

Package ‘AUCcomparison’

April 29, 2021

Title Statistical test - Between group comparison of AUC

Version 0.0.0.9000

Author Marie Alexandre [aut,cre]

Description The goal of AUCcomparison is to propose a statistical test evaluating the difference of Area under the curve (AUC) of a given outcome between two distinct groups of individuals. To this end, longitudinal data obtained for subjects splitted into G distinct groups are fitted with a Mixed-Effects model whose fixed-effects (marginal dynamics) and random-effects (individual dynamics) are respectively modeled by group-structured polynomial or B-splines curves and individual polynomial or B-spline curves.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Imports freeknotsplines,
lme4,
matlib,
splines,
ArgumentCheck,
dplyr,
gtools,
plyr,
mathjaxr,
rmarkdown

RdMacros mathjaxr

Suggests testthat,
knitr

Depends R (>= 2.10),
mathjaxr

VignetteBuilder knitr

R topics documented:

ANOVA_AUC_Group_Specific	2
AUC_Lagrange_Cjp_coefficients	5

AUC_Spline_matrix_A	6
AUC_Spline_matrix_B	7
AUC_time_weights_estimation	8
Group_specific_AUC_estimation	9
Group_specific_Delta_AUC_estimation	11
Group_specific_Var_AUC_estimation	14
Group_specific_Var_Delta_AUC_estimation	17
HIV_Simu_Dataset_Delta0	20
HIV_Simu_Dataset_Delta01	20
HIV_Simu_Dataset_Delta01_cens	21
HIV_Simu_Dataset_Delta025	22
HIV_Simu_Dataset_Delta025_cens	22
HIV_Simu_Dataset_Delta0_cens	23
MEM_Polynomial_Group_structure	24
Optimal_knot_research	29
Stat_test_Delta_AUC_Group_Specific	30

Index	33
--------------	-----------

ANOVA_AUC_Group_Specific

One-way analysis of variance of group specific AUC

Description

This function performs a one-way ANOVA to compare the area under the curves of multiple groups marginal dynamics, modeled by group-structured polynomials or B-spline curve in Mixed-Effects model. Before performing the ANOVA, this function can perform a Bartlett's test to evaluate homoscedasticity. In addition to ANOVA, users can decide to evaluate all the 2 by 2 comparisons.

Usage

```
ANOVA_AUC_Group_Specific(
  MEM_Pol_group, Groups, Time_groups,
  Nb_id_group, common.interval = TRUE,
  method = "trapezoid", Averaged = FALSE,
  conf_level = 0.95, bartlettTest = FALSE,
  data = NULL, twobytwo.comp = TRUE,
  alternative = "two.sided"
)
```

Arguments

- MEM_Pol_group A list with similar structure than the output provided by the function [MEM_Polynomial_Group_struct](#)
- A list containing:
- Model_estimation: a list containing at least 2 elements:
 1. the vector of the marginal (fixed) parameters estimates (at least for the groups whose AUC is to estimate), labeled '*beta*'.
 2. the variance-covariance matrix of these parameters, labeled '*varFix*' (see [MEM_Polynomial_Group_structure](#) for details about the parameter order).

- `Model_features`: a list of at least 2 elements:
 1. `Groups`: a vector indicating the names of the groups whose fixed parameters are given.
 2. `Marginal.dyn.feature`: a list summarizing the features of the marginal dynamics defined in the model:
 - `dynamic.type`: a character scalar indicating the chosen type of marginal dynamics. Options are 'polynomial' or 'spline'.
 - `intercept`: a logical vector summarizing choices about global and group-specific intercepts (Number of groups + 1) elements whose elements are named as ('global.intercept', 'group.intercept1', ..., 'group.interceptG') if G Groups are defined in `MEM_Pol_group`. For each element of the vector, if TRUE, the considered intercept is considered as included in the model (see *Examples*).
- If `dynamic.type` is defined as 'polynomial':
 - `polynomial.degree`: an integer vector indicating the degree of polynomial functions, one value for each group.
- If `dynamic.type` is defined as 'spline':
 - `spline.degree`: an integer vector indicating the degree of B-spline curves, one for each group.
 - `knots`: a list of group-specific internal knots used to build B-spline basis (one numerical vector for each group) (see [bs](#) for more details).
 - `df`: a numerical vector of group-specific degrees of freedom used to build B-spline basis, (one for each group).
 - `boundary.knots`: a list of group-specific boundary knots used to build B-spline basis (one vector for each group) (see [bs](#) for more details).

<code>Groups</code>	a vector indicating the names of the groups belonging to the set of groups involved in <code>MEM_Pol_group</code> we want to include in the ANOVA (a subset or the entire set of groups involved in the model can be considered).
<code>Time_groups</code>	a list of numerical vectors indicating the time points to consider in AUC calculation for each group (as much elements than the number of groups in <code>Groups</code>).
<code>Nb_id_group</code>	a numerical vector indicating the number of individuals belonging to each group (as much elements than the number of groups in <code>Groups</code>).
<code>common.interval</code>	a logical scalar. If TRUE (default) AUCs of all the compared groups are calculated on the same time interval defined as the intersect of all the time interval defined in <code>Time_groups</code> . If FALSE, AUC specific to each group is evaluated on its own interval.
<code>method</code>	a character scalar indicating the interpolation method to use to estimate the AUC. Options are 'trapezoid' (default), 'lagrange' and 'spline'. In this version, the 'spline' interpolation is implemented with "not-a-knot" spline boundary conditions.
<code>Averaged</code>	a logical scalar. If TRUE, AUC are evaluated as normalized AUC (nAUC) where nAUC is computed as the AUC divided by the range of time of calculation. If FALSE (default), the classic AUC is calculated (see Group_specific_AUC_estimation for more details).
<code>conf_level</code>	a numerical value (between 0 and 1) indicating the confidence level of the interval. By default, this variable is fixed at 0.95

<code>bartlettTest</code>	a logical scalar indicating whether a bartlettTest must be performed before the ANOVA. Default: FALSE
<code>data</code>	a dataframe gathering data for the groups involved in the ANOVA that have been fitted by the MEM model summarized in <code>MEM_Po1_group</code> . This dataframe has to contain at least 4 columns: <ul style="list-style-type: none"> • A column labeled 'Group' containing the information of the group for each observation (the same Groups than those defined in <code>Groups</code>). • A column labeled 'id' containing the information of individual identifier. • A column labeled 'time' containing information about the time of observations. • A column labeled 'value' containing the longitudinal observations.
<code>twobytwo.comp</code>	a logical scalar indicating whether all the 2 by 2 comparisons must be evaluated after the ANOVA. Default: TRUE.
<code>alternative</code>	a character scalar specifying the alternative hypothesis for the 2 by 2 comparisons. Options are 'two.sided' (default), 'greater' or 'less'.

Value

A list of three elements elements:

- `bartlettTest`: a list of class "htest" corresponding to the Bartlett's test results (see [bartlett.test](#) for more details). If the test is not performed, a character 'Not performed' is returned.
- `ANOVA_F`: a list containing:
 1. `Followup`: a vector of two numerical values indicating the time interval when `Common.interval = TRUE`. If `common.interval=FALSE`, if variable is not included in the list.
 2. `Estimated_AUCs`: the vector of AUC estimated for each group involved in the ANOVA.
 3. `Estimated_VarAUCs`: the vector of the intra group variance estimated by the MEM (equiv. squared Standard error).
 4. `Between`: a vector gathering the between groups sum of squares, degree of freedom and variance with `var=SS/df`.
 5. `Within`: a vector gathering the within groups sum of squares, degree of freedom and variance with `var=SS/df`.
 6. `ANOVA_F`: the value of the ANOVA statistic F.
 7. `Pvalue`: the value of the Pvalue.
 8. `Reject_H0`: a boolean indicating whether the null hypothesis (all AUC are equals) is rejected.
- `TwobyTwo_Comparison`: a list of K sublists where each sublist k gathers results of the kth comparison. If the 2 by 2 comparison is not performed, a character 'Not performed' is returned.
 1. `Groups`: a vector indicating the 2 compared groups.
 2. `Estimated.AUC`: a vector of the 2 estimated values of AUC.
 3. `Delta_AUC`: the value of the difference of AUC between the 2 compared groups.
 4. `Tstat`: the value of the t-statistic.
 5. `Pvalue`: the P-value (without any adjustment on multiple tests).
 6. `Conf.int`: the confidence interval.
 7. `Adjusted.Pvalue`: the P-value adjusted on multiple testing by "bonferroni" method (see [p.adjust](#) for more details).

See Also

[AUC_time_weights_estimation bartlett.test](#)

Examples

```
# Download of data
data("HIV_Simu_Dataset_Delta01_cens")
data <- HIV_Simu_Dataset_Delta01_cens

colnames(data) <- c("id","time","Group","value","cens")
# Change factors in character vectors
data$id <- as.character(data$id) ; data$Group <- as.character(data$Group)

MEM_estimation <- MEM_Polynomial_Group_structure(y=data$value,x=data$time,Group=data$Group,
                                                Id=data$id,Cens=data$cens)

Groups <- unique(data$Group)
Time_groups <- lapply(seq(1,length(Groups)),function(g)
                      return(unique(data$time[which(data$Group == Groups[g])])))
Nb_id_group <- sapply(seq(1,length(Groups)),function(g)
                     return(length(unique(data$id[which(data$Group == Groups[g])]))))

ANOVA_test <- ANOVA_AUC_Group_Specific(MEM_Pol_group=MEM_estimation,Groups=Groups,
                                       Time_groups=Time_groups,Nb_id_group=Nb_id_group,
                                       common.interval=TRUE,method="trapezoid",Averaged=FALSE,
                                       conf_level=0.95,bartlettTest=TRUE,data=data,
                                       twobytwo.comp=TRUE,alternative="two.sided")
```

AUC_Lagrange_Cjp_coefficients

Time Dependent Coefficients Cjp for AUC Lagrange Interpolation Method

Description

This function calculate the time-dependent coefficients Cjp involved in the calculation of the area under the curve when the Lagrange interpolation method is used.

Usage

```
AUC_Lagrange_Cjp_coefficients(ind_j,ind_p,t)
```

Arguments

ind_j	a numerical scalar indicating the value of the index j.
ind_p	a numerical scalar indicating the value of the index p
t	a numerical vector of time points (x-axis coordinates) to consider for the AUC calculation.

Details

The coefficients C_{jp} involved in the calculation of the AUC are defined as

$$\begin{aligned}
 C_{2p} &= (t_2 - t_1) \prod_{l=0; l \neq p}^{P=2} t_{1+l} - \frac{(t_2^2 - t_1^2)}{2} \sum_{l=0; l \neq p}^{P=2} t_{1+l} + \frac{(t_2^3 - t_1^3)}{3} \\
 C_{mp} &= (t_m - t_{m-1}) \prod_{l=0; l \neq p}^{P=2} t_{m-2+l} - \frac{(t_m^2 - t_{m-1}^2)}{2} \sum_{l=0; l \neq p}^{P=2} t_{m-2+l} + \frac{(t_m^3 - t_{m-1}^3)}{3} \\
 C_{jp} &= -(t_j - t_{j-1}) \prod_{l=0; l \neq p}^{P=3} t_{j-2+l} + \frac{(t_j^2 - t_{j-1}^2)}{2} \sum_{l_1=0; l_1 \neq p}^{P-1=2} \sum_{l_2=l_1+1; l_2 \neq p}^{P=3} t_{j-2+l_1} \cdot t_{j-2+l_2} - \\
 &\quad \frac{(t_j^3 - t_{j-1}^3)}{3} \sum_{l=0; l \neq p}^{P=3} t_{j-2+l} + \frac{(t_j^4 - t_{j-1}^4)}{4}
 \end{aligned}$$

where m is the number of time points in the vector \mathbf{t} .

Value

a numerical scalar corresponding to the coefficient C_{jp} evaluated for $j = \text{ind_j}$ and $p = \text{ind_p}$.

AUC_Spline_matrix_A	<i>Spline Interpolation Method - Matrix of Second Derivative Coefficients</i>
---------------------	---

Description

In the area under the curve calculation using the spline interpolation method, the vector of the second derivative of the outcome of interest Y is expressed as $AY'' = BY + F$. This function calculate calculate the matrix A .

Usage

```
AUC_Spline_matrix_A(time)
```

Arguments

`time` a numerical vector of time points of length m (x-axis coordinates).

Details

The tridiagonal matrix A is defined as (for the "not-a-knot boundary conditions"): The j th line of the matrix, $A_{[j, :]}$ is given by

$$\begin{aligned}
 A_{[j, :]} &= \left(\frac{1}{h_2}, -\left[\frac{1}{h_2} + \frac{1}{h_3} \right], \frac{1}{h_3}, 0, \dots, 0 \right) \text{ if } j = 1 \\
 A_{[j, :]} &= \left(0, \dots, 0, \frac{1}{h_{m-1}}, -\left[\frac{1}{h_{m-1}} + \frac{1}{h_m} \right], \frac{1}{h_m} \right) \text{ if } j = m \\
 A_{[j, :]} &= \left(0_1, \dots, 0_{j-2}, \frac{h_j}{6}, \frac{h_j + h_{j+1}}{3}, \frac{h_{j+1}}{6}, 0_{j+2}, \dots, 0_m \right) \text{ otherwise}
 \end{aligned}$$

Value

a tridiagonal matrix corresponding to the weights of the second derivative of the variable of interest in the spline interpolation method. In this version, the matrix is build considering the "not-a-knot" spline boundary conditions.

AUC_Spline_matrix_B	<i>Spline Interpolation Method - Matrix of the zero order derivative coefficients</i>
---------------------	---

Description

In the area under the curve calculation using the spline interpolation method, the vector of the second derivative of the outcome of interest Y is expressed as $AY'' = BY + F$. This function calculate calculate the matrix B.

Usage

```
AUC_Spline_matrix_B(time)
```

Arguments

time a numerical vector of time points of length m (x-axis coordinates).

Details

The tridiagonal matrix B is defined as (for the "not-a-knot boundary conditions): The j th line of the matrix, $B_{[j, :]}$ is given by

$$B_{[j, :]} = (0, \dots, 0) \text{ if } j = 1$$

$$B_{[j, :]} = (0, \dots, 0) \text{ if } j = m$$

$$B_{[j, :]} = \left(0_1, \dots, 0_{j-2}, \frac{1}{h_j}, -\left[\frac{1}{h_j} + \frac{1}{h_{j+1}} \right], \frac{1}{h_{j+1}}, 0_{j+2}, \dots, 0_m \right) \text{ otherwise}$$

Value

a tridiagonal matrix corresponding to the weights of the variable of interest in the spline interpolation method. In this version, the matrix is build considering the "not-a-knot" spline boundary conditions.

AUC_time_weights_estimation

Weights for AUC Matrix Formulation

Description

In matrix formulation, the area under a curve of interest, named Y , can be expressed as matrix product of a vector of weights W and the vector of the values of Y . This function calculates the weights W when AUC is calculated either by the trapezoid, the Lagrange or the Spline interpolation methods.

Usage

```
AUC_time_weights_estimation(time, method)
```

Arguments

time	a numerical vector of time points of length m (x-axis coordinates for AUC calculation).
method	a character scalar indicating the interpolation method of interest. Options are 'trapezoid', 'lagrange' and 'spline'. In this version the 'spline' interpolation method is implemented with the "not-a-knot" spline boundary conditions.

Details

In matrix formulation, the AUC of the outcome Y can be expressed as $AUC = W \cdot Y$, with W defined by the following expressions for the trapezoid, the Lagrange and the spline interpolation methods.

Trapezoid method:

$$W_j = \frac{t_{j+1} - t_j}{2} \text{ if } j = 1$$

$$W_j = \frac{t_j - t_{j-1}}{2} \text{ if } j = m$$

$$W_j = \frac{t_{j+1} - t_{j-1}}{2} \text{ otherwise}$$

Lagrange method: (see [AUC_Lagrange_Cjp_coefficients](#) for the definition of the Cjp coefficients)

$$W_j = \frac{C_{[2][j-1]}}{\prod_{l=0; l \neq (j-1)}^{P=2} (t_j - t_{j+1+l})} + \sum_{p=0}^{P=3} \frac{C_{[j-1+p][3-p]}}{\prod_{l=0; l \neq (3-p)}^{P=3} (t_j - t_{j-3+p+l})} \text{ if } j = 1, 2, 3$$

$$W_j = \frac{C_{[m][j-(m-2)]}}{\prod_{l=0; l \neq (j-(m-2))}^{P=2} (t_j - t_{j-2+l})} + \sum_{p=0}^{m-j} \frac{C_{[j-1+p][3-p]}}{\prod_{l=0; l \neq (3-p)}^{P=3} (t_j - t_{j-3+p+l})} \text{ if } j = m-2, m-1, m$$

$$W_j = \sum_{p=0}^{m-j} \frac{C_{[j-1+p][3-p]}}{\prod_{l=0; l \neq (3-p)}^{P=3} (t_j - t_{j-3+p+l})} \text{ otherwise}$$

Spline method: (see [AUC_Spline_matrix_A](#) and [AUC_Spline_matrix_B](#) for the definition of Matrices A and B)

$$W_j = \sum_{p=2}^m -\frac{(t_p - t_{p-1})^3}{24}(u_{pj} + u_{p-1j}) + W_j^{trap}.$$

where (u_{pj}) is the element $U(p, j)$ with U a matrix defined as $U = A^{-1}B$.

Value

A numerical scalar with same length than the vector `time` corresponding to the weights W .

Group_specific_AUC_estimation

Area Under The Curve of Group-Specific Polynomial Marginal Dynamics

Description

This function estimates the area under the curve of marginal dynamics modeled by group-structured polynomials or B-spline curves.

Usage

```
Group_specific_AUC_estimation(
  MEM_Pol_group, time, Groups = NULL,
  method = "trapezoid", Averaged = FALSE
)
```

Arguments

- `MEM_Pol_group` A list with similar structure than the output provided by the function [MEM_Polynomial_Group_struct](#)
A list containing:
- `Model_estimation`: a list containing at least 2 elements:
 1. the vector of the marginal (fixed) parameters estimates (at least for the groups whose AUC is to estimate), labeled `'beta'`.
 2. the variance-covariance matrix of these parameters, labeled `'varFix'` (see [MEM_Polynomial_Group_structure](#) for details about the parameter order).
 - `Model_features`: a list of at least 2 elements:
 1. `Groups`: a vector indicating the names of the groups whose fixed parameters are given.
 2. `Marginal.dyn.feature`: a list summarizing the features of the marginal dynamics defined in the model:
 - `dynamic.type`: a character scalar indicating the chosen type of marginal dynamics. Options are `'polynomial'` or `'spline'`.
 - `intercept`: a logical vector summarizing choices about global and group-specific intercepts (Number of groups + 1) elements whose elements are named as (`'global.intercept'`, `'group.intercept1'`, ..., `'group.interceptG'`) if `G` Groups are defined in `MEM_Pol_group`. For each element of the vector, if `TRUE`, the considered intercept is considered as included in the model (see *Examples*).

If `dynamic.type` is defined as 'polynomial':

- `polynomial.degree`: an integer vector indicating the degree of polynomial functions, one value for each group.

If `dynamic.type` is defined as 'spline':

- `spline.degree`: an integer vector indicating the degree of B-spline curves, one for each group.
- `knots`: a list of group-specific internal knots used to build B-spline basis (one numerical vector for each group) (see [bs](#) for more details).
- `df`: a numerical vector of group-specific degrees of freedom used to build B-spline basis, (one for each group).
- `boundary.knots`: a list of group-specific boundary knots used to build B-spline basis (one vector for each group) (see [bs](#) for more details).

<code>time</code>	a numerical vector of time points (x-axis coordinates) or a list of numerical vectors (with as much elements than the number of groups in <code>Groups</code>).
<code>Groups</code>	a vector indicating the names of the groups belonging to the set of groups involved in <code>MEM_Pol_group</code> for which we want to estimate the AUC (a subset or the entire set of groups involved in the model can be considered). If <code>NULL</code> (default), the AUC for all the groups involved the MEM is calculated.
<code>method</code>	a character scalar indicating the interpolation method to use to estimate the AUC. Options are 'trapezoid' (default), 'lagrange' and 'spline'. In this version, the 'spline' interpolation is implemented with the "not-a-knot" spline boundary conditions.
<code>Averaged</code>	a logical scalar. If <code>TRUE</code> , the function return the normalized AUC (nAUC) computed as the AUC divided by the range of the time calculation. If <code>FALSE</code> (default), the classic AUC is calculated.

Details

The area under the curve for the group g of interest is calculated as an approximation of the integral of the expected value of the estimated outcome Y specific to the group g . Assuming a time interval $[0, T_g]$, the AUC is then calculated as

$$AUC_g = \int_0^{T_g} E(\hat{Y}_g)(t) dt$$

Similarly, the normalized AUC (nAUC) for this same group is then defined as

$$nAUC_g = \frac{1}{T_g} \int_0^{T_g} E(\hat{Y}_g)(t) dt$$

Value

A numerical vector containing the estimation of the AUC (or nAUC) for each group defined in the `Groups` vector.

See Also

[bs](#), [MEM_Polynomial_Group_structure](#)

[illegible]

Group_specific_Delta_AUC_estimation

Difference of AUC of Two Group-Specific Polynomial Marginal Dynamics

Description

This function estimates the difference of area under the curve of marginal dynamics from two groups when marginal dynamics are modeled by group-structured polynomials or B-spline curves.

Usage

```
Group_specific_Delta_AUC_estimation(
  MEM_Pol_group, Group1, Group2,
  time.G1, time.G2, common.interval = TRUE,
  method = "trapezoid", Averaged = FALSE
)
```

Arguments

MEM_Pol_group A list with similar structure than the output provided by the function [MEM_Polynomial_Group_struct](#)
A list containing:

- **Model_estimation**: a list containing at least 2 elements:
 1. the vector of the marginal (fixed) parameters estimates (at least for the groups whose AUC is to estimate), labeled *'beta'*.
 2. the variance-covariance matrix of these parameters, labeled *'varFix'* (see [MEM_Polynomial_Group_structure](#) for details about the parameter order).
- **Model_features**: a list of at least 2 elements:
 1. **Groups**: a vector indicating the names of the groups whose fixed parameters are given.
 2. **Marginal.dyn.feature**: a list summarizing the features of the marginal dynamics defined in the model:
 - **dynamic.type**: a character scalar indicating the chosen type of marginal dynamics. Options are *'polynomial'* or *'spline'*.
 - **intercept**: a logical vector summarizing choices about global and group-specific intercepts (Number of groups + 1) elements whose elements are named as (*'global.intercept'*, *'group.intercept1'*, ..., *'group.interceptG'*) if G Groups are defined in MEM_Pol_group. For each element of the vector, if TRUE, the considered intercept is considered as included in the model (see *Examples*).

If **dynamic.type** is defined as *'polynomial'*:

 - **polynomial.degree**: an integer vector indicating the degree of polynomial functions, one value for each group.

If **dynamic.type** is defined as *'spline'*:

 - **spline.degree**: an integer vector indicating the degree of B-spline curves, one for each group.
 - **knots**: a list of group-specific internal knots used to build B-spline basis (one numerical vector for each group) (see [bs](#) for more details).

	<ul style="list-style-type: none"> – <code>df</code>: a numerical vector of group-specific degrees of freedom used to build B-spline basis, (one for each group). – <code>boundary.knots</code>: a list of group-specific boundary knots used to build B-spline basis (one vector for each group) (see bs for more details).
<code>Group1</code>	a character scalar indicating the name of the first group whose marginal dynamics must be considered. This group name must belong to the set of groups involved in the MEM (see <code>Groups</code> vector in <code>MEM_Po1_group</code>).
<code>Group2</code>	a character scalar indicating the name of the second group whose marginal dynamics must be considered. This group name must belong to the set of groups involved in the MEM (see <code>Groups</code> vector in <code>MEM_Po1_group</code>).
<code>time.G1</code>	a numerical vector of time points (x-axis coordinates) to use for the Group1 AUC calculation.
<code>time.G2</code>	a numerical vector of time points (x-axis coordinates) to use for the Group2 AUC calculation.
<code>common.interval</code>	a logical scalar. If FALSE, the difference of AUC is calculated as the difference of AUCs where the AUC of each group is calculated on its specific interval of time. If TRUE (default), the difference of AUC is estimated on a common interval of time defined as the intersect of the two group-specific interval (see @details for more details).
<code>method</code>	a character scalar indicating the interpolation method to use to estimate the AUC. Options are 'trapezoid' (default), 'lagrange' and 'spline'. In this version, the 'spline' interpolation is implemented with "not-a-knot" spline boundary conditions.
<code>Averaged</code>	a logical scalar. If TRUE, the function return the difference of normalized AUC (nAUC) where nAUC is computed as the AUC divided by the range of time of calculation. If FALSE (default), the classic AUC is calculated.

Details

The difference of area under the curve between the two groups of interest is calculated as an approximation of the difference of the integrals of the expected value of the estimated outcome Y specific to the two groups $g1$ and $g2$. Assuming a time interval $[0, T_{g1}]$ for the group 1 and $[0, T_{g2}]$ for the group 2, the difference of AUC is then calculated as

$$\Delta AUC = \int_0^{T_{g2}} E(\hat{Y}_{g2})(t)dt - \int_0^{T_{g1}} E(\hat{Y}_{g1})(t)dt$$

Similarly, the difference of the normalized AUC ($\Delta nAUC$) for these two same groups is then defined as

$$\Delta nAUC = \frac{1}{T_{g2}} \int_0^{T_{g2}} E(\hat{Y}_{g2})(t)dt - \frac{1}{T_{g1}} \int_0^{T_{g1}} E(\hat{Y}_{g1})(t)dt$$

When we want to estimate the difference of AUC (or nAUC) on the same time interval for the two groups (`common.interval = TRUE`), we define $T = \min(T_{g1}, T_{g2})$ and we calculate the difference of AUC as

$$\Delta AUC = \int_0^T E(\hat{Y}_{g2})(t)dt - \int_0^T E(\hat{Y}_{g1})(t)dt$$

$$\Delta nAUC = \frac{1}{T} \int_0^T E(\hat{Y}_{g2})(t)dt - \frac{1}{T} \int_0^T E(\hat{Y}_{g1})(t)dt$$

Value

A numerical scalar defined as $\Delta AUC = AUC_2 - AUC_1$ (or $\Delta nAUC = nAUC_2 - nAUC_1$) with AUC_1 (or $nAUC_1$) and AUC_2 (or $nAUC_2$) being respectively estimated as the AUC (or nAUC) for the Group1 and for the Group2.

See Also

[MEM_Polynomial_Group_structure](#), [Group_specific_AUC_estimation](#)

Examples

```
# Download of data
data("HIV_Simu_Dataset_Delta01_cens")
data <- HIV_Simu_Dataset_Delta01_cens

# Change factors in character vectors
data$id <- as.character(data$id) ; data$Group <- as.character(data$Group)

# Example 1: We consider the variable \code{MEM_Pol_Group} as the output of our function
# \link[AUCcomparison]{MEM_Polynomial_Group_structure}
MEM_estimation_1 <- MEM_Polynomial_Group_structure(y=data$VL,x=data$time,Group=data$Group,
                                                  Id=data$id,Cens=data$cens)

time_group1 <- unique(data$time[which(data$Group=="Group1")])
time_group2 <- unique(data$time[which(data$Group=="Group2")])

Delta_AUC_1 <- Group_specific_Delta_AUC_estimation(MEM_Pol_group=MEM_estimation_1,
                                                  Group1="Group1",Group2="Group2",
                                                  time.G1=time_group1,time.G2=time_group2)

# Example 2: We consider results of MEM estimation from another source.
# We have to give build the variable 'MEM_Pol_group' with the good structure
# We build the variable 'MEM_estimation_2' with the results of MEM estimation obtained for 2 groups
MEM_estimation_2 <- list(Model_estimation=c(1.077,0.858,-0.061,0.0013,0.887,-0.066,0.0014),
                        Model_features=list(Groups=c("Group1","Group2"),
                                             Marginal.dyn.feature=list(dynamic.type="polynomial",
                                                                           intercept=c(global.intercept=TRUE,
                                                                           group.intercept1=FALSE,group.intercept2=FALSE),
                                                                           polynomial.degree=c(3,3))))

Delta_AUC_2 <- Group_specific_Delta_AUC_estimation(MEM_Pol_group=MEM_estimation_2,
                                                  Group1="Group1",Group2="Group2",
                                                  time.G1=time_group1,time.G2=time_group2)
```

Group_specific_Var_AUC_estimation

*Variance of the Area Under The Curve of Group-Specific Polynomial
Marginal Dynamics*

Description

This function calculates the variance of the area under the curve of marginal dynamics modeled by group-structured polynomials or B-spline curves in Mixed-Effects models

Usage

```
Group_specific_Var_AUC_estimation(
  MEM_Pol_group, time, Groups = NULL,
  method = "trapezoid", Averaged = FALSE
)
```

Arguments

- MEM_Pol_group** A list with similar structure than the output provided by the function [MEM_Polynomial_Group_struct](#)
A list containing:
- **Model_estimation**: a list containing at least 2 elements:
 1. the vector of the marginal (fixed) parameters estimates (at least for the groups whose AUC is to estimate), labeled '*beta*'.
 2. the variance-covariance matrix of these parameters, labeled '*varFix*' (see [MEM_Polynomial_Group_structure](#) for details about the parameter order).
 - **Model_features**: a list of at least 2 elements:
 1. **Groups**: a vector indicating the names of the groups whose fixed parameters are given.
 2. **Marginal.dyn.feature**: a list summarizing the features of the marginal dynamics defined in the model:
 - **dynamic.type**: a character scalar indicating the chosen type of marginal dynamics. Options are 'polynomial' or 'spline'.
 - **intercept**: a logical vector summarizing choices about global and group-specific intercepts (Number of groups + 1) elements whose elements are named as ('global.intercept', 'group.intercept1', ..., 'group.interceptG') if G Groups are defined in MEM_Pol_group. For each element of the vector, if TRUE, the considered intercept is considered as included in the model (see *Examples*).

If **dynamic.type** is defined as 'polynomial':

 - **polynomial.degree**: an integer vector indicating the degree of polynomial functions, one value for each group.

If **dynamic.type** is defined as 'spline':

 - **spline.degree**: an integer vector indicating the degree of B-spline curves, one for each group.
 - **knots**: a list of group-specific internal knots used to build B-spline basis (one numerical vector for each group) (see [bs](#) for more details).
 - **df**: a numerical vector of group-specific degrees of freedom used to build B-spline basis, (one for each group).
 - **boundary.knots**: a list of group-specific boundary knots used to build B-spline basis (one vector for each group) (see [bs](#) for more details).
- time** a numerical vector of time points (x-axis coordinates) or a list of numerical vectors (with as much elements than the number of groups in Groups).
- Groups** a vector indicating the names of the groups belonging to the set of groups involved in MEM_Pol_group for which we want to estimate the AUC (a subset or the entire set of groups involved in the model can be considered). If NULL (default), the AUC for all the groups involved the MEM is calculated.

method	a character scalar indicating the interpolation method to use to estimate the AUC. Options are 'trapezoid' (default), 'lagrange' and 'spline'. In this version, the 'spline' interpolation is implemented with the "not-a-knot" spline boundary conditions.
Averaged	a logical scalar. If TRUE, the function return the normalized AUC (nAUC) computed as the AUC divided by the range of the time calculation. If FALSE (default), the classic AUC is calculated. (See Group_specific_AUC_estimation for more details about calculation)

Value

A numerical vector containing the estimation of the variance of the AUC (or nAUC) for each group defined in the Groups vector.

See Also

[bs](#), [MEM_Polynomial_Group_structure](#)

Examples

```
# Download of data
data("HIV_Simu_Dataset_Delta01_cens")
data <- HIV_Simu_Dataset_Delta01_cens

# Change factors in character vectors
data$id <- as.character(data$id) ; data$Group <- as.character(data$Group)

# Example 1: We consider the variable \code{MEM_Pol_Group} as the output of our function
# \link[AUCcomparison]{MEM_Polynomial_Group_structure}
MEM_estimation <- MEM_Polynomial_Group_structure(y=data$VL,x=data$time,Group=data$Group,
                                                Id=data$id,Cens=data$cens)

time_group1 <- unique(data$time[which(data$Group == "Group1")])
time_group2 <- unique(data$time[which(data$Group == "Group2")])

Var_AUC_estimation <- Group_specific_Var_AUC_estimation(MEM_Pol_group=MEM_estimation,
                                                        time=list(time_group1,time_group2))

# Example 2: We consider results of MEM estimation from another source.
# We have to give build the variable 'MEM_Pol_group' with the good structure
# We build the variable 'MEM_Pol_group.1' with the results of MEM estimation obtained for 2 groups
# Generation of random matrix
Covariance_Matrix_1 <- matrix(rnorm(7*7,mean=0,sd=0.01),ncol=7,nrow=7)
# Transform the matrix into symmetric one
Covariance_Matrix_1 <- Covariance_Matrix_1 %*% t(Covariance_Matrix_1)
MEM_Pol_group.1 <- list(Model_estimation=Covariance_Matrix_1,
                       Model_features=list(Groups=c("Group1","Group2"),
                                           Marginal.dyn.feature=list(dynamic.type="polynomial",
                                                                    intercept=c(global.intercept=TRUE,
                                                                    group.intercept1=FALSE,group.intercept2=FALSE),
                                                                    polynomial.degree=c(3,3))))

Var_AUC_estimation_G1.1 <- Group_specific_Var_AUC_estimation(MEM_Pol_group.1,
                                                            time=time_group1,Groups=c("Group1"))

# We build the variable 'MEM_Pol_group.2' with the results of MEM estimation obtained only for the
```



```

# group of interest (extraction)
# Generation of random matrix
Covariance_Matrix_2 <- matrix(rnorm(4*4,mean=0,sd=0.01),ncol=4,nrow=4)
# Transform the matrix into a symmetric one
Covariance_Matrix_2 <- Covariance_Matrix_2 %*% t(Covariance_Matrix_2)
MEM_Pol_group.2 <- list(Model_estimation=Covariance_Matrix_2,
                        Model_features=list(Groups=c("Group1"),
                                             Marginal.dyn.feature=list(dynamic.type="polynomial",
                                                                        intercept=c(global.intercept=TRUE,
                                                                        group.intercept1=FALSE),
                                                                        polynomial.degree=c(3))))

Var_AUC_estimation_G1.2 <- Group_specific_Var_AUC_estimation(MEM_Pol_group=MEM_Pol_group.2,
                                                            time=time_group1)

```

Group_specific_Var_Delta_AUC_estimation

Variance of the Difference of AUC of Two Group-Specific Polynomial Marginal Dynamics

Description

This function calculates the variance of the difference of area under the curve of two marginal dynamics modeled by group-structured polynomials or B-spline curve in Mixed-Effects models.

Usage

```

Group_specific_Var_Delta_AUC_estimation(
  MEM_Pol_group, Group1, Group2,
  time.G1, time.G2, common.interval = TRUE,
  method = "trapezoid", Group.dependence = TRUE,
  Averaged = FALSE
)

```

Arguments

MEM_Pol_group A list with similar structure than the output provided by the function [MEM_Polynomial_Group_struct](#).
A list containing:

- **Model_estimation**: a list containing at least 2 elements:
 1. the vector of the marginal (fixed) parameters estimates (at least for the groups whose AUC is to estimate), labeled '*beta*'.
 2. the variance-covariance matrix of these parameters, labeled '*varFix*' (see [MEM_Polynomial_Group_structure](#) for details about the parameter order).
- **Model_features**: a list of at least 2 elements:
 1. **Groups**: a vector indicating the names of the groups whose fixed parameters are given.
 2. **Marginal.dyn.feature**: a list summarizing the features of the marginal dynamics defined in the model:

- `dynamic.type`: a character scalar indicating the chosen type of marginal dynamics. Options are 'polynomial' or 'spline'.
- `intercept`: a logical vector summarizing choices about global and group-specific intercepts (Number of groups + 1) elements whose elements are named as ('global.intercept', 'group.intercept1', ..., 'group.interceptG') if G Groups are defined in `MEM_Po1_group`. For each element of the vector, if TRUE, the considered intercept is considered as included in the model (see *Examples*).

If `dynamic.type` is defined as 'polynomial':

- `polynomial.degree`: an integer vector indicating the degree of polynomial functions, one value for each group.

If `dynamic.type` is defined as 'spline':

- `spline.degree`: an integer vector indicating the degree of B-spline curves, one for each group.
- `knots`: a list of group-specific internal knots used to build B-spline basis (one numerical vector for each group) (see [bs](#) for more details).
- `df`: a numerical vector of group-specific degrees of freedom used to build B-spline basis, (one for each group).
- `boundary.knots`: a list of group-specific boundary knots used to build B-spline basis (one vector for each group) (see [bs](#) for more details).

Group1	a character scalar indicating the name of the first group whose marginal dynamics must be considered. This group name must belong to the set of groups involved in the MEM (see Groups vector in <code>MEM_Po1_group</code>).
Group2	a character scalar indicating the name of the second group whose marginal dynamics must be considered. This group name must belong to the set of groups involved in the MEM (see Groups vector in <code>MEM_Po1_group</code>).
time.G1	a numerical vector of time points (x-axis coordinates) to use for the variance of the Group1 AUC calculation.
time.G2	a numerical vector of time points (x-axis coordinates) to use for the variance of the Group2 AUC calculation.
common.interval	a logical scalar. If FALSE, the variance of difference of AUC is calculated as the variance of the difference of AUCs where the AUC of each group is calculated on its specific interval of time. If TRUE (default), the variance is estimated on a common interval of time defined as the intersect of the two group-specific interval (see Group_specific_Delta_AUC_estimation for more details about calculation.).
method	a character scalar indicating the interpolation method to use to estimate the AUC. Options are 'trapezoid' (default), 'lagrange' and 'spline'. In this version, the 'spline' interpolation is implemented with "not-a-knot" spline boundary conditions.
Group.dependence	a logical scalar indicating whether the two groups, whose the difference of AUC (ΔAUC) is studied, are considered as dependent. By default, this variable is defined as TRUE.
Averaged	a logical scalar. If TRUE, the function return the difference of normalized AUC (nAUC) where nAUC is computed as the AUC divided by the range of time of calculation. If FALSE (default), the classic AUC is calculated.

Value

A numerical scalar corresponding to the variance of the difference of AUC (ΔAUC) between the Group1 and the Group2. If the two groups are considered as dependent (Group.dependence=TRUE), the variance of ΔAUC is calculated as $Var(AUC_1) + Var(AUC_2) - 2Cov(AUC_1, AUC_2)$. Otherwise, only the sum of the two variance is used.

See Also

[bs](#), [Group_specific_Var_AUC_estimation](#), [MEM_Polynomial_Group_structure](#)

Examples

```
# Download of data
data("HIV_Simu_Dataset_Delta01_cens")
data <- HIV_Simu_Dataset_Delta01_cens

# Change factors in character vectors
data$id <- as.character(data$id) ; data$Group <- as.character(data$Group)

# Example 1: We consider the variable \code{MEM_Pol_Group} as the output of our function
# \link[AUCcomparison]{MEM_Polynomial_Group_structure}
MEM_estimation <- MEM_Polynomial_Group_structure(y=data$VL,x=data$time,Group=data$Group,
                                                Id=data$id,Cens=data$cens)

time_group1 <- unique(data$time[which(data$Group=="Group1")])
time_group2 <- unique(data$time[which(data$Group=="Group2")])

Var_Delta_AUC_estimation <- Group_specific_Var_Delta_AUC_estimation(
  MEM_Pol_group=MEM_estimation,
  Group1="Group1",Group2="Group2",
  time.G1=time_group1,time.G2=time_group2)

# Example 2: We consider results of MEM estimation from another source.
# We have to give build the variable 'MEM_Pol_group' with the good structure
# We build the variable 'MEM_Pol_group.1' with the results of MEM estimation obtained for 2 groups
# Generation of random matrix
Covariance_Matrix_1 <- matrix(rnorm(7*7,mean=0,sd=0.01),ncol=7,nrow=7)
# Transform the matrix into symmetric one
Covariance_Matrix_1 <- Covariance_Matrix_1 %*% t(Covariance_Matrix_1)
MEM_Pol_group.1 <- list(Model_estimation=Covariance_Matrix_1,
  Model_features=list(Groups=c("Group1","Group2"),
    Marginal.dyn.feature=list(dynamic.type="polynomial",
      intercept=c(global.intercept=TRUE,
        group.intercept1=FALSE,group.intercept2=FALSE),
        polynomial.degree=c(3,3))))

Var_Delta_AUC_estimation_2 <- Group_specific_Var_Delta_AUC_estimation(
  MEM_Pol_group=MEM_Pol_group.1,
  Group1="Group1",Group2="Group2",
  time.G1=time_group1,time.G2=time_group2)
```

HIV_Simu_Dataset_Delta0

HIV RNA Load Simulated Data for Delta AUC = 0

Description

This dataset gives simulated longitudinal log-transformed HIV RNA load dynamics for two distinct groups of patients who interrupted their antiretroviral treatment at baseline (time=0). The group 'Group1' was simulated to mimic a control group while the group 'Group2' gathers patients assigned to a treatment group. Each group accounts for 20 patients and each of them were followed for 24 weeks with one measure every week. This dataset was simulated to obtain a similar area under the mean HIV RNA load curves, despite the difference of the mean dynamics shape.

Usage

HIV_Simu_Dataset_Delta0

Format

A data frame with 4 columns and 1000 rows:

id	The subject id.
time	Time of HIV RNA load measurements.
Group	Treatment group.
VL	The HIV RNA load measures.

Details

A dataset containing simulated data used in the accompanying paper to this package.

Examples

HIV_Simu_Dataset_Delta0

HIV_Simu_Dataset_Delta01

HIV RNA Load Simulated Data for Delta AUC = -0.1

Description

This dataset gives simulated longitudinal log-transformed HIV RNA load dynamics for two distinct groups of patients who interrupted their antiretroviral treatment at baseline (time=0). The group 'Group1' was simulated to mimic a control group while the group 'Group2' gathers patients assigned to a treatment group. Each group accounts for 20 patients and each of them were followed for 24 weeks with one measure every week. This dataset was simulated to obtain a difference of area under the mean HIV RNA load curves between the two groups (AUC2 - AUC1) equals to -0.1 log10 cp/ml.

Usage

```
HIV_Simu_Dataset_Delta01
```

Format

A data frame with 4 columns and 1000 rows:

id	The subject id.
time	Time of HIV RNA load measurements.
Group	Treatment group.
VL	The HIV RNA load measures.

Details

A dataset containing simulated data used in the accompanying paper to this package.

Examples

```
HIV_Simu_Dataset_Delta01
```

```
HIV_Simu_Dataset_Delta01_cens
```

HIV RNA Load Simulated Data for Delta AUC = -0.1 With Left-censored data

Description

This dataset gives simulated longitudinal log-transformed HIV RNA load dynamics for two distinct groups of patients who interrupted their antiretroviral treatment at baseline (time=0). The group 'Group1' was simulated to mimic a control group while the group 'Group2' gathers patients assigned to a treatment group. Each group accounts for 20 patients and each of them were followed for 24 weeks with one measure every week. HIV RNA load values are left-censored due to a limit of detection (LOD), considering $\log_{10}(50)$ as threshold value. This dataset was simulated to obtain a difference of area under the mean HIV RNA load curves between the two groups (AUC2 - AUC1) equals to $-0.1 \log_{10}$ cp/ml.

Usage

```
HIV_Simu_Dataset_Delta01_cens
```

Format

A data frame with 4 columns and 1000 rows:

id	The subject id.
time	Time of HIV RNA load measurements.
Group	Treatment group.
VL	The HIV RNA load measures. Measures are either equals to their true value (if $VL > LOD$) or to the LOD value
Cens	The indicator of left-censoring: 1 if $VL_{true} \leq LOD$ and 0 otherwise.

Details

A dataset containing simulated data used in the accompanying paper to this package.

Examples

HIV_Simu_Dataset_Delta01_cens

HIV_Simu_Dataset_Delta025

HIV RNA Load Simulated Data for Delta AUC = -0.25

Description

This dataset gives simulated longitudinal log-transformed HIV RNA load dynamics for two distinct groups of patients who interrupted their antiretroviral treatment at baseline (time=0). The group 'Group1' was simulated to mimic a control group while the group 'Group2' gathers patients assigned to a treatment group. Each group accounts for 20 patients and each of them were followed for 24 weeks with one measure every week. This dataset was simulated to obtain a difference of area under the mean HIV RNA load curves between the two groups (AUC2 - AUC1) equals to -0.25 log10 cp/ml.

Usage

HIV_Simu_Dataset_Delta025

Format

A data frame with 4 columns and 1000 rows:

id	The subject id.
time	Time of HIV RNA load measurements.
Group	Treatment group.
VL	The HIV RNA load measures.

Details

A dataset containing simulated data used in the accompanying paper to this package.

Examples

HIV_Simu_Dataset_Delta025

HIV_Simu_Dataset_Delta025_cens

HIV RNA Load Simulated Data for Delta AUC = -0.25 With Left-censored data

Description

This dataset gives simulated longitudinal log-transformed HIV RNA load dynamics for two distinct groups of patients who interrupted their antiretroviral treatment at baseline (time=0). The group 'Group1' was simulated to mimic a control group while the group 'Group2' gathers patients assigned to a treatment group. Each group accounts for 20 patients and each of them were followed for 24 weeks with one measure every week. HIV RNA load values are left-censored due to a limit of detection (LOD), considering log10(50) as threshold value. This dataset was simulated to obtain a difference of area under the mean HIV RNA load curves between the two groups (AUC2 - AUC1) equals to -0.25 log10 cp/ml.

Usage

HIV_Simu_Dataset_Delta025_cens

Format

A data frame with 4 columns and 1000 rows:

id	The subject id.
time	Time of HIV RNA load measurements.
Group	Treatment group.
VL	The HIV RNA load measures. Measures are either equals to their true value (if VL > LOD) or to the LOD value
Cens	The indicator of left-censoring: 1 if VLtrue <= LOD and 0 otherwise.

Details

A dataset containing simulated data used in the accompanying paper to this package.

Examples

HIV_Simu_Dataset_Delta025_cens

HIV_Simu_Dataset_Delta0_cens	<i>HIV RNA Load Simulated Data for Delta AUC = 0 With Left-censored data</i>
------------------------------	--

Description

This dataset gives simulated longitudinal log-transformed HIV RNA load dynamics for two distinct groups of patients who interrupted their antiretroviral treatment at baseline (time=0). The group 'Group1' was simulated to mimic a control group while the group 'Group2' gathers patients assigned to a treatment group. Each group accounts for 20 patients and each of them were followed for 24 weeks with one measure every week. HIV RNA load values are left-censored due to a limit of detection (LOD), considering log10(50) as threshold value. This dataset was simulated to obtain a similar area under the mean HIV RNA load curves, despite the difference of the mean dynamics shape (without considering the left-censorship).

Usage

HIV_Simu_Dataset_Delta0_cens

Format

A data frame with 4 columns and 1000 rows:

id	The subject id.
time	Time of HIV RNA load measurements.
Group	Treatment group.
VL	The HIV RNA load measures. Measures are either equals to their true value (if $VL > LOD$) or to the LOD value
Cens	The indicator of left-censoring: 1 if $VL_{true} \leq LOD$ and 0 otherwise.

Details

A dataset containing simulated data used in the accompanying paper to this package.

Examples

```
HIV_Simu_Dataset_Delta0_cens
```

```
MEM_Polynomial_Group_structure
```

Polynomial Mixed-Effects Models with Censored and Group-Structured Responses

Description

This function fits a mixed-effects model (MEM) to potentially censored data structured by group when marginal and individual dynamics are described either by polynomials or B-spline curves.

Usage

```
MEM_Polynomial_Group_structure(
  y,x = NULL,Group = NULL,Id = NULL,Cens = NULL,
  marginal_dyn_type = "polynomial",ind_dyn_type = "polynomial",
  global_intercept = TRUE,group_intercept = FALSE,
  degree_group = 3,Adaptive = "none",knotnumcrit = "AIC",
  min_knots_group = 2,max_knots_group = 2,
  knots_group = NULL,df_group = NULL,Boundary.knots_group = NULL,
  ind_intercept = FALSE,degree_ind = 3,
  min_knots_ind = 2,max_knots_ind = 2,
  same_base_group_ind = FALSE,knots_ind = NULL,
  df_ind = NULL,Boundary.knots_ind = NULL,
  ...
)
```

Arguments

y observed responses described either as a data frame containing at least a column named 'y' and possibly the columns 'x', 'Group', 'Id' and 'Cens' (among others), or as a vector of numerical values.

x	a numerical vector representing the x-axis coordinates corresponding to the observed responses (e.g. time of observations) which can be defined if y is a vector or a data frame without x column. By default, this variable is defined as NULL.
Group	a vector of group indicator for each observed responses which can be defined if y is a vector or a data frame without Group column. If this variable is defined as NULL (default) and y does not contain group information, all observed data are assumed to belong to the same group.
Id	a vector of individual ID for each observed responses which can be defined if y is a vector or a data frame without 'Id' column. By default, this variable is defined as NULL.
Cens	a vector of censoring indicator (if y >= ytrue, then Cens == 1). If this variable is defined as NULL (default) and y does not contain 'Cens' column, observed data are assumed as uncensored.
marginal_dyn_type	a character variable indicating the type of marginal dynamics. Options are 'polynomial' (default) and 'spline'.
ind_dyn_type	a character variable indicating the type of individual dynamics (random effects). Options are 'polynomial' (default) or 'spline'.
global_intercept	a logical scalar. If TRUE (default) a global intercept (no group-specific) is included in the marginal dynamics.
group_intercept	a logical scalar (same option for all groups) or vector. For each group, if TRUE, a group-specific intercept is included in the marginal dynamics. By default, this variable is defined as FALSE
degree_group	an integer scalar (same option for all groups) or vector. The variable indicates for each group either the degree of polynomial functions or spline curves describing marginal dynamics. By default, the variable is fixed at 3.
Adaptive	an optional character variable whether B-spline curves are built with internal knot positions optimally estimated according to data (see Optimal_knot_research and @details for more details). Options are 'none' (default), 'group', 'individual', and 'both'.
knotnumcrit	an optional character variable indicating the criterion to be used for determining the number of internal knots in B-splines. See @details for more details.
min_knots_group	an optional integer scalar indicating the minimum number of internal knots to consider in the research of optimal knots for marginal dynamics. By default, this variable fixed at 2. See @details for more details.
max_knots_group	an optional integer scalar indicating the maximum number of internal knots to consider in the research of optimal knots for marginal dynamics. By default, this variable fixed at 2. See @details for more details.
knots_group	a numerical vector (same option for all groups) or a list of either numerical vectors or NULL (one for each group) indicating the internal knots for group-specific B-spline curves. By default, this variable is defined as NULL (see bs and @details for more details).
df_group	an integer scalar (same option for all groups) or vector indicating the degrees of freedom to consider to build marginal B-spline curves. One can specify df_group rather than knots_group (see bs for more details). By default, this variable is defined as NULL. See @details for more details.

Boundary.knots_group	a numerical vector (same option for all groups) or a list of either numerical vectors or NULL (one for each group) indicating the boundary knots for group-specific B-spline curves. By default, this variable is defined as NULL (see bs and @details for more details).
ind_intercept	a logical scalar. If TRUE, an intercept is included in the individual dynamics (random effects). By default, this variable is defined as FALSE.
degree_ind	an integer scalar indicating either the degree of the polynomial functions or the B-spline curves describing individual dynamics. By default, this variable is fixed at 2.
min_knots_ind	an optional interger scalar indicating the minimum number of internal knots to consider in the research of optimal knots for individual dynamics. By default, this variable is fixed at 2. See @details for more details.
max_knots_ind	an optional interger scalar indicating the maximum number of internal knots to consider in the research of optimal knots for individual dynamics. By default, this variable is fixed at 2. See @details for more details.
same_base_group_ind	an optional logical scalar indicating whether or not the same B-spline basis must be considered in group-specific and individual dynamics. If TRUE, each individual B-spline basis will be build as the corresponding group-specific B-spline basis evaluated at the individual predictor variable. By default, this variable is defined as FALSE. See @details for more details.
knots_ind	a numerical vector (same option for all individuals) or a list of numerical values or NULL indicating the internal knots for individual-specific B-spline curves. If this variable is defined as a list, internal knots can either be defined individually (one vector or NULL for each Id value), equivalent for each individual belong to the same group (one vector or NULL for each Group), or equivalent for each individual (one vector or NULL). By default, this variable is defined as NULL (see bs and @details for more details).
df_ind	an integer scalar (same option for all individuals) or vector indicating the degrees of freedom to consier to build individual B-splines curves. This variable can be choosen different for each individual or equivalent for each individual belonging to the same group (one value for each group). By default, this variable is defined as NULL(see bs and @details for more details).
Boundary.knots_ind	a numerical vector indicating the boundary knots for individual-specific B-spline curves. By default, this variable is defined as NULL (see bs and @details for more details).
...	Further arguments to be passed (see lmec for more details).

Details

The variable `adaptive` can take 4 differents values:

- 'none' (default): internal knot positions defining B-spline curves in the model are not optimally estimated according to data, whether for fixed or random effects. Knots are then defined by users.
- 'group': only internal knot positions defining B-spline curves in fixed effects are optimally estimated, even if random effects involve B-splines.
- 'individual': only internal knot positions defining B-spline curves in random effects are optimally estimated, even if fixed effects involve B-splines.

- 'both': internal knot positions defining B-spline curves for both fixed and random effects are optimally estimated.

At group level (fixed effects),

- the variables Adaptive, knotnumcrit, min_knots_group, max_knots_group, knots_group, df_group and Boundary.knots_group will be used only if the variable marginal_dyn_type has been chosen as 'spline'.
 - among them, the variables knotnumcrit, min_knots_group and max_knots_group will be used only if adaptive knots are requested (the variable Adaptive chosen as 'group' or 'both').
 - the variables knots_group, df_group and Boundary.knots_group will be used without adaptive knots.

Similarly, at individual level (random effects),

- the variables Adaptive, knotnumcrit, min_knots_ind, max_knots_ind, knots_ind, df_ind and Boundary.knots_ind will be used only if the variable ind_dyn_type has been chosen as 'spline'.
 - among them, the variables knotnumcrit, min_knots_ind and max_knots_ind will be used only if adaptive knots are requested (the variable Adaptive chosen as 'individual' or 'both').
 - the variables knots_ind, df_ind and Boundary.knots_ind will be used without adaptive knots.
- the variable same_base_group_ind can be used only if both variables marginal_dyn_type and ind_dyn_type are chosen as 'spline'.

The Mixed-Effects model describing the outcome of interest Y_{ij,g_i} of the subject i in the group g_i at the j th time point (x-axis) is given by

$$Y_{ij,g_i} = \gamma_0 + \sum_{g=1}^G 1_{g_i=g} \times F_g(t_{ij,g}) + h_i(t_{ij,g_i}) + \varepsilon_{ij}$$

where γ_0 is the global (non group-specific) intercept and the functions F_g and h_i are the non-linear smooth functions describing respectively the marginal group-specific dynamics and the individual dynamics (random effects). Through this function, the group-specific functions are defined as following, where β_0^g is the optional group_intercept:

$$F_g(t_{ij,g}) = \beta_0^g + \sum_{k=1}^{K_g} \beta_k^g f_g^k(t_{ij,g})$$

If marginal_dyn_type is defined as 'polynomial', $K_g = \text{degree_group}$ and $f_g^k(t_{ij,g}) = t_{ij,g}^k$ and if marginal_dyn_type is defined as 'spline', $K_g = \text{df_group}$ (see [bs](#) for more details) and $f_g^k(t_{ij,g})$ is the k th group-specific spline basis. Similarly, the individual dynamics are described by the following functions, with b_{0i} as optional ind_intercept:

$$h_i(t_{ij,g}) = b_{0i} + \sum_{k=1}^{K_i} b_{ki} \Psi_k^i(t_{ij,g})$$

If ind_dyn_type is defined as 'polynomial', $K_i = \text{degree_ind}$ and $\Psi_k^i(t_{ij,g}) = t_{ij,g}^k$ and if ind_dyn_type is defined as 'spline', $K_i = \text{df_ind}$ (see [bs](#) for more details) and $\Psi_k^i(t_{ij,g})$ is the k th individual spline basis.

Value

A list containing:

- `Model_estimation`: a list containing the results of the model estimation provided by the function `lmec`. In this list, the vector of fixed parameters called `beta`, the parameters are returned in the following order: $\beta = (\gamma_0, \beta_0^1, \dots, \beta_{K_1}^1, \dots, \beta_0^g, \dots, \beta_{K_g}^g, \dots, \beta_0^G, \dots, \beta_{K_G}^G)$.
- `Model_features`: a list of 3 elements:

1. `Groups`: a vector indicating the names of the groups involved in the model.
2. `Marginal.dyn.feature`: a list summarizing the features of the marginal dynamics defined in the model (through input arguments):
 - `dynamic.type`: a character scalar indicating the chosen type of marginal dynamics.
 - `intercept`: a logical vector summarizing choices about global and group-specific intercepts (Number of groups + 1 values).

For 'polynomial' marginal dynamics:

- `polynomial.degree`: an integer vector indicating the degree of polynomial functions.

For 'spline' marginal dynamics:

- `spline.degree`: an integer vector indicating the degree of B-spline curves.
- `adaptive.splines`: a logical scalar indicating whether or not adaptive internal knots have been considered.
- `knots`: a list of group-specific internal knots used to build B-spline basis. If the degrees of freedom were equals to the spline degrees, then NULL.
- `df`: a numerical vector of group-specific degrees of freedom used to build B-spline basis.
- `boundary.knots`: a list of group-specific boundary knots used to build B-spline basis.

3. `Individual.dyn.feature`: a list summarizing the features of the individual dynamics defined in the model (through input arguments)

- `dynamic.type`: a character scalar indicating the chosen type of individual dynamics.
- `intercept`: a logical scalar indicating whether a random intercept was included in the model.

For 'polynomial' individual dynamics:

- `polynomial.degree`: an integer scalar indicating the degree of polynomial functions.

For 'spline' marginal dynamics:

- `spline.degree`: an integer scalar indicating the degree of B-spline curves
- `adaptive.splines`: a logical scalar indicating whether or not adaptive internal knots have been considered.
- `knots`: a data frame of individually estimated internal knots (if Adaptive chosen as 'individual' or 'both'), or a list of chosen individual internal knots.
- `df`: a numerical vector of individual degrees of freedom.
- `boundary.knots`: a numerical vector of individual boundary knots.

See Also

[bs](#), [lmec](#), [Optimal_knot_research](#)

Examples

```
# Download of data
data("HIV_Simu_Dataset_Delta01_cens")
data <- HIV_Simu_Dataset_Delta01_cens

# Change factors in character vectors
data$id <- as.character(data$id) ; data$Group <- as.character(data$Group)

# We call the function considering the variable 'y' as a vector (we need to specify the groups )
MEM_Pol_Group <- MEM_Polynomial_Group_structure(y=data$VL,x=data$time,Group=data$Group,
                                              Id=data$id,Cens=data$cens)

# We call the function considering the variable 'y' as a data frame
colnames(data)[4] <- "y"
MEM_Pol_Group <- MEM_Polynomial_Group_structure(y=data,x=data$time,Cens=data$cens,Id=data$id)
```

Optimal_knot_research *Optimal Research B-Spline Internal Knots*

Description

This function uses AIC criterion to estimate optimal position and number of knots by fitting free-knot splines to data with one independent and one dependent variable. It is assumed that knots are estimated for least-squares splines with no penalty using genetic algorithm.

Usage

```
Optimal_knot_research(
  data,degree = 3,minknot = 2,maxknot = 2,
  criteria = "AIC",...
)
```

Arguments

<code>data</code>	a data frame containing the independent variable (y) and the dependent one (x).
<code>degree</code>	an integer scalar indicating the degree of the spline it. By default, this value is fixed at 3
<code>minknot</code>	an integer scalar indicating the minimum number of knots to consider. By default, this variable is fixed at 2
<code>maxknot</code>	an integer scalar indicating the maximum number of knots to consider. By default, this variable is fixed at 2.
<code>criteria</code>	a character variable indicating the criterion to be used for determining the number and the positions of knots. Choices are "AIC" for Akaike information criterion (by default), "AICc" for corrected AIC, "BIC" for Bayesian information criterion, "adjAIC" for an adjusted version of AIC, "GCV" for generalized cross-validation and "adjGCV" for an adjusted version of GCV.
<code>...</code>	Further arguments to be passed (see freenotfit for more details).

Value

A numerical vector of optimal knots whose number can varied from `minknot` to `maxknot`

See Also

[freeknotfit](#) for more details about the method.

Stat_test_Delta_AUC_Group_Specific

T-statistic of the Difference of AUC of Two Group-Specific Polynomial Marginal Dynamics

Description

This function performs the t-test evaluating whether the difference of area under the curve of two marginal dynamics, modeled by group-structured polynomials or B-spline curve in Mixed-Effects model, is null.

Usage

```
Stat_test_Delta_AUC_Group_Specific(
  MEM_Pol_group, Group1, Group2,
  time.G1, time.G2, common.interval = TRUE,
  method = "trapezoid", Group.dependence = TRUE,
  Averaged = FALSE, conf_level = 0.95,
  alternative = "two.sided"
)
```

Arguments

`MEM_Pol_group` A list with similar structure than the output provided by the function [MEM_Polynomial_Group_struct](#)
A list containing:

- `Model_estimation`: a list containing at least 2 elements:
 1. the vector of the marginal (fixed) parameters estimates (at least for the groups whose AUC is to estimate), labeled `'beta'`.
 2. the variance-covariance matrix of these parameters, labeled `'varFix'` (see [MEM_Polynomial_Group_structure](#) for details about the parameter order).
- `Model_features`: a list of at least 2 elements:
 1. `Groups`: a vector indicating the names of the groups whose fixed parameters are given.
 2. `Marginal.dyn.feature`: a list summarizing the features of the marginal dynamics defined in the model:
 - `dynamic.type`: a character scalar indicating the chosen type of marginal dynamics. Options are `'polynomial'` or `'spline'`.
 - `intercept`: a logical vector summarizing choices about global and group-specific intercepts (Number of groups + 1) elements whose elements are named as (`'global.intercept'`, `'group.intercept1'`, ..., `'group.interceptG'`) if G Groups are defined in `MEM_Pol_group`. For each element of the vector, if TRUE, the considered intercept is considered as included in the model (see *Examples*).

	<p>If <code>dynamic.type</code> is defined as 'polynomial':</p> <ul style="list-style-type: none"> – <code>polynomial.degree</code>: an integer vector indicating the degree of polynomial functions, one value for each group. <p>If <code>dynamic.type</code> is defined as 'spline':</p> <ul style="list-style-type: none"> – <code>spline.degree</code>: an integer vector indicating the degree of B-spline curves, one for each group. – <code>knots</code>: a list of group-specific internal knots used to build B-spline basis (one numerical vector for each group) (see bs for more details). – <code>df</code>: a numerical vector of group-specific degrees of freedom used to build B-spline basis, (one for each group). – <code>boundary.knots</code>: a list of group-specific boundary knots used to build B-spline basis (one vector for each group) (see bs for more details).
Group1	a character scalar indicating the name of the first group whose marginal dynamics must be considered. This group name must belong to the set of groups involved in the MEM (see Groups vector in MEM_Pol_group).
Group2	a character scalar indicating the name of the second group whose marginal dynamics must be considered. This group name must belong to the set of groups involved in the MEM (see Groups vector in MEM_Pol_group).
time.G1	a numerical vector of time points (x-axis coordinates) to use for the Group1 AUC calculation.
time.G2	a numerical vector of time points (x-axis coordinates) to use for the Group2 AUC calculation.
common.interval	a logical scalar. If FALSE, the difference of AUC is calculated as the difference of AUCs where the AUC of each group is calculated on its specific interval of time. If TRUE (default), the difference of AUC is estimated on a common interval of time defined as the intersect of the two group-specific interval (see Group_specific_Delta_AUC_estimation for more details about calculation).
method	a character scalar indicating the interpolation method to use to estimate the AUC. Options are 'trapezoid' (default), 'lagrange' and 'spline'. In this version, the 'spline' interpolation is implemented with "not-a-knot" spline boundary conditions.
Group.dependence	a logical scalar indicating whether the two groups, whose the difference of AUC (Δ AUC) is studied, are considered as dependent. By default, this variable is defined as TRUE.
Averaged	a logical scalar. If TRUE, the function return the difference of normalized AUC (nAUC) where nAUC is computed as the AUC divided by the range of time of calculation. If FALSE (default), the classic AUC is calculated.
conf_level	a numerical value (between 0 and 1) indicating the confidence level of the interval. By default, this variable is fixed at 0.95
alternative	a character scalar specifying the alternative hypothesis. Options are 'two.sided' (default), 'greater' or 'less'.

Value

A list containing:

- Tstat: the value of the t-statistic.

- Pvalue: the P-value.
- Conf.int: the confidence interval.
- Delta_AUC: the estimated value of the difference of AUC between the two groups (nAUC2 - nAUC1) (see [Group_specific_Delta_AUC_estimation](#) for more details).
- AUCs: the estimated values of the Group-specific AUC (AUC1 and AUC 2) (see [Group_specific_AUC_estimation](#) for more details).

See Also

[MEM_Polynomial_Group_structure](#), [Group_specific_Delta_AUC_estimation](#),
[Group_specific_AUC_estimation](#)

Examples

```
# Download of data
data("HIV_Simu_Dataset_Delta01_cens")
data <- HIV_Simu_Dataset_Delta01_cens

# Change factors in character vectors
data$id <- as.character(data$id) ; data$Group <- as.character(data$Group)

MEM_estimation <- MEM_Polynomial_Group_structure(y=data$VL,x=data$time,Group=data$Group,
                                                Id=data$id,Cens=data$cens)

time_group1 <- unique(data$time[which(data$Group=="Group1")])
time_group2 <- unique(data$time[which(data$Group=="Group2")])
Test <- Stat_test_Delta_AUC_Group_Specific(MEM_Pol_group=MEM_estimation,
                                           Group1="Group1",Group2="Group2",
                                           time.G1=time_group1,time.G2=time_group2)
```


Index

*Topic **datasets**

- HIV_Simu_Dataset_Delta0, [20](#)
- HIV_Simu_Dataset_Delta01, [20](#)
- HIV_Simu_Dataset_Delta01_cens, [21](#)
- HIV_Simu_Dataset_Delta025, [22](#)
- HIV_Simu_Dataset_Delta025_cens, [22](#)
- HIV_Simu_Dataset_Delta0_cens, [23](#)

- ANOVA_AUC_Group_Specific, [2](#)
- AUC_Lagrange_Cjp_coefficients, [5](#), [8](#)
- AUC_Spline_matrix_A, [6](#), [9](#)
- AUC_Spline_matrix_B, [7](#), [9](#)
- AUC_time_weights_estimation, [5](#), [8](#)

- bartlett.test, [4](#), [5](#)
- bs, [3](#), [10](#), [12](#), [13](#), [15](#), [16](#), [18](#), [19](#), [25–28](#), [31](#)

- freesknotfit, [29](#), [30](#)

- Group_specific_AUC_estimation, [3](#), [9](#), [14](#),
[16](#), [32](#)
- Group_specific_Delta_AUC_estimation,
[11](#), [18](#), [31](#), [32](#)
- Group_specific_Var_AUC_estimation, [14](#),
[19](#)
- Group_specific_Var_Delta_AUC_estimation,
[17](#)

- HIV_Simu_Dataset_Delta0, [20](#)
- HIV_Simu_Dataset_Delta01, [20](#)
- HIV_Simu_Dataset_Delta01_cens, [21](#)
- HIV_Simu_Dataset_Delta025, [22](#)
- HIV_Simu_Dataset_Delta025_cens, [22](#)
- HIV_Simu_Dataset_Delta0_cens, [23](#)

- lmec, [26](#), [28](#)

- MEM_Polynomial_Group_structure, [2](#), [9](#), [10](#),
[12](#), [14–17](#), [19](#), [24](#), [30](#), [32](#)

- Optimal_knot_research, [25](#), [28](#), [29](#)

- p.adjust, [4](#)

- Stat_test_Delta_AUC_Group_Specific, [30](#)