

Package ‘TimeDepFrail’

May 7, 2026

Type Package

Title Time Dependent Shared Frailty Cox Model

Version 0.1.0

Description Fits time-dependent shared frailty Cox model (specifically the adapted Paik et al.'s Model) based on the paper “Centre-Effect on Survival After Bone Marrow Transplantation: Application of Time-Dependent Frailty Models”, by C.M. Wintrebert, H. Putter, A.H. Zwinderman and J.C. van Houwelingen (2004) <[doi:10.1002/bimj.200310051](https://doi.org/10.1002/bimj.200310051)>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

LazyData true

URL <https://alessandragni.github.io/TimeDepFrail/>

NeedsCompilation no

Author Alessandra Ragni [aut, cre] (ORCID: <<https://orcid.org/0000-0002-3647-7340>>),
Giulia Romani [aut],
Chiara Masci [aut] (ORCID: <<https://orcid.org/0000-0002-9208-3194>>)

Maintainer Alessandra Ragni <alessandra.ragni@polimi.it>

Repository CRAN

Date/Publication 2025-04-07 14:20:01 UTC

Contents

AdPaikModel	2
AdPaik_1D	6
bas_hazard	9
coef.AdPaik	10
coefseAdPaik	11
confint.AdPaik	12

data_dropout	13
extractAIC.AdPaik	14
frailty_sd	14
logLik.AdPaik	15
nobs.AdPaik	16
plot.AdPaik	16
plot_bas_hazard	17
plot_frailty_sd	19
plot_post_frailty_est	20
plot_post_frailty_var	22
plot_survivalAdPaik	24
post_frailty_confint	25
post_frailty_est	27
post_frailty_var	28
print.AdPaik	29
print.summary.AdPaik	30
summary.AdPaik	31
survivalAdPaik	31
Index	33

AdPaikModel	<i>Adapted Paik et al.'s Model: Time-Dependent Shared Frailty Cox Model</i>
-------------	---

Description

Function for applying the 'Adapted Paik et al.'s Model', a Cox Model with time-dependent frailty, shared by individuals belonging to the same group/cluster.

To generate time-dependence, the temporal domain is divided into a certain number of intervals. The model log-likelihood function depends on a certain number of parameters and is maximized with respect to all of them, using a reinterpretation of the 'Powell's method in multidimension', that is a multi-dimensional optimization method based on repeated one-dimensional optimization of the log-likelihood function (with respect to one parameter at the time). In this context, the one-dimensional optimization is performed through the 'optimize' R function. For more information about the unknown model parameters, their type and numerosity refer to Details.

Several quantities are estimated at the end of the optimization phase, such as optimal parameters, baseline hazard, frailty dispersion (standard deviation and variance), posterior frailty estimates, with their variance and confidence interval, conditional survival function, Akaike Information Criterion (AIC), ...

Usage

```
AdPaikModel(
  formula,
  data,
  time_axis,
```

```

categories_range_min,
categories_range_max,
n_extrarun = 60,
tol_ll = 1e-06,
tol_optimize = 1e-06,
h_dd = 0.001,
verbose = FALSE,
print_previous_ll_values = c(TRUE, 3)
)

```

Arguments

formula	Formula object having on the left hand side the time-to-event variable, that is the time-instant in which the individual failed. On the right hand side, it has the regressors and the cluster variable.
data	Dataset in which all variables of the formula object must be found and contained. This dataset can also contain other variables, but they will not be considered. It can be either a dataframe or a matrix, but in both cases the name of each column must correspond to the name of the formula variables. It is not necessary to attach it (in case of a data.frame)
time_axis	Temporal domain
categories_range_min	Vector containing the minimum value assumable by each parameter category.
categories_range_max	Vector containing the maximum value assumable by each parameter category.
n_extrarun	Total number of runs (iterations) are obtained summing to the number of parameters and n_extrarun.
tol_ll	Tolerance on the log-likelihood value.
tol_optimize	Internal tolerance for the one-dimensional optimization through 'optimize' R function.
h_dd	Discretization step used for the numerical approximation of the second derivative of the log-likelihood function.
verbose	Logical. If TRUE, detailed progress messages will be printed to the console. Defaults to FALSE.
print_previous_ll_values	If we want to print the previous values of the log-likelihood function. This can be useful for controlling that the optimization procedure is proceeding in a monotone way and it does not oscillate. This argument is composed of two elements: TRUE/FALSE if we want or not to print the previous values and how many values we want to print on the console. Default is (TRUE, 3), so that only the previous 3 values of the log-likelihood are printed.

Details

Two observation needs to made about the time-domain:

- The time domain may coincide with the follow-up or it may be contained in it. Indeed, the left boundary can be greater than the beginning of the follow-up and, for instance, it can coincide with the time-instants in which the events begin to happen; conversely, the right boundary of the two must be the same.
- The partition of the time domain into intervals can be made according to two selected criteria: (1) using an already existent partition of the follow-up (2) using the shape of the baseline hazard function for a time independent model as reference: divide the time-domain according to regions in which it has a peak or a plateau.

The parameters with respect to which the log-likelihood function must be optimized are:

- baseline log-hazard (number of parameters = number of intervals of the time-domain)
- data regressors
- μ_1, ν : parameters of the gamma distribution of α_j (time-independent/constant) (2 parameters)
- γ_k : parameters of the gamma distribution of ϵ_{jk} (time-dependent) (number of parameters = number of intervals) Another model parameter is μ_2 and it is get imposing the constraint that $\mu_1 + \mu_2 = 1$. As it can be notice, some parameters can be grouped into the same category (regressors, baseline log-hazard and so on) and we can easily constraint them assigning each category both a minimum and maximum range. The vector is structured as follows: (baseline log-hazard, regressors, μ_1, ν, γ_k) with dimension (n_intervals, n_regressors, 1, 1, n_intervals).

The output of the model call 'AdPaikModel(...)' is a S3 object of class 'AdPaik', composed of the following quantities:

- formula: formula object provided in input by the user and specifying the relationship between the time-to-event, the covariates of the dataset (regressors) and the cluster variable.
- dataset: matrix of the dataset containing the regressors and the dummy variables of the categorical covariates.
- Regressors: categorical vector of length R, with the name of the regressors. They could be different from the original covariates of the dataset in case of categorical covariates. Indeed, each categorical covariate with n levels needs to be transformed into (n-1) dummy variables and, therefore, (n-1) new regressors.
- NRegressors: number of regressors (R)
- ClusterVariable: name of the variable with respect to which the individuals can be grouped.
- NClusters: number of clusters/centres (also indicated with N).
- ClusterCodes: vector of length equal to the number of clusters, containing the codes of the clusters.
- TimeDomain: vector of the time-domain partition.
- NIntervals: number of intervals of the time-domain, also called with L.
- NObservations: number of observations of the dataset.
- NParameters: number of parameters of the model. It can be computed as: $n_p = 2L + R + 2$.
- ParametersCategories: Numerical vector of length 5, containing the numerosity of each parameter category.
- ParametersRange: S3 object of class 'ParametersRange', containing ParametersRangeMin and ParametersRangeMax, two numerical vectors of length n_p , giving the minimum and the maximum range of each parameter, respectively.

- Loglikelihood: value of the maximized log-likelihood function, at the optimal estimated parameters.
- AIC: 'Akaike Information Criterion': it can be computed as $AIC = 2n_p - 2ll_{optimal}$. It quantifies the loss of information related to the model fitting and output. The smaller, the less the loss of information and the better the model accuracy.
- Status: Logical value. TRUE if the model reaches convergence, FALSE otherwise.
- NRun: Number of runs necessary to reach convergence. If the model does not reach convergence, such number is equal to the maximum number of imposed runs.
- OptimalParameters: numerical vector of length n_p , containing the optimal estimated parameters (the parameters that maximize the log-likelihood function).
- StandardErrorParameters: numerical vector of length n_p , corresponding to the standard error of each estimated parameters.
- ParametersCI: S3 object of class 'ParametersCI', composed of two numerical vector of length equal to n_p : the left and right 95\ interval of each estimated parameter of given level.
- BaselineHazard: numerical vector of length equal to L, containing the baseline hazard step-function.
- FrailtyDispersion: S3 object of class 'FrailtyDispersion', containing two numerical vectors of length equal to L with the standard deviation and the variance of the frailty. numerical vector of length equal to L (i.e. number of intervals of the time-domain), reporting the standard deviation of the frailty.
- PosteriorFrailtyEstimates: S3 object of class 'PFE.AdPaik'. The object of class 'PFE.AdPaik' contains the Posterior Frailty Estimates computed with the procedure indicated in the reference paper and it is composed of three elements:
 - 'alpha': posterior frailty estimates for $\alpha_j, \forall j$. It is a vector of length equal to the number of centres.
 - 'eps': posterior frailty estimates for $\epsilon_{jk}, \forall j, k$. Matrix of dimension (N, L).
 - 'Z': posterior frailty estimates for $Z_{jk} = \alpha_j + \epsilon_{jk}, \forall j, k$. Matrix of dimension (N, L).
- PosteriorFrailtyVariance: S3 object of class 'PFV.AdPaik'. The object of class 'PFV.AdPaik' contains the Posterior Frailty Variances computed as indicated in the reference papaer and it is composed of three elements:
 - 'alphaVar': posterior frailty variance for $\alpha_j, \forall j$. It is a vector of length equal to the number of centres.
 - 'epsVar': posterior frailty variance for $\epsilon_{jk}, \forall j, k$. Matrix of dimension (N, L).
 - 'ZVar': posterior frailty variance for $Z_{jk} = \alpha_j + \epsilon_{jk}, \forall j, k$. Matrix of dimension (N, L).

Value

S3 object of class 'AdPaik', composed of several elements. See Details.

Examples

```
# Consider the 'Academic Dropout dataset'
data(data_dropout)

# Define the variables needed for the model execution
```

```

formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)

# Call the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max, TRUE)

```

AdPaik_1D

One-Dimensional Analysis of log-Likelihood Function

Description

Function for studying the log-likelihood function from the point of view of a single parameter and, therefore, in a single direction. It performs both the optimization of the log-likelihood with respect to this parameter and the evaluation of the log-likelihood in several samples of the same parameter, while the other parameters can assume a constant assigned value or can vary in their range.

Usage

```

AdPaik_1D(
  formula,
  data,
  time_axis,
  index_param_to_vary,
  flag_optimal_params = FALSE,
  optimal_params = NULL,
  categories_range_min,
  categories_range_max,
  n_iter = 5,
  tol_optimize = 1e-06,
  flag_plot = FALSE,
  n_points = 150,
  cex = 0.7,
  cex_max = 0.8,
  color_bg = "black",
  color_max_bg = "red",
  pch = 21
)

```

Arguments

`formula` Formula object indicating the response variable, the covariates and the cluster variable.

<code>data</code>	Dataset in which the variables of the formula object are located.
<code>time_axis</code>	Partitioned time-domain.
<code>index_param_to_vary</code>	Index of the parameter, in the parameter vector, with respect to which the log-likelihood function is maximized in a one-dimensional way. The index <code>s</code> provided to identify the parameter under consideration inside the vector, avoiding providing its name or value.
<code>flag_optimal_params</code>	Are the other parameters extracted from the optimal vector of parameters? If so, the flag should be equal to <code>TRUE</code> . Otherwise, the flag is equal to <code>FALSE</code> .
<code>optimal_params</code>	Vector of optimal parameters, determined through an entire multi-dimensional maximization of the log-likelihood function. The default value (<code>NULL</code>) indicates that no vector is provided and the parameters are randomly extracted in their range.
<code>categories_range_min</code>	Vector containing the minimum value assumed by each parameter category.
<code>categories_range_max</code>	Vector containing the maximum value assumed by each parameter category.
<code>n_iter</code>	Number of times the one-dimensional analysis with respect to the indicated parameter must be executed. Default value is 5. See details for more information.
<code>tol_optimize</code>	Tolerance used in the <code>optimize</code> R function for the one-dimensional optimization of the log-likelihood function.
<code>flag_plot</code>	Logical value for plotting the trend of the log-likelihood function with respect to the parameter under consideration. A plot for each iteration (<code>n_iter</code>) is reported. Defaults to <code>FALSE</code> . Be careful that if the optimal parameters are provided, then the trend may be always the same and therefore it may be sufficient to set <code>n_iter</code> = 1. On the other hand, if optimal parameters are not provided, then it is recommended to impose a higher <code>n_iter</code> .
<code>n_points</code>	Number of internal points in which the log-likelihood function must be evaluated, to plot it.
<code>cex</code>	Dimension of the points in the plot.
<code>cex_max</code>	Dimension of the optimal point in the plot.
<code>color_bg</code>	Color used in the plot for the points.
<code>color_max_bg</code>	Color used for the optimal point in the plot.
<code>pch</code>	Shape to be used for the points.

Details

The one-dimensional analysis of the log-likelihood function can be performed in two ways, with two different aims and results:

- Keeping fixed the other parameters (all the parameters in the vector, except for the one under consideration) to their optimal value (`flag_optimal_params` = `TRUE`), determined through the multi-dimensional optimization. In this way, the optimized value of the parameter coincides with the one get with the general and global approach and, therefore, there is no need to repeat

this procedure several times ($n_iter = 1$). However, this approach is really useful if we want to check the trend the log-likelihood function and to observe if it increases, decreases or is constant.

- Letting the other parameters vary in their range ($flag_optimal_params = FALSE$). The optimized parameter value will always assume a different value, because it depends on the value of the other parameters, and it is suggested to repeat the procedure several times ($n_iter \geq 5$), so that it is possible to identify a precise existence region for such parameter

Value

If the flag for the plot has been activated, the function returns both the plot of the one-dimensional log-likelihood function and a class S3 object. Otherwise, only a S3 object of class 'AdPaik_1D'. This class object is composed of:

- numerical vector of length $@n_iter$ containing the optimal estimated parameter.
- numerical vector of length $@n_iter$ containing the associated one-dimensional optimized log-likelihood value

Examples

```
# Consider the 'Academic Dropout dataset'
data(data_dropout)
# Define the variables needed for the model execution
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10

# Identify a parameter existence range
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0.5, 1 - eps, 1, 10)
index_param_to_vary <- 1
analysis_1D_opt <- AdPaik_1D(formula, data_dropout,
                             time_axis, index_param_to_vary,
                             flag_optimal_params = FALSE,
                             optimal_params = NULL,
                             flag_plot = TRUE,
                             categories_range_min, categories_range_max,
                             n_iter = 5)

# or Study the log-likelihood behaviour
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0.4, 1 - eps, 1, 10)
index_param_to_vary <- 14
# Call the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max, TRUE)
analysis_1D_opt <- AdPaik_1D(formula, data_dropout, time_axis,
                             index_param_to_vary, flag_optimal_params = TRUE,
                             flag_plot = TRUE, optimal_params = result$OptimalParameters,
                             categories_range_min, categories_range_max, n_iter = 1)
```

bas_hazard	<i>Baseline Hazard Step-Function</i>
------------	--------------------------------------

Description

The method computes the baseline hazard step-function in each interval of the time-domain, using the estimated parameters $\phi_k, \forall k$

Usage

```
bas_hazard(object)
```

Arguments

object S3 object of class 'AdPaik' returned by the main model output, that contains all the information for the computation of the frailty standard deviation.

Value

Numerical vector of length equal to the number of intervals of the time-domain, with the value of the baseline hazard step-function.

Examples

```
# Consider the 'Academic Dropout dataset'
data(data_dropout)

# Define the variables needed for the model execution
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)

# Call the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max)

bas_hazard(result)
```

coef.AdPaik

*Extract the Coefficients for the 'Adapted Paik et Al.' Model***Description**

Extracts the optimal β obtained with the time-dependent frailty model proposed in the 'Adapted Paik et al.' framework.

Usage

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

Arguments

object	An S3 object of class AdPaik, returned by the main model function (AdPaikModel). This object contains all the optimal parameter estimates.
...	Additional arguments (ignored).

Details

The coef.AdPaik function extracts the coefficients from the OptimalParameters field in object.

The function validates the structure of object and ensures compatibility with the expected model output. It throws an error if the object is malformed or inconsistent.

Value

A named list containing the coefficients.

Examples

```
# Example using the 'Academic Dropout' dataset
data(data_dropout)

# Define the formula and time axis for the model
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)

# Run the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max, TRUE)

# Extract the coefficients
coef(result)
```

coefseAdPaik	<i>Extracts the Standard Errors of the Coefficients for the 'Adapted Paik et Al.' Model</i>
--------------	---

Description

Extracts the standard errors for β obtained with the time-dependent frailty model proposed in the 'Adapted Paik et al.' framework.

Usage

```
coefseAdPaik(object)
```

Arguments

`object` An S3 object of class `AdPaik`, returned by the main model function (`AdPaikModel`). This object contains all the optimal parameter estimates.

Details

The `se.coef` function extracts the standard errors for the estimated parameters from the `StandardErrorParameters` field in `object`.

The function validates the structure of `object` and ensures compatibility with the expected model output. It throws an error if the object is malformed or inconsistent.

Value

A named list containing the categories of the standard errors for the optimal parameters.

Examples

```
# Example using the 'Academic Dropout' dataset
data(data_dropout)

# Define the formula and time axis for the model
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)

# Run the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max, TRUE)

# Extract the coefficients
coefseAdPaik(result)
```

confint.AdPaik	<i>Extracts the Confidence Intervals for the Coefficients for the 'Adapted Paik et Al.' Model</i>
----------------	---

Description

Extracts the confidence intervals for β obtained with the time-dependent frailty model proposed in the 'Adapted Paik et al.' framework.

Usage

```
## S3 method for class 'AdPaik'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	An S3 object of class AdPaik, returned by the main model function (AdPaikModel). This object contains all the optimal parameter estimates.
parm	A specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. Defaults to NULL, and all parameters are considered. Changing it is not supported for this model. It will be ignored.
level	The confidence level required. Defaults to 0.95.
...	Additional arguments to be passed to other methods.

Details

The `confint.AdPaik` function extracts the standard errors for the beta coefficients from the `ParametersCI` field in `object`.

The function validates the structure of `object` and ensures compatibility with the expected model output. It throws an error if the object is malformed or inconsistent.

Value

A named list containing the categories of the standard errors for the optimal parameters.

Examples

```
# Example using the 'Academic Dropout' dataset
data(data_dropout)

# Define the formula and time axis for the model
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)
```

```
# Run the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max, TRUE)

# Extract the coefficients
confint(result)
```

data_dropout

Dropout Dataset

Description

This dataset is extracted from an administrative database provided by a non-specified university and tracks students enrolled in 2012 over three academic years (or 6 semesters). We are interested in understanding what factors lead to students dropping out. Dropout students with a time-instant in the first semester have been removed, for internal reasons (the university cannot take preventive action to reduce or avoid their withdrawal). The students are followed for at most 3 academic years or, equivalently, 6 semesters (follow-up periods), from the first day of lecture up to the time-instant of withdrawal (i.e. survival event) or the end of the academic year.

Usage

data_dropout

Format

A data frame with 4448 rows and 4 columns:

Gender Categorical covariate (Male or Female).

CFUP Standardized numerical covariate indicating the number of credits \ passed by the students by the end of the first semester.

time_to_event Time (in semesters) at which a student decides to leave the university. \ A value greater than 6.0 indicates the student did not drop out during the follow-up (e.g. 6.1 semesters)

group Categorical variable indicating the student's course of study, with 16 different levels from CosA, CosB, ... , CosP.

Source

Data for demonstration purposes.

extractAIC.AdPaik	<i>Extract AIC for AdPaik Objects</i>
-------------------	---------------------------------------

Description

Computes the AIC for an AdPaik model.

Usage

```
## S3 method for class 'AdPaik'
extractAIC(fit, scale = NULL, k = 2, ...)
```

Arguments

fit	An AdPaik model object.
scale	Changing it is not supported for this model. It will be ignored.
k	Penalty parameter (default is 2 for AIC).
...	Additional arguments (ignored).

Value

A numeric vector with the number of parameters and AIC value.

frailty_sd	<i>Frailty Standard Deviation and Variance for the 'Adapted Paik et Al.'s Model'</i>
------------	--

Description

The function computes both the standard deviation and the variance of the time-dependent frailty of the 'Adapted Paik et al.'s Model'.

Recalling the frailty structure $Z_{jk} = \alpha_j + \epsilon_{jk}$ as being composed by a constant group-dependent term (α_j) and a time and group dependent term (ϵ_{jk}), the frailty variance (and standard deviation) can be computed in two different way:

- Considering only the time-dependent spread of the clusters/groups/centre: $var(Z_{jk}) = \mu_2 * \gamma_k$. In this case, the flag_full should be FALSE and flag_variance should be TRUE.
- Considering both the time-dependent and constant spread of the clusters: $var(Z_{jk}) = \mu_1 * \nu + \mu_2 * \gamma_k$. The new added term only moves upward the other case and the flag_full should be TRUE and flag_variance should be TRUE.

Usage

```
frailty_sd(object, flag_full = TRUE, flag_variance = FALSE)
```

Arguments

object	S3 object of class 'AdPaik' returned by the main model output, that contains all the information for the computation of the frailty standard deviation.
flag_full	A boolean flag indicating whether to get the full standard deviation (TRUE) or only the time-dependent component (FALSE). Default to TRUE.
flag_variance	A boolean flag indicating whether to get the frailty variance (TRUE) or the frailty standard deviation (FALSE). Default to FALSE.

Value

Numerical vector of length equal to the number of intervals of the time-domain, with the value of the frailty standard deviation or variance (either full or only the time-dependent component).

Examples

```
# Consider the 'Academic Dropout dataset'
data(data_dropout)

# Define the variables needed for the model execution
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)

# Call the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max)

frailty_sd(result)
```

logLik.AdPaik

Extract Log-Likelihood for AdPaik Objects

Description

Returns the log-likelihood of the fitted AdPaik model.

Usage

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

Arguments

object	An AdPaik model object.
...	Additional arguments (ignored).

Value

A log-likelihood object with degrees of freedom (df).

nobs.AdPaik	<i>Extract Number of Observations for AdPaik</i>
-------------	--

Description

Returns the number of observations used in the model.

Usage

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

Arguments

object	An AdPaik model object.
...	Additional arguments (ignored).

Value

Integer: Number of observations.

plot.AdPaik	<i>Plots Related to the the 'Adapted Paik et Al.' Model</i>
-------------	---

Description

Plots Related to the the 'Adapted Paik et Al.' Model

Usage

```
## S3 method for class 'AdPaik'
plot(
  x,
  which = c(1, 2),
  captions = c("Plot 1: Baseline Hazard", "Plot 2: Posterior Frailty Estimate"),
  ...
)
```

Arguments

x	An object of class 'AdPaik'.
which	A numeric vector indicating which plots to display. Choices: 1 = Baseline Hazard, 2 = Posterior Frailty Estimate.
captions	A character vector with captions for each plot.
...	Additional arguments to be passed to other methods.

Value

No return value. This function generates plots.

Examples

```
# Import data
data(data_dropout)

# Define the variables needed for the model execution
eps_paik <- 1e-10
categories_range_min <- c(-8, -2, eps_paik, eps_paik, eps_paik)
categories_range_max <- c(-eps_paik, 0.4, 1 - eps_paik, 1, 10)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
formula <- time_to_event ~ Gender + CFUP + cluster(group)

# Call the main model function

result <- AdPaikModel(formula, data_dropout, time_axis, categories_range_min, categories_range_max)

plot(result)
```

plot_bas_hazard *Plot the Baseline Hazard Step-Function*

Description

This function plots the baseline hazard step-function based on the estimated parameters from the Adapted Paik et al.'s model.

Usage

```
plot_bas_hazard(
  result,
  xlim = c(min(result$TimeDomain), max(result$TimeDomain)),
  ylim = c(0, max(result$BaselineHazard)),
  xlab = "Time",
  ylab = "Values",
```

```

    main = "Baseline hazard step-function",
    color = "black",
    pch = 21,
    bg = "black",
    cex_points = 0.7
  )

```

Arguments

<code>result</code>	S3 object of class 'AdPaik', returned by the method call 'AdPaikModel(...)'.
<code>xlim</code>	A numeric vector specifying the x-axis limits. Default is set to the interval min-max of the time-domain.
<code>ylim</code>	A numeric vector specifying the y-axis limits. Default is 0 to the maximum value of the baseline hazard.
<code>xlab, ylab</code>	String giving the x and y axis name. Default values are 'x' and 'y'.
<code>main</code>	Title of the plot. Default title is 'Baseline hazard step-function'.
<code>color</code>	Color used for plotting the horizontal segments of the step-function. Default one is 'black'.
<code>pch</code>	Symbol for marking the boundaries of each segment. Default is a dot (value 21).
<code>bg</code>	Color for the boundary symbols. Default matches the plot color ('black').
<code>cex_points</code>	Size of the boundary symbols. Default is 0.7.

Details

The function plots a horizontal segment for each interval of the time domain, representing the baseline hazard. The boundaries of each segment are marked with colored dots, and subsequent segments are intentionally left unconnected to reflect the discrete nature of the intervals.

Value

Plot of the baseline hazard step-function and value of the function in each interval.

Examples

```

# Import data
data(data_dropout)

# Define the variables needed for the model execution
eps_paik <- 1e-10
categories_range_min <- c(-8, -2, eps_paik, eps_paik, eps_paik)
categories_range_max <- c(-eps_paik, 0.4, 1 - eps_paik, 1, 10)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
formula <- time_to_event ~ Gender + CFUP + cluster(group)

# Call the main model function
result <- AdPaikModel(formula, data_dropout, time_axis, categories_range_min, categories_range_max)

```

```
plot_bas_hazard(result)
```

```
plot_frailty_sd      Plot for the Frailty Standard Deviation or Variance
```

Description

This function generates a plot of either the frailty standard deviation or the frailty variance for the intervals in the time-domain.

Usage

```
plot_frailty_sd(
  result,
  flag_full = TRUE,
  flag_variance = FALSE,
  xlim = c(min(result$TimeDomain), max(result$TimeDomain)),
  ylim = NULL,
  xlab = "Time",
  ylab = "Values",
  main = NULL,
  pch = 21,
  color_bg = "blue",
  cex_points = 0.7
)
```

Arguments

result	An S3 object of class 'AdPaik', returned by the main model call 'AdPaik-Model(...)'.
flag_full	A boolean flag indicating whether to plot the full standard deviation (TRUE) or only the time-dependent one (FALSE). Default is TRUE.
flag_variance	A boolean flag indicating whether to plot the frailty variance (TRUE) or the frailty standard deviation (FALSE). Default is FALSE.
xlim	A numeric vector specifying the range for the x-axis (intervals). If NULL, default is set to the interval min-max of the time-domain.
ylim	A numeric vector specifying the range for the y-axis (intervals). If NULL, default is 0 to the maximum value of the frailty variance/standard deviation.
xlab	A string for the x-axis label. Default is 'Intervals'.
ylab	A string for the y-axis label. Default is 'Values'.
main	A string for the plot title. Default title is 'Frailty Standard Deviation' or 'Frailty Variance' according to the produced plot (flag_variance).
pch	A numeric or character symbol used for plotting the frailty standard deviation values. Default is a dot (21).
color_bg	A string specifying the color used for the symbols. Default is 'blue'.
cex_points	A numeric value indicating the size of the plotting symbols. Default is 0.7.

Details

The plot represents the values of the frailty standard deviation or variance for each time interval (represented by its mid point). It connects these points to illustrate the trend of the chosen metric.

This function supports plotting the full or only time dependent frailty standard deviation or variance retrieved from the main model (contained in the S3 object of class 'AdPaik').

Value

A plot displaying either the frailty standard deviation or variance across the specified intervals.

Examples

```
# Import data
data(data_dropout)

# Define the variables needed for the model execution
eps_paik <- 1e-10
categories_range_min <- c(-8, -2, eps_paik, eps_paik, eps_paik)
categories_range_max <- c(-eps_paik, 0.4, 1 - eps_paik, 1, 10)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
formula <- time_to_event ~ Gender + CFUP + cluster(group)

# Call the main model function

result <- AdPaikModel(formula, data_dropout, time_axis, categories_range_min, categories_range_max)

plot_frailty_sd(result)
```

plot_post_frailty_est *Plot the Posterior Frailty Estimates*

Description

This function plots the posterior frailty estimates for each group in each time interval (represented by its mid point). Each group's estimates are represented by a sequence of points connected by straight lines. The function can plot either the entire posterior frailty estimate or its time-independent and time-dependent components based on user-specified flags.

Usage

```
plot_post_frailty_est(
  result,
  flag_eps = FALSE,
  flag_alpha = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = "Time",
```

```

ylab = "Values",
main = "Posterior frailty estimates",
cex = 0.7,
pch_type = seq(1, length(result$ClusterCodes)),
color_bg = rep("black", length(result$ClusterCodes)),
cex_legend = 0.7,
pos_legend = "topright"
)

```

Arguments

result	S3 object of class 'AdPaik', returned by the method call 'AdPaikModel(...)'.
flag_eps	Logical flag indicating whether to plot only the time-dependent posterior frailty estimates. Default is FALSE.
flag_alpha	Logical flag indicating whether to plot only the time-independent posterior frailty estimates. Default is FALSE.
xlim	A numeric vector specifying the range for the x-axis (intervals). If NULL, default is set to the interval min-max of the time-domain, plus space for the legend. If flag_alpha = TRUE, the plot is produced around 1 (defaults to 0.8-1.4).
ylim	A numeric vector specifying the range for the y-axis (intervals). If NULL, default is min-max value of the posterior frailty estimate.
xlab, ylab	String giving the x and y axis name. Default values are 'Time' and 'Values'.
main	Title of the plot. Default title is 'Posterior frailty estimates'.
cex	Dimension of the points used for plotting the estimates.
pch_type	Numerical vector of length equal to the number of clusters in the data, giving the symbol to be used for plotting the estimates. Default symbol (circle, 21) is the same for all clusters.
color_bg	Numerical vector of length equal to the number of clusters in the data, giving the color to be used for plotting the symbols for the estimates. Default ('black') is the same for all faculties. On the other hand, the same color is used throughout the intervals for the same faculty.
cex_legend	Dimension of the symbol in the legend. Default is 0.7.
pos_legend	Either a numeric vector providing the x and y coordinates for the legend or a string specifying the legend's position (e.g., 'bottomright', 'bottom', 'bottom-left', 'left', 'topleft', 'top', 'topright', 'right', 'center').

Details

Recalling the frailty structure as $Z_{jk} = \alpha_j + \epsilon_{jk}, \forall j, k$ and the posterior frailty estimate as $\hat{Z}_{jk} = \hat{\alpha}_j / \hat{\alpha}_{max} + \hat{\epsilon}_{jk} / \hat{\epsilon}_{max}$, this function allows plotting either the entire posterior frailty estimate \hat{Z}_{jk} or its time-independent $\frac{\hat{\alpha}_j}{\hat{\alpha}_{max}}$ or time-dependent $\frac{\hat{\epsilon}_{jk}}{\hat{\epsilon}_{max}}$ components. The user can control which components to display using the flag_eps and flag_alpha parameters. Only one of these flags can be set to TRUE at a time.

Value

The plot of the posterior frailty estimates.

Examples

```

# Import data
data(data_dropout)

# Define the variables needed for the model execution
eps_paik <- 1e-10
categories_range_min <- c(-8, -2, eps_paik, eps_paik, eps_paik)
categories_range_max <- c(-eps_paik, 0.4, 1 - eps_paik, 1, 10)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
formula <- time_to_event ~ Gender + CFUP + cluster(group)

# Call the main model function

result <- AdPaikModel(formula, data_dropout, time_axis, categories_range_min, categories_range_max)

# Define variables for plotting the estimates
pch_type <- c(21, seq(21,25,1), seq(21,25,1), seq(21,25,1))
color_bg <- c("darkblue", rep("red", 5), rep("purple", 5), rep("green",5))

plot_post_frailty_est(result, pch_type = pch_type, color_bg = color_bg)

```

plot_post_frailty_var *Plot the Posterior Frailty Variances*

Description

This function plots the posterior frailty variances for each group in each time interval (represented by its mid point). Each group's estimates are represented by a sequence of points connected by straight lines. The function can plot either the entire posterior frailty variance or its time-independent and time-dependent components based on user-specified flags.

Usage

```

plot_post_frailty_var(
  result,
  flag_eps = FALSE,
  flag_alpha = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = "Time",
  ylab = "Values",
  main = "Posterior frailty variances",
  cex = 0.7,
  pch_type = seq(1, length(result$ClusterCodes)),
  color_bg = rep("black", length(result$ClusterCodes)),
  cex_legend = 0.7,

```

```

    pos_legend = "topright"
  )

```

Arguments

result	S3 object of class 'AdPaik', returned by the method call 'AdPaikModel(...)'.
flag_eps	Logical flag indicating whether to plot only the time-dependent posterior frailty estimates. Default is FALSE.
flag_alpha	Logical flag indicating whether to plot only the time-independent posterior frailty estimates. Default is FALSE.
xlim	A numeric vector specifying the range for the x-axis (intervals). If NULL, default is set to the interval min-max of the time-domain, plus space for the legend. If flag_alpha = TRUE, the plot is produced around 1 (defaults to 0.8-1.4).
ylim	A numeric vector specifying the range for the y-axis (intervals). If NULL, default is min-max value of the posterior frailty estimate.
xlab, ylab	String giving the x and y axis name. Default values are 'Time' and 'Values'.
main	Title of the plot. Default title is 'Posterior frailty estimates'.
cex	Dimension of the points used for plotting the estimates.
pch_type	Numerical vector of length equal to the number of clusters in the data, giving the symbol to be used for plotting the estimates. Default symbol (circle, 21) is the same for all clusters.
color_bg	Numerical vector of length equal to the number of clusters in the data, giving the color to be used for plotting the symbols for the estimates. Default ('black') is the same for all faculties. On the other hand, the same color is used throughout the intervals for the same faculty.
cex_legend	Dimension of the symbol in the legend. Default is 0.7.
pos_legend	Either a numeric vector providing the x and y coordinates for the legend or a string specifying the legend's position (e.g., 'bottomright', 'bottom', 'bottom-left', 'left', 'topleft', 'top', 'topright', 'right', 'center').

Details

Recalling the frailty structure as $Z_{jk} = \alpha_j + \epsilon_{jk}$, $\forall j, k$ and the posterior frailty variance as $var(\hat{Z}_{jk}) = var(\hat{\alpha}_j / \hat{\alpha}_{max}) + var(\hat{\epsilon}_{jk} / \hat{\epsilon}_{max})$, this function allows plotting either the entire posterior frailty variance $var(\hat{Z}_{jk})$ or its time-independent $var(\frac{\hat{\alpha}_j}{\hat{\alpha}_{max}})$ or time-dependent $var(\frac{\hat{\epsilon}_{jk}}{\hat{\epsilon}_{max}})$ components. The user can control which components to display using the flag_eps and flag_alpha parameters. Only one of these flags can be set to TRUE at a time.

Value

The plot of the posterior frailty variances.

Examples

```

# Import data
data(data_dropout)

# Define the variables needed for the model execution
eps_paik <- 1e-10
categories_range_min <- c(-8, -2, eps_paik, eps_paik, eps_paik)
categories_range_max <- c(-eps_paik, 0.4, 1 - eps_paik, 1, 10)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
formula <- time_to_event ~ Gender + CFUP + cluster(group)

# Call the main model function

result <- AdPaikModel(formula, data_dropout, time_axis, categories_range_min, categories_range_max)

# Define variables for plotting the variances
pch_type <- c(21, seq(21,25,1), seq(21,25,1), seq(21,25,1))
color_bg <- c("darkblue", rep("red", 5), rep("purple", 5), rep("green",5))

plot_post_frailty_var(result, pch_type = pch_type, color_bg = color_bg)

```

plot_survivalAdPaik *Plot of Conditional Survival Function*

Description

Plots the conditional survival function based on the 'Adapted Paik et al.' model's estimated coefficients and frailty effects, for each unit in each time interval (represented by its mid point).

Usage

```

plot_survivalAdPaik(
  result,
  lwd = 1,
  xlim = c(min(result$TimeDomain), max(result$TimeDomain)),
  ylim = c(0, 1),
  xlab = "Time",
  ylab = "Values",
  main = "Conditional Survival",
  cex = 0.2,
  cexlegend = 0.8
)

```

Arguments

result S3 object of class 'AdPaik' containing model results.

lwd	The line width of the plot. Default is 1.
xlim	A numeric vector specifying the range for the x-axis (intervals). Default is min-max value of the time domain.
ylim	A numeric vector specifying the range for the y-axis (intervals). Default is the range 0-1.
xlab, ylab	String giving the x and y axis name. Default values are 'Time' and 'Values'.
main	Title of the plot. Default title is 'Survival'.
cex	Dimension of the points used for plotting the estimates. Defaults to 0.2.
cexlegend	Dimension of the text used for the legend. Defaults to 0.9.

Value

The plot of the conditional survival function.

Examples

```
# Import data
data(data_dropout)

# Define the variables needed for the model execution
eps_paik <- 1e-10
categories_range_min <- c(-8, -2, eps_paik, eps_paik, eps_paik)
categories_range_max <- c(-eps_paik, 0.4, 1 - eps_paik, 1, 10)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
formula <- time_to_event ~ Gender + CFUP + cluster(group)

# Call the main model function

result <- AdPaikModel(formula, data_dropout, time_axis, categories_range_min, categories_range_max)

plot_survivalAdPaik(result)
```

post_frailty_confint *Posterior Frailty Confidence Intervals*

Description

Function for computing the posterior frailty confidence intervals of the time-dependent shared frailty Cox model.

Recalling the structure of the frailty $Z_{jk} = \alpha_j + \epsilon_{jk}, \forall j, k$ with $k = 1, \dots, L$ and $j = 1, \dots, N$ as being composed by the sum of two independent gamma distributions:

- $\alpha_j \sim \text{gamma}(\mu_1/\nu, 1/\nu), \forall j$
- $\epsilon_{jk} \sim \text{gamma}(\mu_2/\gamma_k, 1/\gamma_k), \forall j, k$

The posterior frailty estimate is $\hat{Z}_{jk} = \hat{\alpha}_j / \hat{\alpha}_{max} + \hat{\epsilon}_{jk} / \hat{\epsilon}_{max}$. This function allows to get the confidence intervals of either the entire posterior frailty estimates \hat{Z}_{jk} or its time-independent $\frac{\hat{\alpha}_j}{\hat{\alpha}_{max}}$ or time-dependent $\frac{\hat{\epsilon}_{jk}}{\hat{\epsilon}_{max}}$ components. The user can control which components to display using the flag_eps and flag_alpha parameters. Only one of these flags can be set to TRUE at a time.

Usage

```
post_frailty_confint(
  object,
  level = 0.95,
  flag_eps = FALSE,
  flag_alpha = FALSE
)
```

Arguments

object	S3 object of class 'AdPaik' returned by the main model output, that contains all the information for the computation of the frailty standard deviation.
level	A numeric value representing the confidence level for the posterior frailty confidence intervals. Default is 0.95 for 95% confidence.
flag_eps	Logical flag indicating whether to extract only the time-dependent posterior frailty estimates. Default is FALSE.
flag_alpha	Logical flag indicating whether to extract only the time-independent posterior frailty estimates. Default is FALSE.

Value

A list for posterior frailty confidence intervals, depending on the flag_eps and flag_alpha values. Specifically:

- A list of length equal to the N containing posterior frailty confidence intervals for $\alpha_j, \forall j$. In this case the flag_eps must be FALSE and the flag_alpha must be TRUE.
- A list of length equal to the NxL containing posterior frailty confidence intervals for $\epsilon_{jk}, \forall j, k$. In this case the flag_eps must be TRUE and the flag_alpha must be FALSE.
- A list of length equal to the NxL containing posterior frailty confidence intervals for $Z_{jk} \forall j, k$. In this case the flag_eps must be FALSE and the flag_alpha must be FALSE.

Examples

```
# Consider the 'Academic Dropout dataset'
data(data_dropout)

# Define the variables needed for the model execution
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)
```

```
# Call the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max)

post_frailty_confint(result)
```

post_frailty_est *Posterior Frailty Estimates*

Description

Function for computing the posterior frailty estimates of the time-dependent shared frailty Cox model.

Recalling the structure of the frailty $Z_{jk} = \alpha_j + \epsilon_{jk}, \forall j, k$ with $k = 1, \dots, L$ and $j = 1, \dots, N$ as being composed by the sum of two independent gamma distributions:

- $\alpha_j \sim \text{gamma}(\mu_1/\nu, 1/\nu), \forall j$
- $\epsilon_{jk} \sim \text{gamma}(\mu_2/\gamma_k, 1/\gamma_k), \forall j, k$

The posterior frailty estimate is $\hat{Z}_{jk} = \hat{\alpha}_j / \hat{\alpha}_{max} + \hat{\epsilon}_{jk} / \hat{\epsilon}_{max}$. This function allows to get either the entire posterior frailty estimate \hat{Z}_{jk} or its time-independent $\frac{\hat{\alpha}_j}{\hat{\alpha}_{max}}$ or time-dependent $\frac{\hat{\epsilon}_{jk}}{\hat{\epsilon}_{max}}$ components. The user can control which components to display using the `flag_eps` and `flag_alpha` parameters. Only one of these flags can be set to TRUE at a time.

Usage

```
post_frailty_est(object, flag_eps = FALSE, flag_alpha = FALSE)
```

Arguments

<code>object</code>	S3 object of class 'AdPaik' returned by the main model output, that contains all the information for the computation of the frailty standard deviation.
<code>flag_eps</code>	Logical flag indicating whether to extract only the time-dependent posterior frailty estimates. Default is FALSE.
<code>flag_alpha</code>	Logical flag indicating whether to extract only the time-independent posterior frailty estimates. Default is FALSE.

Value

Vector or matrix of posterior frailty estimates, depending on the `flag_eps` and `flag_alpha` values. Specifically:

- It is a vector of length equal to the N containing posterior frailty estimates for $\alpha_j, \forall j$. In this case the `flag_eps` must be FALSE and the `flag_alpha` must be TRUE.

- Matrix of dimension (N, L) containing posterior frailty estimates for $\epsilon_{jk}, \forall j, k$. In this case the flag_eps must be TRUE and the flag_alpha must be FALSE.
- Matrix of dimension (N, L) containing posterior frailty estimates for $Z_{jk} = \alpha_j + \epsilon_{jk}, \forall j, k$. In this case the flag_eps must be FALSE and the flag_alpha must be FALSE.

Examples

```
# Consider the 'Academic Dropout dataset'
data(data_dropout)

# Define the variables needed for the model execution
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)

# Call the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max)

post_frailty_est(result)
```

post_frailty_var *Posterior Frailty Variances*

Description

Function for computing the posterior frailty variances of the time-dependent shared frailty Cox model.

Recalling the structure of the frailty $Z_{jk} = \alpha_j + \epsilon_{jk}, \forall j, k$ with $k = 1, \dots, L$ and $j = 1, \dots, N$ as being composed by the sum of two independent gamma distributions:

- $\alpha_j \sim \text{gamma}(\mu_1/\nu, 1/\nu), \forall j$
- $\epsilon_{jk} \sim \text{gamma}(\mu_2/\gamma_k, 1/\gamma_k), \forall j, k$

The posterior frailty variance is $\text{var}(\hat{Z}_{jk}) = \text{var}(\hat{\alpha}_j/\hat{\alpha}_{max}) + \text{var}(\hat{\epsilon}_{jk}/\hat{\epsilon}_{max})$. This function allows to get either the entire posterior frailty variance $\text{var}(\hat{Z}_{jk})$ or its time-independent $\text{var}(\frac{\hat{\alpha}_j}{\hat{\alpha}_{max}})$ or time-dependent $\text{var}(\frac{\hat{\epsilon}_{jk}}{\hat{\epsilon}_{max}})$ components. The user can control which components to display using the flag_eps and flag_alpha parameters. Only one of these flags can be set to TRUE at a time.

Usage

```
post_frailty_var(object, flag_eps = FALSE, flag_alpha = FALSE)
```

Arguments

object	S3 object of class 'AdPaik' returned by the main model output, that contains all the information for the computation of the frailty standard deviation.
flag_eps	Logical flag indicating whether to extract only the time-dependent posterior frailty estimates. Default is FALSE.
flag_alpha	Logical flag indicating whether to extract only the time-independent posterior frailty estimates. Default is FALSE.

Value

Vector or matrix of posterior frailty variances, depending on the flag_eps and flag_alpha values. Specifically:

- It is a vector of length equal to the N containing posterior frailty variances for $\alpha_j, \forall j$. In this case the flag_eps must be FALSE and the flag_alpha must be TRUE.
- Matrix of dimension (N, L) containing posterior frailty variances for $\epsilon_{jk}, \forall j, k$. In this case the flag_eps must be TRUE and the flag_alpha must be FALSE.
- Matrix of dimension (N, L) containing posterior frailty variances for $Z_{jk}, \forall j, k$. In this case the flag_eps must be FALSE and the flag_alpha must be FALSE.

Examples

```
# Consider the 'Academic Dropout dataset'
data(data_dropout)

# Define the variables needed for the model execution
formula <- time_to_event ~ Gender + CFUP + cluster(group)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
eps <- 1e-10
categories_range_min <- c(-8, -2, eps, eps, eps)
categories_range_max <- c(-eps, 0, 1 - eps, 1, 10)

# Call the main model
result <- AdPaikModel(formula, data_dropout, time_axis,
                      categories_range_min, categories_range_max)

post_frailty_var(result)
```

print.AdPaik

Print method for AdPaik objects

Description

This function prints a summary of the optimal parameters estimated in an AdPaik object.

Usage

```
## S3 method for class 'AdPaik'  
print(x, ...)
```

Arguments

x	An object of class AdPaik.
...	Additional arguments (not used).

Value

Prints a summary of the AdPaik object and returns it invisibly.

print.summary.AdPaik *Print Method for summary.AdPaik Objects*

Description

print method for objects of class "summary.AdPaik". Formats and prints the model summary.

Usage

```
## S3 method for class 'summary.AdPaik'  
print(x, ...)
```

Arguments

x	An object of class "summary.AdPaik".
...	Additional arguments (currently unused).

Value

Prints a structured summary of the AdPaik model.

summary.AdPaik	<i>Summary Method for AdPaik Objects</i>
----------------	--

Description

summary method for objects of class "AdPaik". It prepares a structured summary of the results.

Usage

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

Arguments

object	An object of class "AdPaik", returned by the main model function.
...	Additional arguments (currently unused).

Value

An object of class "summary.AdPaik" containing structured model summary information.

survivalAdPaik	<i>Compute the Conditional Survival Function</i>
----------------	--

Description

Computes the conditional survival function based on the 'Adapted Paik et al.' model's given the estimated coefficients and frailty effects.

Usage

```
survivalAdPaik(result)
```

Arguments

result	S3 object of class 'AdPaik' containing model results.
--------	---

Value

A dataset where each row corresponds to an individual unit in the dataset, and the columns represent the survival function values over time interval, with the first column indicating the cluster to which the individual belongs.

Examples

```
# Import data
data(data_dropout)

# Define the variables needed for the model execution
eps_paik <- 1e-10
categories_range_min <- c(-8, -2, eps_paik, eps_paik, eps_paik)
categories_range_max <- c(-eps_paik, 0.4, 1 - eps_paik, 1, 10)
time_axis <- c(1.0, 1.4, 1.8, 2.3, 3.1, 3.8, 4.3, 5.0, 5.5, 5.8, 6.0)
formula <- time_to_event ~ Gender + CFUP + cluster(group)

# Call the main model function

result <- AdPaikModel(formula, data_dropout, time_axis, categories_range_min, categories_range_max)

survivalAdPaik(result)
```

Index

* datasets

- data_dropout, [13](#)

- AdPaik_1D, [6](#)
- AdPaikModel, [2](#)

- bas_hazard, [9](#)

- coef.AdPaik, [10](#)
- coefseAdPaik, [11](#)
- confint.AdPaik, [12](#)

- data_dropout, [13](#)

- extractAIC.AdPaik, [14](#)

- frailty_sd, [14](#)

- logLik.AdPaik, [15](#)

- nobs.AdPaik, [16](#)

- plot.AdPaik, [16](#)
- plot_bas_hazard, [17](#)
- plot_frailty_sd, [19](#)
- plot_post_frailty_est, [20](#)
- plot_post_frailty_var, [22](#)
- plot_survivalAdPaik, [24](#)
- post_frailty_confint, [25](#)
- post_frailty_est, [27](#)
- post_frailty_var, [28](#)
- print.AdPaik, [29](#)
- print.summary.AdPaik, [30](#)

- summary.AdPaik, [31](#)
- survivalAdPaik, [31](#)