Package 'RISCA'

May 2, 2022

Type Package

Title Causal Inference and Prediction in Cohort-Based Analyses

Version 1.0.1

Depends R (>= 3.5.0), splines, survival, relsurv, reticulate

Imports date, graphics, nlme, MASS, mvtnorm, statmod, parallel,
doParallel, foreach, nnet, kernlab, glmnet, caret,
SuperLearner, zoo, flexsurv, randomForestSRC, survivalmodels,
prodlim, hdnom, glmnetUtils, dplyr, dvmisc, mosaic, mosaicCalc,
cubature, timeROC, rpart

Description Numerous functions for cohort-based analyses, either for prediction or causal infer-
ence. For causal inference, it includes Inverse Probability Weighting and G-
computation for marginal estimation of an exposure effect when confounders are ex-
pected. We deal with binary outcomes, times-to-events, competing events, and multi-
state data. For multistate data, semi-Markov model with interval censoring may be consid-
ered, and we propose the possibility to consider the excess of mortality related to the disease com-
pared to reference lifetime tables. For predictive studies, we propose a set of functions to esti-
mate time-dependent receiver operating characteristic (ROC) curves with the possible considera-
tion of right-censoring times-to-events or the presence of confounders. Finally, several func-
tions are available to assess time-dependent ROC curves or survival curves from aggregated data.

License GPL (>= 2)

LazyLoad yes

URL www.labcom-risca.com

NeedsCompilation no

Author Yohann Foucher [aut, cre] (<https://orcid.org/0000-0003-0330-7457>),
Florent Le Borgne [aut],
Etienne Dantan [aut],
Florence Gillaizeau [aut],
Arthur Chatton [aut],
Christophe Combescure [aut],
Camille Sabathe [aut]

Maintainer Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Repository CRAN

Date/Publication 2022-05-02 13:40:02 UTC

R **topics documented:**

aft.gamma	*Library of the Super Learner for an accelerated failure time (AFT) parametric model with a gamma distribution*

Description

Fit an AFT parametric model with a gamma distribution.

Usage

```
aft.gamma(times, failures, group, cov.quanti, cov.quali, data)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).

Details

The model is obtained by using the dist="gamma" in the flexsurvreg package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled times and corresponds to the observed follow-up times. The second column is entitled failures and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Jackson, C. (2016). flexsurv: A Platform for Parametric Survival Modeling in R. Journal of Statistical Software, 70(8), 1-33. doi:10.18637/jss.v070.i08

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-aft.gamma(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
    col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

| aft.ggamma | *Library of the Super Learner for an accelerated failure time (AFT) parametric model with a generalized gamma distribution* |

Description

Fit an AFT parametric model with a generalized gamma distribution.

Usage

```
aft.ggamma(times, failures, group, cov.quanti, cov.quali,
data)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.

data   A data frame for training the model in which to look for the variables related
                    to the status of the follow-up time (`times`), the event (`failures`), the optional
                    treatment/exposure (`group`) and the covariables included in the previous model
                    (`cov.quanti` and `cov.quali`).

Details

The model is obtained by using the `dist="gengamma"` in the `flexsurvreg` package.

Value

model   The estimated model.

group   The name of the variable related to the exposure/treatment.

cov.quanti   The name(s) of the variable(s) related to the possible quantitative covariates.

cov.quali   The name(s) of the variable(s) related to the possible qualitative covariates.

data   The data frame used for learning. The first column is entitled `times` and corre-
                    sponds to the observed follow-up times. The second column is entitled `failures`
                    and corresponds to the event indicators. The other columns correspond to the
                    predictors.

times   A vector of numeric values with the times of the `predictions`.

hazard   A vector of numeric values with the values of the cummulative baseline hazard
                    function at the prediction `times`.

predictions   A matrix with the predictions of survivals of each subject (lines) for each ob-
                    served times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Jackson, C. (2016). flexsurv: A Platform for Parametric Survival Modeling in R. Journal of Statistical Software, 70(8), 1-33. doi:10.18637/jss.v070.i08

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-aft.ggamma(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

aft.llogis	*Library of the Super Learner for an accelerated failure time (AFT) parametric model with a log logistic distribution*

Description

Fit an AFT parametric model with a log logistic distribution.

Usage

```
aft.llogis(times, failures, group, cov.quanti, cov.quali, data)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).

Details

The model is obtained by using the dist="llogis" in the flexsurvreg package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.

data	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the `predictions`.
hazard	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction `times`.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Jackson, C. (2016). flexsurv: A Platform for Parametric Survival Modeling in R. Journal of Statistical Software, 70(8), 1-33. doi:10.18637/jss.v070.i08

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-aft.llogis(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

aft.weibull	*Library of the Super Learner for an accelerated failure time (AFT)*
	parametric model with a Weibull distribution

Description

Fit an AFT parametric model with a Weibull distribution.

Usage

```
aft.weibull(times, failures, group, cov.quanti, cov.quali=NULL,
data, newdata, newtimes)
```

Arguments

`times`	The name of the variable related the numeric vector with the follow-up times.
`failures`	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
`group`	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
`cov.quanti`	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
`cov.quali`	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
`data`	A data frame for training the model in which to look for the variables related to the status of the follow-up time (`times`), the event (`failures`), the optional treatment/exposure (`group`) and the covariables included in the previous model (`cov.quanti` and `cov.quali`).
`newdata`	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in `cov.quanti` and `cov.quali`. The default value is NULL, the predicted values are computed for the subjects of the training sample.
`newtimes`	The times at which to produce predicted values. The default value isNULL, the predicted values are computed for the observed times in the training data frame.

Details

The model is obtained by using the `dist="weibull"` in the `flexsurvreg` package.

Value

`model`	The estimated model.
`group`	The name of the variable related to the exposure/treatment.
`cov.quanti`	The name(s) of the variable(s) related to the possible quantitative covariates.
`cov.quali`	The name(s) of the variable(s) related to the possible qualitative covariates.
`data`	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
`times`	A vector of numeric values with the times of the `predictions`.
`hazard`	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction `times`.
`predictions`	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Jackson, C. (2016). flexsurv: A Platform for Parametric Survival Modeling in R. Journal of Statistical Software, 70(8), 1-33. doi:10.18637/jss.v070.i08

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-aft.weibull(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

auc   *Area Under ROC Curve From Sensitivities And Specificities.*

Description

This function computes the area under ROC curve by using the trapezoidal rule.

Usage

```
auc(sens, spec)
```

Arguments

sens	A numeric vector with the sensitivities
spec	A numeric vector with the specificities

Details

This function computes the area under ROC curve using the trapezoidal rule from two vectors of sensitivities and specificities. The value of the area is directly returned.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

Examples

```
se.temp <- c(0, 0.5, 0.5, 1)
sp.temp <- c(1, 0.5, 0.5, 0)
auc(se.temp, sp.temp)
```

---

cox.aic	*Library of the Super Learner for Cox univariate significant model*

Description

Fit a Cox regression for a selection of covariate

Usage

```
cox.aic(times, failures, group,cov.quanti, cov.quali,
data, final.model.cov)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali)
final.model.cov	
	The final covariable to adjust the fit

Details

The univariate Cox regression is obtained by using the survival package.

Value

`model`	The estimated model.
`group`	The name of the variable related to the exposure/treatment.
`cov.quanti`	The name(s) of the variable(s) related to the possible quantitative covariates.
`cov.quali`	The name(s) of the variable(s) related to the possible qualitative covariates.
`data`	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
`times`	A vector of numeric values with the times of the `predictions`.
`hazard`	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction `times`.
`predictions`	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-cox.aic(times="times", failures="failures", data=dataDIVAT2,
 final.model.cov=c("age"), cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

cox.all *Library of the Super Learner for Cox Regression*

Description

Fit a Cox regression for all covariates to be used in the Super Learner

Usage

```
cox.all(times, failures, group, cov.quanti, cov.quali,
data)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).

Details

The Cox regression is obtained by using the survival package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.

data	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the `predictions`.
hazard	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction `times`.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Terry M. Therneau(2021). A Package for Survival Analysis in R. R package version 3.2-13, https://CRAN.R-project.org/package=survival.

Terry M. Therneau, Patricia M. Grambsch (2000). Modeling Survival Data: Extending the Cox Model. Springer, New York. ISBN 0-387-98784-3.

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-cox.all(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

cox.en   *Library of the Super Learner for Elastic Net Cox Regression*

Description

Fit an elastic net Cox regression for a given value of the regularization parameter.

Usage

```
cox.en(times, failures, group, cov.quanti, cov.quali,
data, alpha, lambda)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
alpha	The value of the regularization parameter alpha for penalizing the partial likelihood.
lambda	The value of the regularization parameter lambda for penalizing the partial likelihood.

Details

The elastic net Cox regression is obtained by using the glmnet package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled times and corresponds to the observed follow-up times. The second column is entitled failures and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-cox.en(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), lambda=.1, alpha=.1)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

cox.lasso   *Library of the Super Learner for Lasso Cox Regression*

Description

Fit a Lasso Cox regression for a given value of the regularization parameter.

Usage

```
cox.lasso(times, failures, group, cov.quanti, cov.quali,
data, lambda)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.

cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
lambda	The value of the regularization parameter lambda for penalizing the partial likelihood.

Details

The Lasso Cox regression is obtained by using the glmnet package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled times and corresponds to the observed follow-up times. The second column is entitled failures and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-cox.lasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), lambda=1)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

cox.ridge   *Library of the Super Learner for Ridge Cox Regression*

Description

Fit a ridge Cox regression for a given value of the regularization parameter.

Usage

```
cox.ridge(times, failures, group, cov.quanti, cov.quali,
data, lambda)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
lambda	The value of the regularization parameter lambda for penalizing the partial likelihood.

Details

The ridge Cox regression is obtained by using the `glmnet` package.

Value

`model`	The estimated model.
`group`	The name of the variable related to the exposure/treatment.
`cov.quanti`	The name(s) of the variable(s) related to the possible quantitative covariates.
`cov.quali`	The name(s) of the variable(s) related to the possible qualitative covariates.
`data`	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
`times`	A vector of numeric values with the times of the `predictions`.
`hazard`	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction `times`.
`predictions`	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-cox.ridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), lambda=1)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

dataCSL   *CSL Liver Chirrosis Data.*

Description

Survival status for the liver chirrosis patients of Schlichting et al.

Usage

```
data(dataCSL)
```

Format

This data frame contains the following columns:

id	a numeric vector related to the subject ID.
time	A numeric vector with the time of measurement.
prot	A numeric vector with the prothrombin level at measurement time.
dc	A numeric vector code. 0: censored observation, 1: died at eventT.
eventT	A numeric vector with the time of event (death).
treat	A numeric vector code. 0: active treatment of prednisone, 1: placebo treatment.
sex	A numeric vector code. 0: female, 1: male.
age	A numeric vector with the age of subject at inclusion time subtracted 60.
prot.base	A numeric vector with the prothrombin base level before entering the study
prot.prev	A numeric vector with the level of prothrombin at previous measurement time.
lt	A numeric vector with the starting time for the time-interval.
rt	A numeric vector with the stopping time for the time-interval.

Source

The timreg package

References

Schlichting et al. Prognostic factors in cirrhosis identified by Cox's regression model. Hepatology. 1983 Nov-Dec;3(6):889-95. <doi https://doi.org/ 10.1002/ hep.1840030601>

Examples

```
data(dataCSL)
names(dataCSL)
```

---

dataDIVAT1   *A First Sample From The DIVAT Data Bank.*

Description

A data frame with 5943 French kidney transplant recipients from the DIVAT cohort.

Usage

```
data(dataDIVAT1)
```

Format

A data frame with 5943 observations for the 7 following variables:

trajectory A numeric vector with the sequences of observed states. The patient evolution can be
described according to a 4-state structure: X=1 represents the healthy state, X=2 represents
the acute rejection episode, X=3 the definitive return to dialysis and X=4 the death. These
times can be right-censored. A vector of covariates is also collected at the transplantation, i.e.
the baseline of the cohort.

time1 A numeric vector with the times (in days) between the transplantation and the first clinical
event (acute rejection episode, return to dialysis, or death with a functioning graft), or the
times to censoring if trajectory=1.

time2 A numeric vector with the time between the transplantation and the second clinical event (re-
turn to dialysis or death with a functioning graft), or the the time to censoring if trajectory=12.

ageR A numeric vector with the recipient age (in years) at the transplantation.

sexR A character vector with the recipient gender.

year.tx A numeric vector with the calendar year of the transplantation.

z A numeric vector represents the explicative variable under interest, i.e. the delayed graft function
(1=yes, 0=no).

Source

URL: www.divat.fr

Examples

```
data(dataDIVAT1)

### a description of transitions
table(dataDIVAT1$trajectory)

### patient-graft survival (first event between the return to dialysis and the patient
### death with a functioning graft)

dataDIVAT1$failure<-1*(dataDIVAT1$trajectory!=1 & dataDIVAT1$trajectory!=12)
```

```
dataDIVAT1$time<-NA
dataDIVAT1$time<-ifelse(dataDIVAT1$trajectory %in% c(1,12,13,14),
dataDIVAT1$time1,dataDIVAT1$time1+dataDIVAT1$time2)

plot(survfit(Surv(time/365.24, failure) ~ 1 , data=dataDIVAT1), mark.time=FALSE,
        xlim=c(0,12), ylim=c(0,1), cex=1.5, col=1, lwd=2, lty=1,
        xlab="Times after the transplantation (years)",
        ylab="Patient-graft survival")
```

---

dataDIVAT2   *A Second Sample From the DIVAT Data Bank.*

Description

A data frame with 1912 French kidney transplant recipients from the DIVAT cohort.

Usage

```
data(dataDIVAT2)
```

Format

A data frame with the 4 following variables:

age This numeric vector provides the age of the recipient at the transplantation (in years).

hla This numeric vector provides the indicator of transplantations with at least 4 HLA incompatibilities between the donor and the recipient (1 for high level and 0 otherwise).

retransplant This numeric vector provides the indicator of re-transplantation (1 for more than one transplantation and 0 for first kidney transplantation).

ecd The Expended Criteria Donor (1 for transplantations from ECD and 0 otherwise). ECD are defined by widely accepted criteria, which includes donors older than 60 years of age or 50-59 years of age with two of the following characteristics: history of hypertension, cerebrovascular accident as the cause of death or terminal serum creatinine higher than 1.5 mg/dL.

times This numeric vector is the follow up times of each patient.

failures This numeric vector is the event indicator (0=right censored, 1=event). An event is considered when return in dialysis or patient death with functioning graft is observed.

Source

URL: www.divat.fr

References

Le Borgne F, Giraudeau B, Querard AH, Giral M and Foucher Y. Comparisons of the performances of different statistical tests for time-to-event analysis with confounding factors: practical illustrations in kidney transplantation. Statistics in medicine. 30;35(7):1103-16, 2016. <doi:10.1002/sim.6777>

Examples

```
data(dataDIVAT2)

# Compute the non-adjusted Hazard Ratio related to the ECD versus SCD
cox.ecd<-coxph(Surv(times, failures) ~ ecd, data=dataDIVAT2)
summary(cox.ecd) # Hazard Ratio = 1.97
```

---

dataDIVAT3   *A Third Sample From the DIVAT Data Bank.*

Description

A data frame with 4267 French kidney transplant recipients.

Usage

```
data(dataDIVAT3)
```

Format

A data frame with 4267 observations for the 8 following variables.

ageR This numeric vector represents the age of the recipient (in years)

sexeR This numeric vector represents the gender of the recipient (1=men, 0=female)

year.tx This numeric vector represents the year of the transplantation

ante.diab This numeric vector represents the diabetes statute (1=yes, 0=no)

pra This numeric vector represents the pre-graft immunization using the panel reactive antibody (1=detectable, 0=undetectable)

ageD This numeric vector represents the age of the donor (in years)

death.time This numeric vector represents the follow up time in days (until death or censoring)

death This numeric vector represents the death indicator at the follow-up end (1=death, 0=alive)

Source

URL: www.divat.fr

References

Le Borgne et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. Manuscript submitted. Stat Methods Med Res. 27(11):3397-3410, 2018. <doi: 10.1177/ 0962280217702416.>

Examples

```
data(dataDIVAT3)

### a short summary of the recipient age at the transplantation
summary(dataDIVAT3$ageR)

### Kaplan and Meier estimation of the recipient survival
plot(survfit(Surv(death.time/365.25, death) ~ 1, data = dataDIVAT3),
 xlab="Post transplantation time (in years)", ylab="Patient survival",
 mark.time=FALSE)
```

---

dataDIVAT4 *A Fourth Sample From the DIVAT Data Bank.*

Description

A data frame with 6648 French kidney transplant recipients from the DIVAT cohort. According to this data set, patient and graft survival can be computed for each hospital. This database was used by Combsecure et al. (2014).

Usage

```
data(dataDIVAT4)
```

Format

A data frame with the 3 following variables:

hospital This numeric vector represents the hospital in which the kidney transplantation was performed. Six hospitals were included.

time For right censored data, this numeric vector represents the follow up time (in days). When the event is observed, this numeric vector represents the exact time-to-event.

status This indicator describes the end of the follow-up. The status equals 0 for right censored data and 1 if the event is observed.

Details

The data were extracted from the prospective multicentric DIVAT cohort (www.divat.fr). Patients transplanted between 2000 and 2012 in 6 French centers were included. Only adult patients receiving a single kidney transplant and treated by Calcineurin inhibitors and Mycophenolate Mofetil for maintenance therapy after transplantation were considered. In parallel, patients who received multi-organ transplantation were excluded from the study. The time-to-event under interest is the time between the kidney transplantation and the graft failure, i.e. the first event between the return in dialysis or the death of the patient with functional kidney.

References

Combescure et al. The multivariate DerSimonian and Laird's methodology applied to meta-analysis of survival curves. 10;33(15):2521-37, 2014. Statistics in Medicine. <doi:10.1002/sim.6111>

Examples

```
data(dataDIVAT4)

divat.surv <- survfit(Surv(time/365.24, status) ~ hospital, data = dataDIVAT4)

plot(divat.surv, lty = 1:6, col=1:6, lwd=2, mark.time=FALSE,
 xlab="Post transplantation time (days)", ylab="Patient and graft survival")
```

---

dataDIVAT5   *The Aggregated Kidney Graft Survival Stratified By The 1-year Serum Creatinine.*

Description

A data frame that presents the aggregated outcomes per center of 5943 French kidney transplant recipients from the DIVAT cohort. It was used by Combescure et al. (2016).

Usage

```
data(dataDIVAT5)
```

Format

A data frame with 106 observations for the 8 following variables:

classe This numeric vector represents the groups of recipients defined using the 1-year serum creatinine. 1 is the first group with the lowest values of 1-year serum creatinine.

n This numeric vector represents the number of recipients at the baseline (date of the transplantation) in each group.

year This numeric vector represents the post transplant time (in years).

surv This numeric vector represents the survival probabilities at each year (obtained using the Kaplan and Meier estimator from the individual data).

n.risk This numeric vector represents the number of subjects at-risk of the event at the corresponding year.

proba This numeric vector represents the proportion of the patients in a center which belong to the corresponding group.

marker.min This numeric vector represents the minimum value of the interval of the 1-year serum creatinine (in $\mu mol/l$).

marker.max This numeric vector represents the maximum value of the interval of the 1-year serum creatinine (in $\mu mol/l$).

centre.num This numeric vector represents the centers.

Details

The immunology and nephrology department of the Nantes University hospital constituted a data bank with the monitoring of medical records for kidney and/or pancreas transplant recipients. Here, we considered a subpopulation of 4195 adult patients and who had received a first kidney graft between January 1996 and Jun 2008. Five centers participated. A total of 511 graft failures were observed (346 returns to dialysis and 165 deaths with a functional kidney). Based on this database, we constructed an aggregated dataset to perform a meta-analysis on 5 published monocentric studies. The medical objective was to evaluate whether 1-year serum creatinine (Cr) is a good predictive marker of graft failure. Cr is a breakdown product and is removed from the body by the kidneys. If kidney function is abnormal, blood Cr levels increase.

Source

URL: www.divat.fr

References

Combescure et al. A literature-based approach to evaluate the predictive capacity of a marker using time-dependent Summary Receiver Operating Characteristics. Stat Methods Med Res, 25(2):674-85, 2016. <doi: 10.1177/ 0962280212464542>

Examples

```
data(dataDIVAT5)

# Kaplan Meier estimations of the graft survival in the first center
plot(dataDIVAT5$year[dataDIVAT5$centre.num==1],
 dataDIVAT5$surv[dataDIVAT5$centre.num==1],
 xlab="Post transplantation time (years)", ylab="Graft survival",
 ylim=c(0.7,1), xlim=c(0, 9), type="n")

# Goup 1
lines(c(0, dataDIVAT5$year[dataDIVAT5$centre.num==1 &
 dataDIVAT5$classe==1]),
 c(1, dataDIVAT5$surv[dataDIVAT5$centre.num==1 &
 dataDIVAT5$classe==1]),
 type="b", col=1, lty=1, lwd=2)

# Goup 2
lines(c(0, dataDIVAT5$year[dataDIVAT5$centre.num==1 &
 dataDIVAT5$classe==2]),
 c(1, dataDIVAT5$surv[dataDIVAT5$centre.num==1 &
 dataDIVAT5$classe==2]),
 type="b", col=2, lty=2, lwd=2)

# legend
legend("bottomleft", c("group #1 (1-year Cr<4.57)",
 "group #2 (1-year Cr>4.57)"), col=c(1, 2),
 lty=c(1, 2), lwd=c(2, 2))
```

dataFTR	*Data for First Kidney Transplant Recipients.*

Description

Data were extracted from the DIVAT cohort. It corresponds to the reference sample constituted by first transplant recipients (FTR).

Usage

```
data(dataFTR)
```

Format

A data frame with the 4 following variables:

Tps.Evt This numeric vector provides the post-transplantation time (in days).

Evt This numeric vector provides the indicator of graft failure at the end of the follow-up (1 for failure and 0 for right censoring).

ageR2cl This numeric vector provides the recipient age at transplantation (1 for older than 55 years and 0 otherwise).

sexeR This numeric vector provides the recipient gender (1 for men and 0 for women).

Details

First transplant recipients (FTR) constituted the reference group. Recipients older than 18 years at the date of transplantation between 1996 and 2010 were selected from the French DIVAT (www.divat.fr/en) multicentric prospective cohort. Only recipients with a maintenance therapy with calcineurin inhibitors, mammalian target of rapamycin inhibitors or belatacept, in addition to mycophenolic acid and steroid were included. Simultaneous transplantations were excluded. Two explicative variables are proposed: recipient age at transplantation and recipient gender.

References

K. Trebern-Launay, M. Giral, J. Dantal and Y. Foucher. Comparison of the risk factors effects between two populations: two alternative approaches illustrated by the analysis of first and second kidney transplant recipients. BMC Med Res Methodol. 2013 Aug 6;13:102. <doi: 10.1186/1471-2288-13-102>

Examples

```
data(dataFTR)

# Compute a Cox PH model with both explicative variables
summary(coxph(Surv(Tps.Evt/365.24, Evt) ~ ageR2cl + sexeR, data=dataFTR))
```

dataHepatology	*The Data Extracted From The Meta-Analysis By Cabibbo et al. (2010).*

Description

Data were extracted from the studies included in the meta-analysis by Cabibbo et al. which aimed to assess the survival rate in untreated patients with hepatocellular carcinoma.

Usage

```
data(dataHepatology)
```

Format

A data frame with with the 8 following variables:

study This numeric vector represents number of the study.

first.author This vector represents the name of the first author.

year.pub This numeric vector represents the publication year.

time This numeric vector represents the times for which the survival rates are collected in years.

survival This numeric vector represents the survival rates for each value of time

n.risk This numeric vector represents the number of at-risk patients for each value of time

location This factor indicates the location of the study (Asia, North Amercia or Europe)

design This factor indicates if the study is monocentric ou multicentric.

Details

The survival probabilities were extracted from the published survival curves each month during the first six months and then by step of three months. The pictures of the curves were digitalized using the R package ReadImage and the probabilities were extracted using the package digitize proposed by Poisot. The numbers of at-risk patients for each interval of time were derived from the numbers of at-risk patients reported in the studies, and using the methods of Parmar or Williamson to account for censorship. Studies have different length of follow-up. For each study, survival probabilities and the numbers of at-risk patients were collected at all points in time before the end of follow-up.

References

Cabibbo, G., et al., A meta-analysis of survival rates of untreated patients in randomized clinical trials of hepatocellular carcinoma. Hepatology, 2010. 51(4): p. 1274-83.

Poisot, T., The digitize Package: Extracting Numerical Data from Scatter-plots. The R Journal, 2011. 3(1): p. 25-26.

Parmar, M.K., V. Torri, and L. Stewart, Extracting summary statistics to perform meta-analyses of the published literature for survival endpoints. Stat Med, 1998. 17(24): p. 2815-34

Examples

```
data(dataHepatology)
times <- dataHepatology$time
survival <- dataHepatology$survival
study  <- dataHepatology$study

plot(times, survival, type="n",
 ylim=c(0,1), xlab="Time",ylab="Survival")

for (i in unique(sort(study)))
{
lines(times[study==i], survival[study==i], type="l", col="grey")
points(max(times[study==i]),
 survival[study==i & times == max( times[study==i])], pch=15)
}
```

---

dataKi67   *The Aggregated Data Published By de Azambuja et al. (2007).*

Description

The aggregated data from the meta-analysis proposed by Azambuja et al. (2007).

Usage

```
data(dataKi67)
```

Format

A data frame with 406 observations (rows) with the 10 following variables (columns).

classe This numeric vector represents the groups of patients defined using KI-67. 1 is the first group which is defined by the lowest KI-67 values.

n This numeric vector represents the number of recipients at the baseline (date of KI-67 collection) in each group.

year This numeric vector represents the survival time (in years).

surv This numeric vector represents the survival probabilities at each year (obtained using the Kaplan and Meier estimator from the published papers).

nrisk This numeric vector represents the number of subjects at-risk of the event at the corresponding year.

proba This numeric vector represents the proportion of the patients for a given paper which belong to the corresponding group.

log.marker.min This numeric vector represents the logarithm of the minimum value of the KI-67 interval.

log.marker.max This numeric vector represents the logarithm of the maximum value of the KI-67 interval.

study.num This numeric vector identifies the studies.

author This character vector identifies the first author of the paper.

year.paper This numeric vector identifies the year of publication.

Details

KI-67 is a marker of the proliferative activity of breast cancer, but its prognostic capacity is still unclear. In their meta-analysis, de Azambuja et al. (2007) concluded that KI-67 positivity conferred a worse survival. This work focused on the 35 evaluable studies of the relationship between KI-67 and the overall survival. 23 studies described survival curves according to the level of KI-67. Survival probabilities were measured every year.

References

de Azambuja et al. Ki-67 as prognostic marker in early breast cancer: a meta-analysis of published studies involving 12 155 patients. British Journal of Cancer. 96:1504-1513, 2007. <doi: 10.1038/sj.bjc.6603756>

Examples

```
data(dataKi67)

# Kaplan Meier estimations of graft survivals in Wintzer et al. (1991)
plot(dataKi67$year[dataKi67$study.num==1],
 dataKi67$surv[dataKi67$study.num==1],
 xlab="Post transplantation time (years)",
 ylab="Graft survival", ylim=c(0.6,1), xlim=c(0, 4), type="n")

# Goup 1
lines(c(0, dataKi67$year[dataKi67$study.num==1 & dataKi67$classe==1]),
 c(1, dataKi67$surv[dataKi67$study.num==1 & dataKi67$classe==1]),
 type="b", col=1, lty=1, lwd=2)

# Goup 2
lines(c(0, dataKi67$year[dataKi67$study.num==1 & dataKi67$classe==2]),
 c(1, dataKi67$surv[dataKi67$study.num==1 & dataKi67$classe==2]),
 type="b", col=2, lty=2, lwd=2)

# legend
legend("bottomleft", c("group #1 (log Ki67 < 2.49)",
 "group #2 (log Ki67 > 2.49)"), col=c(1, 2), lty=c(1, 2), lwd=c(2, 2))
```

---

dataKTFS   *A Sixth Sample Of The DIVAT Cohort.*

Description

A data frame with 2169 French kidney transplant recipients for who the Kidney Transplant Failure Score (KTFS) was collected. The KTFS is a score proposed by Foucher et al. (2010) to stratify the recipients according to their risk of return in dialysis.

Usage

```
data(dataKTFS)
```

Format

A data frame with 2169 observations for the 3 following variables:

time This numeric vector represents the follow up time in years (until return in dialyis or censoring)

failure This numeric vector represents the graft failure indicator at the follow-up end (1=return, 0=censoring)

score This numeric vector represents the KTFS values.

Source

URL: www.divat.fr

References

Foucher Y. al. A clinical scoring system highly predictive of long-term kidney graft survival. Kidney International, 78:1288-94, 2010. <doi:10.1038/ki.2010.232>

Examples

```
data(dataKTFS)

### a short summary of the recipient age at the transplantation
summary(dataKTFS$score)

### Kaplan and Meier estimation of the recipient survival
plot(survfit(Surv(time, failure) ~ I(score>4.17), data = dataKTFS),
 xlab="Post transplantation time (in years)", ylab="Patient survival",
 mark.time=FALSE, col=c(2,1), lty=c(2,1))

legend("bottomleft", c("Recipients in the high-risk group",
 "Recipients in the low-risk group"), col=1:2, lty=1:2)
```

dataOFSEP *A Simulated Sample From the OFSEP Cohort.*

Description

A data frame with 1300 simulated French patients with multiple sclerosis from the OFSEP cohort. The baseline is 1 year after the initiation of the first-line treatment.

Usage

```
data(dataOFSEP)
```

Format

A data frame with 1300 observations for the 3 following variables:

time This numeric vector represents the follow up time in years (until disease progression or censoring)

event This numeric vector represents the disease progression indicator at the follow-up end (1=progression, 0=censoring)

age This numeric vector represents the patient age (in years) at baseline.

duration This numeric vector represents the disease duration (in days) at baseline.

period This numeric vector represents the calendar period: 1 in-between 2014 and 2018, and 0 otherwise.

gender This numeric vector represents the gender: 1 for women.

relapse This numeric vector represents the diagnosis of at least one relapse since the treatment initiation : 1 if at leat one event, and 0 otherwise.

edss This vector of character string represents the EDSS level : "miss" for missing, "low" for EDSS between 0 to 2, and "high" otherwise.

t1 This vector of character string represents the new gadolinium-enhancing T1 lesion : "missing", "0" or "1+" for at least 1 lesion.

t2 This vector of character string represents the new T2 lesions : "no" or "yes".

rio This numeric vector represents the modified Rio score.

References

Sabathe C et al. SuperLearner for survival prediction from censored data: extension of the R package RISCA. Submited.

Examples

```
data(dataOFSEP)

### Kaplan and Meier estimation of the disease progression free survival
plot(survfit(Surv(time, event) ~ 1, data = dataOFSEP),
     ylab="Disease progression free survival",
     xlab="Time after the first anniversary of the first-line treatment in years")
```

dataSTR *Data for Second Kidney Transplant Recipients.*

Description

Data were extracted from the DIVAT cohort. It corresponds to the relative sample constituted by second transplant recipients (STR).

Usage

```
data(dataSTR)
```

Format

A data frame with the 6 following variables:

Tps.Evt This numeric vector provides the post transplantation time (in days).

Evt This vector provides the indicators of graft failure at the end of the follow-up (1 for failures and 0 for right censoring).

ageR2cl This numeric vector provides the recipient age at transplantation (1 for older than 55 years and 0 otherwise).

sexeR This numeric vector provides the recipient gender (1 for men and 0 for women).

ageD2cl This numeric vector provides the donor age (1 for older than 55 years and 0 otherwise).

Tattente2cl This numeric vector provides the waiting time in dialysis between the two consecutive transplantations (1 for more than 3 years and 0 otherwise).

Details

Second transplant recipients (STR) constituted the relative group of interest. Recipients older than 18 years at the date of transplantation between 1996 and 2010 were selected from the French DIVAT (www.divat.fr/en) multicentric prospective cohort. Only recipients with a maintenance therapy with calcineurin inhibitors, mammalian target of rapamycin inhibitors or belatacept, in addition to mycophenolic acid and steroid were included. Simultaneous transplantations were excluded. Two explicative variables are common with the reference group: recipient age at transplantation and recipient gender. Two explicative variables are specific to STR: donor age and waiting time in dialysis between the two consecutive transplantations.

References

K. Trebern-Launay, M. Giral, J. Dantal and Y. Foucher. Comparison of the risk factors effects between two populations: two alternative approaches illustrated by the analysis of first and second kidney transplant recipients. BMC Med Res Methodol. 2013 Aug 6;13:102. <doi: 10.1186/1471-2288-13-102>

Examples

```
data(dataSTR)

# Compute a Cox PH model
summary(coxph(Surv(Tps.Evt/365.24, Evt) ~ ageR2cl + Tattente2cl, data=dataSTR))
```

---

differentiation   *Numerical Differentiation with Finite Differences.*

Description

This function allows to approximate the derivative of a function by the finite differences method.

Usage

```
differentiation(x, fx)
```

Arguments

x   A numeric vector with the x values.

fx   A numeric vector with the $f(x)$ values.

Details

Numerical differentiation is a method of approximating the derivative of a function f at particular value x.

Examples

```
x <- c(0.0, 0.2, 0.4)
fx <- c(0.00000, 0.74140, 1.3718)
differentiation(x, fx)
```

---

expect.utility1   *Cut-Off Estimation Of A Prognostic Marker (Only One Observed
                    Group).*

Description

This function allows the estimation of a cut-off for medical decision making between two treatments
A and B from a prognostic marker by maximizing the expected utility in a time-dependent context.
Only the observations of one group are available.

Usage

```
expect.utility1(times, failures, variable, pro.time, u.A0, u.A1, u.B0, u.B1,
 n.boot, rmst.change)
```

Arguments

times
: A numeric vector with the follow up times for the patients receiving the treatment B.

failures
: A numeric vector with the event indicator for the patients receiving the treatment B (0=right censoring, 1=event).

variable
: A numeric vector with the observed values of the marker under interest X for the patients receiving the treatment B. This variable is collected at the baseline (times=0). By convention, we assume that patients with X>k will preferentially receive A, k being the optimal cut-off. In contrast, patients with X<k will receive preferentially B.

pro.time
: The prognostic time for which the prognostic capacities of the marker and the patient outcomes are considered in the same unit than the one used in the argument times.

u.A0
: A value of the utility of a patient receiving the treatment A before the event occurrence. This value should respect the 0-1 scale (from death to perfect health).

u.A1
: A value of the utility of a patient receiving the treatment A after the event occurrence. This value should respect the 0-1 scale.

u.B0
: A value of the utility of a patient receiving the treatment B before the event occurrence. This value should respect the 0-1 scale.

u.B1
: A value of the utility of a patient receiving the treatment B after the event occurrence. This value should respect the 0-1 scale.

n.boot
: Number of bootstrap iterations to compute the 95% confidence interval of the optimal cut-off. The default value is NULL: no confidence interval is estimated.

rmst.change
: A numeric vector with the expected relative change in the Restricted Mean Survival Time (RMST) by using the treatment A instead of the treatment B among patients with X>k.

Details

The user observes a cohort of patients receiving the treatment B. She(he) assumes that an alternative treatment A would be more convenient for patients with high-values of the marker X. She(he) aims to compute the optimal cut-off value for a future stratified medical decision rule: treatment A for patients with X>k and treatment B for patients with X<k. The user has to enter the observed cohort of patients with the treatment B. Additional to the assumptions related to health-state utilities, the user have to specify in rmst.change, i.e. the expected relative change in terms of RMST between the two treatments. For instance, if the observed life expectancy of a patient with treatment B over the next 8 years (value entered in pro.time) is 6.70 years, and assuming that the treatment A increases this life expectancy during the next 8 years by 1.33 years, the expected relative change in RMST is 0.20 (=1.33/6.7).

Value

estimation	This is a single value if `n.boot=NULL`, which corresponds to the estimated cut-off that maximizes the time-dependent expected utility of the medical decision. If this value corresponds to the minimum of the marker, all the patients should be treated with A. If this value corresponds to the maximum of the marker, all the patients should conserve the treatment B. When `n.boot` is not null, two additional values are returned: `CIinf` is the lower bound of the 95% confidence interval and `CIsup` is the upper bound of the 95% confidence interval.
max.eu	This value corresponds to the maximum expected utility associated with the `estimation`.
table	This data frame is composed by 8 columns representing respectively the cut-off values, the time-dependent expected utilities (`utility`), the proportions of patients with a marker value higher (`pA`) and lower (`pB`) than the cut-off value, the numbers of QALYs for patients with a marker value higher (`qA`) and lower (`qB`) than the cut-off value, the RMST for patients with a marker value higher (`eA`) and lower (`eB`) than the cut-off value.
delta.rmst	This value represents the expected RMST for patients with a marker higher than the `estimation` (treated with A) minus the observed RMST for patients with a marker higher than the `estimation`(treated with B).
delta.qaly	This value represents the number of QALYs for patients with a marker higher than the `estimation` (treated with A) minus the observed number of QALYs for patients with a marker higher than the `estimation` (treated with B).
missing	Number of deleted observations due to missing data.

Author(s)

Etienne Dantan <Etienne.Dantan@univ-nantes.fr>

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Dantan et al. Optimal threshold estimator of a prognostic marker by maximizing a time-dependent expected utility function for a patient-centered stratified medicine. Statistical Methods in Medical Research, 27(6) :1847-1859. 2016. <doi:10.1177/ 0962280216671161>

Examples

```
data(dataKTFS)

# to respect the CRAN policy (run times < 5s), we reduced the database
# to the first 1500 patients. Removed the first line to use the antire sample.

dataKTFS <- dataKTFS[1:1500,]
dataKTFS$score <- round(dataKTFS$score, 1)

# the expected utility function for a prognostic up to 8 years
```

```
EUt.obj <- expect.utility1(dataKTFS$time, dataKTFS$failure, dataKTFS$score,
 pro.time=8, u.A0=0.81*0.95, u.A1=0.53, u.B0=0.81, u.B1=0.53, rmst.change=0.2)

plot(EUt.obj$table$cut.off, EUt.obj$table$utility, type="l",
 xlab="Cut-off values", ylab="Expected utility", col=1, lty=1)

segments(EUt.obj$estimation, 0, EUt.obj$estimation, EUt.obj$max.eu, lty=3)
segments(0, EUt.obj$max.eu, EUt.obj$estimation, EUt.obj$max.eu, lty=3)

text(EUt.obj$estimation-0.2, 6.22,
 paste("Optimal cut-off=", round(EUt.obj$estimation,2)), srt=90, cex=0.8)
 text(min(dataKTFS$score)+1.4, EUt.obj$max.eu-0.006,
 paste("Expected utility=", round(EUt.obj$max.eu, 2)), cex=0.8)

# the optimal cut-off: patients with an higher value should receive the treatment A

EUt.obj$estimation
```

| expect.utility2 | *Cut-Off Estimation Of A Prognostic Marker (Two Groups Are observed).* |

Description

This function allows the estimation of a cut-off of a prognostic marker for medical decision making between two treatments A and B by maximizing the expected utility in a time-dependent context.

Usage

```
expect.utility2(times, failures, variable, treatment, pro.time,
 u.A0, u.A1, u.B0, u.B1,  n.boot)
```

Arguments

times	A numeric vector with the follow up times.
failures	A numeric vector with the event indicator (0=right censoring, 1=event).
variable	A numeric vector with the observed values of the marker/variable under interest X. This variable is collected at the baseline (times=0). By convention, we assume that patients with X>k will preferentially receive A, k being the optimal cut-off. In contrast, patients with X<k will preferentially receive B.
treatment	A character vector with the observed treatment. Only character strings "A" and "B" are allowed.
pro.time	The prognostic time for which the capacities of the marker and the patient outcomes are considered in the same unit than the one used in the argument times.
u.A0	A value of the utility of a patient receiving the treatment A before the event occurrence. This value should respect the 0-1 scale (from death to perfect health).

u.A1	A value of the utility of a patient receiving the treatment A after the event occurrence. This value should respect the 0-1 scale.
u.B0	A value of the utility of a patient receiving the treatment B before the event occurrence. This value should respect the 0-1 scale.
u.B1	A value of the utility of a patient receiving the treatment B after the event occurrence. This value should respect the 0-1 scale.
n.boot	Number of bootstrap iterations to compute the 95% confidence interval of the optimal cut-off. The default value is NULL: no confidence interval is estimated.

Details

The treatments A and B are allocated independently to the value of the marker X. We assume that the treatment A will be more convenient for patients with high value of the marker X. The aim is to compute the optimal cut-off value for a future stratified medical decision rule: treatment A for patients with X>k and treatment B for patients with X<k.

Value

estimation	This is a single value if n.boot=NULL, which corresponds to the estimated cut-off that maximizes the time-dependent expected utility of the medical decision. If this value corresponds to the minimum of the marker, all the patients should be treated with A. If this value corresponds to the maximum of the marker, all the patients should conserve the treatment B. When n.boot is not null, two additional values are returned: CIinf is the lower bound of the 95% confidence interval and CIsup is the upper bound of the 95% confidence interval.
max.eu	This value corresponds to the maximum expected utility associated with the estimation.
table	This data frame is composed by 8 columns representing respectively the cut-off values (cut.off), the time-dependent expected utilities (utility), the proportions of patients with a marker value higher (pA) and lower (pB) than the cut-off value, the numbers of QALYs for patients with a marker value higher (qA) and lower (qB) than the cut-off value, the RMST for patients with a marker value higher (eA) and lower (eB) than the cut-off value.
delta.rmst	This is a vector with two values. The first value, entitled high.level, represents the RMST for patients with a marker higher than the estimation and treated with A minus the RMST for patients with a marker higher than the estimation and treated with B. The second value, entitled low.level, represents the RMST for patients with a marker lower than or equal to the estimation and treated by B minus the RMST for patients with a marker lower than or equal to the estimation and treated with A.
delta.qaly	This is a vector with two values. The first value, entitled high.level, represents the number of QALYs for patients with a marker higher than the estimation and treated by A minus the number of QALYs for patients with a marker higher than the estimation and treated with B. The second value, entitled low.level, represents the number of QALYs for patients with a marker lower than or equal to the estimation and treated with B minus the number of QALYs for patients with a marker lower than or equal to the estimation and treated with A.
missing	Number of deleted observations due to missing data.

Author(s)

Etienne Dantan <Etienne.Dantan@univ-nantes.fr>

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Dantan et al. Optimal threshold estimator of a prognostic marker by maximizing a time-dependent expected utility function for a patient-centered stratified medicine. Statistical Methods in Medical Research, 27(6) :1847-1859. 2016. <doi:10.1177/ 0962280216671161>

Examples

```
# import and attach the data example

data(dataCSL)
dataCSL <- dataCSL[order(dataCSL$id, dataCSL$time),]
dataCSL$ordre <-0
for (i in unique(dataCSL$id))
 {dataCSL$ordre[dataCSL$id==i] <- 1:sum(dataCSL$id==i)}

dataCSL$ttt[dataCSL$treat==0]<-"A"
dataCSL$ttt[dataCSL$treat==1]<-"B"

dataCSL0 <- dataCSL[dataCSL$ordre==1,]
dataCSL0<-dataCSL0[,c(1,4,5,14,9)]

# the expected utility function for a prognostic up to 8 years

EUt.obj <- expect.utility2(dataCSL0$eventT, dataCSL0$dc,
 dataCSL0$prot.base, treatment= dataCSL0$ttt,
 pro.time=8, u.A0=0.75*0.95, u.A1=0, u.B0=0.75, u.B1=0)

plot(EUt.obj$table$cut.off, EUt.obj$table$utility, type="l",
 xlab="Cut-off values", ylab="Expected utility",col=1, lty=1)

segments(EUt.obj$estimation, 0, EUt.obj$estimation, EUt.obj$max.eu, lty=3)
segments(0, EUt.obj$max.eu, EUt.obj$estimation, EUt.obj$max.eu, lty=3)

text(EUt.obj$estimation-2, 3.38,
 paste("Optimal cut-off=",round(EUt.obj$estimation,2)), srt=90, cex=0.8)
 text(min(dataCSL0$prot.base)+40, EUt.obj$max.eu-0.005,
 paste("Expected utility=",round(EUt.obj$max.eu,2)), cex=0.8)

# the optimal cut-off: patients with an higher value should receive the treatment A

EUt.obj$estimation
```

fr.ratetable   *Expected Mortality Rates of the General French Population*

Description

An object of class ratetable for the expected mortality of the French population. It is an array with three dimensions: age, sex and year.

Usage

```
data(fr.ratetable)
```

Format

The format is "ratetable". The attributes are:

dim	A numeric vector with the length of each dimension.
dimnames	A character vector with the names of each variable of the three dimensions.
dimid	A character vector with the identification of the dimensions: age, year and sex.
factor	A numeric vector of indicators equals to 1 if the corresponding dimension does not vary according to the time. Only the dimension related to sex equals 1.
cutpoints	A list of the thresholds to identify the changes in the mortality rates according to the time-dependent dimensions (NULL for sex).
class	The class of the object: ratetable.

Details

The organization of a ratetable object is described in details by Therneau (1999) and Pohar (2006). The original data and updates can be downloaded from the Human Life-Table Database (HMD, The Human Mortality Database).

Source

URL: www.mortality.org

References

Therneau and Offord. Expected Survival Based on Hazard Rates (Update), Technical Report, Section of Biostatistics, Mayo Clinic 63, 1999.

Pohar and Stare. Relative survival analysis in R. Computer methods and programs in biomedicine, 81: 272-278, 2006.

Examples

```
data(fr.ratetable)

is.ratetable(fr.ratetable)
```

---

gc.logistic   *Marginal Effect for Binary Outcome by G-computation.*

Description

This function allows to estimate the marginal effect of an exposure or a treatment by G-computation for binary outcomes.

Usage

```
gc.logistic(glm.obj, data, group, effect, var.method, iterations, n.cluster, cluster.type)
```

Arguments

glm.obj   A glm object obtained by using the function glm with the argument `family = binomial(link=logit)` to obtain a multivariate logistic regression. It shall be included the exposition/treatment of interest variable and at least one covariate.

data   A data frame in which to look for the variable related to the outcome, the treatment/exposure and the covariables included in the previous logistic regression `glm.obj`. The covariates' names have to be identical than the ones included in the logistic model.

group   The name or the variable related to the exposure/treatment variable. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed patients.

effect   The type of the marginal effect to be estimated. Three types are possible (see details): "ATE" (by default), "ATT" and "ATU".

var.method   The method to estimate the variances and the confidence intervals. Two methods are possible: "simulations" (by default) which consists in parametric simulation based on the maximum likelihood estimates of the multivariate logistic regression `glm.obj` and "bootstrap" which consists in bootstrap resampling of the database `data`.

iterations   The number of iterations (simulations or resamples depending on the argument in `var.method`) to estimate of the variances and confidence intervals.

n.cluster   The number of cores to use, i.e., the upper limit for the number of child processes that run simultaneously (1 by default).

cluster.type A character string with the type of parallelization. The default type is "PSOCK" (it calls makePSOCKcluster, faster on MacOS or Linux platforms). An alternative is "FORK" (it calls makeForkCluster, it does not work on Windows platforms).

Details

The ATE corresponds to Average Treatment effect on the Entire population, i.e. the marginal effect if all the sample is treated versus all the sample is untreated. The ATT corresponds to Average Treatment effect on the Treated, i.e. the marginal effect if the treated patients (group = 1) would

have been untreated. The ATU corresponds to Average Treatment effect on the Untreated , i.e. the marginal effect if the untreated patients (group = 0) would have been treated. Simulation method for variance estimation has a shorter computing time than the boostrap method, but bootstrap is more accurate.

Value

effect	A character string with the type of the marginal effect.
p0	A table related to the average proportion of events in the unexposed/untreated sample: estimate is the estimated value, ci.lower and ci.upper represent the 95% confidence interval.
p1	A table related to the average proportion of events in the exposed/treated sample: estimate is the estimated value, ci.lower and ci.upper represent the 95% confidence interval.
delta	A table related to the difference between the average proportions of events in the exposed/treated sample minus in the unexposed/untreated sample: estimate is the estimated value, std.error is the corresponding standard error, ci.lower and ci.upper represent the 95% confidence interval.
logOR	A table related to the logarithm of the average Odds Ratio (OR): estimate is the estimated value, std.error is the corresponding standard error, ci.lower and ci.upper represent the 95% confidence interval.
p.value	The p-value of the bilateral test of the null hypothesis p0 = p1, i.e. OR = 1.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Arthur Chatton <Arthur.Chatton@etu.univ-nantes.fr>

References

Chatton A, Le Borgne F, Gillaizeau F, Leyrat C, Barbin L, Laplaud D, Leger M, Giraudeau B, Foucher Y. G-computation and propensity score-based methods with binary outcome: a comparative study to evaluate their performances for causal inference. Manuscript submitted. 2019.

Examples

```
#data simulation
#treatment = 1 if the patients have been the exposure or treatment of interest and 0 otherwise
treatment <- rbinom(600, 1, prob=0.5)

covariate <- rnorm(600, 0, 1)
covariate[treatment==1] <- rnorm(sum(treatment==1), 0.3, 1)

outcome <- rbinom(600, 1, prob=
exp(-2+0.26*treatment+0.7*covariate)/(1+exp(-2+0.26*treatment+0.7*covariate)))

tab <- data.frame(outcome, treatment, covariate)

#Raw effect of the treatment
```

```
glm.raw <- glm(outcome ~ treatment, data=tab, family = binomial(link=logit))
summary(glm.raw)

#Conditional effect of the treatment
glm.multi <- glm(outcome ~ treatment + covariate, data=tab, family = binomial(link=logit))
summary(glm.multi)

#Marginal effects of the treatment (ATE)
gc.ate <- gc.logistic(glm.obj=glm.multi, data=tab, group="treatment", effect="ATE",
 var.method="simulations", iterations=1000, n.cluster=1)

#Sum-up of the 3 ORs
data.frame( raw=exp(glm.raw$coefficients[2]),
conditional=exp(glm.multi$coefficients[2]),
marginal.ate=exp(gc.ate$logOR[,1]) )
```

---

gc.sl.binary	*Marginal Effect for Binary Outcome by Super Learned G-computation.*

Description

This function allows to estimate the marginal effect of an exposure or a treatment by G-computation for binary outcomes, the outcome prediction (Q-model) is obtained by using a super learner.

Usage

```
gc.sl.binary(outcome, group, cov.quanti, cov.quali, data, effect,
 tuneLength, cv, iterations, n.cluster, cluster.type)
```

Arguments

outcome	The name of the variable related to the binary outcome variable. This numeric variable must have only two levels: 0 and 1 (for instance: 0=healthy, 1=diseased).
group	The name of the variable related to the exposure/treatment variable. This numeric variable must have only two levels: 0 for the untreated/unexposed patients and 1 for the treated/exposed patients.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame in which to look for the variable related to the outcome, the treatment/exposure and the covariables.
effect	The type of the marginal effect to be estimated. Three types are possible (see details): "ATE" (by default), "ATT" and "ATU".

tuneLength	It defines the total number of parameter combinations that will be evaluated in the machine learning techniques. The default value is 10.
cv	The number of splits for cross-validation. The default value is 10.
iterations	The number of bootstrap resamples to estimate of the variances and confidence intervals.
n.cluster	The number of cores to use, i.e., the upper limit for the number of child processes that run simultaneously (1 by default).
cluster.type	A character string with the type of parallelization. The default type is "PSOCK" (it calls makePSOCKcluster, faster on MacOS or Linux platforms). An alternative is "FORK" (it calls makeForkCluster, it does not work on Windows platforms).

Details

The ATE corresponds to Average Treatment effect on the Entire population, i.e. the marginal effect if all the sample is treated versus all the sample is untreated. The ATT corresponds to Average Treatment effect on the Treated, i.e. the marginal effect if the treated patients (group = 1) would have been untreated. The ATU corresponds to Average Treatment effect on the Untreated , i.e. the marginal effect if the untreated patients (group = 0) would have been treated. The Super Learner includes the following machine learning techniques: logistic regression with Lasso penalization, logistic regression with Elasticnet penalization, neural network with one hidden layer, and support vector machine with radial basis, as explained in details by Chatton et al. (2020).

Value

missing	Number of deleted observations due to missing data.
effect	A character string with the type of the marginal effect.
p0	A table related to the average proportion of events in the unexposed/untreated sample: estimate is the estimated value, ci.lower and ci.upper represent the 95% confidence interval.
p1	A table related to the average proportion of events in the exposed/treated sample: estimate is the estimated value, ci.lower and ci.upper represent the 95% confidence interval.
delta	A table related to the difference between the average proportions of events in the exposed/treated sample minus in the unexposed/untreated sample: estimate is the estimated value, std.error is the corresponding standard error, ci.lower and ci.upper represent the 95% confidence interval.
logOR	A table related to the logarithm of the average Odds Ratio (OR): estimate is the estimated value, std.error is the corresponding standard error, ci.lower and ci.upper represent the 95% confidence interval.
p.value	The p-value of the bilateral test of the null hypothesis p0 = p1, i.e. OR = 1.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Chatton A et al. Improving Causal Inference Through the Super Learned G-computation. Manuscript submitted. 2020.

Examples

```
#data simulation

#treatment = 1 if the patients have been the exposure or treatment of interest and 0 otherwise
#treatment <- rbinom(200, 1, prob=0.5)

#one quantitative covariate

#covariate1 <- rnorm(200, 0, 1)
#covariate1[treatment==1] <- rnorm(sum(treatment==1), 0.3, 1)

#one qualitative covariate

#covariate2 <- rbinom(200, 1, prob=0.5)
#covariate2[treatment==1] <- rbinom(sum(treatment==1), 1, prob=0.4)

#outcome <- rbinom(200, 1, prob = exp(-2+0.26*treatment+0.4*covariate1-

#0.4*covariate2)/(1+exp( -2+0.26*treatment+0.4*covariate1-0.4*covariate2)))

#tab <- data.frame(outcome, treatment, covariate1, covariate2)

#Raw effect of the treatment

#glm.raw <- glm(outcome ~ treatment, data=tab, family = binomial(link=logit))
#summary(glm.raw)

#Conditional effect of the treatment

#glm.multi <- glm(outcome ~ treatment + covariate1 + covariate2,
# data=tab, family = binomial(link=logit))
#summary(glm.multi)

#Marginal effects of the treatment (ATE) by using logistic regression as the Q-model

#gc.ate1 <- gc.logistic(glm.obj=glm.multi, data=tab, group="treatment", effect="ATE",
# var.method="bootstrap", iterations=1000, n.cluster=1)

#Marginal effects of the treatment (ATE) by using a super learner as the Q-model

#gc.ate2 <- gc.sl.binary(outcome="outcome", group="treatment", cov.quanti="covariate1",
# cov.quali="covariate2", data=tab, effect="ATE", tuneLength=10, cv=3,
# iterations=1000, n.cluster=1)

#Sum-up of the 3 ORs

#data.frame( raw=exp(glm.raw$coefficients[2]),
```

```
#conditional=exp(glm.multi$coefficients[2]),
#marginal.ate.logistic=exp(gc.ate1$logOR[,1]),
#marginal.ate.sl=exp(gc.ate2$logOR[,1]) )
```

---

gc.sl.time   *Marginal Effect for Censored Outcome by G-computation with a Super*
                                    Learner for the Outcome Model.

Description

This function allows to estimate the (population-average) marginal effect of an exposure or a treatment by G-computation for a censored times-to-event, the Q-model being specified by a Super Learner.

Usage

```
gc.sl.time(object, max.time, effect, estim.tune, estim.weights,
   methods, conditional,  iterations, length.out)
```

Arguments

object	An object returned by the function `sl.time`. It shall be included the exposition/treatment of interest in the argument `group` of the `sl.time` function.
max.time	The maximum time of follow-up to estimate of restricted mean survival time (RMST). The default value is `NULL`, which corresponds to `object$pro.time` if this argument is not NULL, or the time at which half of the sample is still at-risk otherwise.
effect	The type of marginal effect to be estimated. Three types are possible (see details): "ATE" (by default), "ATT" and "ATU".
estim.tune	A logical value. If `TRUE`, the tuning parameters of the algorithms of the Super Learner are estimated at each boostraap iteration. The default is `TRUE`.
estim.weights	A logical value. If `TRUE`, the weights of the algorithms of the Super Learner are estimated at each bootstrap iteration. The default is `TRUE`.
methods	A vector of character with the methods for which the marginal effect are computing. The methods included in `object$methods` or "sl" are allowed. You can use "all" if you want to estimate the marginal effect of the Super Learner and all the included algorithmsor models. You can use "sl" if you want only the estimation for the Super Learner. The default value is "sl".
conditional	A logical value. If `TRUE`, the conditional (subject-specific) Hazard Ratio (HR) is also computing. See the help related to the `hr.sl.time` function for more details on this option. The default is `FALSE`.
iterations	The number of bootstrap resamples to estimate of the variances and the confidence intervals.
length.out	An non-negative number argument related to the desired length of time intervals to compute the marginal estimands. The default is 100.

Details

The ATE corresponds to Average Treatment effect on the Entire population, i.e. the marginal effect if all the sample is treated versus all the sample is untreated. The ATT corresponds to Average Treatment effect on the Treated, i.e. the marginal effect if the treated patients (group = 1) would have been untreated. The ATU corresponds to Average Treatment effect on the Untreated , i.e. the marginal effect if the untreated patients (group = 0) would have been treated. The RMST is the mean survival time of all subjects in the study population followed up to max.time.

Value

effect	A character string with the type of selected effect.
max.time	A scalar related to the maximum time of follow-up for the RMST computation.
RMST0	A table related to the RMST in the unexposed/untreated sample: estimate is the estimated value, ci.lower and ci.upper represent the 95% confidence interval.
RMST1	A table related to the RMST in the exposed/treated sample: estimate is the estimated value, ci.lower and ci.upper represent the 95% confidence interval.
delta	A table related to the difference between the RMST in the exposed/treated sample minus in the unexposed/untreated one: estimate is the estimated value, ci.lower and ci.upper represent the 95% confidence interval, and p.value is the p-value of the bilateral test of the null hypothesis RMST0 = RMST1.
logHR	A table related to the logarithm of the average HR: estimate is the estimated value, std.error is the corresponding standard error, ci.lower and ci.upper represent the 95% confidence interval, and p.value is the p-value of the bilateral test of the null hypothesis HR = 1.
table.surv	This data frame presents the survival probabilities (survival) in each group (variable) according to the times. The number of individuals at risk (n.risk) is also provided.
logHR.conditional.values	
	A vector with the estimated individual HRs (only if conditional = TRUE).
logHR.conditional	
	A table related to the logarithm of the conditional HR (only if conditional = TRUE): estimate is the estimated value, std.error is the corresponding standard error, ci.lower and ci.upper represent the 95% confidence interval, and p.value is the p-value of the bilateral test of the null hypothesis HR = 1.
learner	A list with the same arguments (RMST0, RMST1, etc.) for the other algorithms/model included in the Super Learner ans declared in the argument methods (only if methods=NULL).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Chatton et al. G-computation versus inverse probability weighting for time-to-event analyses. Manuscript in progress. 2019.

Examples

```
data(dataDIVAT2)

#The outcome model base on a Super Learner and the first 150 individuals of the data base
sl1<-sl.time( methods=c("aft.gamma", "ph.gompertz"),  metric="ibs",
  data=dataDIVAT2[1:150,],  times="times", failures="failures", group="ecd",
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant"), cv=3)

#Marginal effect of the treatment (ATE): use 1000 iterations instead of 2
gc.ate <- gc.sl.time(sl1, max.time=12, effect="ATE", iterations=2,
  estim.tune=FALSE, estim.weights=FALSE)

#Plot the survival curves
plot(gc.ate, ylab="Confounder-adjusted survival",
 xlab="Time post-transplantation (years)", col=c(1,2))
```

---

gc.survival	*Marginal Effect for Censored Outcome by G-computation with a Cox Regression for the Outcome Model.*

Description

This function allows to estimate the marginal effect of an exposure or a treatment by G-computation for a censored times-to-event, the Q-model being specified by a Cox model.

Usage

```
gc.survival(object, data, group, times, failures, max.time, effect,
iterations, n.cluster, cluster.type)
```

Arguments

object	A coxph object obtained by using the function coxph with the argument x=TRUE must be specified. It shall be included the exposition/treatment of interest variable and at least one covariate.
data	A data frame in which to look for the variables related to the status of the event (observed or censored), the follow-up time, the treatment/exposure and the co-variables included in the previous model object. The covariates' names have to be identical than the ones included in object.
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones.
times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).

max.time	The maximum time of follow-up to estimate of restricted mean survival time (RMST).
effect	The type of marginal effect to be estimated. Three types are possible (see details): "ATE" (by default), "ATT" and "ATU".
iterations	The number of bootstrap resamples to estimate of the variances and the confidence intervals.
n.cluster	The number of cores to use, i.e., the upper limit for the number of child processes that run simultaneously (1 by default).
cluster.type	A character string with the type of parallelization. The default type is "PSOCK" (it calls makePSOCKcluster, faster on MacOS or Linux platforms). An alternative is "FORK" (it calls makeForkCluster, it does not work on Windows platforms).

Details

The ATE corresponds to Average Treatment effect on the Entire population, i.e. the marginal effect if all the sample is treated versus all the sample is untreated. The ATT corresponds to Average Treatment effect on the Treated, i.e. the marginal effect if the treated patients (group = 1) would have been untreated. The ATU corresponds to Average Treatment effect on the Untreated , i.e. the marginal effect if the untreated patients (group = 0) would have been treated. The RMST is the mean survival time of all subjects in the study population followed up to max.time.

Value

table.surv	This data frame presents the survival probabilities (survival) in each group (variable) according to the times. The number of individuals at risk (n.risk) is also provided.
effect	A character string with the type of selected effect.
max.time	A scalar related to the maximum time of follow-up.
RMST0	A table related to the RMST in the unexposed/untreated sample: estimate is the estimated value, std.error is the standard error, ci.lower and ci.upper represent the 95% confidence interval.
RMST1	A table related to the RMST in the exposed/treated sample: estimate is the estimated value, std.error is the standard error, ci.lower and ci.upper represent the 95% confidence interval.
delta	A table related to the difference between the RMST in the exposed/treated sample minus in the unexposed/untreated one: estimate is the estimated value, std.error is the standard error, ci.lower and ci.upper represent the 95% confidence interval, and p.value is the p-value of the bilateral test of the null hypothesis RMST0 = RMST1.
logHR	A table related to the logarithm of the average Hazard Ratio (HR): estimate is the estimated value, std.error is the standard error, ci.lower and ci.upper represent the 95% confidence interval, and p.value is the p-value of the bilateral test of the null hypothesis HR = 1.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Arthur Chatton <Arthur.Chatton@etu.univ-nantes.fr>

References

Chatton et al. G-computation versus inverse probability weighting for time-to-event analyses. Manuscript in progress. 2019.

Examples

```
data(dataDIVAT2)

#Raw effect of the treatment
cox.raw <- coxph(Surv(times,failures) ~ ecd, data=dataDIVAT2, x=TRUE)
summary(cox.raw)

#Conditional effect of the treatment
cox.cdt <- coxph(Surv(times,failures) ~ ecd + age + retransplant,
 data=dataDIVAT2, x=TRUE)
summary(cox.cdt)

#Marginal effect of the treatment (ATE): use 1000 iterations instead of 10
#We restricted to 10 to respect the CRAN policy in terms of time for computation
gc.ate <- gc.survival(object=cox.cdt, data=dataDIVAT2, group="ecd", times="times",
 failures="failures", max.time=max(dataDIVAT2$times), iterations=10, effect="ATE",
 n.cluster=1)
gc.ate

#Sum-up of the 3 HRs
data.frame( raw=exp(cox.raw$coefficients),
conditional=exp(cox.cdt$coefficients[1]),
marginal.ate=exp(gc.ate$logHR[,1]) )

#Plot the survival curves
plot(gc.ate, ylab="Confounder-adjusted survival",
 xlab="Time post-transplantation (years)", col=c(1,2))
```

---

hr.sl.time	*Conditionnal Effect for Censored Outcome with a Super Learner for the Outcome Model.*

Description

This function allows to estimate the conditional (subject-specific) effect of an exposure or a treatment for a censored times-to-event by using a Super Learner (SL).

Usage

```
hr.sl.time(object, pro.time, iterations, n.cluster, cluster.type, conda.env)
```

Arguments

object	An object returned by the function `sl.time`. It shall be included the exposition/treatment of interest in the argument `group` of the `sl.time` function.
pro.time	The maximum time of follow-up to estimate of restricted mean survival time (RMST). The default value is `NULL`, which corresponds to `object$pro.time` if this argument is not NULL, or the time at which half of the sample is still at-risk otherwise.
iterations	The number of bootstrap resamples to estimate of the variances and the confidence intervals.
n.cluster	The number of cores to use, i.e., the upper limit for the number of child processes that run simultaneously (1 by default).
cluster.type	A character string with the type of parallelization. The default type is "PSOCK" (it calls makePSOCKcluster, faster on MacOS or Linux platforms). An alternative is "FORK" (it calls makeForkCluster, it does not work on Windows platforms).
conda.env	The name of the Conda environment to use if Python is used via reticulate. The default is `NULL`, which corresponds th the default environment by default located at ~/.virtualenvs.

Details

The conditional effect is estimated by a conditional hazard ratio (HR), which corresponds to average of the individual HR on the studied population, i.e. the subject-specific effect if the individual is treated (group = 1) versus untreated (group = 0).

Value

logHR.conditional.values

A vector with the estimated individual HRs.

logHR.conditional

A table related to the logarithm of the conditional HR: `estimate` is the estimated value, `std.error` is the corresponding standard error, `ci.lower` and `ci.upper` represent the 95% confidence interval, and `p.value` is the p-value of the bilateral test of the null hypothesis HR = 1.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Foucher et al. Manuscript in progress. 2021.

Examples

```
data(dataDIVAT2)

#The outcome model base on a Super Learner from the first 150 individuals
sl1<-sl.time( methods=c("aft.gamma", "ph.gompertz"), metric="bs",
  data=dataDIVAT2[1:150,], times="times", failures="failures", group="ecd",
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant"), cv=3)

#Set the prognostic time, for instance:
pro.time <- min(c(
  max(dataDIVAT2$times[dataDIVAT2$ecd==0]),
  max(dataDIVAT2$times[dataDIVAT2$ecd==1]) ))

#Conditional effect of the ECD by using a SL: use 1000 iterations instead of 2
hr.sl <- hr.sl.time(sl1, pro.time=pro.time, iterations=2, n.cluster=1)

#Conditional effect of the ECD by using a Cox model
cox.cdt <- coxph(Surv(times,failures) ~ ecd + age + retransplant,
  data=dataDIVAT2)

data.frame( rbind(
summary(cox.cdt)$coef[1,c(1,2,5)],
hr.sl$logHR.conditional[,c(1,2,5)] ))
```

---

ipw.log.rank   *Log-Rank Test for Adjusted Survival Curves.*

Description

The user enters individual survival data and the weights previously calculated (by using logistic regression for instance). The usual log-rank test is adapted to the corresponding adjusted survival curves.

Usage

```
ipw.log.rank(times, failures, variable, weights)
```

Arguments

times	A numeric vector with the follow up times.
failures	A numeric vector with the event indicator (0=right censored, 1=event).
variable	A numeric vector with the binary variable under interest (only two groups).
weights	The weights for correcting the contribution of each individual. By default, the weights are all equaled to 1 and the survival curves correspond to the usual Kaplan-Meier estimator.

Details

For instance, the `weights` may be equal to 1/p, where p is the estimated probability of the individual to be in its group. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the group. The possible confounding factors are the explanatory variables of this model.

Value

statistic   The value of the log-rank statistic.

p.value   The p-value associated to the previous log-rank statistic.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Jun Xie <junxie@purdue.edu>

Florant Le Borgne <fleborgne@idbc.fr>

References

Le Borgne F, Giraudeau B, Querard AH, Giral M and Foucher Y. Comparisons of the performances of different statistical tests for time-to-event analysis with confounding factors: practical illustrations in kidney transplantation. Statistics in medicine. 30;35(7):1103-16, 2016. <doi:10.1002/sim.6777>

Jun Xie and Chaofeng Liu. Adjusted Kaplan-Meier estimator and log-rank test with inverse probability of treatment weighting for survival data. Statistics in medicine, 24(20):3089-3110, 2005. <doi:10.1002/sim.2174>

Examples

```
data(dataDIVAT2)

# adjusted log-rank test
Pr0 <- glm(ecd ~ 1, family = binomial(link="logit"), data=dataDIVAT2)$fitted.values[1]
Pr1 <- glm(ecd ~ age + hla + retransplant, data=dataDIVAT2,
 family=binomial(link = "logit"))$fitted.values
W <- (dataDIVAT2$ecd==1) * (1/Pr1) + (dataDIVAT2$ecd==0) * (1)/(1-Pr1)

ipw.log.rank(dataDIVAT2$times, dataDIVAT2$failures, dataDIVAT2$ecd, W)
```

---

ipw.survival	*Adjusted Survival Curves by Using IPW.*

Description

This function allows to estimate confounder-adjusted survival curves by weighting the individual contributions by the inverse of the probability to be in the group (IPW).

Usage

```
ipw.survival(times, failures, variable, weights)
```

Arguments

times	A numeric vector with the follow up times.
failures	A numeric vector with the event indicators (0=right censored, 1=event).
variable	A numeric vector with the binary variable under interest (only two groups).
weights	The weights for correcting the contribution of each individual. By default, the weights are all equaled to 1 and the survival curves correspond to the usual Kaplan-Meier estimator.

Details

For instance, the weights may be equal to 1/p, where p is the estimated probability of the individual to be in its group. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the group. The possible confounding factors are the covariates of this model.

Value

table.surv	This data frame presents the survival probabilities (survival) in each group (variable) according to the times. The number of individuals at risk (n.risk) and the number of observed events are also provided (n.event).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florent Le Borgne <fleborgne@idbc.fr>

References

Le Borgne et al. Comparisons of the performances of different statistical tests for time-to-event analysis with confounding factors: practical illustrations in kidney transplantation. Statistics in medicine. 30;35(7):1103-16, 2016. <doi:10.1002/ sim.6777>

Examples

```
data(dataDIVAT2)

# adjusted Kaplan-Meier estimator by IPW
 Pr0 <- glm(ecd ~ 1, family = binomial(link="logit"), data=dataDIVAT2)$fitted.values[1]
 Pr1 <- glm(ecd ~ age + hla + retransplant, data=dataDIVAT2,
 family=binomial(link = "logit"))$fitted.values
 W <- (dataDIVAT2$ecd==1) * (1/Pr1) + (dataDIVAT2$ecd==0) * (1)/(1-Pr1)
 res.akm <-ipw.survival(times=dataDIVAT2$times, failures=dataDIVAT2$failures,
  variable=dataDIVAT2$ecd, weights=W)

plot(res.akm, ylab="Confounder-adjusted survival",
 xlab="Time post-transplantation (years)", col=c(1,2), grid.lty=1)
```

---

lines.rocrisca   *Add Lines to a ROC Plot*

Description

Used to add an additionnal ROC curve to ROC plot generated with `plot.rocrisca`.

Usage

```
## S3 method for class 'rocrisca'
lines(x, ...)
```

Arguments

x	An object of class `rocrisca`, returned by the functions `roc.binary`, `roc.net`, `roc.summary`, and `roc.time`.
...	Additional arguments affecting the plot line.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Le Borgne et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. Manuscript submitted. Stat Methods Med Res. 27(11):3397-3410, 2018. <doi: 10.1177/ 0962280217702416.>

Examples

```
# import and attach the data example

data(dataDIVAT3)

# A subgroup analysis to reduce the time needed for this exemple

dataDIVAT3 <- dataDIVAT3[1:400,]

# The ROC curve to evaluate the crude capacities of the recipient age for the
# prognosis of post kidney transplant mortality (we ignore the censoring process)

roc1 <- roc.binary(status="death", variable="ageR", confounders=~1,
data=dataDIVAT3, precision=seq(0.1,0.9, by=0.1) )

# The standardized and weighted ROC curve to evaluate the capacities
# of the recipient age for the prognosis of post kidney transplant
# mortality by taking into account the donor age and the recipient
# gender (we ignore the censoring process).

# 1. Standardize the marker according to the covariates among the controls
lm1 <- lm(ageR ~ ageD + sexeR, data=dataDIVAT3[dataDIVAT3$death == 0,])
dataDIVAT3$ageR_std <- (dataDIVAT3$ageR - (lm1$coef[1] + lm1$coef[2] *
 dataDIVAT3$ageD + lm1$coef[3] * dataDIVAT3$sexeR)) / sd(lm1$residuals)

# 2. Compute the sensitivity and specificity from the proposed IPW estimators
roc2 <- roc.binary(status="death", variable="ageR_std",
 confounders=~bs(ageD, df=3) + sexeR, data=dataDIVAT3, precision=seq(0.1,0.9, by=0.1))

# The corresponding ROC graph

plot(roc2, type="b", col=2, pch=2, lty=2)

lines(roc1, type="b", col=1, pch=1)

legend("bottomright", lty=1:2, lwd=1, pch=1:2, col=1:2,
 c(paste("Crude estimation, (AUC=", round(roc1$auc, 2), ")", sep=""),
 paste("Adjusted estimation, (AUC=", round(roc2$auc, 2), ")", sep="") ) )
```

---

lrs.multistate   *Likelihood Ratio Statistic to Compare Embedded Multistate Models*

Description

This function computes a Likelihood Ratio Statistic to compare two embedded multistate models.

Usage

```
lrs.multistate(model1, model0)
```

Arguments

model1   A list containing the results after using a function included in the present RISCA
                package.

model0   A list containing the results after using a function included in the present RISCA
                package. The function used to obtained the model0 have to be the same than the
                one used to obtain the model1. The model0 have to be embedded in the model1.

Value

statistic   The value of the statistic.

ddl   The degrees of freedom.

pvalue   The p-value.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

Examples

```
data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d3<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 250, replace = FALSE),]

# To illustrate the use of a 3-state model, individuals with trajectory 13 and 123 are
# censored at the time of transition into state X=3

d3$trajectory[d3$trajectory==13]<-1
d3$trajectory[d3$trajectory==123]<-12
d3$trajectory[d3$trajectory==14]<-13
d3$trajectory[d3$trajectory==124]<-123

# 3-state parametric semi-Markov model : does 'z' influence both the
# transition 1->3 ? We only reduced the precision and the number of iteration
# to save time in this example, prefer the default values.

m1 <- semi.markov.3states(times1=d3$time1, times2=d3$time2,
  sequences=d3$trajectory, dist=c("E","E","E"),
  ini.dist.12=c(9.93), ini.dist.13=c(11.54), ini.dist.23=c(10.21),
  cov.12=d3$z, init.cov.12=c(-0.13), names.12=c("beta12_z"),
  cov.13=d3$z, init.cov.13=c(1.61),  names.13=c("beta13_z"),
  conf.int=TRUE, silent=FALSE, precision=0.001)

m1

m0 <- semi.markov.3states(times1=d3$time1, times2=d3$time2,
```

```
  sequences=d3$trajectory, dist=c("E","E","E"),
  ini.dist.12=c(9.93), ini.dist.13=c(11.54), ini.dist.23=c(10.21),
  cov.12=d3$z, init.cov.12=c(-0.13), names.12=c("beta12_z"),
  conf.int=TRUE, silent=FALSE, precision=0.001)

m0

lrs.multistate(model1=m1, model0=m0)
```

---

markov.3states   *3-State Time-Inhomogeneous Markov Model*

Description

The 3-state Markov model includes an initial state (X=1), a transient state (X=2) and an absorbing
state (X=3). Usually, X=1 corresponds to disease-free or remission, X=2 to relapse, and X=3 to
death. In this illness-death model, the possible transitions are: 1->2, 1->3 and 2->3.

Usage

```
markov.3states(times1, times2, sequences, weights=NULL, dist,
cuts.12=NULL, cuts.13=NULL, cuts.23=NULL,
ini.dist.12=NULL, ini.dist.13=NULL, ini.dist.23=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times (in days) from baseline to the first transition (in X=2 or X=3) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times (in days) from baseline to the second transition (in X=3) or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3.
sequences	A numeric vector with the sequences of observed states. Four possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 123 (individual who transited from X=1 to X=3 through X=2).
weights	A numeric vector with the weights for correcting the contribution of each individual. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.

cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

$$
\begin{array}{rl}
\text{Exponential distribution} & \lambda(t) = 1/\sigma \\
\text{Weibull distribution} & \lambda(t) = \nu(\frac{1}{\sigma})^{\nu} t^{\nu-1} \\
\text{Generalized Weibull distribution} & \lambda(t) = \frac{1}{\theta} \left(1 + \left(\frac{t}{\sigma}\right)^{\nu} \right)^{\frac{1}{\theta}-1} \nu \left(\frac{1}{\sigma}\right)^{\nu} t^{\nu-1}
\end{array}
$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in `ini.dist.12`, `ini.dist.13` and `ini.dist.23`.

To estimate the marginal effect of a binary exposure, the `weights` may be equal to `1/p`, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

`object`	The character string indicating the estimated model: "markov.3states (3-state time-inhomogeneous markov model)".
`dist`	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3.
`cuts.12`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
`cuts.13`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
`cuts.23`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
`covariates`	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3 and 2->3.
`table`	A data frame containing the estimated parameters of the model (Estimate). When the option `conf.int=TRUE` is specified, this data.frame includes three additional columns: the Standard Errors of parameters (`Std.Error`), the value of the Wald statistic (`t.value`), and the related p-value (`Pr(>|t|)`).
`cov.matrix`	A data frame corresponding to variance-covariance matrix of the parameters.
`LogLik`	A numeric value corresponding to the (weighted) log-likelihood of the model.
`AIC`	A numeric value corresponding to the Akaike Information Criterion of the model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Huszti et al. Relative survival multistate Markov model. Stat Med. 10;31(3):269-86, 2012. <DOI: 10.1002/sim.4392>

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Examples

```
# import the observed data
# X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, and X=4 to death with a functioning graft

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d3<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 300, replace = FALSE),]

# Individuals with trajectory 13 and 123 are
# censored at the time of transition into state X=3

d3$trajectory[d3$trajectory==13]<-1
d3$trajectory[d3$trajectory==123]<-12
d3$trajectory[d3$trajectory==14]<-13
d3$trajectory[d3$trajectory==124]<-123

# 3-state parametric Markov model including one explicative variable
# (z is the delayed graft function) on the transition 1->2. We only reduced
# the precision and the number of iteration to save time in this example,
# prefer the default values.

markov.3states(times1=d3$time1, times2=d3$time2, sequences=d3$trajectory, weights=NULL,
  dist=c("E","E","E"), ini.dist.12=c(9.93),
  ini.dist.13=c(11.54), ini.dist.23=c(10.21),
  cov.12=d3$z, init.cov.12=c(-0.13), names.12=c("beta12_z"),
  conf.int=TRUE,  silent=FALSE, precision=0.001)
```

---

markov.3states.rsadd   *3-state Relative Survival Markov Model with Additive Risks*

Description

The 3-state Markov relative survival model includes an initial state (X=1), a transient state (X=2), and the death (X=3). The possible transitions are: 1->2, 1->3 and 2->3. Assuming additive risks, the observed mortality hazard is the sum of two components: the expected population mortality (X=P) and the excess mortality related to the disease under study (X=E). The expected population

mortality hazard (X=P) can be obtained from the death rates provided by life tables of statistical national institutes. These tables indicate the proportion of people dead in a calendar year stratified by birthdate and gender.

Usage

```
markov.3states.rsadd(times1, times2, sequences, weights=NULL, dist,
cuts.12=NULL, cuts.13=NULL, cuts.23=NULL,
ini.dist.12=NULL, ini.dist.13=NULL, ini.dist.23=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
p.age, p.sex, p.year, p.rate.table,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times in days from baseline to the first transition (X=2 or X=3) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times in days from baseline to the second transition or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3.
sequences	A numeric vector with the sequences of observed states. Four possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 123 (individual who transited from X=1 to X=3 through X=2).
weights	A numeric vector with the weights for correcting the contribution of each individual. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the

distribution is not piecewise. Piecewise model is only allowed for exponential distribution.

ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
p.age	A numeric vector with the patient ages in days at baseline (X=1).
p.sex	A character vector with the genders: male or female.
p.year	A numeric vector with the entry dates in the study respecting the date format, i.e. in number of days since 01.01.1960.
p.rate.table	A list containing the information related to the expected rates of mortality. This list is organized as a ratetable object.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

Exponential distribution	$\lambda(t) = 1/\sigma$
Weibull distribution	$\lambda(t) = \nu(\frac{1}{\sigma})^{\nu} t^{\nu-1}$
Generalized Weibull distribution	$\lambda(t) = \frac{1}{\theta}\left(1 + \left(\frac{t}{\sigma}\right)^{\nu}\right)^{\frac{1}{\theta}-1} \nu \left(\frac{1}{\sigma}\right)^{\nu} t^{\nu-1}$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in `ini.dist.12`, `ini.dist.13` and `ini.dist.23`.

To estimate the marginal effect of a binary exposure, the `weights` may be equal to `1/p`, where `p` is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities `p` are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

object	The character string indicating the estimated model: "markov.3states.rsadd (3-state time-inhomogeneous markov model)".
dist	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
covariates	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3 and 2->3.
table	A data frame containing the estimated parameters of the model (Estimate). When the option `conf.int=TRUE` is specified, this data frame includes three additional columns: the Standard Errors of parameters (Std.Error), the value of the Wald statistic (t.value), and the related p-value (Pr(>|t|)).
cov.matrix	A data frame corresponding to variance-covariance matrix of the parameters.
LogLik	A numeric value corresponding to the log-likelihood of the estimated model.
AIC	A numeric value corresponding to the Akaike Information Criterion of the estimated model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Huszti et al. Relative survival multistate Markov model. Stat Med. 10;31(3):269-86, 2012. <DOI: 10.1002/sim.4392>

Gillaizeau et al. A multistate additive relative survival semi-Markov model. Stat Methods Med Res. 26(4):1700-1711, 2017. <doi: 10.1177/ 0962280215586456>.

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Examples

```
# import the observed data
# (X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft)

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d3<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 200, replace = FALSE),]

# Individuals with trajectory 13 and 123 are
# censored at the time of transition into state X=3

d3$trajectory[d3$trajectory==13]<-1
d3$trajectory[d3$trajectory==123]<-12
d3$trajectory[d3$trajectory==14]<-13
d3$trajectory[d3$trajectory==124]<-123

# import the expected mortality rates

data(fr.ratetable)

# 3-state Markov model with additive risks including one explicative variable
# (z is the delayed graft function) on all transitions. We only reduced
# the precision and the number of iteration to save time in this example,
# prefer the default values.

markov.3states.rsadd(times1=d3$time1, times2=d3$time2, sequences=d3$trajectory,
  dist=c("E","E","E"), ini.dist.12=c(8.34),
  ini.dist.13=c(10.70), ini.dist.23=c(11.10),
  cov.12=d3$z, init.cov.12=c(0.04), names.12=c("beta12_z"),
  cov.13=d3$z, init.cov.13=c(1.04), names.13=c("beta1E_z"),
  cov.23=d3$z, init.cov.23=c(0.29), names.23=c("beta2E_z"),
  p.age=d3$ageR*365.24, p.sex=d3$sexR,
  p.year=date::as.date(paste("01","01",d3$year.tx), order = "mdy"),
  p.rate.table=fr.ratetable,  conf.int=TRUE,
  silent=FALSE, precision=0.001)
```

---

markov.4states   *4-State Time-Inhomogeneous Markov Model*

Description

The 4-state Markov model includes an initial state (X=1), a transient state (X=2) and two absorbing states (X=3 and X=4). The possible transitions are: 1->2, 1->3, 1->4, 2->3 and 2->4.

Usage

```
markov.4states(times1, times2, sequences,  weights=NULL, dist,
cuts.12=NULL, cuts.13=NULL, cuts.14=NULL, cuts.23=NULL, cuts.24=NULL,
ini.base.12=NULL, ini.base.13=NULL, ini.base.14=NULL,
ini.base.23=NULL, ini.base.24=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
cov.14=NULL, init.cov.14=NULL, names.14=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
cov.24=NULL, init.cov.24=NULL, names.24=NULL,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times in days from baseline to the first transition (X=2, X=3 or X=4) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times in days from baseline to the second transition or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3 or X=4.
sequences	a numeric vector with the sequences of observed states. Six possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 14 (individual who directly transited from X=1 to X=4), 123 (individual who transited from X=1 to X=3 through X=2), 124 (individual who transited from X=1 to X=4 through X=2).
weights	A numeric vector with the weights for correcting the contribution of each individual. When the vector is completed, the IPW estimator is implemented. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.

cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.14	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.24	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.base.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.14	A numeric vector of initial values for the distribution from X=1 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.24	A numeric vector of initial values for the distribution from X=2 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.

names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.14	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=4.
init.cov.14	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.14. Default initial value is 0.
names.14	An optional character vector with name of explicative variables associated to cov.14.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
cov.24	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=4.
init.cov.24	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.24. Default initial value is 0.
names.24	An optional character vector with name of explicative variables associated to cov.24.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

$$
\begin{array}{ll}
\text{Exponential distribution} & \lambda(t) = 1/\sigma \\
\text{Weibull distribution} & \lambda(t) = \nu(\tfrac{1}{\sigma})^{\nu} t^{\nu-1} \\
\text{Generalized Weibull distribution} & \lambda(t) = \frac{1}{\theta} \left(1 + \left(\tfrac{t}{\sigma}\right)^{\nu}\right)^{\frac{1}{\theta}-1} \nu \left(\tfrac{1}{\sigma}\right)^{\nu} t^{\nu-1}
\end{array}
$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in ini.base.12, ini.base.13 and ini.base.23.

To estimate the marginal effect of a binary exposure, the weights may be equal to 1/p, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

`object`	A character string indicating the estimated model: "markov.4states (4-state time-inhomogeneous markov model)".
`dist`	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3, 1->4, 2->3, and 2->4.
`cuts.12`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
`cuts.13`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
`cuts.14`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4.
`cuts.23`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
`cuts.24`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4.
`covariates`	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3, 1->4, 2->3, and 2->4.
`table`	A data frame containing the estimated parameters of the model (Estimate). When the option `conf.int=TRUE` is specified, this data frame includes three additional columns: the Standard Errors of parameters (Std.Error), the value of the Wald statistic (t.value), and the related p-value (Pr(>|t|)).
`cov.matrix`	A data frame corresponding to variance-covariance matrix of the parameters.
`LogLik`	A numeric value corresponding to the log-likelihood of the estimated model.
`AIC`	A numeric value corresponding to the Akaike Information Criterion of the estimated model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Huszti et al. Relative survival multistate Markov model. Stat Med. 10;31(3):269-86, 2012. <DOI: 10.1002/sim.4392>

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Examples

```
# import the observed data
# (X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft)

data(dataDIVAT1)
```

```
# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d4<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 200, replace = FALSE),]

# 4-state parametric Markov model including one explicative variable ('z')
# on the trainsition from X=1 to X=2. We only reduced
# the precision and the number of iteration to save time in this example,
# prefer the default values.

markov.4states(times1=d4$time1, times2=d4$time2, sequences=d4$trajectory,
  dist=c("E","E","E","E","E"),
  ini.base.12=c(8.31), ini.base.13=c(10.46), ini.base.14=c(10.83),
  ini.base.23=c(9.01), ini.base.24=c(10.81),
  cov.12=d4$z, init.cov.12=c(-0.02), names.12=c("beta12_z") ,
  conf.int=TRUE, silent=FALSE, precision=0.001)$table
```

---

markov.4states.rsadd   *4-state Relative Survival Markov Model with Additive Risks*

Description

The 4-state Markov relative survival model includes an initial state (X=1), a transient state (X=2), and two absorbing states including death (X=3, and X=4 for death). The possible transitions are: 1->2, 1->3, 1->4, 2->3 and 2->4. Assuming additive risks, the observed mortality hazard (X=4) is the sum of two components: the expected population mortality (X=P) and the excess mortality related to the disease under study (X=E). The expected population mortality hazard (X=P) can be obtained from the death rates provided by life tables of statistical national institutes. These tables indicate the proportion of people dead in a calendar year stratified by birthdate and gender.

Usage

```
markov.4states.rsadd(times1, times2, sequences, weights=NULL, dist,
cuts.12=NULL, cuts.13=NULL, cuts.14=NULL, cuts.23=NULL,
cuts.24=NULL, ini.dist.12=NULL, ini.dist.13=NULL,
ini.dist.14=NULL, ini.dist.23=NULL, ini.dist.24=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
cov.14=NULL, init.cov.14=NULL, names.14=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
cov.24=NULL, init.cov.24=NULL, names.24=NULL,
p.age, p.sex, p.year, p.rate.table,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times in days from baseline to the first transition (X=2, X=3 or X=4) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times in days from baseline to the second transition or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3 or X=4.
sequences	A numeric vector with the sequences of observed states. Six possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 14 (individual who directly transited from X=1 to X=4), 123 (individual who transited from X=1 to X=3 through X=2), 124 (individual who transited from X=1 to X=4 through X=2).
weights	a numeric vector with the weights for correcting the contribution of each individual. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.14	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.24	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.

ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.14	A numeric vector of initial values for the distribution from X=1 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.24	A numeric vector of initial values for the distribution from X=2 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.14	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=4.
init.cov.14	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.14. Default initial value is 0.
names.14	An optional character vector with name of explicative variables associated to cov.14.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
cov.24	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=4.
init.cov.24	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.24. Default initial value is 0.
names.24	An optional character vector with name of explicative variables associated to cov.24.
p.age	A numeric vector with the patient ages in days at baseline (X=1).
p.sex	A character vector with the genders: male or female.

p.year	A numeric vector with the entry dates in the study respecting the date format, i.e. in number of days since 01.01.1960.
p.rate.table	A list containing the information related to the expected rates of mortality. This list is organized as a `ratetable` object.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

$$\begin{array}{ll}
\text{Exponential distribution} & \lambda(t) = 1/\sigma \\
\text{Weibull distribution} & \lambda(t) = \nu(\tfrac{1}{\sigma})^{\nu} t^{\nu-1} \\
\text{Generalized Weibull distribution} & \lambda(t) = \tfrac{1}{\theta}\left(1 + \left(\tfrac{t}{\sigma}\right)^{\nu}\right)^{\frac{1}{\theta}-1} \nu\left(\tfrac{1}{\sigma}\right)^{\nu} t^{\nu-1}
\end{array}$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in `ini.dist.12`, `ini.dist.13` and `ini.dist.23`.

To estimate the marginal effect of a binary exposure, the `weights` may be equal to `1/p`, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

object	A character string indicating the estimated model: "markov.4states.rsadd (4-state relative survival markov model with additive risks)".
dist	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3, 1->4, 2->3, and 2->4.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
cuts.14	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
cuts.24	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4.

covariates	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3, 1->4, 2->3, and 2->4.
table	A data frame containing the estimated parameters of the model (Estimate). When the option conf.int=TRUE is specified, this data.frame includes three additional columns: the Standard Errors of parameters (Std.Error), the value of the Wald statistic (t.value), and the related p-value (Pr(>|t|)).
cov.matrix	A data frame corresponding to variance-covariance matrix of the parameters.
LogLik	A numeric value corresponding to the log-likelihood of the estimated model.
AIC	A numeric value corresponding to the Akaike Information Criterion of the estimated model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Huszti et al. Relative survival multistate Markov model. Stat Med. 10;31(3):269-86, 2012. <DOI: 10.1002/sim.4392>

Gillaizeau et al. A multistate additive relative survival semi-Markov model. Stat Methods Med Res. 26(4):1700-1711, 2017. <doi: 10.1177/ 0962280215586456>.

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Examples

```
# import the observed data
# (X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft)

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d3<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 200, replace = FALSE),]

# import the expected mortality rates

data(fr.ratetable)

# 4-state parametric additive relative survival Markov model including one
# explicative variable ('z') on the transition 1->2. We only reduced
# the precision and the number of iteration to save time in this example,
# prefer the default values.

markov.4states.rsadd(times1=d3$time1, times2=d3$time2, sequences=d3$trajectory,
```

```
dist=c("E","E","E","E","E"),
ini.dist.12=c(8.34), ini.dist.13=c(10.44), ini.dist.14=c(10.70),
ini.dist.23=c(9.43), ini.dist.24=c(11.11),
cov.12=d3$z, init.cov.12=c(0.04), names.12=c("beta12_z"),
p.age=d3$ageR*365.24, p.sex=d3$sexR,
p.year=date::as.date(paste("01","01",d3$year.tx), order = "mdy"),
p.rate.table=fr.ratetable, conf.int=TRUE,
silent=FALSE, precision=0.001)
```

---

metric   *Metrics to Evaluate the Prognostic Capacities*

Description

Compute several metrics to evaluate the prognostic capacities with time-to-event data.

Usage

```
metric(times, failures, data, prediction.matrix, prediction.times, metric,
pro.time, ROC.precision)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
data	A data frame for in which to look for the variables related to the status of the follow-up time (`times`) and the event (`failures`).
prediction.matrix	
	A matrix with the predictions of survivals of each subject (lines) for each prognostic times (columns).
prediction.times	
	A vector of numeric values with the times of the `predictions` (same length than the number of columns of `prediction.matrix`).
metric	The metric to comput. See details.
pro.time	This optional value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument times. Not used for the following metrics: "loglik", "ibs", "bll", and "ibll". Default value is the time at which half of the subjects are still at risk.
ROC.precision	An optional argument with the percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. Only used when `metric="auc"`. 0 (min) and 1 (max) are not allowed. By default, the precision is `seq(.01,.99,.01)`.

Details

The following metrics can be used: "brier" for the Brier score at the prognostic time `pro.time`, "loglik" for the Log-likelihood, "ibs" for the Integrated Brier score up to the last observed time of event, "ibll" for the Integrated Binomial Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "ribs" for the restricted Integrated Brier score up to the prognostic time `pro.time`, "ribll" for the restricted Integrated Binomial Log-likelihood Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "auc" for the area under the time-dependent ROC curve up to the prognostic time `pro.time`.

Value

A numeric value with the metric estimation.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-cox.ridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), lambda=1)

# The apparent AUC at 10-year post-transplanation
metric(times="times", failures="failures", data=dataDIVAT2,
  prediction.matrix=model$predictions, prediction.times=model$times,
  metric="auc", pro.time=10)

# The integrated Brier score up to 10 years post-transplanation
metric(times="times", failures="failures", data=dataDIVAT2,
  prediction.matrix=model$predictions, prediction.times=model$times,
  metric="ribs", pro.time=10)
```

---

mixture.2states   *Horizontal Mixture Model for Two Competing Events*

Description

The 2-state mixture model which includes an initial state (X=1) and two absorbing states in competition (X=2 and X=3). Parameters are estimated by (weighted) Likelihood maximization.

Usage

```
mixture.2states(times, sequences, weights=NULL, dist, cuts.12=NULL, cuts.13=NULL,
 ini.dist.12=NULL, ini.dist.13=NULL, cov.12=NULL, init.cov.12=NULL,
 names.12=NULL, cov.13=NULL, init.cov.13=NULL, names.13=NULL,
 cov.p=NULL, init.cov.p=NULL, names.p=NULL, init.intercept.p=NULL,
 conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times	A numeric vector with the observed times in days from baseline to the last observation.
sequences	A numeric vector with the sequence of observed states. Three possible values are allowed: 1 (the individual is right-censored in X=1), 12 (the individual transits to X=2) and 13 (the individual transits to X=3).
weights	A numeric vector with the weights for correcting the contribution of each individual. When the vector is completed, the IPW estimator is implemented. Default is NULL which means that no weighting is applied.
dist	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2 and 1->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.

init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to `cov.13`. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to `cov.13`.
cov.p	A matrix (or data frame) with the explicative time-fixed variable(s) related to the probability P(X=2), which is regressing according to a logistic function.
init.cov.p	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to `cov.p`. Default initial value is 0.
names.p	An optional character vector with name of explicative variables associated to `cov.p`.
init.intercept.p	
	A numeric value to iniate the intercept of the logit of P(X=2). Default value is 0.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is `TRUE`.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is `TRUE`, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

$$
\begin{array}{ll}
\text{Exponential distribution} & \lambda(t) = 1/\sigma \\
\text{Weibull distribution} & \lambda(t) = \nu(\frac{1}{\sigma})^{\nu} t^{\nu-1} \\
\text{Generalized Weibull distribution} & \lambda(t) = \frac{1}{\theta}\left(1 + \left(\frac{t}{\sigma}\right)^{\nu}\right)^{\frac{1}{\theta}-1} \nu \left(\frac{1}{\sigma}\right)^{\nu} t^{\nu-1}
\end{array}
$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in `ini.dist.12`, `ini.dist.13` and `ini.dist.23`.

To estimate the marginal effect of a binary exposure, the `weights` may be equal to `1/p`, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

object	The character string indicating the estimated model: "mixture.2states (mixture model with two competing events)".
dist	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2 and 1->3.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.

cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
covariates	A numeric vector indicating the numbers of covariates respectively related to the time to the event X=2, the time to the event X=3, the long-term probability P(X=2).
table	A data frame containing the estimated parameters of the model (Estimate). When the option conf.int=TRUE is specified, this data frame includes three additional columns: the Standard Errors of parameters (Std.Error), the value of the Wald statistic (t.value), and the p-value for the Wald test (Pr(>|t|)).
cov.matrix	A data frame corresponding to variance-covariance matrix of the parameters.
LogLik	A numeric value corresponding to the (weighted) log-likelihood of the model.
AIC	A numeric value corresponding to the Akaike Information Criterion of the model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Trebern-Launay K, Kessler M, Bayat-Makoei S, Querard AH, Briancon S, Giral M, Foucher Y. Horizontal mixture model for competing risks: a method used in waitlisted renal transplant candidates. European Journal of Epidemiology. 33(3):275-286, 2018. <doi: 10.1007/s10654-017-0322-3>.

Examples

```
# import the observed data
# X=1 corresponds to initial state with a functioning graft,
# X=2 to acute rejection episode (transient state),
# X=3 to return to dialysis, X=4 to death with a functioning graft

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d2<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 300, replace = FALSE),]

# Data-management: two competing events
# the patient death is now X=2
# the return in dialysis is now X=3

d2$time<-NA
d2$time[d2$trajectory==1]<-d2$time1[d2$trajectory==1]

d2$time[d2$trajectory==12]<-d2$time2[d2$trajectory==12]
d2$trajectory[d2$trajectory==12]<-1

d2$time[d2$trajectory==13]<-d2$time1[d2$trajectory==13]
```

```
d2$time[d2$trajectory==123]<-d2$time2[d2$trajectory==123]
d2$trajectory[d2$trajectory==123]<-13

d2$time[d2$trajectory==14]<-d2$time1[d2$trajectory==14]

d2$time[d2$trajectory==124]<-d2$time2[d2$trajectory==124]
d2$trajectory[d2$trajectory==124]<-14

d2$trajectory[d2$trajectory==14]<-12

table(d2$trajectory)

# Univariable horizontal mixture model one binary explicative variable
# z is 1 if delayed graft function and 0 otherwise

mm2.test <- mixture.2states(times=d2$time, sequences=d2$trajectory, weights=NULL,
  dist=c("E","W"), cuts.12=NULL, cuts.13=NULL,
  ini.dist.12=c(9.28), ini.dist.13=c(9.92, -0.23),
  cov.12=d2$z, init.cov.12=0.84, names.12="beta_12",
  cov.13=d2$z, init.cov.13=0.76, names.13="beta_13",
  cov.p=NULL, init.cov.p=NULL, names.p=NULL, init.intercept.p=-0.75,
  conf.int=TRUE, silent=FALSE)

mm2.test$table
```

---

nnet.time   *Library of the Super Learner for Survival Neural Network*

Description

Fit a 1-layer neural network based on the partial likelihood from a Cox proportional hazards model.

Usage

```
nnet.time(times, failures, group, cov.quanti, cov.quali,
data, n.nodes, decay, batch.size, epochs)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.

cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
n.nodes	The number of hidden nodes.
decay	The value of the weight decay.
batch.size	The value of batch size.
epochs	The value of epochs.

Details

The survival neural network is obtained by using the deepsurv function from the survivalmodels package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled times and corresponds to the observed follow-up times. The second column is entitled failures and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Medical Research Methodology, 18(1), 24. https://doi.org/10.1186/s12874-018-0482-1

Examples

```
# On a linux machine

#Install Python

#sudo apt install python3-pip

#Install Python Miniconda for CPU computaion on a linux machine

#Download the file from https://docs.conda.io/en/latest/miniconda.html#linux-installers

#chmod +x Miniconda3-latest-Linux-x86_64.sh
#./Miniconda3-latest-Linux-x86_64.sh
#conda create --name torch python=3.8
#conda activate torch
#conda install pytorch torchvision torchaudio cpuonly -c pytorch
#conda install -c conda-forge pycox

#export PATH="/home/username/.local/bin:$PATH"

# remotes::install_github("RaphaelS1/survivalmodels")
# install_pycox(method = "auto",  conda = "auto",  pip = TRUE,  install_torch = TRUE )
# install_keras(method = "auto",  conda = "auto",  pip = TRUE,  install_tensorflow = TRUE)

# use_condaenv(condaenv = "~/miniconda3/envs/torch", conda = "auto", required = FALSE)

#data(dataDIVAT2)

# The estimation of the model from the first 500 individuals
#model<-nnet.time(times="times", failures="failures", data=dataDIVAT2[1:500,],
# cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), n.nodes=5, decay=.01,
#                  batch.size=256L, epochs = 1L)

# The predicted survival of the first subject of the training sample
#plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
#     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

ph.exponential   *Library of the Super Learner for an proportional hazards (PH) para-*
                                  metric model with an Exponential distribution

---

Description

Fit an PH parametric model with an Exponential distribution.

Usage

```
ph.exponential(times, failures, group, cov.quanti, cov.quali=NULL,
data, newdata, newtimes)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in cov.quanti and cov.quali. The default value is NULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value isNULL, the predicted values are computed for the observed times in the training data frame.

Details

The model is obtained by using the dist="exp" in the flexsurvreg package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled times and corresponds to the observed follow-up times. The second column is entitled failures and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the predictions.
hazard	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Jackson, C. (2016). flexsurv: A Platform for Parametric Survival Modeling in R. Journal of Statistical Software, 70(8), 1-33. doi:10.18637/jss.v070.i08

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-ph.exponential(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

ph.gompertz	*Library of the Super Learner for an proportional hazards (PH) parametric model with a Gompertz distribution*

Description

Fit an PH parametric model with a Gompertz distribution.

Usage

```
ph.gompertz(times, failures, group, cov.quanti, cov.quali=NULL,
data, newdata, newtimes)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.

cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (`times`), the event (`failures`), the optional treatment/exposure (`group`) and the covariables included in the previous model (`cov.quanti` and `cov.quali`).
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in `cov.quanti` and `cov.quali`. The default value is NULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value isNULL, the predicted values are computed for the observed times in the training data frame.

Details

The model is obtained by using the `dist="gompertz"` in the `flexsurvreg` package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the `predictions`.
hazard	A vector of numeric values with the values of the cummulative baseline hazard function at the prediction `times`.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Jackson, C. (2016). flexsurv: A Platform for Parametric Survival Modeling in R. Journal of Statistical Software, 70(8), 1-33. doi:10.18637/jss.v070.i08

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-ph.gompertz(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

plot.rocrisca	*Plot Method for 'rocrisca' Objects*

Description

A plot of ROC curves is produced. In the RISCA package, it concerns the functions roc.binary, roc.net, roc.summary, and roc.time.

Usage

```
## S3 method for class 'rocrisca'
plot(x, ..., information=TRUE)
```

Arguments

x	An object of class rocrisca, returned by the functions roc.binary, roc.net, roc.summary, and roc.time.
...	Additional arguments affecting the plot.
information	A logical value indicating whether the non-information line is plotted. The default values is TRUE.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Le Borgne et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. Manuscript submitted. Stat Methods Med Res. 27(11):3397-3410, 2018. <doi: 10.1177/ 0962280217702416.>

Examples

```
# import and attach the data example

data(dataDIVAT3)

# A subgroup analysis to reduce the time needed for this exemple

dataDIVAT3 <- dataDIVAT3[1:400,]

# The ROC curve to evaluate the crude capacities of the recipient age for the
# prognosis of post kidney transplant mortality (we ignore the censoring process)

roc1 <- roc.binary(status="death", variable="ageR", confounders=~1,
data=dataDIVAT3, precision=seq(0.1,0.9, by=0.1) )

# The corresponding ROC graph with color and symbols

plot(roc1, type="b", xlab="1-specificity", ylab="sensibility", col=2, pch=2)
```

---

plot.sl.time *Caliration Plot for Super Learner*

Description

A calibration plot of a Super Learner obtained by the function sl.time.

Usage

```
## S3 method for class 'sl.time'
plot(x, ..., col=1, lty=1, lwd=1, type="b", pch = 16, ylab=NULL, xlab=NULL, method="sl",
 n.groups=5, pro.time=NULL, newdata=NULL, times=NULL, failures=NULL)
```

Arguments

x	An object returned by the function sl.time.
...	Additional arguments affecting the plot.
col	A numeric value with the color of the survival curves. The default is 1 for black.
lty	A numeric value with the type of the survival curves. The default is 1.
lwd	A numeric value with the type of the survival curves. The default is 1.
type	A character string giving the type of plot desired. The default is "b" for both the lines and points
pch	A numeric value for plotting characters or symbols. The default is 16.
xlab	The label for the x axis. The default is "Predicted survival".
ylab	The label for the y axis. The default is "Observed survival".

method	A character string with the name of the algorithm included in the SL for wich the calibration plot is performed. The defaut is "sl" for the Super Learner.
n.groups	A numeric value with the number of groups by their class probabilities. The default is 5.
pro.time	The prognostic time at wich the calibration plot of the survival probabilities.
newdata	An optional data frame containing the new sample for validation with covariate values, folow-up times, and event status. The default value isNULL, the calibration plot is performed from the same subjects of the training sample.
times	The name of the variable related the numeric vector with the follow-up times in newdata (optional argument only necessary when newdata is not NULL).
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in newdata (optional argument only necessary when newdata is not NULL).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
data(dataDIVAT2)

dataDIVAT2$train <- 1*rbinom(n=dim(dataDIVAT2)[1], size = 1, prob=1/3)

# The training of the super learner with 2 models
sl<-sl.time(method=c("aft.gamma", "ph.exponential"),
  metric="ribs",  data=dataDIVAT2[dataDIVAT2$train==1,],
  times="times", failures="failures", pro.time = 12,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),  cv=3)

# The calibration plot from the validation sample
plot(sl, method="sl", n.groups=5, pro.time=12, col=2,
    xlab="Predicted 12-year survival", ylab="Observed 12-year survival",
    newdata=dataDIVAT2[dataDIVAT2$train==0,], times="times", failures="failures")
```

---

plot.survrisca   *Plot Method for 'survrisca' Objects*

Description

A plot of survival curves is produced. In the RISCA package, it concerns the functions ipw.survival and gc.survival.

Usage

```
## S3 method for class 'survrisca'
plot(x, ..., col=1, lty=1, lwd=1, type="s", max.time=NULL,
min.y=0, max.y=1, grid.lty=NULL)
```

Arguments

x	An object of class survrisca, returned by the functions ipw.survival gc.survival.
...	Additional arguments affecting the plot.
col	A numeric vector with the color of the survival curves. The default is 1 for black.
lty	A numeric vector with the type of the survival curves. The default is 1.
lwd	A numeric vector with the type of the survival curves. The default is 1.
type	A character string giving the type of plot : "p" for points, "l" for lines, "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines. The default is "s" for step function.
max.time	The maximum time of the x-asis. The default is NULL, it corresponds to the maximum follow-up time observed in the database from which the survrisca object was estimated.
min.y	The minimum of the y-axis. The default is 0.
max.y	The maximum of the y-axis. The default is 1.
grid.lty	A character or (integer) numeric with the line type of the grid lines. The default is NULL for no grid.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florent Le Borgne <fleborgne@idbc.fr>

Examples

```
data(dataDIVAT2)

res.km <- ipw.survival(times=dataDIVAT2$times, failures=dataDIVAT2$failures,
  variable=dataDIVAT2$ecd, weights=NULL)

plot(res.km, ylab="Graft and patient survival",
 xlab="Time post-transplantation (years)", col=c(1,2), grid.lty=1)
```

---

port | *POsitivity-Regression Tree (PoRT) Algorithm to Identify Positivity Vi-olations.*

Description

This function allows to identify potential posivity violations by using the PoRT algorithm.

Usage

```
port(group, cov.quanti, cov.quali, data, alpha, beta, gamma, pruning,
  minbucket, minsplit, maxdepth)
```

Arguments

group	A character string with the name of the exposure in `data`: 0 for the untreated/unexposed patients and 1 for the treated/exposed patients.
cov.quanti	A character string with the names of the quantitative predictors in `data`.
cov.quali	A character string with the names of the qualitative predictors in `data`.
data	A data frame in which to look for the variables related to the treatment/exposure and the predictors.
alpha	The minimal proportion of the whole sample size to consider a problematic sub-group. The default value is 0.05.
beta	The exposed or unexposed proportion under which one can consider a positivity violation. The default value is 0.05.
gamma	The maximum number of predictors used to define the subgroup. The default value is 2. See 'Details'.
pruning	If `TRUE`, provide only the violations contained between two values for quantitative predictors. The default value is `FALSE`.
minbucket	An `rpart` parameter: minimum number of observations in any leaf. The default value is 6.
minsplit	An `rpart` parameter: minimum number of observations that must exist in a node in order for a split to be attempted. If only one of `minbucket` or `minsplit` is specified, the code either sets `minsplit` to `minbucket*3` or `minbucket` to `minsplit/3`, as appropriate. The default value is 20.
maxdepth	An `rpart` parameter. Set the maximum depth of any node of the final tree, with the root node counted as depth 0. Values greater than 30 rpart will give nonsense results on 32-bit machines. The default value is 30.

Details

In a first step, the PoRT algorithm estimates one tree for each predictor and memorises the leaves corresponding to problematic subgroups according to the hyperparameters `alpha` and `beta` (i.e., the subgroup must at least include `alpha*100` percent of the whole sample, and the exposure prevalence

in the subgroup must be superior to `1-beta` or inferior to `beta`). If gamma=1, the algorithm stops. Otherwise, if at least one problematic subgroup is identified in this first step, the corresponding predictor(s) is(are) not considered in the second step, which estimates one tree for all possible couples of remaining predictors and memorizes the leaves corresponding to problematic subgroups. If gamma=2, the algorithm stops; otherwise, the third step consists of building one tree for all possible trios of remaining covariates not involved in the previously identified subgroups, etc.

Value

The `port` function returns a characters string summarising all the subgroups identified as violating the positivity assumption, and provides for each of these subgroups the exposure prevalence, the subgroup size and the relative subgroup size (with respect to the sample size).

Author(s)

Arthur Chatton <Arthur.Chatton@univ-nantes.fr>

References

Danelian G, Foucher Y, Leger M, Le Borgne F, and Chatton A. Identification of positivity violations' using regression trees: The PoRT algorithm. Manuscript submitted. 2022.

Examples

```
data("dataDIVAT2")

# PoRT with default hyperparameters
port(group="ecd", cov.quanti="age", cov.quali=c("hla", "retransplant"),
data=dataDIVAT2)

# Illustration of the 'pruning' argument
port(group="ecd", cov.quanti="age", cov.quali=c("hla", "retransplant"),
    data=dataDIVAT2, beta=0.01)

port(group="ecd", cov.quanti="age", cov.quali=c("hla", "retransplant"),
    data=dataDIVAT2, beta=0.01, pruning=TRUE)
```

---

pred.mixture.2states   *Cumulative Incidence Function Form Horizontal Mixture Model With Two Competing Events*

Description

This function allows to estimate a cumulative incidence function (CIF) from an horizontal mixture model with two competing events, i.e. the results obtained from the function `mixture.2states`.

Usage

```
pred.mixture.2states(model, failure, times, cov.12=NULL, cov.13=NULL, cov.p=NULL)
```

Arguments

model	A list obtained by using the function `mixture.2states`.
failure	A numeric value for identifying the event for which the CIF has to be computed. Two possible values are allowed: 2 (for the CIF related to X=2) and 3 (for the CIF related to X=3).
times	A numeric vector with positive values related to the times for which the CIF has to be computed.
cov.12	A vector, matrix or data frame in which to look for variables related to the time from X=1 to X=2 with which to predict the CIF.
cov.13	A vector, matrix or data frame in which to look for variables related to the time from X=1 to X=3 with which to predict the CIF.
cov.p	A vector, matrix or data frame in which to look for variables related to the probability P(X=2).

Details

The covariates has to be identical than the ones included in the mixture model declared in the argument `model`. More precisely, the columns of `cov.12`, `cov.13` and `cov.p` must correspond to the same variables.

Value

times	A numeric vector with the times for which the CIF has to be computed.
cif	A matrix with the predicted CIF for the `times` in columns and the individuals in rows.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

References

Trebern-Launay K, Kessler M, Bayat-Makoei S, Querard AH, Briancon S, Giral M, Foucher Y. Horizontal mixture model for competing risks: a method used in waitlisted renal transplant candidates. European Journal of Epidemiology. 33(3):275-286, 2018. <doi: 10.1007/s10654-017-0322-3>.

Examples

```
# import the observed data
# X=1 corresponds to initial state with a functioning graft,
# X=2 to acute rejection episode (transient state),
# X=3 to return to dialysis, X=4 to death with a functioning graft

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
```

```
set.seed(2)
d2<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 300, replace = FALSE),]

# Data-management: two competing events
# the patient death is now X=2
# the return in dialysis is now X=3

d2$time<-NA
d2$time[d2$trajectory==1]<-d2$time1[d2$trajectory==1]

d2$time[d2$trajectory==12]<-d2$time2[d2$trajectory==12]
d2$trajectory[d2$trajectory==12]<-1

d2$time[d2$trajectory==13]<-d2$time1[d2$trajectory==13]

d2$time[d2$trajectory==123]<-d2$time2[d2$trajectory==123]
d2$trajectory[d2$trajectory==123]<-13

d2$time[d2$trajectory==14]<-d2$time1[d2$trajectory==14]

d2$time[d2$trajectory==124]<-d2$time2[d2$trajectory==124]
d2$trajectory[d2$trajectory==124]<-14

d2$trajectory[d2$trajectory==14]<-12

table(d2$trajectory)

# Univariable horizontal mixture model one binary explicative variable
# z is 1 if delayed graft function and 0 otherwise

mm2.model <- mixture.2states(times=d2$time, sequences=d2$trajectory,
  weights=NULL, dist=c("E","W"), cuts.12=NULL, cuts.13=NULL,
  ini.dist.12=c(9.28), ini.dist.13=c(9.92, -0.23),
  cov.12=d2$z, init.cov.12=0.84, names.12="beta_12",
  cov.13=d2$z, init.cov.13=0.76, names.13="beta_13",
  cov.p=NULL, init.cov.p=NULL, names.p=NULL, init.intercept.p=-0.75,
  conf.int=TRUE, silent=FALSE)

cif2.mm2 <- pred.mixture.2states(mm2.model, failure=2, times=seq(0, 4000, by=30),
 cov.12=c(0,1), cov.13=c(0,1), cov.p=NULL)

plot(cif2.mm2$times/365.25, cif2.mm2$cif[1,], col = 1, type="l", lty = 1,
 ylim=c(0,1), lwd =2, ylab="Cumulative Incidence Function",
 xlab="Times (years)", main="", xlim=c(0, 11), legend=FALSE)

lines(cif2.mm2$times/365.25, cif2.mm2$cif[2,], lwd=2, col=2)
```

---

predict.cox   *Prediction from a Penalized Cox Regression*

Description

Predict the survival of new observations based on a penalized Cox regression estimated by unsing a model of the class cox.

Usage

```
## S3 method for class 'cox'
predict(object, ..., newdata, newtimes)
```

Arguments

object	An object returned by one of the following functions: cox.lasso, cox.ridge, or cox.en.
...	Further arguments passed.
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in cov.quanti and cov.quali included in the training sample. The default value is NULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value isNULL, the predicted values are computed for the observed times in the training data frame.

Value

times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
data(dataDIVAT2)

# The estimation of the training model
model<-cox.lasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), lambda=.01)

# Predicted survival from the validation sample
pred <- predict(model,
  newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions[1,], x=pred$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)
```

```
legend("bottomright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))
```

---

predict.flexsurv   *Prediction from an Flexible Parametric Model*

Description

Predict the survival of new observations based on an flexible parametric model from an object of the class flexsurv.

Usage

```
## S3 method for class 'flexsurv'
predict(object, ..., newdata, newtimes)
```

Arguments

object	An object returned by the function aft.ggamma, aft.gamma, aft.weibull, ph.exp, or ph.gompertz.
...	Further arguments passed.
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in cov.quanti and cov.quali included in the training sample. The default value isNULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value isNULL, the predicted values are computed for the observed times in the training data frame.

Details

The model object is obtained from the flexsurvreg package.

Value

times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
data(dataDIVAT2)

# The estimation of the training model
model<-ph.gompertz(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

# Predicted survival for subjects of the training sample
pred <- predict(model)

# Predicted survival for 2 new subjects
pred <- predict(model,
  newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions[1,], x=pred$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

legend("bottomright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))
```

---

predict.nnet.time   *Prediction from a Survival Neural Network*

Description

Predict the survival of new observations based on a neural network estimated by using nnet.time.

Usage

```
## S3 method for class 'nnet.time'
predict(object, ..., newdata, newtimes)
```

Arguments

object	An object returned by the function nnet.time.
...	Further arguments passed.
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in cov.quanti and cov.quali included in the training sample. The default value isNULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value isNULL, the predicted values are computed for the observed times in the training data frame.

Value

times	A vector of numeric values with the times of the `predictions`.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
# On a linux machine

#Install Python

#sudo apt install python3-pip

#Install Python Miniconda for CPU computaion on a linux machine

#Download the file from https://docs.conda.io/en/latest/miniconda.html#linux-installers

#chmod +x Miniconda3-latest-Linux-x86_64.sh
#./Miniconda3-latest-Linux-x86_64.sh
#conda create --name torch python=3.8
#conda activate torch
#conda install pytorch torchvision torchaudio cpuonly -c pytorch
#conda install -c conda-forge pycox

#export PATH="/home/username/.local/bin:$PATH"

# remotes::install_github("RaphaelS1/survivalmodels")
# install_pycox(method = "auto", conda = "auto", pip = TRUE, install_torch = TRUE )
# install_keras(method = "auto", conda = "auto", pip = TRUE, install_tensorflow = TRUE)

# use_condaenv(condaenv = "~/miniconda3/envs/torch", conda = "auto", required = FALSE)

#data(dataDIVAT2)

# The estimation of the training model
#model<-nnet.time(times="times", failures="failures", data=dataDIVAT2,
# cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), n.nodes=10, decay=0,
# batch.size=128, epochs=10)

# Predicted survival from the validation sample
#pred <- predict(model,
# newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

#plot(y=pred$predictions[1,], x=pred$times, xlab="Time (years)", ylab="Predicted survival",
#    col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

```
#lines(y=pred$predictions[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

#legend("bottomright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))
```

---

predict.rf.time   *Prediction from a Suvival Random Forest Tree*

Description

Predict the survival of new observations based on a random forest tree estimated by using `rf.time`.

Usage

```
## S3 method for class 'rf.time'
predict(object, ..., newdata, newtimes)
```

Arguments

object	An object returned by the function `rf.time`.
...	Further arguments passed.
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in `cov.quanti` and `cov.quali` included in the training sample. The default value is `NULL`, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value is`NULL`, the predicted values are computed for the observed times in the training data frame.

Value

times	A vector of numeric values with the times of the `predictions`.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
data(dataDIVAT2)

# The estimation of the training model
model<-rf.time(times="times", failures="failures", data=dataDIVAT2,
 cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), nodesize=10, mtry=2, ntree=100)
```

```
# Predicted survival from the validation sample
pred <- predict(model,
  newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions[1,], x=pred$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

legend("bottomright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))
```

---

predict.sl.time	*Prediction from an Super Learner (SL) for Censored Outcomes*

Description

Predict the survival of new observations based on an SL by unsing the `sl.time` function.

Usage

```
## S3 method for class 'sl.time'
predict(object, ..., newdata, newtimes)
```

Arguments

object	An object returned by the function sl.time.
...	Further arguments passed.
newdata	An optional data frame containing covariate values at which to produce predicted values. There must be a column for every covariate included in cov.quanti and cov.quali included in the training sample. The default value isNULL, the predicted values are computed for the subjects of the training sample.
newtimes	The times at which to produce predicted values. The default value isNULL, the predicted values are computed for the observed times in the training data frame.

Value

times	A vector of numeric values with the times of the predictions.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
data(dataDIVAT2)

dataDIVAT2$train <- 1*rbinom(n=dim(dataDIVAT2)[1], size = 1, prob=1/3)

# The training of the super learner with 2 algorithms
sl<-sl.time(method=c("cox.ridge", "aft.ggamma"),  metric="ribs",
 data=dataDIVAT2[dataDIVAT2$train==1,], times="times", failures="failures", pro.time = 12,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), cv=3)

# Individual prediction for 2 new subjects
pred <- predict(sl,
  newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions$sl[1,], x=pred$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions$sl[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

legend("bottomright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))
```

---

rf.time   *Library of the Super Learner for Survival Random Forest Tree*

Description

Fit survival random forest tree for given values of the regularization parameters.

Usage

```
rf.time(times, failures, group, cov.quanti, cov.quali,
data, nodesize, mtry, ntree)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.

cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (`times`), the event (`failures`), the optional treatment/exposure (`group`) and the covariables included in the previous model (`cov.quanti` and `cov.quali`).
nodesize	The value of the node size.
mtry	The number of variables randomly sampled as candidates at each split.
ntree	The number of trees.

Details

The survival random forest tree is obtained by using the `randomForestSRC` package.

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates.
data	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
times	A vector of numeric values with the times of the `predictions`.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-rf.time(times="times", failures="failures", data=dataDIVAT2,
```

```
  cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), nodesize=10, mtry=2, ntree=100)

# The predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

rmst   *Restricted Mean Survival Times.*

Description

This function allows to estimate the Restricted Mean Survival Times (RMST).

Usage

```
rmst(times, surv.rates, max.time, type)
```

Arguments

times	A numeric vector with the times.
surv.rates	A numeric vector with the survival rates.
max.time	The maximum follow-up time.
type	The type of input survival cirves. Two possible arguments: "s" for a step function or "l" for a continous function.

Details

RMST represents an interesting alternative to the hazard ratio in order to estimate the effect of an exposure. The RMST is the mean survival time in the population followed up to `max.time`. It corresponds to the area under the survival curve up to `max.time`.

References

Royston R, Parmar M. Restricted mean survival time: an alternative to the hazard ratio for the design and analysis of randomized trials with a time-to-event outcome. BMC Medical Research Methodology 2013;13:152. <DOI: 10.1186/ 1471-2288-13-152>

Examples

```
data(dataDIVAT2)

#Survival according to the donor status (ECD versus SCD)
res <- summary(survfit(Surv(times,failures) ~ ecd, data=dataDIVAT2))

#The mean survival time in ECD recipients followed-up to 10 years
rmst(times = res$time[as.character(res$strata)=="ecd=1"],
 surv.rates = res$surv[as.character(res$strata)=="ecd=1"],
```

```
  max.time = 10, type = "s")

#The mean survival time in SCD recipients followed-up to 10 years
rmst(times=res$time[as.character(res$strata)=="ecd=0"],
 surv.rates=res$surv[as.character(res$strata)=="ecd=0"],
 max.time=10, type = "s")
```

---

roc.binary	*ROC Curves For Binary Outcomes.*

Description

This function allows for the estimation of ROC curve by taking into account possible confounding factors. Two methods are implemented: i) the standardized and weighted ROC based on an IPW estimator, and ii) the placement values ROC.

Usage

```
roc.binary(status, variable, confounders,  data, precision, estimator)
```

Arguments

status	A character string with the name of the variable in `data` which represents the disease status indicator (for instance: 0=healthy, 1=diseased).
variable	A character string with the name of the variable in `data` which represents the diagnostic/prognostic variable under interest. For the standardized and weighted ROC method, the variable must be previously standardized according to the covariates among the controls as proposed by Le Borgne et al. (2017).
confounders	An object of class "formula". More precisely only the right part with an expression of the form ~ `model`, where `model` is the linear predictor of the logistic regressions performed for each cut-off value. The user can use ~1 to obtain the crude estimation.
data	An object of the class `data.frame` containing the variables previously detailed.
precision	A numeric vector with values between 0 and 1. The values represent the x-axis (1-specificity) of the ROC graph for which the user want to obtain the corresponding sensitivities. 0 and 1 are not allowed.
estimator	Two possible estimators can be used: "ipw" and "pv". IPW is based on the Inverse Probability Weigthing theory as proposed by Le Borgne et al. (2017). This estimator applied on a variable standardized according to the covariates among the controls allows to obtain a standardized and weighted ROC. The IPW estimator is selected by default. The user can also use the placement values (pv) estimator as proposed by Pepe and Cai (Biometrics, 2004).

Details

This function computes confounder-adjusted ROC curve for uncensored data. We adapted the usual estimator by considering the individual probabilities to be diseased, given the possible confounding factors. The standardized and weighted ROC is obtained by both providing a variable under interest standardized according to the possible confounding factors and using "ipw" in the option `estimator`. The user can also use the estimator first proposed by Pepe and Cai (2004) which is based on placement values.

Value

`table`	This data frame presents the sensitivities and specificities.
`auc`	The area under the ROC curve.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

References

Blanche et al. (2013) Review and comparison of roc curve estimators for a time-dependent outcome with marker-dependent censoring. Biometrical Journal, 55, 687-704. <doi:10.1002/bimj.201200045>

Pepe and Cai. (2004) The analysis of placement values for evaluating discriminatory measures. Biometrics, 60(2), 528-35. <doi:10.1111/j.0006-341X.2004.00200.x>

Le Borgne et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. Manuscript submitted. Stat Methods Med Res. 27(11):3397-3410, 2018. <doi: 10.1177/ 0962280217702416.>

Examples

```
# import and attach the data example

data(dataDIVAT3)

# A subgroup analysis to reduce the time needed for this exemple

dataDIVAT3 <- dataDIVAT3[1:400,]

# The ROC curve to evaluate the crude capacities of the recipient age for the
# prognosis of post kidney transplant mortality (we ignore the censoring process)

roc1 <- roc.binary(status="death", variable="ageR", confounders=~1,
data=dataDIVAT3, precision=seq(0.1,0.9, by=0.1) )

# The corresponding ROC graph with basic options

plot(roc1, xlab="1-specificity", ylab="sensibility")

# The corresponding ROC graph with color and symbols
```

```
plot(roc1, col=2, pch=2, type="b", xlab="1-specificity", ylab="sensibility")

# The standardized and weighted ROC curve to evaluate the capacities
# of the recipient age for the prognosis of post kidney transplant
# mortality by taking into account the donor age and the recipient
# gender (we ignore the censoring process).

# 1. Standardize the marker according to the covariates among the controls
lm1 <- lm(ageR ~ ageD + sexeR, data=dataDIVAT3[dataDIVAT3$death == 0,])
dataDIVAT3$ageR_std <- (dataDIVAT3$ageR - (lm1$coef[1] + lm1$coef[2] *
 dataDIVAT3$ageD + lm1$coef[3] * dataDIVAT3$sexeR)) / sd(lm1$residuals)

# 2. Compute the sensitivity and specificity from the proposed IPW estimators
roc2 <- roc.binary(status="death", variable="ageR_std",
 confounders=~bs(ageD, df=3) + sexeR, data=dataDIVAT3, precision=seq(0.1,0.9, by=0.1))

# The corresponding ROC graph

plot(roc2, col=2, pch=2, lty=2, type="b", xlab="1-specificity", ylab="sensibility")

lines(roc1, col=1, pch=1, type="b")

legend("bottomright", lty=1:2, lwd=1, pch=1:2, col=1:2,
 c(paste("Crude estimation, (AUC=", round(roc1$auc, 2), ")", sep=""),
 paste("Adjusted estimation, (AUC=", round(roc2$auc, 2), ")", sep="") ) )
```

---

roc.net	*Net Time-Dependent ROC Curves With Right Censored Data.*

Description

This function performs the characteristics of a net time-dependent ROC curve (Lorent, 2013) based on k-nearest neighbor's (knn) estimator or only based on the Pohar-Perme estimator (Pohar, 2012).

Usage

```
roc.net(times, failures, variable, p.age, p.sex, p.year,
 rate.table, pro.time, cut.off, knn=FALSE, prop=NULL)
```

Arguments

times	A numeric vector with the follow up times.
failures	A numeric vector with the event indicator (0=right censored, 1=event).
variable	A numeric vector with the prognostic variable. This variable is collected at the baseline.
p.age	A numeric vector with the age of the individuals at the baseline (in days).
p.sex	A character vector with the gender the individuals ('male' or 'female').

p.year	A numeric vector with the calendar year at the baseline (number of days from the January 1, 1960).
rate.table	A rate-table object with the expected mortality rates by age, sex, and cohort year. The same units used in `p.age`, `p.sex`, `p.year`
pro.time	The value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument `time`.
cut.off	The cut-off values of the variable used to define the possible binary tests.
knn	A logical value indicating whether k-nearest neighbor's estimator should be used.
prop	This is the proportion of the nearest neighbors. The estimation will be based on 2*prop (both right and left proportions) of the total sample size.

Details

This function computes net time-dependent ROC curve with right-censored data using estimator defined by Pohar-Perm et al. (2012) and the k-nearest neighbor's (knn) estimator. The aim is to evaluate the capacity of a variable (measured at the baseline) to predict the excess of mortality of a studied population compared to the general population mortality. Using the knn estimator ensures a monotone and increasing ROC curve, but the computation time may be long. This approach may thus be avoided if the sample size is large because of computing time.

Value

table	This data frame presents the sensitivities and specificities associated with the cut-off values. One can observe NA if the value cannot be computed.
auc	The area under the time-dependent ROC curve for a prognostic up to prognostic time.
missing	Number of deleted observations due to missing data.
warning	This message indicates possible difficulties in the computation of the net ROC curve, for instance if the net survival was not lower or equal1 to 1 for particular cut-off values or times.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

References

Pohar M., Stare J., Esteve J. (2012) On Estimation in Relative Survival. Biometrics, 68, 113-120. <doi:10.1111/ j.1541-0420.2011.01640.x>

Lorent M., Giral M., Foucher Y. (2013) Net time-dependent ROC curves: a solution for evaluating the accuracy of a marker to predict disease-related mortality. Statistics in Medicine, 33, 2379-89. <doi:10.1002/sim.6079>

Examples

```
# import the observed data

data(dataDIVAT3)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT3 <- dataDIVAT3[1:400,]

# import the expected mortality rates

data(fr.ratetable)

# the values of recipient age used for computing the sensibilities and
# specificities (choose more values in practice)

age.cut <- quantile(dataDIVAT3$ageR, probs=seq(0.1, 0.9, by=0.1))

# recoding of the variables for matching with the ratetable

dataDIVAT3$sex <- "male"
dataDIVAT3$sex[dataDIVAT3$sexeR==0] <- "female"
dataDIVAT3$year <- date::mdy.date(month=01, day=01, year=dataDIVAT3$year.tx,
 nineteen = TRUE, fillday = FALSE, fillmonth = FALSE)
dataDIVAT3$age <- dataDIVAT3$ageR*365

# the ROC curve (without correction by the knn estimator) to
# reduce the time for computing this example. In practice, the
# correction should by used in case of non-montone results.

roc1 <- roc.net(times=dataDIVAT3$death.time,
 failures=dataDIVAT3$death,  variable=dataDIVAT3$ageR,
 p.age=dataDIVAT3$age, p.sex=dataDIVAT3$sex, p.year=dataDIVAT3$year,
 rate.table=fr.ratetable, pro.time=3000, cut.off=age.cut, knn=FALSE)

# the sensibilities and specificities associated with the cut off values

roc1$table

# the traditional ROC graph

plot(roc1, col=2, pch=2, lty=2, type="b", xlab="1-specificity", ylab="sensibility")

legend("bottomright", paste("Without knn, (AUC=",
 round(roc1$auc, 2), ")", sep=""),lty=1, lwd=2, col=2)
```

---

roc.prognostic.aggregate

Prognostic ROC Curve Based on Survival Probabilities

Description

Prognostic ROC curve is an alternative graphical approach to represent the discriminative capacity of the marker: a receiver operating characteristic (ROC) curve by plotting 1 minus the survival in the high-risk group against 1 minus the survival in the low-risk group. The area under the curve (AUC) corresponds to the probability that a patient in the low-risk group has a longer lifetime than a patient in the high-risk group. The prognostic ROC curve provides complementary information compared to survival curves. The AUC is assessed by using the trapezoidal rules. When survival curves do not reach 0, the prognostic ROC curve is incomplete and the extrapolations of the AUC are performed by assuming pessimist, optimist and non-informative situations. The user enters the survival according to the model she/he chooses. The area under the prognostic ROC curve is assessed by using the trapezoidal rules. The extrapolated areas (when survival curves do not reach 0) are performed by assuming pessimist, optimist and non-informative situation.

Usage

```
roc.prognostic.aggregate(time.lr, surv.lr, time.hr, surv.hr)
```

Arguments

time.lr	A numeric vector with the survival times in the low risk group.
surv.lr	A numeric vector with the survival probabilities corresponding to time.lr
time.hr	A numeric vector with the survival times in the high risk group.
surv.hr	A numeric vector with the survival probabilities corresponding to time.hr.

Details

The maximum prognostic time is the minimum between the maximum of time.lr and the maximum of time.hr.

Value

max.time	This is the maximum prognostic time used for the analysis
table	This data frame presents the different time cut-offs associated with the coordinates of the ROC curves.
auc	This data frame presents the different estimations of the area under the prognostic ROC curve: the lower bound, the pessimist, the non-informative, the optimist and the upper bound.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr> and

C. Combescure <Christophe.Combescure@hcuge.ch>

References

Combescure C, Perneger TV, Weber DC, Daures JP and Foucher Y. Prognostic ROC curves: a method for representing the overall discriminative capacity of binary markers with right-censored time-to-event endpoints. Epidemiology 2014 Jan;25(1):103-9. <doi: 10.1097/EDE.0000000000000004>

Examples

```
# example of two survival curves using exponential distributions
time.hr <- seq(0, 600, by=5)
time.lr <- seq(0, 500, by=2)
surv.hr <- exp(-0.005*time.hr)
surv.lr <- exp(-0.003*time.lr)

# Illustration of both survival curves
plot(time.hr, surv.hr, xlab="Time (in days)",
 ylab="Patient survival", lwd=2, type="l")
lines(time.lr, surv.lr, lty=2, col=2, lwd=2)
legend("topright", c("High-Risk Group", "Low-Risk Group"), lwd=2,
 col=1:2, lty=1:2)

# Computation of the prognostic ROC curve
proc.result <- roc.prognostic.aggregate(time.lr, surv.lr, time.hr, surv.hr)

# Representation of the prognostic ROC curve
plot(proc.result$table$x, proc.result$table$y, type="l",
 lwd=2, xlim=c(0,1), ylim=c(0,1),
 xlab="1-Survival in the low risk group",
 ylab="1-Survival in the high risk group")
abline(c(0,0), c(1,1), lty=2)

# The pessimist value of the area under the curve
proc.result$auc$pessimist
```

```
roc.prognostic.individual
```
Prognostic ROC Curve based on Individual Data

Description

The user enters individual survival data. The area under the prognostic ROC curve is assessed by using the trapezoidal rules. The extrapolated areas (when survival curves do not reach 0) are performed by assuming pessimist, optimist and non-informative situation. The confidence intervals are obtained by non-parametric bootstrapping.

Usage

```
roc.prognostic.individual(times, failures, variable, iterations)
```

Arguments

times	A numeric vector with the follow up times.
failures	A numeric vector with the event indicator (0=right censored, 1=event).

variable   A numeric vector with the result of the binary test (only two groups). The variable equals 1 for the high risk group and 0 for the low risk group.

iterations   The number of bootstrap samples to compute the confidence intervals. The default value is 0, which corresponds to no confidence interval computation.

Details

The maximum prognostic time is the minimum between the two last observed times of failure in both groups. Prognostic ROC curve is an alternative graphical approach to represent the discriminative capacity of the marker: a receiver operating characteristic (ROC) curve by plotting 1 minus the survival in the high-risk group against 1 minus the survival in the low-risk group. The area under the curve (AUC) corresponds to the probability that a patient in the low-risk group has a longer lifetime than a patient in the high-risk group. The prognostic ROC curve provides complementary information compared to survival curves. The AUC is assessed by using the trapezoidal rules. When survival curves do not reach 0, the prognostic ROC curve is incomplete and the extrapolations of the AUC are performed by assuming pessimist, optimist and non-informative situations.

Value

max.time   This is the maximum prognostic time used for the analysis

table   This data frame presents the different time cut-offs associated with the coordinates of the ROC curves.

auc   This data frame presents the different estimations of the area under the prognostic ROC curve: the lower bound, the pessimist, the non-informative, the optimist and the upper bound.

CI.95   This data frame presents the 95 percentage confidence intervals of the area under the prognostic ROC curve: the lower bound, the pessimist, the non-informative, the optimist and the upper bound.

auc.boot   This data frame presents the different estimations of the area under the prognostic ROC curve for each of the B bootstrap samples: the lower bound, the pessimist, the non-informative, the optimist and the upper bound.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr> and

C. Combescure <Christophe.Combescure@hcuge.ch>

References

Combescure C, Perneger TV, Weber DC, Daures JP and Foucher Y. Prognostic ROC curves: a method for representing the overall discriminative capacity of binary markers with right-censored time-to-event endpoints. Epidemiology 2014 Jan;25(1):103-9. <doi: 10.1097/EDE.0000000000000004>

Examples

```
####################################################################
# example of two samples with different exponential distributions #
```

```
####################################################################

n1 <- 200
n2 <- 200
grp  <- c(rep(1, n1), rep(0, n2))
time.evt <- c(rexp(n1, rate = 1.2), rexp(n2, rate = 0.5))
time.cen <- rexp(n1+n2, rate = 0.2)
time <- pmin(time.evt, time.cen)
evt  <- 1*(time.evt < time.cen)

# Illustration of both survival curves
surv.temp <- survfit(Surv(time, evt) ~ grp)
plot(surv.temp, lty = 2:3)

# Computation of the prognostic ROC curve
proc.result <- roc.prognostic.individual(time, evt, grp, iterations=50)
  # Use more than 50 bootstrap samples for real applications

# Representation of the prognostic ROC curve
plot(proc.result$table$x, proc.result$table$y, type="l",
 lwd=2, xlim=c(0,1), ylim=c(0,1),
 xlab="1-Survival in the low risk group",
 ylab="1-Survival in the high risk group")
abline(c(0,0), c(1,1), lty=2)

# The corresponding 95% CI of the pessimist value
proc.result$CI.95$pessimist
```

---

roc.summary	*Summary ROC Curve For Aggregated Data.*

Description

This function computes summary ROC curve (Combescure et al., 2016).

Usage

```
roc.summary(study.num, classe, n, year, surv, nrisk, proba, marker.min,
 marker.max, init.nlme1, precision, pro.time, time.cutoff)
```

Arguments

study.num	A numeric vector (1,2,3,...) with the study identification.
classe	A numeric vector with integers (1,2,3,...) for identifying the groups defined using the studied marker. 1 is the first group with the lowest values of the marker.
n	A numeric vector with the number of subjects at the baseline (date of marker collection).
year	A numeric vector with the survival times.

surv   A numeric vector with the survival probabilities corresponding to the previous
                     times (often obtained graphically using the published survival curves).

nrisk   A numeric vector with the number of subjects at-risk of the event at the corre-
                     sponding year.

proba   This numeric vector represents the proportion of the patients in a center which
                     belong to the corresponding group.

marker.min   A numeric vector with the minimum values of the marker interval corresponding
                     to the previous class.

marker.max   A numeric vector with the maximum values of the marker interval corresponding
                     to the previous class.

init.nlme1   A numeric vector with the initiate values (mean, sd) of the maker distribution
                     which is assumed to be Gaussian. Default is (0,1).

precision   A numeric vector with the initiate values (mean, sd) of the maker distribution
                     which is assumed to be Gaussian. Default is 10^{-6}.

pro.time   The value of prognostic time is the maximum delay for which the capacity of
                     the variable is evaluated. The same unit than the one used in the argument time.

time.cutoff   The value of internal thresholds for the definition of the piecewise hazard func-
                     tion (3 values for a 4-piece constant function and 4 values for a 5-piece constant
                     function).

Details

This function computes summary ROC curve. The hazard function associated with the time-to-
event was defined as a 4-piece or a 5-piece constant function with a specific association with the
marker at each interval. The maker distribution is assumed to be Gaussian distributed.

Value

nlme1   An object of class nlme representing the nonlinear mixed-effects model of the
                     marker distribution. The marker is assumed Gaussian distributed. mu and sigma
                     represent the mean and the standard deviation. The inter-study variability is
                     modeled with a random effect on the mean. See nlmeObject for the components
                     of the fit.

nlme2   An object of class nlme representing the nonlinear mixed-effects model of the
                     time distribution. The hazard function is a stepwise function with 5 intervals.
                     exp(beta0.1) and exp(beta0.2) represent the baseline hazard and the haz-
                     ard ratio in the first interval. exp(beta1.1) and exp(beta1.2) represent the
                     corrections of these parameters for the second interval... The inter-study vari-
                     ability is modeled with a random effect on the baseline parameter beta0.1. See
                     nlmeObject for the components of the fit.

table   This data frame presents the sensitivities (se) and specificities (sp) associated
                     with the cut-off values (cut.off).

auc   The area under the SROC curve for a prognostic up to prognostic time.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Christophe Combescure <christophe.combescure@hcuge.ch>

References

Combescure et al. A literature-based approach to evaluate the predictive capacity of a marker using time-dependent Summary Receiver Operating Characteristics. Stat Methods Med Res, 25(2):674-85, 2016. <doi: 10.1177/ 0962280212464542>

Examples

```
# The example is too long to compute for a submission on the CRAN
# Remove the characters '#'

### import and attach the data example
data(dataKi67)

### Compute the SROC curve for a prognostic up to 9 years
roc9y<-roc.summary(dataKi67$study.num, dataKi67$classe, dataKi67$n,
 dataKi67$year, dataKi67$surv, dataKi67$nrisk, dataKi67$proba,
 dataKi67$log.marker.min, dataKi67$log.marker.max,
 init.nlme1=c(2.55, -0.29), precision=50, pro.time=9,
 time.cutoff=c(2, 4, 8))

### The ROC graph associated to these to SROC curves
plot(roc9y, col=1, lty=1, lwd=2, type="l", xlab="1-specificity", ylab="sensibility")

### Check of the goodness-of-fit: the observed proportions of
### patients in the $g$th interval of the study $k$ versus the
### fitted proportions (equation 3).

plot(roc9y$data.marker$proba, roc9y$data.marker$fitted,
 xlab="Observed probabilities", ylab="Fitted probabilities",
 ylim=c(0,1), xlim=c(0,1))
abline(0,1)

### Check of the goodness-of-fit: the observed bivariate
### probabilities versus the fitted bivariate
### probabilities (equation 4).

plot(roc9y$data.surv$p.joint, roc9y$data.surv$fitted,
 xlab="Observed probabilities", ylab="Fitted probabilities",
 ylim=c(0,1), xlim=c(0,1))
abline(0,1)

### Check of the goodness-of-fit: the residuals of the bivariate
### probabilities (equation 4) versus the times.

plot(roc9y$data.surv$year, roc9y$data.surv$resid,
```

```
  xlab="Survival time (years)", ylab="Residuals")
lines(lowess(roc9y$data.surv$year,
 I(roc9y$data.surv$resid), iter=0))
```

---

roc.time   *Time-Dependent ROC Curves With Right Censored Data.*

Description

This function allows for the estimation of time-dependent ROC curve by taking into account possible confounding factors. This method is implemented by standardizing and weighting based on an IPW estimator.

Usage

```
roc.time(times, failures, variable, confounders, data,
 pro.time, precision)
```

Arguments

times	A character string with the name of the variable in data which represents the follow up times.
failures	A character string with the name of the variable in data which represents the event indicator (0=right censored, 1=event).
variable	A character string with the name of the variable in data which represents the prognostic variable under interest. This variable is collected at the baseline. The variable must be previously standardized according to the covariates among the controls as proposed by Le Borgne et al. (2017).
confounders	An object of class "formula". More precisely only the right part with an expression of the form ~ model, where model is the linear predictor of the logistic regressions performed for each cut-off value. The user can use ~1 to obtain the crude estimation.
data	An object of the class data.frame containing the variables previously detailed.
pro.time	The value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument times.
precision	The quintiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. 0 (min) and 1 (max) are not allowed.

Details

This function computes confounder-adjusted time-dependent ROC curve with right-censored data. We adapted the naive IPCW estimator as explained by Blanche, Dartigues and Jacqmin-Gadda (2013) by considering the probability of experiencing the event of interest before the fixed prognostic time, given the possible confounding factors.

Value

table	This data frame presents the sensitivities and specificities associated with the cut-off values.
auc	The area under the time-dependent ROC curve for a prognostic up to `pro.time`.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

References

Blanche et al. (2013) Review and comparison of roc curve estimators for a time-dependent outcome with marker-dependent censoring. Biometrical Journal, 55, 687-704. <doi:10.1002/ bimj.201200045>

Le Borgne F. et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. Manuscript submitted. Stat Methods Med Res. 27(11):3397-3410, 2018. <doi: 10.1177/ 0962280217702416.>

Examples

```
# import and attach the data example
data(dataDIVAT3)

# A subgroup analysis to reduce the time needed for this exemple

dataDIVAT3 <- dataDIVAT3[1:400,]

# The standardized and weighted time-dependent ROC curve to evaluate the
# capacities of the recipient age for the prognosis of post kidney
# transplant mortality up to 2000 days by taking into account the
# donor age and the recipient gender.

# 1. Standardize the marker according to the covariates among the controls
lm1 <- lm(ageR ~ ageD + sexeR, data=dataDIVAT3[dataDIVAT3$death.time >= 2500,])
dataDIVAT3$ageR_std <- (dataDIVAT3$ageR - (lm1$coef[1] + lm1$coef[2] * dataDIVAT3$ageD +
 lm1$coef[3] * dataDIVAT3$sexeR)) / sd(lm1$residuals)

# 2. Compute the sensitivity and specificity from the proposed IPW estimators
roc2 <- roc.time(times="death.time", failures="death", variable="ageR_std",
confounders=~bs(ageD, df=3) + sexeR, data=dataDIVAT3, pro.time=2000,
precision=seq(0.1,0.9, by=0.2))

# The corresponding ROC graph
plot(roc2, col=2, pch=2, lty=1, type="b", xlab="1-specificity", ylab="sensibility")


# The corresponding AUC
roc2$auc
```

---

semi.markov.3states   *3-State Semi-Markov Model*

Description

The 3-state SM model includes an initial state (X=1), a transient state (X=2) and an absorbing state (X=3). Usually, X=1 corresponds to disease-free or remission, X=2 to relapse, and X=3 to death. In this illness-death model, the possible transitions are: 1->2, 1->3 and 2->3.

Usage

```
semi.markov.3states(times1, times2, sequences, weights=NULL,
dist, cuts.12=NULL, cuts.13=NULL, cuts.23=NULL,
ini.dist.12=NULL, ini.dist.13=NULL, ini.dist.23=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times in days from baseline to the first transition (X=2 or X=3) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times in days from baseline to the second transition or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3.
sequences	A numeric vector with the sequences of observed states. Four possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 123 (individual who transited from X=1 to X=3 through X=2).
weights	A numeric vector with the weights for correcting the contribution of each individual. When the vector is completed, the IPW estimator is implemented. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.

cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

$$\begin{array}{rl} \text{Exponential distribution} & \lambda(t) = 1/\sigma \\ \text{Weibull distribution} & \lambda(t) = \nu(\frac{1}{\sigma})^\nu t^{\nu-1} \\ \text{Generalized Weibull distribution} & \lambda(t) = \frac{1}{\theta}\left(1 + \left(\frac{t}{\sigma}\right)^\nu\right)^{\frac{1}{\theta}-1} \nu\left(\frac{1}{\sigma}\right)^\nu t^{\nu-1} \end{array}$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in `ini.dist.12`, `ini.dist.13` and `ini.dist.23`.

To estimate the marginal effect of a binary exposure, the `weights` may be equal to `1/p`, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

`object`	The character string indicating the estimated model: "semi.markov.3states (3-state semi-markov model)".
`dist`	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3.
`cuts.12`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
`cuts.13`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
`cuts.23`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
`covariates`	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3 and 2->3.
`table`	A data frame containing the estimated parameters of the model (`Estimate`). When the option `conf.int=TRUE` is specified, this data frame includes three additional columns: the Standard Errors of parameters (`Std.Error`), the value of the Wald statistic (`t.value`), and the related p-value for the Wald test (`Pr(>|t|)`).
`cov.matrix`	A data frame corresponding to variance-covariance matrix of the parameters.
`LogLik`	A numeric value corresponding to the (weighted) log-likelihood of the model.
`AIC`	A numeric value corresponding to the Akaike Information Criterion of the model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Gillaizeau et al. A multistate additive relative survival semi-Markov model. Statistical methods in medical research. 26(4):1700-1711, 2017. <DOI: 10.1177/ 0962280215586456>

Examples

```
# import the observed data
# (X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft)

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d3<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 250, replace = FALSE),]

# To illustrate the use of a 3-state model, individuals with trajectory 13 and 123 are
# censored at the time of transition into state X=3

d3$trajectory[d3$trajectory==13]<-1
d3$trajectory[d3$trajectory==123]<-12
d3$trajectory[d3$trajectory==14]<-13
d3$trajectory[d3$trajectory==124]<-123

# 3-state parametric semi-Markov model including one explicative variable
# on the transition 1->2 (z is 1 if delayed graft function and 0 otherwise).
# We only reduced the precision and the number of iteration to save time in this example,
# prefer the default values.

semi.markov.3states(times1=d3$time1, times2=d3$time2, sequences=d3$trajectory,
 dist=c("E","E","E"), ini.dist.12=c(9.93), ini.dist.13=c(11.54), ini.dist.23=c(10.21),
  cov.12=d3$z, init.cov.12=c(-0.13), names.12=c("beta12_z"),
  conf.int=TRUE, silent=FALSE, precision=0.001)$table
```

```
semi.markov.3states.ic
```
3-State Semi-Markov Model With Interval-Censored Data

Description

The 3-state SM model includes an initial state (X=1), a transient state (X=2) and an absorbing state (X=3). Usually, X=1 corresponds to disease-free or remission, X=2 to relapse, and X=3 to death. In this illness-death model, the possible transitions are: 1->2, 1->3 and 2->3. The time from X=1 to X=2 is interval-censored. Parameters are estimated by (weighted) Likelihood maximization.

Usage

```
semi.markov.3states.ic(times0, times1, times2, sequences, weights=NULL,
dist, cuts.12=NULL, cuts.13=NULL, cuts.23=NULL,
ini.dist.12=NULL, ini.dist.13=NULL, ini.dist.23=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
```

```
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
conf.int=TRUE, silent=TRUE, precision=10^(-6),
legendre=30, homogeneous=TRUE)
```

Arguments

times0	A numeric vector with the observed times in days from baseline to the last observation time in X=1.
times1	A numeric vector with the observed times in days from baseline to the first observation time in X=2. NA for individuals right-censored in X=1 or individuals who are directly in X=3 after X=1 (without any observation in X=2).
times2	A numeric vector with the observed times in days from baseline to the last follow-up.
sequences	A numeric vector with the sequences of observed states. Four possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly observed in X=3 after X=3, without any observation of X=2), 123 (individual who transited from X=1 to X=3 through X=2).
weights	A numeric vector with the weights for correcting the contribution of each individual. When the vector is completed, the IPW estimator is implemented. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.

ini.dist.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.
legendre	A numeric value indicating the number of knots and weights for Gaussian quadrature used in convolution products. Default is 30.
homogeneous	A logical value specifying if the time spent in the state X=1 is considered as non-associated with the distribution of the time from the entry in the state X=2 to the transition in the state X=3. Default is TRUE, assuming no association.

Details

Hazard functions available are:

$$
\begin{array}{ll}
\text{Exponential distribution} & \lambda(t) = 1/\sigma \\
\text{Weibull distribution} & \lambda(t) = \nu(\frac{1}{\sigma})^\nu t^{\nu-1} \\
\text{Generalized Weibull distribution} & \lambda(t) = \frac{1}{\theta}\left(1 + \left(\frac{t}{\sigma}\right)^\nu\right)^{\frac{1}{\theta}-1} \nu\left(\frac{1}{\sigma}\right)^\nu t^{\nu-1}
\end{array}
$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in ini.dist.12, ini.dist.13 and ini.dist.23.

To estimate the marginal effect of a binary exposure, the weights may be equal to 1/p, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Two kinds of model can be estimated: homogeneous and non-homogeneous semi-Markov model. In the first one, the hazard functions only depend on the times spent in the corresponding state. Note that for the transitions from the state X=1, the time spent in the state corresponds to the chronological time from the baseline of the study, as for Markov models. In the second one, the hazard function of the transition from the state X=2 to X=3 depends on two time scales: the time spent in the state 2 which is the random variable of interest, and the time spend in the state X=1 as a covariate.

Value

object	The character string indicating the model: "semi.markov.3states.ic (3-state semi-markov model with interval-censored data)".
dist	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
covariates	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3 and 2->3.
table	A data frame containing the estimated parameters of the model (Estimate). When the option conf.int=TRUE is specified, this data frame includes three additional columns: the Standard Errors of parameters (Std.Error), the value of the Wald statistic (t.value), and the related p-value for the Wald test (Pr(>|t|)).
cov.matrix	A data frame corresponding to variance-covariance matrix of the parameters.
LogLik	A numeric value corresponding to the (weighted) log-likelihood of the model.
AIC	A numeric value corresponding to the Akaike Information Criterion of the model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Examples

```
# The example is too long to compute for a submission on the CRAN
# Remove the characters '#'

# import the observed data (read the application in Gillaizeau et al. for more details)
# X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft

# data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

# dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
# set.seed(2)
# d3<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 100, replace = FALSE),]

# To illustrate the use of a 3-state model, the return in dialysis are right-censored

# d3$trajectory[d3$trajectory==13]<-1
# d3$trajectory[d3$trajectory==123]<-12
# d3$trajectory[d3$trajectory==14]<-13
# d3$trajectory[d3$trajectory==124]<-123
# table(d3$trajectory)

# X=2 is supposed to be interval-censored between 'times0' and 'times1' because
# health examinations take place each year after inclusion

# d3$times0<-NA
# d3$times1<-NA
# d3$time2_<-NA

# i<-d3$trajectory==1
# d3$times0[i]<-trunc(d3$time1[i]/365.24)*365.24+1
# d3$times1[i]<-NA
# d3$times2[i]<- d3$time1[i]+1

# i<-d3$trajectory==12
# d3$times0[i]<-trunc(d3$time1[i]/365.24)*365.24+1
# d3$times1[i]<-(trunc(d3$time1[d3$trajectory==12]/365.24)+1)*365.24
# d3$times2[i]<-pmax(d3$time2[i], (trunc(d3$time1[i]/365.24)+2)*365.24)

# i<-d3$trajectory==13
# d3$times0[i]<-trunc(d3$time1[i]/365.24)*365.24+1
# d3$times1[i]<-NA
# d3$times2[i]<-d3$time1[i]

# i<-d3$trajectory==123
# d3$times0[i]<-trunc(d3$time1[i]/365.24)*365.24+1
# d3$times1[i]<-(trunc(d3$time1[i]/365.24)+1)*365.24
# d3$times2[i]<- pmax(d3$time2[i], (trunc(d3$time1[i]/365.24)+2)*365.24)
```

```
# 3-state homogeneous semi-Markov model with interval-censored data
# including one binary explicative variable (z is 1 if delayed graft function and
# 0 otherwise).
# Estimation of the marginal effect of z on the transition from X=1 to X=2
# by adjusting for 2 possible confounding factors (age and gender)
# We only reduced the precision and the number of iteration to save time in this example,
# prefer the default values.

# propensity.score <- glm(z ~ ageR + sexR, family=binomial(link="logit"),data=d3)
# d3$fit<-propensity.score$fitted.values

# p1<-mean(d3$z)
# d3$w <- p1/d3$fit
# d3$w[d3$z==0]<-(1-p1)/(1-d3$fit[d3$z==0])

# semi.markov.3states.ic(times0=d3$times0, times1=d3$times1,
# times2=d3$times2, sequences=d3$trajectory,
# weights=d3$w, dist=c("E","E","E"), cuts.12=NULL, cuts.13=NULL, cuts.23=NULL,
# ini.dist.12=c(8.23), ini.dist.13=c(10.92), ini.dist.23=c(10.67),
# cov.12=d3$z, init.cov.12=c(0.02), names.12=c("beta12_z"),
# conf.int=TRUE, silent=FALSE, precision=0.001, legendre=20)$table
```

---

semi.markov.3states.rsadd

3-State Relative Survival Semi-Markov Model With Additive Risks

---

Description

The 3-state SMRS model includes an initial state (X=1), a transient state (X=2), and the death (X=3). The possible transitions are: 1->2, 1->3 and 2->3. Assuming additive risks, the observed mortality hazard is the sum of two components: the expected population mortality (X=P) and the excess mortality related to the disease under study (X=E). The expected population mortality hazard (X=P) can be obtained from the death rates provided by life tables of statistical national institutes. These tables indicate the proportion of people dead in a calendar year stratified by birthdate and gender.

Usage

```
semi.markov.3states.rsadd(times1, times2, sequences, weights=NULL,  dist,
cuts.12=NULL, cuts.13=NULL, cuts.23=NULL,
ini.dist.12=NULL, ini.dist.13=NULL, ini.dist.23=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
p.age, p.sex, p.year, p.rate.table,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times in days from baseline to the first transition (X=2 or X=3) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times in days from baseline to the second transition or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3.
sequences	A numeric vector with the sequences of observed states. Four possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 123 (individual who transited from X=1 to X=3 through X=2).
weights	A numeric vector with the weights for correcting the contribution of each individual. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.

cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
p.age	A numeric vector with the patient ages in days at baseline (X=1).
p.sex	A character vector with the genders: male or female.
p.year	A numeric vector with the entry dates in the study respecting the date format, i.e. in number of days since 01.01.1960.
p.rate.table	A list contaning the information related to the expected rates of mortality. This list is organized as a ratetable object.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

$$
\begin{array}{ll}
\text{Exponential distribution} & \lambda(t) = 1/\sigma \\
\text{Weibull distribution} & \lambda(t) = \nu(\frac{1}{\sigma})^\nu t^{\nu-1} \\
\text{Generalized Weibull distribution} & \lambda(t) = \frac{1}{\theta}\left(1 + \left(\frac{t}{\sigma}\right)^\nu\right)^{\frac{1}{\theta}-1} \nu \left(\frac{1}{\sigma}\right)^\nu t^{\nu-1}
\end{array}
$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in ini.dist.12, ini.dist.13 and ini.dist.23.

To estimate the marginal effect of a binary exposure, the weights may be equal to 1/p, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

`object`	The character string indicating the estimated model: "semi.markov.3states.rsadd (3-state relative survival semi-Markov model with additive risks)".
`dist`	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3.
`cuts.12`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
`cuts.13`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
`cuts.23`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
`covariates`	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3 and 2->3.
`table`	A data frame containing the estimated parameters of the model (Estimate). When the option `conf.int=TRUE` is specified, this data frame includes three additional columns: the Standard Errors of parameters (`Std.Error`), the value of the Wald statistic (`t.value`), and the related p-value (`Pr(>|t|)`).
`cov.matrix`	A data frame corresponding to variance-covariance matrix of the parameters.
`LogLik`	A numeric value corresponding to the log-likelihood of the estimated model.
`AIC`	A numeric value corresponding to the Akaike Information Criterion of the estimated model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Gillaizeau et al. A multistate additive relative survival semi-Markov model. Statistical methods in medical research. 26(4):1700-1711, 2017. <DOI: 10.1177/ 0962280215586456>

Examples

```
# import the observed data
# (X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft)

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
```

```
d3<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 150, replace = FALSE),]

# To use a 3-state model, individuals with trajectory 13 and 123 are censored at the time
# of transition into state X=3

d3$trajectory[d3$trajectory==13]<-1
d3$trajectory[d3$trajectory==123]<-12
d3$trajectory[d3$trajectory==14]<-13
d3$trajectory[d3$trajectory==124]<-123

# import the expected mortality rates

data(fr.ratetable)

# 3-state parametric additive relative survival semi-Markov model including one
# explicative variable (z is the delayed graft function) on all transitions. We only reduced
# the precision and the number of iteration to save time in this example,
# prefer the default values.

# Note: a semi-Markovian process with sojourn times exponentially distributed
# is a time-homogeneous Markov process

semi.markov.3states.rsadd(times1=d3$time1, times2=d3$time2, sequences=d3$trajectory,
  dist=c("E","E","E"),
  ini.dist.12=c(10.70), ini.dist.13=c(11.10), ini.dist.23=c(0.04),
  cov.12=d3$z, init.cov.12=c(0.04), names.12=c("beta12_z"),
  cov.13=d3$z, init.cov.13=c(1.04), names.13=c("beta1E_z"),
  cov.23=d3$z, init.cov.23=c(0.29), names.23=c("beta2E_z"),
  p.age=d3$ageR*365.24,  p.sex=d3$sexR,
  p.year=as.date(paste("01","01",d3$year.tx),order = "mdy"),
  p.rate.table=fr.ratetable,
  conf.int=TRUE, silent=FALSE, precision=0.005)
```

---

semi.markov.4states   *4-State Semi-Markov Model*

Description

The 4-state SM model includes an initial state (X=1), a transient state (X=2) and two absorbing states (X=3 and X=4). Usually, X=1 corresponds to disease-free or remission and X=4 to death. The possible transitions are: 1->2, 1->3, 1->4, 2->3 and 2->4.

Usage

```
semi.markov.4states(times1, times2, sequences, weights=NULL, dist,
cuts.12=NULL, cuts.13=NULL, cuts.14=NULL, cuts.23=NULL, cuts.24=NULL,
ini.base.12=NULL, ini.base.13=NULL, ini.base.14=NULL,
ini.base.23=NULL, ini.base.24=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
```

```
cov.14=NULL, init.cov.14=NULL, names.14=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
cov.24=NULL, init.cov.24=NULL, names.24=NULL,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times in days from baseline to the first transition (X=2, X=3 or X=4) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times in days from baseline to the second transition or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3 or X=4.
sequences	a numeric vector with the sequences of observed states. Six possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 14 (individual who directly transited from X=1 to X=4), 123 (individual who transited from X=1 to X=3 through X=2), 124 (individual who transited from X=1 to X=4 through X=2).
weights	A numeric vector with the weights for correcting the contribution of each individual. When the vector is completed, the IPW estimator is implemented. Default is NULL which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are "E" for Exponential distribution, "PE" for the piecewise exponential distribution, "W" for Weibull distribution or "WG" for Generalized Weibull distribution. When the user choose "PE", the arguments "cut.XX" have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.14	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the

	distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.24	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4. Only internal timepoints are allowed: timepoints cannot be 0 or Inf. Default is NULL which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.base.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.14	A numeric vector of initial values for the distribution from X=1 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.base.24	A numeric vector of initial values for the distribution from X=2 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.
init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.14	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=4.
init.cov.14	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.14. Default initial value is 0.
names.14	An optional character vector with name of explicative variables associated to cov.14.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
cov.24	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=4.

init.cov.24	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.24. Default initial value is 0.
names.24	An optional character vector with name of explicative variables associated to cov.24.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

$$
\begin{array}{rl}
\text{Exponential distribution} & \lambda(t) = 1/\sigma \\
\text{Weibull distribution} & \lambda(t) = \nu(\frac{1}{\sigma})^{\nu} t^{\nu-1} \\
\text{Generalized Weibull distribution} & \lambda(t) = \frac{1}{\theta}\left(1 + \left(\frac{t}{\sigma}\right)^{\nu}\right)^{\frac{1}{\theta}-1} \nu \left(\frac{1}{\sigma}\right)^{\nu} t^{\nu-1}
\end{array}
$$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in ini.base.12, ini.base.13 and ini.base.23.

To estimate the marginal effect of a binary exposure, the weights may be equal to 1/p, where p is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities p are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

object	A character string indicating the estimated model: "semi.markov.4states (4-state semi-Markov model)".
dist	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3, 1->4, 2->3, and 2->4.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
cuts.14	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
cuts.24	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4.

covariates	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3, 1->4, 2->3, and 2->4.
table	A data frame containing the estimated parameters of the model (Estimate). When the option conf.int=TRUE is specified, this data frame includes three additional columns: the Standard Errors of parameters (Std.Error), the value of the Wald statistic (t.value), and the related p-value for the Wald test (Pr(>|t|)).
cov.matrix	A data frame corresponding to variance-covariance matrix of the parameters.
LogLik	A numeric value corresponding to the (weighted) log-likelihood of the model.
AIC	A numeric value corresponding to the Akaike Information Criterion of the model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Gillaizeau et al. A multistate additive relative survival semi-Markov model. Statistical methods in medical research. 26(4):1700-1711, 2017. <DOI: 10.1177/ 0962280215586456>

Examples

```
# import the observed data
# (X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft)

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d4<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 200, replace = FALSE),]

# 4-state parametric semi-Markov model including one explicative variable
# (z is the delayed graft function) on the transition from X=1 to X=2

# Note: a semi-Markovian process with sojourn times exponentially distributed
# is a time-homogeneous Markov process

# We only reduced the precision and the number of iteration to save time in this example,
# prefer the default values.

semi.markov.4states(times1=d4$time1, times2=d4$time2, sequences=d4$trajectory,
  dist=c("E","E","E","E","E"),
  ini.base.12=c(8.31), ini.base.13=c(10.46), ini.base.14=c(10.83),
  ini.base.23=c(9.01), ini.base.24=c(10.81),
```

```
      cov.12=d4$z, init.cov.12=c(-0.02), names.12=c("beta12_z"),
      conf.int=TRUE,  silent=FALSE,  precision=0.001)$table
```

---

semi.markov.4states.rsadd

4-State Relative Survival Semi-Markov Model With Additive Risks

---

Description

The 4-state SMRS model includes an initial state (X=1), a transient state (X=2) and two absorbing states (X=3 X=4 for death). The possible transitions are: 1->2, 1->3, 1->4, 2->3 and 2->4. Assuming additive risks, the observed mortality hazard is the sum of two components: the expected population mortality (X=P) and the excess mortality related to the disease under study (X=E). The expected population mortality hazard (X=P) can be obtained from the death rates provided by life tables of statistical national institutes. These tables indicate the proportion of people dead in a calendar year stratified by birthdate and gender.

Usage

```
semi.markov.4states.rsadd(times1, times2, sequences, weights=NULL, dist,
cuts.12=NULL, cuts.13=NULL, cuts.14=NULL, cuts.23=NULL, cuts.24=NULL,
ini.dist.12=NULL, ini.dist.13=NULL, ini.dist.14=NULL,
ini.dist.23=NULL, ini.dist.24=NULL,
cov.12=NULL, init.cov.12=NULL, names.12=NULL,
cov.13=NULL, init.cov.13=NULL, names.13=NULL,
cov.14=NULL, init.cov.14=NULL, names.14=NULL,
cov.23=NULL, init.cov.23=NULL, names.23=NULL,
cov.24=NULL, init.cov.24=NULL, names.24=NULL,
p.age, p.sex, p.year, p.rate.table,
conf.int=TRUE, silent=TRUE, precision=10^(-6))
```

Arguments

times1	A numeric vector with the observed times in days from baseline to the first transition (X=2, X=3 or X=4) or to the right-censoring (in X=1 at the last follow-up).
times2	A numeric vector with the observed times in days from baseline to the second transition or to the right censoring (in X=2 at the last follow-up). NA for individuals right-censored in X=1 or individuals who directly transited from X=1 to X=3 or X=4.
sequences	a numeric vector with the sequences of observed states. Six possible values are allowed: 1 (individual right-censored in X=1), 12 (individual right-censored in X=2), 13 (individual who directly transited from X=1 to X=3), 14 (individual who directly transited from X=1 to X=4), 123 (individual who transited from X=1 to X=3 through X=2), 124 (individual who transited from X=1 to X=4 through X=2).

weights	A numeric vector with the weights for correcting the contribution of each individual. When the vector is completed, the IPW estimator is implemented. Default is `NULL` which means that no weighting is applied.
dist	A character vector with three arguments describing respectively the distributions of duration time for transitions 1->2, 1->3 and 2->3. Arguments allowed are `"E"` for Exponential distribution, `"PE"` for the piecewise exponential distribution, `"W"` for Weibull distribution or `"WG"` for Generalized Weibull distribution. When the user choose `"PE"`, the arguments `"cut.XX"` have also to be defined.
cuts.12	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2. Only internal timepoints are allowed: timepoints cannot be `0` or `Inf`. Default is `NULL` which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.13	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3. Only internal timepoints are allowed: timepoints cannot be `0` or `Inf`. Default is `NULL` which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.14	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4. Only internal timepoints are allowed: timepoints cannot be `0` or `Inf`. Default is `NULL` which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.23	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3. Only internal timepoints are allowed: timepoints cannot be `0` or `Inf`. Default is `NULL` which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
cuts.24	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4. Only internal timepoints are allowed: timepoints cannot be `0` or `Inf`. Default is `NULL` which means that the distribution is not piecewise. Piecewise model is only allowed for exponential distribution.
ini.dist.12	A numeric vector of initial values for the distribution from X=1 to X=2. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.13	A numeric vector of initial values for the distribution from X=1 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.14	A numeric vector of initial values for the distribution from X=1 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.23	A numeric vector of initial values for the distribution from X=2 to X=3. The logarithm of the parameters have to be declared. Default value is 1.
ini.dist.24	A numeric vector of initial values for the distribution from X=2 to X=4. The logarithm of the parameters have to be declared. Default value is 1.
cov.12	A matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=2.

init.cov.12	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.12. Default initial value is 0.
names.12	An optional character vector with name of explicative variables associated to cov.12.
cov.13	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=3.
init.cov.13	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.13. Default initial value is 0.
names.13	An optional character vector with name of explicative variables associated to cov.13.
cov.14	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=1 to X=4.
init.cov.14	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.14. Default initial value is 0.
names.14	An optional character vector with name of explicative variables associated to cov.14.
cov.23	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=3.
init.cov.23	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.23. Default initial value is 0.
names.23	An optional character vector with name of explicative variables associated to cov.23.
cov.24	A numeric matrix (or data frame) with the explicative time-fixed variable(s) related to the time from X=2 to X=4.
init.cov.24	A numeric vector of initial values for regression coefficients (logarithm of the cause-specific hazards ratios) associated to cov.24. Default initial value is 0.
names.24	An optional character vector with name of explicative variables associated to cov.24.
p.age	A numeric vector with the patient ages in days at baseline (X=1).
p.sex	A character vector with the genders: male or female.
p.year	A numeric vector with the entry dates in the study respecting the date format, i.e. in number of days since 01.01.1960.
p.rate.table	A list contaning the information related to the expected rates of mortality. This list is organized as a ratetable object.
conf.int	A logical value specifying if the pointwise confidence intervals for parameters and the variance-covariance matrix should be returned. Default is TRUE.
silent	A logical value specifying if the log-likelihood value should be returned at each iteration. Default is TRUE, which corresponds to silent mode (no display).
precision	A numeric positive value indicating the required precision for the log-likelihood maximization between each iteration. Default is 10^{-6}.

Details

Hazard functions available are:

Exponential distribution	$\lambda(t) = 1/\sigma$
Weibull distribution	$\lambda(t) = \nu(\frac{1}{\sigma})^{\nu} t^{\nu-1}$
Generalized Weibull distribution	$\lambda(t) = \frac{1}{\theta} \left(1 + \left(\frac{t}{\sigma}\right)^{\nu}\right)^{\frac{1}{\theta}-1} \nu \left(\frac{1}{\sigma}\right)^{\nu} t^{\nu-1}$

with σ, ν, and $\theta > 0$. The parameter σ varies for each interval when the distribution is piecewise Exponential. We advise to initialize the logarithm of these parameters in `ini.dist.12`, `ini.dist.13` and `ini.dist.23`.

To estimate the marginal effect of a binary exposure, the `weights` may be equal to `1/p`, where `p` is the estimated probability that the individual belongs to his or her own observed group of exposure. The probabilities `p` are often estimated by a logistic regression in which the dependent binary variable is the exposure. The possible confounding factors are the explanatory variables of this logistic model.

Value

`object`	A character string indicating the estimated model: `"semi.markov.4states.rsadd (4-state relative survival semi-Markov model with additive risks)"`.
`dist`	A character vector with two arguments describing respectively the distributions of duration time for transitions 1->2, 1->3, 1->4, 2->3, and 2->4.
`cuts.12`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=2.
`cuts.13`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=3.
`cuts.14`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=1 to X=4.
`cuts.23`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=3.
`cuts.24`	A numeric vector indicating the timepoints in days for the piecewise exponential distribution related to the time from X=2 to X=4.
`covariates`	A numeric vector indicating the numbers of covariates respectively related to the transition 1->2, 1->3, 1->4, 2->3, and 2->4.
`table`	A data frame containing the estimated parameters of the model (Estimate). When the option `conf.int=TRUE` is specified, this data frame includes three additional columns: the Standard Errors of parameters (`Std.Error`), the value of the Wald statistic (`t.value`), and the related p-value for the Wald test (`Pr(>|t|)`).
`cov.matrix`	A data frame corresponding to variance-covariance matrix of the parameters.
`LogLik`	A numeric value corresponding to the log-likelihood of the estimated model.
`AIC`	A numeric value corresponding to the Akaike Information Criterion of the estimated model.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Florence Gillaizeau <Florence.Gillaizeau@univ-nantes.fr>

References

Gillaizeau et al. Inverse Probability Weighting to control confounding in an illness-death model for interval-censored data. Stat Med. 37(8):1245-1258, 2018. <doi: 10.1002/sim.7550>.

Gillaizeau et al. A multistate additive relative survival semi-Markov model. Statistical methods in medical research. 26(4):1700-1711, 2017. <DOI: 10.1177/ 0962280215586456>

Examples

```
# import the observed data
# (X=1 corresponds to initial state with a functioning graft, X=2 to acute rejection episode,
# X=3 to return to dialysis, X=4 to death with a functioning graft)

data(dataDIVAT1)

# A subgroup analysis to reduce the time needed for this example

dataDIVAT1$id<-c(1:nrow(dataDIVAT1))
set.seed(2)
d4<-dataDIVAT1[dataDIVAT1$id %in% sample(dataDIVAT1$id, 300, replace = FALSE),]

# import the expected mortality rates

data(fr.ratetable)

# 4-state parametric additive relative survival semi-Markov model including one
# explicative variable (z is the delayed graft function) on the transition from X=1 to X=2

# Note: a semi-Markovian process with sojourn times exponentially distributed
# is a time-homogeneous Markov process

# We only reduced the precision and the number of iteration to save time in this example,
# prefer the default values.

semi.markov.4states.rsadd(times1=d4$time1, times2=d4$time2, sequences=d4$trajectory,
  dist=c("E","E","E","E","E"),
  ini.dist.12=c(8.34),  ini.dist.13=c(10.44),  ini.dist.14=c(10.70),
  ini.dist.23=c(9.43),  ini.dist.24=c(11.11),
  cov.12=d4$z, init.cov.12=c(0.04), names.12=c("beta12_z"),
  p.age=d4$ageR*365.24,  p.sex=d4$sexR,
  p.year=as.date(paste("01","01",d4$year.tx), order = "mdy"),
  p.rate.table=fr.ratetable, conf.int=TRUE,
  silent=FALSE, precision=0.001)
```

---

sl.time | *Super Learner for Censored Outcomes*

Description

This function allows to compute a Super Learner (SL) to predict survival outcomes.

Usage

```
sl.time(methods, metric, data, times, failures, group, cov.quanti, cov.quali, cv,
param.tune, pro.time, optim.local.min, ROC.precision, param.weights.fix,
 param.weights.init, keep.predictions, verbose)
```

Arguments

methods	A vector of characters with the names of the algorithms included in the SL. At least two algorithms have to be included.
metric	The loss function used to estimate the weights of the algorithms in the SL. See details.
data	A data frame in which to look for the variables related to the status of the follow-up time (`times`), the event (`failures`), the optional treatment/exposure (`group`) and the covariables included in the previous model (`cov.quanti` and `cov.quali`).
times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
cv	The number of splits for cross-validation. The default value is 10.
param.tune	A list with a length equals to the number of algorithms included in `methods`. If NULL, the tunning parameters are estimated (see details).
pro.time	This optional value of prognostic time represents the maximum delay for which the capacity of the variable is evaluated. The same unit than the one used in the argument times. Not used for the following metrics: "loglik", "ibs", "bll", and "ibll". Default value is the time at which half of the subjects are still at risk.
optim.local.min	
	An optional logical value. If TRUE, the optimization is performed twice to better ensure the estimation of the weights. If FALSE (default value), the optimization is performed once.
ROC.precision	The percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. Only used when `metric="auc"`. 0 (min) and 1 (max) are not allowed. By default, the precision is `seq(.01,.99,.01)`.

param.weights.fix

> A vector with the parameters of the multinomial logistic regession which generates the weigths of the algorithms declared in methods. When completed, the related parameters are not estimated. The default value is NULL: the parameters are estimated by a cv-fold cross-validation. See details.

param.weights.init

> A vector with the initial values of the parameters of the multinomial logistic regession which generates the weigths of the algorithms declared in methods. The default value is NULL: the initial values are equaled to 0. See details.

keep.predictions

> A logical value specifying if all the predictions for all the methods are saved. If FALSE, only the predictions related to the SL are saved (for space saving). The default is TRUE.

verbose

> A logical value specifying if SuperLearner indicates whether to print progress (TRUE) in the fitting process to the console. The default is TRUE

Details

Each object of the list declared in param.tune must have the same name than the names of the methods included in the SL. If param.tune = NULL, the tunning parameters of each algorithm are estimated by cv-fold cross-validation. Otherwise, the user can propose a tunning grid for each method, as explained in the following table. The following metrics can be used: "brier" for the Brier score at the prognostic time pro.time, "loglik" for the Log-likelihood, "ibs" for the Integrated Brier score up to the last observed time of event, "ibll" for the Integrated Binomial Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "ribs" for the restricted Integrated Brier score up to the prognostic time pro.time, "ribll" for the restricted Integrated Binomial Log-likelihood Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "auc" for the area under the time-dependent ROC curve up to the prognostic time pro.time.

Methods:

Names	Description	Package	assumption
"aft.gamma"	Gamma	flexsurv	AFT
"aft.ggamma"	Generalized Gamma	flexsurv	AFT
"aft.weibull"	Weibull	flexsurv	AFT
"ph.exponential"	Exponential	flexsurv	PH
"ph.gompertz"	Gompertz	flexsurv	PH
"cox.en"	Elastic Net Cox	glmnet	PH
"cox.lasso"	Lasso Cox	glmnet	PH
"cox.ridge"	Ridge Cox	glmnet	PH
"rf.time"	Survival Random Forest	randomForestSRC	RF
"nnet.time"	Neural Network	survivalmodels	PH

Loss Function metric:

- Brier Score ("bs")
- Binomial log likelihood ("bll")

- Integrated brier score (`"ibs"`)
- Integrated binomial log likelihood (`"ibll"`)
- Restricted Integrated Brier Score (`"ribs"`)
- Restricted Integrated Binomial Log-Likelihood (`"ribll"`)

Value

`times`	A vector of numeric values with the times of the `predictions`.
`predictions`	A list of matrices with the predictions of survivals of each subject (lines) for each observed times (columns). Each matrix corresponds to the included `methods` and the resulted SL (the last item entitled "sl"). If `keep.predictions=TRUE`, it corresponds to a matrix with predictions related to the SL.
`data`	The data frame used for learning. The first column is entitled `times` and corresponds to the observed follow-up times. The second column is entitled `failures` and corresponds to the event indicators. The other columns correspond to the predictors.
`predictors`	A list with the predictors involved in `group`, `cov.quanti` and `cov.quali`.
`ROC.precision`	The percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve.
`cv`	The number of splits for cross-validation.
`pro.time`	The maximum delay for which the capacity of the variable is evaluated.
`models`	A list with the estimated models/algorithms included in the SL.
`weights`	A list composed by two vector: the regressions `coefficients` of the logistic multinomial regression and the resulting weights' `values`
`metric`	The loss function used to estimate the weights of the algorithms in the SL.
`param.tune`	The estimated tunning parameters.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Polley E and van der Laanet M. Super Learner In Prediction. http://biostats.bepress.com/ucbbiostat/paper266. 2010.

Examples

```
data(dataDIVAT2)

dataDIVAT2$train <- 1*rbinom(n=dim(dataDIVAT2)[1], size = 1, prob=1/3)

# The training of the super learner with 2 algorithms
sl<-sl.time(
```

```
   methods=c("cox.lasso", "aft.ggamma"), metric="ribs",
   data=dataDIVAT2[dataDIVAT2$train==1,],  times="times",
   failures="failures", pro.time = 12, cov.quanti=c("age"),
   cov.quali=c("hla", "retransplant", "ecd"), cv=5)

# Individual prediction
pred <- predict(sl, newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions$sl[1,], x=pred$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))

lines(y=pred$predictions$sl[2,], x=pred$times, col=2, type="l", lty=1, lwd=2)

legend("topright", col=c(1,2), lty=1, lwd=2, c("Subject #1", "Subject #2"))
```

---

summary.sl.time   *Summaries of a Super Learner*

Description

Return goodness-of-fit indicators of a Super Learner obtained by the function sl.time.

Usage

```
## S3 method for class 'sl.time'
summary(object, ..., method, pro.time, newdata, times, failures)
```

Arguments

object	An object returned by the function sl.time.
...	Additional arguments affecting the plot.
method	A character string with the name of the algorithm included in the SL for wich the calibration plot is performed. The defaut is "sl" for the Super Learner.
pro.time	The prognostic time up to which the time-dependent indicators are estimated.
newdata	An optional data frame containing the new sample for validation with covariate values, folow-up times, and event status. The default value isNULL, the calibration plot is performed from the same subjects of the training sample.
times	The name of the variable related the numeric vector with the follow-up times in newdata (optional argument only necessary when newdata is not NULL).
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in newdata (optional argument only necessary when newdata is not NULL).

Details

The following metrics are returned: "brier" for the Brier score at the prognostic time `pro.time`, "loglik" for the Log-likelihood, "ibs" for the Integrated Brier score up to the last observed time of event, "ibll" for the Integrated Binomial Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "ribs" for the restricted Integrated Brier score up to the prognostic time `pro.time`, "ribll" for the restricted Integrated Binomial Log-likelihood Log-likelihood up to the last observed time of event, "bll" for the binomial Log-likelihood, "auc" for the area under the time-dependent ROC curve up to the prognostic time `pro.time`.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

Examples

```
data(dataDIVAT2)

dataDIVAT2$train <- 1*rbinom(n=dim(dataDIVAT2)[1], size = 1, prob=1/3)


# The training of the super learner with 2 algorithms
sl<-sl.time(method=c("aft.gamma", "ph.gompertz"),  metric="ibs",
 data=dataDIVAT2[dataDIVAT2$train==1,], times="times", failures="failures", pro.time = 12,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),  cv=3)

# The prognostic capacities from the validation sample
summary(sl, method="sl", pro.time=12,
     newdata=dataDIVAT2[dataDIVAT2$train==0,],
     times="times", failures="failures")
```

---

survival.mr	*Multiplicative-Regression Model to Compare the Risk Factors Between Two Reference and Relative Populations*

Description

Compute a multiplicative-regression model to compare the risk factors between a reference and a relative population.

Usage

```
survival.mr(times, failures, cov.relative, data,
cox.reference, cov.reference, ini, iterations)
```

Arguments

`times`	The column name in `data`, in which the time of follow-up of each individual is collected.
`failures`	The column name in `data`, in which the indicator of event at the end of follow-up is collected (1 if the event is observed and 0 if right censoring).
`cov.relative`	The column(s) name(s) in `data` in order to declare the explicative variable included in the multiplicative relative model.
`data`	A data frame with the variables (columns) of the individuals (raw) of the relative sample.
`cox.reference`	The results of the Cox model performed in the reference sample, i.e an object obtained by the coxph function.
`cov.reference`	The column(s) name(s) in `data` in order to declare the explicative variable corresponding to those included in the Cox model `cox.reference`. Please, note that the order of these variables is important and have to be similar with the order in `cox.reference`.
`ini`	A vector with the same length than `cov.relative` with the initial values for the parameters to be optimized.
`iterations`	The number of iterations of the bootstrap resampling.

Details

We proposed here an adaptation of a multiplicative-regression model for relative survival to study the heterogeneity of risk factors between two groups of patients. Estimation of parameters is based on partial likelihood maximization and Monte-Carlo simulations associated with bootstrap re-sampling yields to obtain the corresponding standard deviations. The expected hazard ratios are obtained by using a PH Cox model.

Value

`matrix.coef`	A matrix containing the parameters estimations at each of the B iterations.
`estim.coef`	A numerical vector containing the mean of the previous estimation
`lower95.coef`	A numerical vector containing the lower bounds of the 95% confidence intervals.
`upper95.coef`	A numerical vector containing the upper bounds of the 95% confidence intervals.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

K. Trebern-Launay <katygre@yahoo.fr>

References

K. Trebern-Launay, M. Giral, J. Dantal and Y. Foucher. Comparison of the risk factors effects between two populations: two alternative approaches illustrated by the analysis of first and second kidney transplant recipients. BMC Med Res Methodol. 2013 Aug 6;13:102. <doi: 10.1186/1471-2288-13-102>

Examples

```
# import and attach both samples
data(dataFTR)
data(dataSTR)

# We reduce the dimension to save time for this example (CRAN policies)
# Compute the Cox model in the First Kidney Transplantations (FTR)
cox.FTR<-coxph(Surv(Tps.Evt, Evt)~ ageR2cl + sexeR, data=dataFTR[1:100,])
summary(cox.FTR)

# Compute the multiplicative relative model
# for Second Kidney Transplantations  (STR)
# Choose iterations>>5 for real applications
mrs.STR <- survival.mr(times="Tps.Evt", failures="Evt",
 cov.relative=c("ageR2cl", "Tattente2cl"), data=dataSTR[1:100,],
 cox.reference=cox.FTR, cov.reference=c("ageR2cl", "sexeR"),
 ini=c(0,0), iterations=5)


# The parameters estimations (mean of the values)
mrs.STR$estim.coef

# The 95 percent. confidence intervals
cbind(mrs.STR$lower95.coef, mrs.STR$upper95.coef)
```

---

survival.summary   *Summary Survival Curve From Aggregated Data*

Description

Estimation of the summary survival curve from the survival rates and the numbers of at-risk individuals extracted from studies of a meta-analysis.

Usage

```
survival.summary(study, time, n.risk, surv.rate, confidence)
```

Arguments

study	A numeric vector with the numbering of the studies included in the meta-analysis. The numbering of a study is repeated for each survival probabilities extracted from this study.
time	A numeric vector with the time at which the survival probabilities are collected.
n.risk	A numeric vector with the number of at-risk patients in the study for each value of thr time.
surv.rate	A numeric vector with the survival rates collected per study for each value of time.

confidence   A text argument indicating the method to calculate the 95% confidence interval of the summary survival probabilities: "Greenwood" or "MonteCarlo".

Details

The survival probabilities have to be extracted at the same set of points in time for all studies. Missing data are not allowed. The studies included in the meta-analysis can have different length of follow-up. For a study ending after the time t, all survival probabilities until t have to be entered in data. The data are sorted by study and by time. The conditional survival probabilities are arc-sine transformed and thus pooled assuming fixed effects or random effects. A correction of 0.25 is applied to the arc-sine transformation. For random effects, the multivariate methodology of DerSimonian and Laird is applied and the between-study covariances are accounted. The summary survival probabilities are obtained by the product of the pooled conditional survival probabilities. The mean and median survival times are derived from the summary survival curve assuming a linear interpolation of the survival between the points.

Value

verif.data   A data frame in which the first column (study) correspond to the number of the study and the second column (check) equals 1 if the time of collection for this study respects the other times for the other studies and 0 otherwise. Remember that the times of survival rates have to be identical between studies. The end of each study can be different. If at least one study did not respect this format, the other arguments values are non-attributed (NA).

summary.fixed   A matrix containing the summarized survival probabilities assuming fixed effects. The first column contains the time at which the summary survivals are computed. The second column contains the estimations of the summary survival probabilities. The third and fourth columns contain the lower and the upper bound of the 95% confidence interval, computed by either the Greenwood or the Monte Carlo approach as specified by the user.

median.fixed   A numerical vector containing the estimated median survival time computed from the summary survival curve assuming fixed effects and the lower and upper bounds of the 95% confidence interval computed by a Monte Carlo approach.

mean.fixed   A numerical vector containing the estimated mean survival time computed from the summary survival curve assuming fixed effects and the lower and upper bounds of the 95% confidence interval computed by a Monte Carlo approach.

heterogeneity   A numerical vector containing the value of the Q statistic for the heterogeneity, the H index and the I-squared index (in percentage).

summary.random A matrix containing the summarized survival probabilities assuming random effects. The first column contains the time at which the summary survivals are computed. The second column contains the estimations of the summary survival probabilities. The third and fourth columns contain the lower and the upper bound of the 95% confidence interval around the summary survival probabilities, computed by either the Greenwood or the Monte Carlo approach as specified by the user.

median.random   A numerical vector containing the estimated median survival time computed from the summary survival curve assuming random effects and the lower and

upper bounds of the 95% confidence interval computed by a Monte Carlo approach.

mean.random   A numerical vector containing the estimated mean survival time computed from the summary survival curve assuming random effects and the lower and upper bounds of the 95% confidence interval computed by a Monte Carlo approach.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

D. Jackson <daniel.jackson@mrc-bsu.cam.ac.uk>

C. Combescure <Christophe.Combescure@hcuge.ch>

References

Combescure et al. The multivariate DerSimonian and Laird's methodology applied to meta-analysis of survival curves. 10;33(15):2521-37, 2014. Statistics in Medicine. <doi:10.1002/sim.6111>

Examples

```
# import and attach the data example
data(dataHepatology)
attach(dataHepatology)

# computation of the summary survivals

results<-survival.summary(study, time, n.risk, survival, confidence="Greenwood")
results

# plot the estimated summary survival curve against the extracted ones

RandomEffectSummary<- results$summary.random

plot(time, survival, type="n", col="grey", ylim=c(0,1),xlab="Time",
 ylab="Survival")

for (i in unique(sort(study)))
{
lines(time[study==i], survival[study==i], type="l", col="grey")
points(max(time[study==i]),
 survival[study==i & time==max(time[study==i])], pch=15)
}

lines(RandomEffectSummary[,1], RandomEffectSummary[,2], type="l",
 col="red", lwd=3)
points(RandomEffectSummary[,1], RandomEffectSummary[,3], type="l",
 col="red", lty=3, lwd=3)
points(RandomEffectSummary[,1], RandomEffectSummary[,4], type="l",
 col="red", lty=3, lwd=3)
```

---

survival.summary.strata

Summary Survival Curve And Comparison Between Strata.

Description

Estimation of the summary survival curve from the survival rates and the numbers of at-risk individuals extracted from studies of a meta-analysis and comparisons between strata of studies.

Usage

```
survival.summary.strata(study, time, n.risk, surv.rate, confidence, strata)
```

Arguments

study
: A numeric vector with the numbering of the studies included in the meta-analysis. The numbering of a study is repeated for each survival probabilities extracted from this study.

time
: A numeric vector with the time at which the survival probabilities are collected.

n.risk
: A numeric vector with the number of at-risk patients in the study for each value of thr `time`.

surv.rate
: A numeric vector with the survival rates collected per study for each value of `time`.

confidence
: A text argument indicating the method to calculate the 95% confidence interval of the summary survival probabilities: "Greenwood" or "MonteCarlo".

strata
: A factor designing the strata. Each stratum has to contain at least two studies.

Details

The survival probabilities have to be extracted at the same set of points in time for all studies. Missing data are not allowed. The studies included in the meta-analysis can have different length of follow-up. For a study ending after the time t, all survival probabilities until t have to be entered in data. The data are sorted by study and by time. The conditional survival probabilities are arcsine transformed and thus pooled assuming fixed effects or random effects. A correction of 0.25 is applied to the arc-sine transformation. For random effects, the multivariate methodology of DerSimonian and Laird is applied and the between-study covariances are accounted. The summary survival probabilities are obtained by the product of the pooled conditional survival probabilities. The mean and median survival times are derived from the summary survival curve assuming a linear interpolation of the survival between the points. The summary survival curve is assessed in each stratum. The duration of follow-up is the greatest duration for which each stratum contains at least two studies reporting the survival at this duration. The between-strata is assessed and tested.

Value

verif.data	A data frame in which the first column (study) correspond to the number of the study and the second column (check) equals 1 if the time of collection for this study respects the other times for the other studies and 0 otherwise. Remember that the times of survival rates have to be identical between studies. The end of each study can be different. If at least one study did not respect this format, the other arguments values are non-attributed (NA).
summary.fixed	A list list of matrix. Each matrix contains the summarized survival probabilities assuming fixed effects. Each matrix provides the results for one stratum. The first column contains the time at which the summary survivals are computed. The second column contains the estimations of the summary survival probabilities. The third and fourth columns contain the lower and the upper bound of the 95% confidence interval, computed by either the Greenwood or the Monte Carlo approach as specified by the user. The last element of the list is the summary survival when all strata are pooled.
summary.random	A list object containing the summarized survival probabilities in each stratum assuming random effects. The results are presented similarly as summary.fixed.
median.fixed	A numerical vector containing the estimated median survival time computed from the summary survival curve assuming fixed effects and the lower and upper bounds of the 95% confidence interval computed by a Monte Carlo approach.
mean.fixed	A numerical vector containing the estimated mean survival time computed from the summary survival curve assuming fixed effects and the lower and upper bounds of the 95% confidence interval computed by a Monte Carlo approach.
heterogeneity	A numerical vector containing the value of the Q statistic for the heterogeneity, the H index and the I-squared index (in percentage).
p.value	The p-value of the test for the null hypothesis that the between-strata heterogeneity is null.

Author(s)

Y. Foucher <Yohann.Foucher@univ-nantes.fr>

D. Jackson <daniel.jackson@mrc-bsu.cam.ac.uk>

C. Combescure <Christophe.Combescure@hcuge.ch>

References

Combescure et al. The multivariate DerSimonian and Laird's methodology applied to meta-analysis of survival curves. 10;33(15):2521-37, 2014. Statistics in Medicine. <doi:10.1002/sim.6111>

Examples

```
# import and attach the data example
data(dataHepatology)

attach(dataHepatology)
```

```
# computation of the summary survivals

results<-survival.summary.strata(study = study, time = time, n.risk = n.risk,
 surv.rate = survival, confidence="Greenwood", strata = location)

results
```

---

tune.cox.aic   *Tune cox step AIC with forward selection*

Description

This function finds the model which minimize the AIC of a cox PH model

Usage

```
tune.cox.aic(times, failures, group, cov.quanti,
cov.quali, data, mini.model.cov, maxi.model.cov)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
mini.model.cov	the minimal covariate
maxi.model.cov	the maximal covariate

Details

The function runs the stepAIC function of the MASS package.

Value

optimal	The names of covariate to adjuste the fit
results	The result of the stepAIC process

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth edition. Springer.

Examples

```
data(dataDIVAT2)

tune.model<-tune.cox.aic(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"))

tune.model$optimal$final.model.cov # the covariate to include in the model with the best AIC

# The estimation of the training modelwith the corresponding lambda value
model<-cox.aic(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
  final.model.cov=tune.model$optimal$final.model.cov)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tune.cox.en	*Tune Elastic Net Cox Regression*

Description

This function finds the optimal lambda and alpha parameters for an elastic net Cox regression.

Usage

```
tune.cox.en(times, failures, group, cov.quanti,
cov.quali, data, cv, parallel, alpha, lambda)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
cv	The value of the number of folds. The default value is 10.
parallel	If TRUE, use parallel foreach to fit each fold. The default is FALSE.
alpha	The values of the regularization parameter alpha optimized over.
lambda	The values of the regularization parameter lambda optimized over.

Details

The function runs the cv.glmnet function of the glmnet package.

Value

optimal	The value of lambda that gives the minimum mean cross-validated error.
results	The data frame with the mean cross-validated errors for each lambda values.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

tune.model<-tune.cox.en(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), cv=5,
  alpha=seq(.1, 1, by=.1), lambda=seq(.1, 1, by=.1))

tune.model$optimal$lambda # the estimated lambda value

# The estimation of the training modelwith the corresponding lambda value
model<-cox.ridge(times="times", failures="failures", data=dataDIVAT2,
 cov.quanti=c("age"), cov.quali=c("hla", "retransplant", "ecd"), lambda=tune.model$optimal$lambda)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tune.cox.lasso	*Tune Lasso Cox Regression*

Description

This function finds the optimal lambda parameter for a Lasso Cox regression.

Usage

```
tune.cox.lasso(times, failures, group, cov.quanti,
cov.quali, data, cv, parallel, lambda)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.

data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
cv	The value of the number of folds. The default value is 10.
parallel	If TRUE, use parallel foreach to fit each fold. The default is FALSE.
lambda	The values of the regularization parameter lambda optimized over.

Details

The function runs the cv.glmnet function of the glmnet package.

Value

optimal	The value of lambda that gives the minimum mean cross-validated error.
results	The data frame with the mean cross-validated errors for each lambda values.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

tune.model<-tune.cox.lasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
  cv=5, lambda=seq(0, 10, by=.1))

tune.model$optimal$lambda # the estimated lambda value

# The estimation of the training modelwith the corresponding lambda value
model<-cox.lasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
  lambda=tune.model$optimal$lambda)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tune.cox.ridge   *Tune Ridge Cox Regression*

Description

This function finds the optimal lambda parameter for a ridge Cox regression.

Usage

```
tune.cox.ridge(times, failures, group, cov.quanti,
cov.quali, data, cv, parallel, lambda)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
cv	The value of the number of folds. The default value is 10.
parallel	If TRUE, use parallel foreach to fit each fold. The default is FALSE.
lambda	The values of the regularization parameter lambda optimized over.

Details

The function runs the cv.glmnet function of the glmnet package.

Value

optimal	The value of lambda that gives the minimum mean cross-validated error.
results	The data frame with the mean cross-validated errors for each lambda values.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Simon, N., Friedman, J., Hastie, T. and Tibshirani, R. (2011) Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent, Journal of Statistical Software, Vol. 39(5), 1-13, https://www.jstatsoft.org/v39/i05/

Examples

```
data(dataDIVAT2)

tune.model<-tune.cox.ridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
  cv=5, lambda=seq(0, 10, by=.1))

tune.model$optimal$lambda # the estimated lambda value

# The estimation of the training modelwith the corresponding lambda value
model<-cox.ridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
  lambda=tune.model$optimal$lambda)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

| tune.nnet.time | *Tune a 1-Layer Survival Neural Network* |

Description

This function finds the optimal n.nodes, decay, batch.size, and epochs parameters for a survival neural network

Usage

```
tune.nnet.time(times, failures, group, cov.quanti,
cov.quali, data, cv, n.nodes, decay, batch.size, epochs)
```

Arguments

`times`	The name of the variable related the numeric vector with the follow-up times.
`failures`	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
`group`	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
`cov.quanti`	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
`cov.quali`	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
`data`	A data frame for training the model in which to look for the variables related to the status of the follow-up time (`times`), the event (`failures`), the optional treatment/exposure (`group`) and the covariables included in the previous model (`cov.quanti` and `cov.quali`).
`cv`	The value of the number of folds. The default value is 10.
`n.nodes`	The number of hidden nodes optimized over.
`decay`	The value of the weight decay optimized over.
`batch.size`	The value of batch size
`epochs`	The value of epochs

Details

The function runs the `deepsurv` function of the `survivalmodels` package.

Value

`optimal`	The value of lambda that gives the minimum mean cross-validated error.
`results`	The data frame with the mean cross-validated errors for each lambda values.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Medical Research Methodology, 18(1), 24. https://doi.org/10.1186/s12874-018-0482-1

Examples

```
# On a linux machine

#Install Python

#sudo apt install python3-pip

#Install Python Miniconda for CPU computaion on a linux machine

#Download the file from https://docs.conda.io/en/latest/miniconda.html#linux-installers

#chmod +x Miniconda3-latest-Linux-x86_64.sh
#./Miniconda3-latest-Linux-x86_64.sh
#conda create --name torch python=3.8
#conda activate torch
#conda install pytorch torchvision torchaudio cpuonly -c pytorch
#conda install -c conda-forge pycox

#export PATH="/home/username/.local/bin:$PATH"

# remotes::install_github("RaphaelS1/survivalmodels")
# install_pycox(method = "auto",  conda = "auto",  pip = TRUE,  install_torch = TRUE )
# install_keras(method = "auto",  conda = "auto",  pip = TRUE,  install_tensorflow = TRUE)

# use_condaenv(condaenv = "~/miniconda3/envs/torch", conda = "auto", required = FALSE)

#data(dataDIVAT2)

# The estimation of the hyperparameters
#tune.model<-tune.nnet.time(times="times", failures="failures", data=dataDIVAT2[1:500,],
# cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
# cv=3, n.nodes=c(1, 2), decay=0, batch.size=256L, epochs=1L)

# the estimated nodesize value
#tune.model$optimal
#tune.model$results

# The estimation of the training modelwith the corresponding lambda value

#model<-nnet.time(times="times", failures="failures", data=dataDIVAT2,
# cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
# n.nodes=tune.model$optimal$n.nodes, decay=0, batch.size=256L, epochs=1L)

# The resulted predicted survival of the first subject of the training sample
#plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
#      col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

---

tune.rf.time   *Tune Survival Random Forest Tree*

Description

This function finds the optimal nodesize, mtry, and ntree parameters for a survival random forest tree

Usage

```
tune.rf.time(times, failures, group, cov.quanti,
cov.quali, data, cv, nodesize, mtry, ntree)
```

Arguments

times	The name of the variable related the numeric vector with the follow-up times.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
group	The name of the variable related to the exposure/treatment. This variable shall have only two modalities encoded 0 for the untreated/unexposed patients and 1 for the treated/exposed ones. The default value is NULL: no specific exposure/treatment is considered. When a specific exposure/treatment is considered, it will be forced in the algorithm or related interactions will be tested when possible.
cov.quanti	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.quali	The name(s) of the variable(s) related to the possible qualitative covariates. These variables must be numeric with two levels: 0 and 1. A complete disjunctive form must be used for covariates with more levels.
data	A data frame for training the model in which to look for the variables related to the status of the follow-up time (times), the event (failures), the optional treatment/exposure (group) and the covariables included in the previous model (cov.quanti and cov.quali).
cv	The value of the number of folds. The default value is 10.
nodesize	The values of the node size optimized over.
mtry	The numbers of variables randomly sampled as candidates at each split optimized over.
ntree	The numbers of trees optimized over.

Details

The function runs the tune.rfsrc function of the randomForestSRC package.

Value

optimal	The value of lambda that gives the minimum mean cross-validated error.
results	The data frame with the mean cross-validated errors for each lambda values.

Author(s)

Yohann Foucher <Yohann.Foucher@univ-nantes.fr>

Camille Sabathe <camille.sabathe@univ-nantes.fr>

References

Ishwaran H. and Kogalur U.B. (2007). Random survival forests for R, Rnews, 7(2):25-31.

Examples

```
data(dataDIVAT2)

tune.model<-tune.rf.time(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"), cv=5,
  nodesize=c(100, 250, 500), mtry=1, ntree=100)

tune.model$optimal # the estimated nodesize value

# The estimation of the training modelwith the corresponding lambda value
model<-rf.time(times="times", failures="failures", data=dataDIVAT2,
  cov.quanti=c("age"),  cov.quali=c("hla", "retransplant", "ecd"),
  nodesize=tune.model$optimal$nodesize, mtry=1, ntree=100)

# The resulted predicted survival of the first subject of the training sample
plot(y=model$predictions[1,], x=model$times, xlab="Time (years)", ylab="Predicted survival",
     col=1, type="l", lty=1, lwd=2, ylim=c(0,1))
```

Index