Package 'BJM'

July 4, 2026

Title Backward Joint Model for the Dynamic Prediction of Both Time-to-Event and Longitudinal Outcomes

Version 0.1.0

Maintainer Wenhao Li <wenhaoli.jlu@gmail.com>

Description Provides tools to fit joint models of multivariate longitudinal data and time-to-event data for dynamic prediction. It allows the joint prediction of both future time-to-event outcomes and future longitudinal outcomes conditional on survival. The models accommodate irregularly measured longitudinal data and competing risks outcomes. The use of the backward joint model enables fast and efficient computation, especially for applications with large sample sizes and many longitudinal variables.

License MIT + file LICENSE

LazyData true

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0), survival

Imports nlme, mvtnorm, ggplot2, Matrix

Author Wenhao Li [aut, cre], Liang Li [aut]

Repository CRAN

Date/Publication 2026-07-04 07:50:02 UTC

NeedsCompilation no

Contents

cmtPlot 2
dynamicPrediction 4
dynamicPredictionBio 6
longitudinalSub 9
pbc2 11
pbc3 12
predictPlot 14

print.dynamicPrediction.BJM 16
print.dynamicPredictionBio.BJM 17
print.longitudinalSub.BJM 18
print.survivalSub.BJM 19
print_BJM 20
riskPlot 20
summary.dynamicPrediction.BJM 21
summary.dynamicPredictionBio.BJM 22
summary.longitudinalSub.BJM 23
summary.survivalSub.BJM 23
survivalSub 24

Index **25**

`cmtPlot` *Plot conditional mean trajectories (CMT)*

Description

This function generates the Conditional Mean Trajectories (CMT) plot. All patients in this plot experience events at the same time point, specified by `condi_time2event`. Several evenly spaced time points between the baseline and `condi_time2event` are selected for plotting. Each point is calculated using the mean value of all patients' biomarker values at that time point. The interval between two time points is defined by `interval_time`

Usage

```
cmtPlot(
  data.plot.all,
  condi_time2event,
  event_type_variable,
  event_type,
  bio_variable,
  time_variable,
  survival_variable,
  interval_time = 1/12,
  id_variable = "id"
)
```

Arguments

`data.plot.all`	A `data.frame` that includes the biomarker used for plotting. It is utilized to plot conditional mean trajectories (CMT).
`condi_time2event`	Conditional event time, indicating that all patients should have events at this time in the plot

event_type_variable
: Competing risks variable indicator name. Set to NULL if there are no competing risks.

event_type
: A vector containing the names of all event types.

bio_variable
: Name of the biomarker variable used for plotting.

time_variable
: The name of time variable in linear mixed model.

survival_variable
: Name of the time-to-event outcomes variable.

interval_time
: The time interval between two time points. Time points are plotted within the baseline to event time.

id_variable
: Name of the patient ID column in `data.plot.all`. Default is `"id"`.

Value

Conditional mean trajectories plot.

Examples

```
# example without competing risks

data(pbc3)

pbc.cmt <- cmtPlot(data.plot.all = pbc3, condi_time2event = 5,
   event_type_variable = NULL, event_type = NULL,
   bio_variable = "serBilir", time_variable = "year",
   survival_variable = "years",
   interval_time = 1/12
)

pbc.cmt


# example with competing risks

data(pbc3)

data.plot.all = pbc3[!is.na(pbc3$status4),]

pbc.cmt.cr <- cmtPlot(data.plot.all, condi_time2event = 5,
   event_type_variable = 'status4', event_type = c("0", "1"),
   bio_variable = "albumin", time_variable = "year",
   survival_variable = "years",
   interval_time = 1/4
)

pbc.cmt.cr
```

`dynamicPrediction` *Dynamic prediction function*

Description

The time values in the prediction data subset must be less than the specified `prediction.time` which is the prediction time. The time points for longitudinal repeated measurements must not surpass the prediction time.

Usage

```
dynamicPrediction(
  data.predict.all,
  long_fit_all,
  survival_fit_all,
  prediction.time,
  horizon,
  time_variable,
  survivalVariableAll,
  survivalTransFunction,
  bandcount1 = 10,
  bandcount2 = 40
)
```

Arguments

`data.predict.all`
: This involves a collection of `data.frame` objects for dynamic prediction, each corresponding to a distinct longitudinal outcome. These data frames should contain the variables specified in `LongSubFixed` and `LongSubRandom`. Utilizing a list structure allows for the incorporation of multiple longitudinal outcomes, each potentially following different measurement protocols. In instances where all longitudinal outcomes are recorded at identical time points across patients, a singular `data.frame` object may be used in a `list`. It is presumed that each data frame is structured in a long format.

`long_fit_all`
: Outputs from the model fitting process using the `nlme` package, encompassing the results and parameters obtained from the analysis.

`survival_fit_all`
: Results and parameters generated from the model fitting procedure, utilizing the `coxph` function. These outputs include the comprehensive findings and variables derived from the analysis.

`prediction.time`
: Time used to make the prediction.

`horizon`
: Prediction horizon.

`time_variable`
: The name of time variable in linear mixed model.

survivalVariableAll
: The name of the transformed time-to-event outcomes variable.

survivalTransFunction
: The transformation function used for time-to-event outcomes, in the order of survivalVariableAll.

bandcount1
: The number of points used to perform the numerical integral, from the prediction time to the prediction time plus the horizon.

bandcount2
: The number of points used to perform the numerical integral, from the prediction time to infinity.

Value

A probability matrix, where the rows (l_i) correspond to specific time points and the columns to individual patients. Each element within the matrix signifies the probability of a future event occurring, as dynamically predicted for each patient at each time point.

Examples

```
data(pbc3)

data.survival.fitting =  pbc3[!duplicated(pbc3$id), ]

formMarginalSurv = Surv(years, status3) ~ age + sex
formConditionalCR = NULL

survival_fit_all = survivalSub(data.survival.fitting, formMarginalSurv,
                               formConditionalCR)

LongSubFixed = list(
  "long1" = serBilir ~ year + age + sex +  (years) + (years) * year,
  "long2" = prothrombin ~ year + age + sex + (years) + (years) * year,
  "long3" = albumin ~ year + age + age * year + sex + (years) + (years) * year,
  "long4" = alkaline ~ year + age + sex + (years) + (years) * year,
  "long5" = SGOT ~ year + age + sex + (years) + (years) * year,
  "long6" = platelets ~ year + age + sex + (years)  + (years) * year)

LongSubRandom =list(
  "long1" =  ~ year| id,
  "long2" =  ~ year| id,
  "long3" =  ~ year| id,
  "long4" =  ~ year| id,
  "long5" =  ~ year| id,
  "long6" =  ~ year| id)

survivalVariableAll = list(
  "Tyears1",  "Tyears2", "Tyears3", "Tyears4"
)

survivalTransFunction = list(
  fun1 = function(x){abs(x - 1)},
```

```
  fun2 = function(x){abs(x - 3)},
  fun3 = function(x){abs(x - 5)},
  fun4 = function(x){abs(x - 7)}
)

# Complete case analysis
data.fit.all = list()
for(i in 1:length(LongSubFixed)){
  data.fit.all[[i]] = pbc3[pbc3$status3 == 1, ]
}

# fitting longitudinal submodel
long_fit_all = longitudinalSub(data.fit.all, LongSubFixed, LongSubRandom)

i_PID = 2
data.raw.predict.1 = pbc3[pbc3$id == i_PID, ]

data.predict.all = list()
for(i in 1:length(LongSubFixed)){
  data.predict.all[[i]] = data.raw.predict.1[data.raw.predict.1$year <= 3,]
}

# predict risk probability
risk.prob = dynamicPrediction(data.predict.all, long_fit_all, survival_fit_all,
                              prediction.time = 3,
                              horizon = 3, time_variable = "year",
                              survivalVariableAll, survivalTransFunction,
                              bandcount1 = 10, bandcount2 = 10)
```

`dynamicPredictionBio` *Dynamic prediction function for future biomarker*

Description

The time values in the prediction data subset must be less than the specified `prediction.time` which is the prediction time. The time points for longitudinal repeated measurements must not surpass the prediction time.

Usage

```
dynamicPredictionBio(
  bio_i,
  data.predict.all,
  long_fit_all,
  survival_fit_all,
  prediction.time,
```

```
  horizon,
  time_variable,
  survivalVariableAll,
  survivalTransFunction,
  bandcount2 = 40,
  bandcount3 = 300
)
```

Arguments

bio_i	Biomarker used to do prediction
data.predict.all	This involves a collection of data.frame objects for dynamic prediction, each corresponding to a distinct longitudinal outcome. These data frames should contain the variables specified in LongSubFixed and LongSubRandom. Utilizing a list structure allows for the incorporation of multiple longitudinal outcomes, each potentially following different measurement protocols. In instances where all longitudinal outcomes are recorded at identical time points across patients, a singular data.frame object may be used in a list. It is presumed that each data frame is structured in a long format.
long_fit_all	Outputs from the model fitting process using the nlme package, encompassing the results and parameters obtained from the analysis.
survival_fit_all	Results and parameters generated from the model fitting procedure, utilizing the coxph function. These outputs include the comprehensive findings and variables derived from the analysis.
prediction.time	Time used to make the prediction
horizon	Prediction horizon
time_variable	The name of time variable in linear mixed model.
survivalVariableAll	The name of the transformed time-to-event outcomes variable.
survivalTransFunction	The transformation function used for time-to-event outcomes, in the order of survivalVariableAll.
bandcount2	The number of points used to perform the numerical integral, from the prediction time to infinity.
bandcount3	The number of points used to calculate the probability density function.

Value

A probability matrix, where the rows (l_i) correspond to specific time points and the columns to individual patients. Each element within the matrix signifies the probability of a future event occurring, as dynamically predicted for each patient at each time point.

Examples

```
data(pbc3)

data.survival.fitting =  pbc3[!duplicated(pbc3$id), ]

formMarginalSurv = Surv(years, status3) ~ age + sex
formConditionalCR = NULL

survival_fit_all = survivalSub(data.survival.fitting, formMarginalSurv,
                               formConditionalCR)

LongSubFixed = list(
  "long1" = serBilir ~ year + age + sex +  (years) + (years) * year,
  "long2" = prothrombin ~ year + age + sex + (years) + (years) * year,
  "long3" = albumin ~ year + age + age * year + sex + (years) + (years) * year,
  "long4" = alkaline ~ year + age + sex + (years) + (years) * year,
  "long5" = SGOT ~ year + age + sex + (years) + (years) * year,
  "long6" = platelets ~ year + age + sex + (years)  + (years) * year)

LongSubRandom =list(
  "long1" =  ~ year| id,
  "long2" =  ~ year| id,
  "long3" =  ~ year| id,
  "long4" =  ~ year| id,
  "long5" =  ~ year| id,
  "long6" =  ~ year| id)

survivalVariableAll = list(
  "Tyears1",  "Tyears2", "Tyears3", "Tyears4"
)

survivalTransFunction = list(
  fun1 = function(x){abs(x - 1)},
  fun2 = function(x){abs(x - 3)},
  fun3 = function(x){abs(x - 5)},
  fun4 = function(x){abs(x - 7)}
)

# Complete case analysis
data.fit.all = list()
for(i in 1:length(LongSubFixed)){
  data.fit.all[[i]] = pbc3[pbc3$status3 == 1, ]
}

# fitting longitudinal submodel
long_fit_all = longitudinalSub(data.fit.all, LongSubFixed, LongSubRandom)

i_PID = 2
data.raw.predict.1 = pbc3[pbc3$id == i_PID, ]

data.predict.all = list()
```

```
for(i in 1:length(LongSubFixed)){
  data.predict.all[[i]] = data.raw.predict.1[data.raw.predict.1$year <= 3,]
}

Y_predict = dynamicPredictionBio(bio_i = 1, data.predict.all, long_fit_all,
                                 survival_fit_all, prediction.time = 3,
                                 horizon = 3, time_variable = "year",
                                 survivalVariableAll, survivalTransFunction,
                                 bandcount2 = 40, bandcount3 = 400)
```

`longitudinalSub` *The process involves estimating parameters for a multivariate linear mixed-effects model, which simultaneously analyzes multiple dependent variables that may be correlated. This approach incorporates both fixed effects, which are consistent across the population, and random effects, accounting for variations within groups or subjects. By fitting this model, one can assess the influence of predictor variables on several longitudinal outcomes while considering the inherent variability in the data due to random effects.*

Description

The process involves estimating parameters for a multivariate linear mixed-effects model, which simultaneously analyzes multiple dependent variables that may be correlated. This approach incorporates both fixed effects, which are consistent across the population, and random effects, accounting for variations within groups or subjects. By fitting this model, one can assess the influence of predictor variables on several longitudinal outcomes while considering the inherent variability in the data due to random effects.

Usage

```
longitudinalSub(data.fit.all, LongSubFixed, LongSubRandom)
```

Arguments

`data.fit.all`	This process requires a set of `data.frame` objects designated for model fitting, with each `data.frame` representing a separate longitudinal outcome. These `data.frame` objects must include the variables identified in `LongSubFixed` and `LongSubRandom`. The use of a `list` arrangement facilitates the inclusion of various longitudinal outcomes, which may adhere to different measurement protocols. When all longitudinal outcomes are measured at the same time points for every patient, a single `data.frame` object can be in a list. It is assumed that every `data.frame` is organized in a long format.

LongSubFixed — This refers to a collection of formulas detailing the fixed effects portion for each longitudinal outcome. On the left side of each formula, the response variable is defined, while the right side outlines the fixed effect terms. Should only a single formula be provided—whether as a list with one item or as a standalone formula—it is inferred that a conventional univariate joint model is being constructed.

LongSubRandom — A list of one-sided formulas that define the model for the random effects of each longitudinal outcome. The number of items in this `list` should match the length of `formLongFixed`.

Value

This structure comprises a list with four components. The initial element, labeled `lfit`, consists of a collection of outcomes from fitting multiple univariate linear mixed models, where each entry within `lfit` corresponds to the results obtained through the application of the `lme` function from the `nlme` package. The second element is the estimated variance-covariance matrix derived from the random effects in a multivariate linear mixed model. The third and fourth elements, `LongSubFixed` and `LongSubRandom`, respectively, mirror the inputs provided to the model.

Examples

```
LongSubFixed = list(
  "long1" = serBilir ~ year + age + sex +  (years) + (years) * year,
  "long2" = prothrombin ~ year + age + sex + (years) + (years) * year,
  "long3" = albumin ~ year + age + age * year + sex + (years) + (years) * year,
  "long4" = alkaline ~ year + age + sex + (years) + (years) * year,
  "long5" = SGOT ~ year + age + sex + (years) + (years) * year,
  "long6" = platelets ~ year + age + sex + (years)  + (years) * year)

LongSubRandom =list(
  "long1" =  ~ year| id,
  "long2" =  ~ year| id,
  "long3" =  ~ year| id,
  "long4" =  ~ year| id,
  "long5" =  ~ year| id,
  "long6" =  ~ year| id)

survivalVariableAll = list(
  "Tyears1",  "Tyears2", "Tyears3", "Tyears4"
)

survivalTransFunction = list(
  fun1 = function(x){abs(x - 1)},
  fun2 = function(x){abs(x - 3)},
  fun3 = function(x){abs(x - 5)},
  fun4 = function(x){abs(x - 7)}
)

# Complete case analysis
data.fit.all = list()
```

```
for(i in 1:length(LongSubFixed)){
  data.fit.all[[i]] = pbc3[pbc3$status3 == 1, ]
}

# fitting longitudinal submodel
long_fit_all = longitudinalSub(data.fit.all, LongSubFixed, LongSubRandom)
```

---

pbc2 *Mayo Clinic primary biliary cirrhosis data*

Description

The dataset originates from the Mayo Clinic trial on primary biliary cirrhosis (PBC) of the liver, carried out from 1974 to 1984. It includes data from 424 PBC patients who were referred to the Mayo Clinic within this decade and met the eligibility requirements for a randomized placebo-controlled trial of D-penicillamine. However, only the initial 312 cases from the dataset were enrolled in the randomized trial. Thus, the dataset specifically pertains to these 312 patients, for whom the data is largely complete.

Usage

```
data(pbc2)
```

Format

A data frame with 1945 observations on the following 20 variables:

`id` patients identifier; in total there are 312 patients.

`years` number of years between registration and the earlier of death, transplantation, or study analysis time.

`status` a factor with levels `alive`, `transplanted` and `dead`.

`drug` a factor with levels `placebo` and `D-penicil`.

`age` at registration in years.

`sex` a factor with levels `male` and `female`.

`year` number of years between enrollment and this visit date, remaining values on the line of data refer to this visit.

`ascites` a factor with levels `No` and `Yes`.

`hepatomegaly` a factor with levels `No` and `Yes`.

`spiders` a factor with levels `No` and `Yes`.

`edema` a factor with levels `No edema` (i.e. no edema and no diuretic therapy for edema), `edema no diuretics` (i.e. edema present without diuretics, or edema resolved by diuretics), and `edema despite diuretics` (i.e. edema despite diuretic therapy).

`serBilir` serum bilirubin in mg/dl.

`serChol` serum cholesterol in mg/dl.

`albumin` albumin in mg/dl.

`alkaline` alkaline phosphatase in U/liter.

`SGOT` SGOT in U/ml.

`platelets` platelets per cubic ml/1000.

`prothrombin` prothrombin time in seconds.

`histologic` histologic stage of disease.

`status2` a numeric vector with the value 1 denoting if the patient was dead, and 0 if the patient was alive or transplanted.

Source

pbc.

References

Fleming T, Harrington D. *Counting Processes and Survival Analysis*. 1991; New York: Wiley.

Therneau T, Grambsch P. *Modeling Survival Data: Extending the Cox Model*. 2000; New York: Springer-Verlag.

`pbc3` *Mayo Clinic primary biliary cirrhosis data used as example code*

Description

The dataset originates from the Mayo Clinic trial on primary biliary cirrhosis (PBC) of the liver, carried out from 1974 to 1984. It includes data from 424 PBC patients who were referred to the Mayo Clinic within this decade and met the eligibility requirements for a randomized placebo-controlled trial of D-penicillamine. However, only the initial 312 cases from the dataset were enrolled in the randomized trial. Thus, the dataset specifically pertains to these 312 patients, for whom the data is largely complete.

Usage

```
data(pbc2)
```

Format

A data frame with 1945 observations on the following 20 variables:

`id` patients identifier; in total there are 312 patients.

`years` number of years between registration and the earlier of death, transplantation, or study analysis time.

`status` a factor with levels `alive`, `transplanted` and `dead`.

`drug` a factor with levels `placebo` and `D-penicil`.

`age` at registration in years.

`sex` a factor with levels `male` and `female`.

`year` number of years between enrollment and this visit date, remaining values on the line of data refer to this visit.

`ascites` a factor with levels `No` and `Yes`.

`hepatomegaly` a factor with levels `No` and `Yes`.

`spiders` a factor with levels `No` and `Yes`.

`edema` a factor with levels `No edema` (i.e. no edema and no diuretic therapy for edema), `edema no diuretics` (i.e. edema present without diuretics, or edema resolved by diuretics), and `edema despite diuretics` (i.e. edema despite diuretic therapy).

`serBilir` serum bilirubin in mg/dl.

`serChol` serum cholesterol in mg/dl.

`albumin` albumin in mg/dl.

`alkaline` alkaline phosphatase in U/liter.

`SGOT` SGOT in U/ml.

`platelets` platelets per cubic ml/1000.

`prothrombin` prothrombin time in seconds.

`histologic` histologic stage of disease.

`status2` a numeric vector with the value 1 denoting if the patient was dead, and 0 if the patient was alive or transplanted.

`status3` a numeric vector with the value 1 denoting if the patient was dead or transplanted, and 0 if the patient was alive.

`status4` a numeric vector with the value 1 denoting if the patient was transplanted, and 0 if the patient was dead. Used for competing risks.

`status5` a numeric vector with the value 2 if the patient was transplanted, 1 denoting if the patient was dead, and 0 if the patient was alive. Used for competing risks with censored.

`Tyears1` a numeric vector with a transformed value of time-to-event outcome.

`Tyears2` a numeric vector with a transformed value of time-to-event outcome.

`Tyears3` a numeric vector with a transformed value of time-to-event outcome.

`Tyears4` a numeric vector with a transformed value of time-to-event outcome.

Source

`pbc`.

References

Fleming T, Harrington D. *Counting Processes and Survival Analysis*. 1991; New York: Wiley.

Therneau T, Grambsch P. *Modeling Survival Data: Extending the Cox Model*. 2000; New York: Springer-Verlag.

predictPlot *Plot of risk and future biomarker with density using dynamic prediction*

Description

This function gives the risk and biomarker prediction plot.

Usage

```
predictPlot(
  data.predict.all.one,
  long_fit_all,
  survival_fit_all,
  prediction.time = 4,
  horizon = seq(0, 3, 0.5),
  time_variable,
  survivalVariableAll,
  survivalTransFunction,
  bandcount1 = 10,
  bandcount2 = 10,
  bandcount3 = 200,
  bio_his = 1,
  bio_pred = 1,
  density = 1
)
```

Arguments

`data.predict.all.one`
: This involves a collection of `data.frame` one object for dynamic prediction and making plots, each corresponding to a distinct longitudinal outcome. These data frames should contain the variables specified in `LongSubFixed` and `LongSubRandom`. Utilizing a list structure allows for the incorporation of multiple longitudinal outcomes, each potentially following different measurement protocols. In instances where all longitudinal outcomes are recorded at identical time points across patients, a singular `data.frame` object may be used in a `list`. It is presumed that each data frame is structured in a long format.

`long_fit_all`
: Outputs from the model fitting process using the `nlme` package, encompassing the results and parameters obtained from the analysis.

`survival_fit_all`
: Results and parameters generated from the model fitting procedure, utilizing the `coxph` function. These outputs include the comprehensive findings and variables derived from the analysis.

`prediction.time`
: Time used to make the prediction.

horizon	Prediction horizon.
time_variable	The name of time variable in linear mixed model.
survivalVariableAll	The name of the transformed time-to-event outcomes variable.
survivalTransFunction	The transformation function used for time-to-event outcomes, in the order of survivalVariableAll.
bandcount1	The number of points used to perform the numerical integral, from the prediction time to the prediction time plus the horizon.
bandcount2	The number of points used to perform the numerical integral, from the prediction time to infinity.
bandcount3	The number of points used to calculate the probability density function.
bio_his	Which biomarker history will be plotted
bio_pred	Indicator, predict future biomarker or not, if NULL do not predict
density	Indicator, plot future biomarker density or not, if NULL do not plot

Value

Plot of risk and future biomarker with density using dynamic prediction.

Examples

```
data(pbc3)

data.survival.fitting =  pbc3[!duplicated(pbc3$id), ]

formMarginalSurv = Surv(years, status3) ~ age + sex
formConditionalCR = NULL

survival_fit_all = survivalSub(data.survival.fitting, formMarginalSurv,
                               formConditionalCR)

LongSubFixed = list(
  "long1" = serBilir ~ year + age + sex +  (years) + (years) * year,
  "long2" = prothrombin ~ year + age + sex + (years) + (years) * year,
  "long3" = albumin ~ year + age + age * year + sex + (years) + (years) * year,
  "long4" = alkaline ~ year + age + sex + (years) + (years) * year,
  "long5" = SGOT ~ year + age + sex + (years) + (years) * year,
  "long6" = platelets ~ year + age + sex + (years)  + (years) * year)

LongSubRandom =list(
  "long1" =  ~ year| id,
  "long2" =  ~ year| id,
  "long3" =  ~ year| id,
  "long4" =  ~ year| id,
  "long5" =  ~ year| id,
  "long6" =  ~ year| id)

survivalVariableAll = list(
```

```
  "Tyears1",  "Tyears2", "Tyears3", "Tyears4"
)

survivalTransFunction = list(
  fun1 = function(x){abs(x - 1)},
  fun2 = function(x){abs(x - 3)},
  fun3 = function(x){abs(x - 5)},
  fun4 = function(x){abs(x - 7)}
)

# Complete case analysis
data.fit.all = list()
for(i in 1:length(LongSubFixed)){
  data.fit.all[[i]] = pbc3[pbc3$status3 == 1, ]
}

# fitting longitudinal submodel
long_fit_all = longitudinalSub(data.fit.all, LongSubFixed, LongSubRandom)

i_PID = 2
data.raw.predict.plot = pbc3[pbc3$id == i_PID, ]
data.predict.all.pre = list(data.raw.predict.plot, data.raw.predict.plot, data.raw.predict.plot,
                    data.raw.predict.plot, data.raw.predict.plot, data.raw.predict.plot)

# plot biomarker 1 history,  predict future biomarker

predictPlot(data.predict.all.pre, long_fit_all, survival_fit_all,
            prediction.time = 5, bio_his = 1, bio_pred = 1,
            horizon = seq(0.5, 3.0, 0.5), time_variable = "year",
            survivalVariableAll, survivalTransFunction,
           bandcount1 = 10, bandcount2 = 10, bandcount3 = 200)
```

`print.dynamicPrediction.BJM` *Print method for* `dynamicPrediction.BJM` *objects*

Description

Automatically called when you type the result of `dynamicPrediction()` at the console.

Usage

```
## S3 method for class 'dynamicPrediction.BJM'
print(
  x,
  prediction.time = NULL,
  horizon = NULL,
```

```
  subject_ids = NULL,
  digits = 4,
  ...
)

print_dynamicPrediction(
  x,
  prediction.time = NULL,
  horizon = NULL,
  subject_ids = NULL,
  digits = 4,
  ...
)
```

Arguments

x	A dynamicPrediction.BJM object.
prediction.time	Landmark time (for display). Default NULL.
horizon	Prediction horizon (for display). Default NULL.
subject_ids	Optional subject ID labels.
digits	Decimal places. Default 4.
...	Additional arguments (currently unused).

Value

Invisibly returns x.

---

print.dynamicPredictionBio.BJM

Print method for dynamicPredictionBio.BJM *objects*

Description

Automatically called when you type the result of dynamicPredictionBio() at the console.

Usage

```
## S3 method for class 'dynamicPredictionBio.BJM'
print(
  x,
  bio_i = NULL,
  long_fit_all = NULL,
  prediction.time = NULL,
  horizon = NULL,
  subject_ids = NULL,
```

```
  digits = 4,
  ...
)

print_dynamicPredictionBio(
  x,
  bio_i = NULL,
  long_fit_all = NULL,
  prediction.time = NULL,
  horizon = NULL,
  subject_ids = NULL,
  digits = 4,
  ...
)
```

Arguments

x	A dynamicPredictionBio.BJM object.
bio_i	Biomarker index (for label lookup). Default NULL.
long_fit_all	longitudinalSub.BJM object for name lookup.
prediction.time	Landmark time (for display). Default NULL.
horizon	Prediction horizon (for display). Default NULL.
subject_ids	Optional subject ID labels.
digits	Decimal places. Default 4.
...	Additional arguments (currently unused).

Value

Invisibly returns x.

---

print.longitudinalSub.BJM

Print method for longitudinalSub.BJM *objects*

Description

Automatically called when you type long_fit_all at the console.

Usage

```
## S3 method for class 'longitudinalSub.BJM'
print(x, digits = 4, ...)

print_longitudinalSub(x, digits = 4, ...)
```

Arguments

x	A longitudinalSub.BJM object returned by longitudinalSub.
digits	Number of significant digits. Default is 4.
...	Additional arguments (currently unused).

Value

Invisibly returns x.

---

print.survivalSub.BJM *Print method for* survivalSub.BJM *objects*

Description

Automatically called when you type survival_fit_all or print(survival_fit_all) at the console. Displays a JMbayes2-style formatted summary of the survival sub-model.

Usage

```
## S3 method for class 'survivalSub.BJM'
print(x, digits = 4, ...)

print_survivalSub(x, digits = 4, ...)
```

Arguments

x	A survivalSub.BJM object returned by survivalSub.
digits	Number of significant digits. Default is 4.
...	Additional arguments (currently unused).

Value

Invisibly returns x.

Examples

```
data(pbc3)
data.survival.fitting <- pbc3[!duplicated(pbc3$id), ]
formMarginalSurv  <- Surv(years, status3) ~ age + sex
formConditionalCR <- status4 ~ years + age + sex
survival_fit_all  <- survivalSub(data.survival.fitting,
                                 formMarginalSurv, formConditionalCR)
survival_fit_all   # triggers print.survivalSub.BJM automatically
```

print_BJM *Combined print summary for a fitted BJM*

Description

Combined print summary for a fitted BJM

Usage

```
print_BJM(long_fit_all, survival_fit_all, digits = 4)
```

Arguments

long_fit_all	Output from longitudinalSub.
survival_fit_all	Output from survivalSub.
digits	Number of significant digits. Default is 4.

Value

Invisibly returns a named list with both fit objects.

riskPlot *Plot of risk using dynamic prediction*

Description

This function gives the risk prediction plot.

Usage

```
riskPlot(
  data.predict.all.pre,
  long_fit_all,
  survival_fit_all,
  prediction.time = NULL,
  bio_i = NULL,
  horizon,
  time_variable,
  survivalVariableAll,
  survivalTransFunction,
  bandcount1 = 10,
  bandcount2 = 10
)
```

Arguments

data.predict.all.pre	This involves a collection of data.frame objects for
long_fit_all	Outputs from the model fitting process using the nlme package, encompassing the results and parameters obtained from the analysis.
survival_fit_all	Results and parameters generated from the model fitting procedure, utilizing the coxph function. These outputs include the comprehensive findings and variables derived from the analysis.
prediction.time	Time used to make the prediction.
bio_i	Biomarker used to do prediction.
horizon	Prediction horizon.
time_variable	The name of time variable in linear mixed model.
survivalVariableAll	The name of the transformed time-to-event outcomes variable.
survivalTransFunction	The transformation function used for time-to-event outcomes, in the order of survivalVariableAll.
bandcount1	The number of points used to perform the numerical integral, from the prediction time to the prediction time plus the horizon.
bandcount2	The number of points used to perform the numerical integral, from the prediction time to infinity.

Value

Plot of risk using dynamic prediction.

---

summary.dynamicPrediction.BJM

Summary method for dynamicPrediction.BJM *objects*

Description

Like print but also shows mean, SD, and range of predicted risks.

Usage

```
## S3 method for class 'dynamicPrediction.BJM'
summary(
  object,
  prediction.time = NULL,
  horizon = NULL,
  subject_ids = NULL,
  digits = 4,
  ...
)
```

Arguments

object	A dynamicPrediction.BJM object.
prediction.time	Landmark time (for display). Default NULL.
horizon	Prediction horizon (for display). Default NULL.
subject_ids	Optional subject ID labels.
digits	Decimal places. Default 4.
...	Additional arguments (currently unused).

Value

Invisibly returns object.

---

summary.dynamicPredictionBio.BJM

Summary method for dynamicPredictionBio.BJM *objects*

Description

Like print but also shows distribution-level summaries across subjects.

Usage

```
## S3 method for class 'dynamicPredictionBio.BJM'
summary(
  object,
  bio_i = NULL,
  long_fit_all = NULL,
  prediction.time = NULL,
  horizon = NULL,
  subject_ids = NULL,
  digits = 4,
  ...
)
```

Arguments

object	A dynamicPredictionBio.BJM object.
bio_i	Biomarker index (for label lookup). Default NULL.
long_fit_all	longitudinalSub.BJM object for name lookup.
prediction.time	Landmark time (for display). Default NULL.
horizon	Prediction horizon (for display). Default NULL.
subject_ids	Optional subject ID labels.
digits	Decimal places. Default 4.
...	Additional arguments (currently unused).

Value

Invisibly returns `object`.

`summary.longitudinalSub.BJM` *Summary method for* `longitudinalSub.BJM` *objects*

Description

Like `print` but adds per-outcome random-effects variance components and the full correlation matrix of D.

Usage

```
## S3 method for class 'longitudinalSub.BJM'
summary(object, digits = 4, ...)
```

Arguments

`object`	A `longitudinalSub.BJM` object.
`digits`	Number of significant digits. Default is 4.
`...`	Additional arguments (currently unused).

Value

Invisibly returns a list of per-outcome `summary.lme` objects.

`summary.survivalSub.BJM` *Summary method for* `survivalSub.BJM` *objects*

Description

Called via `summary(survival_fit_all)`. Returns (and prints) an extended summary including baseline hazard range, BIC, and McFadden R2 for the competing-risks GLM.

Usage

```
## S3 method for class 'survivalSub.BJM'
summary(object, digits = 4, ...)
```

Arguments

`object`	A `survivalSub.BJM` object returned by `survivalSub`.
`digits`	Number of significant digits. Default is 4.
`...`	Additional arguments (currently unused).

Value

Invisibly returns a list with components `cox_summary` and (if competing risks) `glm_summary`.

Examples

```
data(pbc3)
data.survival.fitting <- pbc3[!duplicated(pbc3$id), ]
formMarginalSurv  <- Surv(years, status3) ~ age + sex
formConditionalCR <- status4 ~ years + age + sex
survival_fit_all  <- survivalSub(data.survival.fitting,
                                 formMarginalSurv, formConditionalCR)
summary(survival_fit_all)
```

---

survivalSub *Fitting survival sub-model*

Description

Fitting survival sub-model

Usage

```
survivalSub(data.survival.fitting, formMarginalSurv, formConditionalCR)
```

Arguments

`data.survival.fitting`	Input data containing survival outcomes and baseline covariates.
`formMarginalSurv`	Survival input formats.
`formConditionalCR`	Competing risks input formats.

Value

Model fitting results of survival sub-model with or without competing risks.

Examples

```
data(pbc3)

data.survival.fitting =  pbc3[!duplicated(pbc3$id), ]

formMarginalSurv = Surv(years, status3) ~ age + sex
formConditionalCR = NULL

survival_fit_all = survivalSub(data.survival.fitting, formMarginalSurv,
                               formConditionalCR)
```

Index

∗ **datasets**
pbc2, 11
pbc3, 12

cmtPlot, 2

dynamicPrediction, 4
dynamicPredictionBio, 6

longitudinalSub, 9, *19, 20*

pbc, *12, 13*
pbc2, 11
pbc3, 12
predictPlot, 14
print.dynamicPrediction.BJM, 16
print.dynamicPredictionBio.BJM, 17
print.longitudinalSub.BJM, 18
print.survivalSub.BJM, 19
print_BJM, 20
print_dynamicPrediction
*(*print.dynamicPrediction.BJM*)*, 16
print_dynamicPredictionBio
*(*print.dynamicPredictionBio.BJM*)*, 17
print_longitudinalSub
*(*print.longitudinalSub.BJM*)*, 18
print_survivalSub
*(*print.survivalSub.BJM*)*, 19

riskPlot, 20

summary.dynamicPrediction.BJM, 21
summary.dynamicPredictionBio.BJM, 22
summary.longitudinalSub.BJM, 23
summary.survivalSub.BJM, 23
survivalSub, *19, 20*, *23*, 24