs_ft a factor with 4 levels of household size: small ($1 \leq$ hhs ≤ 4); medium ($5 \leq$ hhs ≤ 8); large (($9 \leq$ hhs ≤ 12)); very large (hhs ≥ 13).

lrconsaeu log of real consumption per adult equivalent units.

lfirewood_pr Firewood log price.

lcharcol_pr Charcoal log price.

lkerosene_pr Kernosene log price.

lelectric_pr Electricity log price.

References

Alem, Y., Beyene, A. D., Köhlin, G., & Mekonnen, A. (2016). "Modeling household cooking fuel choice: A panel multinomial logit approach". *Energy Economics*, **59**, 129-137.

MouseNeurons   *A subset of mouse brain cells data*

Description

A subset of mouse brain cells data used in To el al. (2022). This is used to evaluate the ability of Lamp5 gene to discriminate three types of glutamatergic neurons. A full data is publicly available at https://portal.brain-map.org/atlases-and-data/rnaseq/mouse-v1-and-alm-smart-seq.

Usage

```
data(MouseNeurons)
```

Format

A data frame with 860 observations from 23 clusters and 7 variables:

sample_name name of each observation.

subclass_label a factor with 3 levels (types) of glutamatergic neurons, i.e., L2/3 IT (Layer 2/3 Intratelencephalic), L4 (Layer 4) and L5 PT (Layer 5 Pyramidal Tract) neurons.

genotype_id the mouse genotype (which yield 23 clusters).

sex the gender of mouse.

age_days the age of mouse, in days.

Slc17a7_cpm count per million of Slc17a7 (Solute Carrier Family 17 Member 7) gene expression.

Lamp5_cpm count per million of Lamp5 (Lysosomal Associated Membrane Protein Family Member 5) gene expression.

References

To, D-K., Adimari, G., Chiogna, M. and Risso, D. (2022) "Receiver operating characteristic estimation and threshold selection criteria in three-class classification problems for clustered data". *Statistical Methods in Medical Research*, **7**, 31, 1325-1341.

---

plot.clus_lme *Plot an clus_lme object.*

Description

Diagnostic plots for the linear mixed-effect model, fitted by clus_lme.

Usage

```
## S3 method for class 'clus_lme'
plot(x, file_name = NULL, ...)
```

Arguments

x	an object of class "clus_lme", i.e., a result of [clus_lme](#) call.
file_name	File name to create on disk.
...	further arguments used with [ggexport](#) function, for example, width, height.

Details

plot.clus_lme provides three diagnostic plots: Q-Q plots for residuals, Fitted vs. Residuals values, and Q-Q plot for cluster effects, based on ggplot().

Value

plot.clus_lme returns the diagnostic plots for the linear mixed-effect model, fitted by clus_lme.

See Also

[clus_lme](#)

---

plot.clus_opt_thres3 *Plot of confidence regions for covariate-specific optimal pair of thresholds.*

Description

This function plots confidence regions for covariate-specific optimal pair of thresholds.

Usage

```
## S3 method for class 'clus_opt_thres3'
plot(
  x,
  ci_level = 0.95,
  colors = NULL,
  xlims,
  ylims,
  size_point = 0.5,
  size_path = 0.5,
  names_labels,
  nrow_legend = 1,
  file_name = NULL,
  ...
)
```

Arguments

x	an object of class "clus_opt_thres3", i.e., a result of [clus_opt_thres3](#).
ci_level	confidence level to be used for constructing the confidence regions; default is 0.95.
colors	a string vector for the name(s) specifying color(s) to be used for drawing confidence regions. If specified, the dimension of the vector needs to be equal the number of considered points (each point corresponds to a set of values for the covariates).
xlims, ylims	numeric vectors of dimension 2, giving the limits for x and y axes in the plot.
size_point, size_path	
	numeric values, indicating sizes for point(s) and line(s) in the plot.
names_labels	an optional character vector giving the label name for covariates.
nrow_legend	an optional number of rows in the legend.
file_name	File name to create on disk.
...	further arguments used with [ggexport](#) function, for example, width, height.

Details

plot.clus_opt_thres3 provides plots of confidence regions (and point estimates) of covariate-specific optimal pair of thresholds. The plots are based on ggplot().

Value

plot.clus_opt_thres3 returns plots of confidence regions of covariate-specific optimal pair of thresholds.

See Also

[clus_opt_thres3](#)

---

print.ci_clus_vus   *Print summary results from ci_clus_vus*

Description

print.ci_vus displays the results of the output from [ci_clus_vus](#).

Usage

```
## S3 method for class 'ci_clus_vus'
print(x, digits = 3, ...)
```

Arguments

x	an object of class "ci_clus_vus", a result of [ci_clus_vus](#) call.
digits	minimal number of significant digits, see [print.default](#).
...	further arguments passed to [print](#) method.

Details

print.ci_clus_vus shows a summary table for confidence interval limits for covariate-specific VUS.

Value

print.ci_clus_vus shows a summary table for confidence intervals for covariate-specific VUS.

See Also

[clus_vus](#)

---

print.clus_lme   *Print summary results of an clus_lme object*

Description

print.clus_lme displays results of the output from [clus_lme](#).

Usage

```
## S3 method for class 'clus_lme'
print(x, digits = max(3L, getOption("digits") - 3L), call = TRUE, ...)
```

Arguments

x	an object of class "clus_lme", a result of [clus_lme](#) call.
digits	minimal number of significant digits, see [print.default](#).
call	logical. If set to TRUE, the matched call will be printed.
...	further arguments passed to [print](#) method.

Details

print.clus_lme shows a summary table for the estimated parameters in the cluster-effect model (continuous diagnostic test in three-class setting).

Value

print.clus_lme returns a summary table for the estimated parameters in the cluster-effect model.

See Also

[clus_lme](#)

---

print.clus_opt_thres3 *Print summary results from* clus_opt_thres3

Description

print.clus_opt_thres3 displays the results of the output from [clus_opt_thres3](#).

Usage

```
## S3 method for class 'clus_opt_thres3'
print(x, digits = 3, call = TRUE, ...)
```

Arguments

x	an object of class "clus_opt_thres3", a result of [clus_opt_thres3](#) call.
digits	minimal number of significant digits, see [print.default](#).
call	logical. If set to TRUE, the matched call will be printed.
...	further arguments passed to [print](#) method.

Details

print.clus_opt_thres3 shows a summary table for covariate-specific optimal pair of thresholds estimates.

Value

print.clus_opt_thres3 returns a summary table for results of covariate-specific optimal pair of thresholds estimation.

See Also

[clus_opt_thres3](#)

---

print.clus_tcfs   *Print summary results from clus_tcfs*

Description

print.clus_tcfs displays the results of the output from [clus_tcfs](#).

Usage

```
## S3 method for class 'clus_tcfs'
print(x, digits = 3, call = TRUE, ...)
```

Arguments

x	an object of class "clus_tcfs", a result of [clus_tcfs](#) call.
digits	minimal number of significant digits, see [print.default](#).
call	logical. If set to TRUE, the matched call will be printed.
...	further arguments passed to [print](#) method.

Details

print.clus_tcfs shows a summary table for covariate-specific TCFs estimates.

Value

print.clus_tcfs returns a summary table for covariate-specific TCFs estimates.

See Also

[clus_tcfs](#)

| print.clus_vus | *Print summary results from clus_vus* |

Description

print.clus_vus displays the results of the output from clus_vus.

Usage

```
## S3 method for class 'clus_vus'
print(x, digits = 3, call = TRUE, ...)
```

Arguments

x	an object of class "VUS", a result of clus_vus call.
digits	minimal number of significant digits, see print.default.
call	logical. If set to TRUE, the matched call will be printed.
...	further arguments passed to print method.

Details

print.clus_vus shows a summary table for covariate-specific VUS estimates, containing estimates, standard errors, z-values and p-values for the hypothesis testing $H_0 : VUS = 1/6$ versus an alternative $H_A : VUS > 1/6$.

Value

print.clus_vus returns a summary table for covariate-specific VUS estimates.

See Also

clus_vus

Index