

Package ‘survivalSL’

June 25, 2023

Type Package

Title Super Learner for Survival Prediction from Censored Data

Version 0.91

Depends R (>= 4.0.0), splines, survival, reticulate, tune

Imports date, graphics, MASS, nnet, kernlab, glmnet, caret,
SuperLearner, flexsurv, randomForestSRC, survivalmodels,
prodlim, hdnom, glmnetUtils, mosaic, mosaicCalc, cubature,
timeROC, rpart, methods

Description

Several functions and S3 methods to construct a super learner in the presence of censored times-to-event and to evaluate its prognostic capacities.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Author Yann Foucher [aut, cre] (<<https://orcid.org/0000-0003-0330-7457>>),
Camille Sabathe [aut] (<<https://orcid.org/0000-0001-9353-691X>>)

Maintainer Yann Foucher <yohann.foucher@univ-poitiers.fr>

Repository CRAN

Date/Publication 2023-06-25 14:40:02 UTC

R topics documented:

aft.gamma	3
aft.ggamma	4
aft.llogis	6
aft.weibull	7
auc	9
cox.aic	10
cox.all	11
cox.en	13
cox.lasso	14
cox.ridge	16

dataDIVAT2	18
dataDIVAT3	19
dataOFSEP	20
differentiation	21
lines.rocrisca	21
metric	22
nn.time	24
ph.exponential	25
ph.gompertz	27
plot.cox	28
plot.flexsurv	30
plot.nn.time	31
plot.rf.time	33
plot.rocrisca	34
plot.sl.time	35
predict.cox	36
predict.flexsurv	38
predict.nn.time	39
predict.rf.time	40
predict.sl.time	41
print.cox	42
print.flexsurv	43
print.nn.time	43
print.rf.time	44
print.sl.time	45
rf.time	45
roc.time	47
sl.time	49
summary.cox	52
summary.flexsurv	53
summary.nn.time	55
summary.rf.time	56
summary.sl.time	57
tune.cox.aic	58
tune.cox.en	60
tune.cox.lasso	61
tune.cox.ridge	63
tune.nn.time	64
tune.rf.time	65

aft.gamma	<i>Library of the Super Learner for an Accelerated Failure Time (AFT) Model with a Gamma Distribution</i>
-----------	---

Description

Fit an AFT parametric model with a gamma distribution.

Usage

```
aft.gamma(times, failures, group, cov.quant, cov.qual, 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.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.qual	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.quant and cov.qual).

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.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	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 cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

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 from the first 200 lignes
model<-aft.ggamma(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=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	<i>Library of the Super Learner for an Accelerated Failure Time (AFT) Model with a Generalized Gamma Distribution</i>
------------	---

Description

Fit an AFT parametric model with a generalized gamma distribution.

Usage

```
aft.ggamma(times, failures, group, cov.quant, 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).

<code>group</code>	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.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	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.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time (<code>times</code>), the event (<code>failures</code>), the optional treatment/exposure (<code>group</code>) and the covariables included in the previous model (<code>cov.quant</code> and <code>cov.qual</code>).

Details

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

Value

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

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 from the first 200 lignes
```

```

model<-aft.ggamma(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=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)
Model with a Log Logistic Distribution*

Description

Fit an AFT parametric model with a log logistic distribution.

Usage

```
aft.llogis(times, failures, group, cov.quant, 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.quant	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.quant 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.quant	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 <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> 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 cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed time (columns).

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 from the first 200 lignes
model<-aft.llogis(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=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) Model with a Weibull Distribution*

Description

Fit an AFT parametric model with a Weibull distribution.

Usage

```
aft.weibull(times, failures, group, cov.quant, cov.quali=NULL, data)
```

Arguments

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	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.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	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.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time (<code>times</code>), the event (<code>failures</code>), the optional treatment/exposure (<code>group</code>) and the covariables included in the previous model (<code>cov.quant</code> and <code>cov.qual</code>).

Details

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

Value

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

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 from the first 200 lignes
model<-aft.weibull(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=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	<i>Area Under ROC Curve from Sensitivities and Specificities.</i>
-----	---

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.

Value

The estimation of the area under ROC curve.

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	<i>Library of the Super Learner for a Cox Model with Selected Covariates</i>
---------	--

Description

Fit a Cox regression for a selection of covariate

Usage

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

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.quant	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 covariates included in the previous model (cov.quant and cov.quali)
final.model	The covariates to consider

Value

model	The estimated model.
group	The name of the variable related to the exposure/treatment.
cov.quant	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 cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

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=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.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.qual	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.quant and cov.qual).

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.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	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 cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

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>.

Examples

```
data(dataDIVAT2)

# The estimation of the model
model<-cox.all(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=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))
```

Description

Fit an elastic net Cox regression for fixed values of the regularization parameters.

Usage

```
cox.en(times, failures, group, cov.quant, 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.quant	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.quant 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.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	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 <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> 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 cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

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.quant=c("age"), cov.qual=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 fixed value of the regularization parameter.

Usage

```
cox.lasso(times, failures, group, cov.quant, cov.qual,
  data, lambda)
```

Arguments

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	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.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.quali</code>	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.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time (<code>times</code>), the event (<code>failures</code>), the optional treatment/exposure (<code>group</code>) and the covariables included in the previous model (<code>cov.quant</code> and <code>cov.quali</code>).
<code>lambda</code>	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

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

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.quant=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 fixed value of the regularization parameter.

Usage

```
cox.ridge(times, failures, group, cov.quant, 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.quant	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 (<code>times</code>), the event (<code>failures</code>), the optional treatment/exposure (<code>group</code>) and the covariables included in the previous model (<code>cov.quant</code> and <code>cov.qual</code>).
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.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	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 <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> 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 cumulative baseline hazard function at the prediction times.
predictions	A matrix with the predictions of survivals of each subject (lines) for each observed times (columns).

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.quant=c("age"), cov.qual=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))
```

`dataDIVAT2`*A 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).

`h1a` 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 Expanded 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 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)
```

 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 least 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.

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")
```

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)
```

lines.rocrisca *Add Lines to a ROC Plot*

Description

Used to add an additional 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.

Value

No return value for this S3 method.

Examples

```
# import and attach the data example
data(dataDIVAT3)

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

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.

# 0. Compute the raw sensitivity and specificity
roc1 <- roc.time(times="death.time", failures="death", variable="ageR",
confounders=~1, data=dataDIVAT3, pro.time=2000,
precision=seq(0.1,0.9, by=0.2))

# 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 AUCs
c(roc1$auc, roc2$auc)

# The corresponding ROC graph

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

lines(roc2, type="b", col=2, 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="") )
```

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

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>data</code>	A data frame for in which to look for the variables related to the status of the follow-up time (<code>times</code>) and the event (<code>failures</code>).
<code>prediction.matrix</code>	A matrix with the predictions of survivals of each subject (lines) for each prognostic times (columns).
<code>prediction.times</code>	A vector of numeric values with the times of the predictions (same length than the number of columns of <code>prediction.matrix</code>).
<code>metric</code>	The metric to compute. See details.
<code>pro.time</code>	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 <code>times</code> . 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.
<code>ROC.precision</code>	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 <code>metric="auc"</code> . 0 (min) and 1 (max) are not allowed. By default, the precision is <code>seq(.01, .99, .01)</code> .

Details

The following metrics can be used: "bs" 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 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.

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-transplantation
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-transplantation
metric(times="times", failures="failures", data=dataDIVAT2,
  prediction.matrix=model$predictions, prediction.times=model$times,
  metric="ribs", pro.time=10)

```

nn.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

```
nn.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.quant	The name(s) of the variable(s) related to the possible quantitative covariates.
cov.qual	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 <code>times</code> and corresponds to the observed follow-up times. The second column is entitled <code>failures</code> 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 time (columns).

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>

ph.exponential	<i>Library of the Super Learner for a Proportional Hazards (PH) Model with an Exponential Distribution</i>
----------------	--

Description

Fit a PH model with an Exponential distribution.

Usage

```
ph.exponential(times, failures, group, cov.quant, cov.qual=NULL, data)
```

Arguments

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	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.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	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.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time (<code>times</code>), the event (<code>failures</code>), the optional treatment/exposure (<code>group</code>) and the covariables included in the previous model (<code>cov.quant</code> and <code>cov.qual</code>).

Details

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

Value

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

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 from the first 200 lignes
model<-ph.exponential(times="times", failures="failures", data=dataDIVAT2[1:200,],
  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) Model
with a Gompertz Distribution*

Description

Fit a PH parametric model with a Gompertz distribution.

Usage

```
ph.gompertz(times, failures, group, cov.quanti, cov.quali=NULL, 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="gompertz"` in the `flexsurvreg` package.

Value

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

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 from the first 200 lignes
model<-ph.gompertz(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.qual=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.cox

Calibration Plot for a Cox-like Model

Description

A calibration plot of an object of the class `cox`.

Usage

```
## S3 method for class 'cox'
plot(x, ..., col, lty, lwd, type, pch, ylab, xlab, ylim, xlim,
     cex, cex.lab, cex.axis, cex.main, n.groups, pro.time, newdata, times, failures)
```

Arguments

x	An object returned by the function <code>cox</code> .
...	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".
cex	A number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, etc.
cex.lab	The size of the axis label text with a numeric value of length 1.
cex.axis	The size of the tick label numbers/text with a numeric value of length 1.
cex.main	The size of the title text with a numeric value of length 1.
xlim	A vector with the x limits of the plot.
ylim	A vector with the y limits of the plot.
n.groups	A numeric value with the number of groups by their class probabilities. The default is 5.
pro.time	The prognostic time at which the calibration plot of the survival probabilities.
newdata	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, 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 <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).

Value

No return value for this S3 method.

See Also

[cox.all](#), [cox.en](#), [cox.lasso](#), [cox.ridge](#).

Examples

```

data(dataDIVAT2)
# The estimation of the model from the first 200 lignes
model<-cox.all(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The calibration plot from the validation sample of 150 patients
plot(model, n.groups=5, pro.time=12, col=2,
  xlab="Predicted 12-year survival", ylab="Observed 12-year survival",
  newdata=dataDIVAT2[151:300,], times="times", failures="failures")

```

plot.flexsurv

Calibration Plot for a Parametric Model

Description

A calibration plot of an object of the class flexsurv.

Usage

```

## S3 method for class 'flexsurv'
plot(x, ..., col, lty, lwd, type, pch, ylab, xlab, ylim, xlim,
  cex, cex.lab, cex.axis, cex.main, n.groups, pro.time, newdata, times, failures)

```

Arguments

x	An object returned by the function flexsurv.
...	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".
cex	A number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, etc.
cex.lab	The size of the axis label text with a numeric value of length 1.
cex.axis	The size of the tick label numbers/text with a numeric value of length 1.
cex.main	The size of the title text with a numeric value of length 1.
xlim	A vector with the x limits of the plot.

ylim	A vector with the y limits of the plot.
n.groups	A numeric value with the number of groups by their class probabilities. The default is 5.
pro.time	The prognostic time at which the calibration plot of the survival probabilities.
newdata	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, 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).

Value

No return value for this S3 method.

See Also

[aft.gamma](#), [aft.ggamma](#), [aft.llogis](#), [aft.weibull](#), [ph.exponential](#), [ph.gompertz](#).

Examples

```
data(dataDIVAT2)
# The estimation of the model from the first 200 lignes
model<-aft.gamma(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The calibration plot from the validation sample of 150 patients
plot(model, n.groups=5, pro.time=12, col=2,
  xlab="Predicted 12-year survival", ylab="Observed 12-year survival",
  newdata=dataDIVAT2[151:300,], times="times", failures="failures")
```

plot.nn.time

Calibration Plot for a Survival Neural Network

Description

A calibration plot of an object of the class nn.time.

Usage

```
## S3 method for class 'nn.time'
plot(x, ..., col, lty, lwd, type, pch, ylab, xlab, ylim, xlim,
  cex, cex.lab, cex.axis, cex.main, n.groups, pro.time, newdata, times, failures)
```

Arguments

<code>x</code>	An object returned by the function <code>nn.time</code> .
<code>...</code>	Additional arguments affecting the plot.
<code>col</code>	A numeric value with the color of the survival curves. The default is 1 for black.
<code>lty</code>	A numeric value with the type of the survival curves. The default is 1.
<code>lwd</code>	A numeric value with the type of the survival curves. The default is 1.
<code>type</code>	A character string giving the type of plot desired. The default is "b" for both the lines and points
<code>pch</code>	A numeric value for plotting characters or symbols. The default is 16.
<code>xlab</code>	The label for the x axis. The default is "Predicted survival".
<code>ylab</code>	The label for the y axis. The default is "Observed survival".
<code>cex</code>	A number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, etc.
<code>cex.lab</code>	The size of the axis label text with a numeric value of length 1.
<code>cex.axis</code>	The size of the tick label numbers/text with a numeric value of length 1.
<code>cex.main</code>	The size of the title text with a numeric value of length 1.
<code>xlim</code>	A vector with the x limits of the plot.
<code>ylim</code>	A vector with the y limits of the plot.
<code>n.groups</code>	A numeric value with the number of groups by their class probabilities. The default is 5.
<code>pro.time</code>	The prognostic time at which the calibration plot of the survival probabilities.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).

Value

No return value for this S3 method.

See Also

[nn.time](#).

plot.rf.time

*Calibration Plot for a Survival Random Forest***Description**

A calibration plot of an object of the class `rf.time`.

Usage

```
## S3 method for class 'rf.time'
plot(x, ..., col, lty, lwd, type, pch, ylab, xlab, ylim, xlim,
     cex, cex.lab, cex.axis, cex.main, n.groups, pro.time, newdata, times, failures)
```

Arguments

<code>x</code>	An object returned by the function <code>rf.time</code> .
<code>...</code>	Additional arguments affecting the plot.
<code>col</code>	A numeric value with the color of the survival curves. The default is 1 for black.
<code>lty</code>	A numeric value with the type of the survival curves. The default is 1.
<code>lwd</code>	A numeric value with the type of the survival curves. The default is 1.
<code>type</code>	A character string giving the type of plot desired. The default is "b" for both the lines and points
<code>pch</code>	A numeric value for plotting characters or symbols. The default is 16.
<code>xlab</code>	The label for the x axis. The default is "Predicted survival".
<code>ylab</code>	The label for the y axis. The default is "Observed survival".
<code>cex</code>	A number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, etc.
<code>cex.lab</code>	The size of the axis label text with a numeric value of length 1.
<code>cex.axis</code>	The size of the tick label numbers/text with a numeric value of length 1.
<code>cex.main</code>	The size of the title text with a numeric value of length 1.
<code>xlim</code>	A vector with the x limits of the plot.
<code>ylim</code>	A vector with the y limits of the plot.
<code>n.groups</code>	A numeric value with the number of groups by their class probabilities. The default is 5.
<code>pro.time</code>	The prognostic time at which the calibration plot of the survival probabilities.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> (optional argument only necessary when <code>newdata</code> is not NULL).

Value

No return value for this S3 method.

See Also

[rf.time.](#)

Examples

```
data(dataDIVAT2)
# The estimation of the model from the first 200 lignes
model<-rf.time(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"), nodesize=10,
  mtry=2, ntree=100)

# The calibration plot from the validation sample of 150 patients
plot(model, n.groups=5, pro.time=12, col=2,
  xlab="Predicted 12-year survival", ylab="Observed 12-year survival",
  newdata=dataDIVAT2[151:300,], times="times", failures="failures")
```

plot.rocrisca

Plot Method for 'rocrisca' Objects

Description

A plot of ROC curves is produced.

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.

Value

No return value for this S3 method.

Examples

```

data(dataDIVAT3)

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

dataDIVAT3 <- dataDIVAT3[1:400,]

# The stime-dependent ROC curve to evaluate the
# capacities of the recipient age for the prognosis of post kidney
# transplant mortality up to 2000 days.

# Compute the raw sensitivity and specificity
roc1 <- roc.time(times="death.time", failures="death", variable="ageR",
confounders=~1, data=dataDIVAT3, pro.time=2000,
precision=seq(0.1,0.9, by=0.2))

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

```

plot.sl.time

Calibration 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, lty, lwd, type, pch, ylab, xlab, ylim, xlim, cex, cex.lab,
     cex.axis, cex.main, method, n.groups, pro.time, newdata, times, failures)

```

Arguments

<code>x</code>	An object returned by the function <code>sl.time</code> .
<code>...</code>	Additional arguments affecting the plot.
<code>col</code>	A numeric value with the color of the survival curves. The default is 1 for black.
<code>lty</code>	A numeric value with the type of the survival curves. The default is 1.
<code>lwd</code>	A numeric value with the type of the survival curves. The default is 1.
<code>type</code>	A character string giving the type of plot desired. The default is "b" for both the lines and points
<code>pch</code>	A numeric value for plotting characters or symbols. The default is 16.
<code>xlab</code>	The label for the x axis. The default is "Predicted survival".
<code>ylab</code>	The label for the y axis. The default is "Observed survival".
<code>cex</code>	A number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, etc.

<code>cex.lab</code>	The size of the axis label text with a numeric value of length 1.
<code>cex.axis</code>	The size of the tick label numbers/text with a numeric value of length 1.
<code>cex.main</code>	The size of the title text with a numeric value of length 1.
<code>xlim</code>	A vector with the x limits of the plot.
<code>ylim</code>	A vector with the y limits of the plot.
<code>method</code>	A character string with the name of the algorithm included in the SL for which the calibration plot is performed. The default is "sl" for the Super Learner.
<code>n.groups</code>	A numeric value with the number of groups by their class probabilities. The default is 5.
<code>pro.time</code>	The prognostic time at which the calibration plot of the survival probabilities.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	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).
<code>failures</code>	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).

Value

No return value for this S3 method.

Examples

```
data(dataDIVAT2)

#The outcome model base on a Super Learner from 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)

# The calibration plot from the validation sample of 150 patients
plot(sl1, method="sl", n.groups=5, pro.time=12, col=2,
  xlab="Predicted 12-year survival", ylab="Observed 12-year survival",
  newdata=dataDIVAT2[151:300,], times="times", failures="failures")
```

predict.cox

Prediction from a Penalized Cox Regression

Description

Predict the survival of new observations based on a penalized Cox regression estimated by using 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: <code>cox.lasso</code> , <code>cox.ridge</code> , or <code>cox.en</code> .
...	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 <code>cov.quant</code> and <code>cov.quali</code> included in the training sample. The default value is <code>NULL</code> , 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 <code>NULL</code> , 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 time (columns).

See Also

[cox.all](#), [cox.en](#), [cox.lasso](#), [cox.ridge](#).

Examples

```
data(dataDIVAT2)

# The estimation of the training model
model<-cox.lasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=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 a 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.quant 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.

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 time (columns).

Examples

```
data(dataDIVAT2)

# The estimation of the model from the first 200 lignes
model<-ph.gompertz(times="times", failures="failures", data=dataDIVAT2[1:200,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# 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.nn.time *Prediction from a Survival Neural Network*

Description

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

Usage

```

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

```

Arguments

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

Value

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

See Also

[nn.time](#).

predict.rf.time *Prediction from a Survival Random Survival Forest*

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

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

Value

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

See Also

[rf.time](#).

Examples

```
data(dataDIVAT2)

# The estimation of the training model
model<-rf.time(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=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 a Super Learner for Censored Outcomes*

Description

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

Usage

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

Arguments

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

Value

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

See Also

[sl.time](#).

Examples

```
data(dataDIVAT2)

# The training of the super learner from the first 150 individuals of the data base
sl1<-sl.time(method=c("cox.ridge", "aft.ggamma"), metric="ribs",
  data=dataDIVAT2[1:150,], times="times", failures="failures", pro.time = 12,
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"), cv=3)
```

```
# Individual prediction for 2 new subjects
pred <- predict(s11,
  newdata=data.frame(age=c(52,52), hla=c(0,1), retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions$s1[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$predictions$s1[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"))
```

print.cox

S3 Method for Printing an 'cox' Object

Description

Print the Cox-like model.

Usage

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

Arguments

x An object returned by the function `cox`.
... Further arguments passed.

Value

No return value for this S3 method.

See Also

[cox.all](#), [cox.en](#), [cox.lasso](#), [cox.ridge](#).

Examples

```
data(dataDIVAT2)

model<-cox.all(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

print(model)
```

print.flexsurv	<i>S3 Method for Printing an 'flexsurv' Object</i>
----------------	--

Description

Print the model.

Usage

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

Arguments

x	An object returned by the function flexsurv.
...	Further arguments passed.

Value

No return value for this S3 method.

See Also

[aft.gamma](#), [aft.ggamma](#), [aft.llogis](#), [aft.weibull](#), [ph.exponential](#), [ph.gompertz](#).

Examples

```
data(dataDIVAT2)  
  
model<-aft.gamma(times="times", failures="failures", data=dataDIVAT2[1:100,],  
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))  
  
print(model)
```

print.nn.time	<i>S3 Method for Printing an 'nn.time' Object</i>
---------------	---

Description

Print the survival neural network.

Usage

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

Arguments

x An object returned by the function nn.time.
... Further arguments passed.

Value

No return value for this S3 method.

See Also

[nn.time.](#)

print.rf.time *S3 Method for Printing an 'rf.time' Object*

Description

Print the random survival forest.

Usage

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

Arguments

x An object returned by the function rf.time.
... Further arguments passed.

Value

No return value for this S3 method.

See Also

[rf.time.](#)

Examples

```
data(dataDIVAT2)  
  
model<-rf.time(times="times", failures="failures", data=dataDIVAT2,  
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),  
  nodesize=10, mtry=2, ntree=100)  
  
print(model)
```

print.sl.time	<i>S3 Method for Printing an 'sl.time' Object</i>
---------------	---

Description

Print the contribution of learners included in the Super Learner.

Usage

```
## S3 method for class 'sl.time'
print(x, ..., digits)
```

Arguments

x	An object returned by the function sl.time.
...	Further arguments passed.
digits	An optional integer for the number of digits to print when printing numeric values.

Value

No return value for this S3 method.

Examples

```
data(dataDIVAT2)

s11<-sl.time(method=c("cox.ridge", "aft.ggamma"), metric="ribs",
  data=dataDIVAT2[1:150,], times="times", failures="failures", pro.time = 12,
  cov.quantl=c("age"), cov.quali=c("hla", "retransplant", "ecd"), cv=3)

print(s11, digits=4)
```

rf.time	<i>Library of the Super Learner for Survival Random Survival Forest</i>
---------	---

Description

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

Usage

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

Arguments

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	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.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	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.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time (<code>times</code>), the event (<code>failures</code>), the optional treatment/exposure (<code>group</code>) and the covariables included in the previous model (<code>cov.quant</code> and <code>cov.qual</code>).
<code>nodesize</code>	The value of the node size.
<code>mtry</code>	The number of variables randomly sampled as candidates at each split.
<code>ntree</code>	The number of trees.

Details

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

Value

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

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.quantile=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))

```

roc.time

*Time-Dependent ROC Curves With Right Censored Data.***Description**

This function allows for the estimation of time-dependent ROC curve by considering 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 \sim 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. J represents the Youden index.
auc	The area under the time-dependent ROC curve for a prognostic up to <code>pro.time</code> .

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 et al. Standardized and weighted time-dependent ROC curves to evaluate the intrinsic prognostic capacities of a marker by taking into account confounding factors. *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 example

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
```


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.quant, cov.qual, 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.quant and cov.qual).
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.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.qual	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 tuning 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.

<code>optim.local.min</code>	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.
<code>ROC.precision</code>	The percentiles (between 0 and 1) of the prognostic variable used for computing each point of the time dependent ROC curve. Only used when <code>metric="auc"</code> . 0 (min) and 1 (max) are not allowed. By default: <code>seq(.01, .99, .01)</code> .
<code>param.weights.fix</code>	A vector with the parameters of the multinomial logistic regression which generates the weights of the algorithms declared in <code>methods</code> . 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.
<code>param.weights.init</code>	A vector with the initial values of the parameters of the multinomial logistic regression which generates the weights of the algorithms declared in <code>methods</code> . The default value is NULL: the initial values are equaled to 0. See details.
<code>keep.predictions</code>	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.
<code>verbose</code>	A logical value 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 tuning parameters of each algorithm are estimated by cv-fold cross-validation. Otherwise, the user can propose a tuning 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 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.all"	Usual Cox model	survival	PH
"cox.aic"	Cox model with selected covariates	MASS	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
"nn.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 time (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.quant and cov.qual.
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 vectors: the regressions coefficients of the logistic multinomial regression and the resulting weights' values
metric	A list composed by two vectors: the loss function used to estimate the weights of the algorithms in the SL and its value.
param.tune	The estimated tuning parameters.

References

Polley E and van der Laan M. Super Learner In Prediction. <http://biostats.bepress.com/ucbbiostat/paper266>. 2010.

Examples

```

data(dataDIVAT2)

#The outcome model base on a Super Learner and the first 150 individuals of the data base
s11<-sl.time( methods=c("aft.gamma", "ph.gompertz"), metric="ibs",
  data=dataDIVAT2[1:150,], times="times", failures="failures", group="ecd",
  cov.quant=c("age"), cov.quali=c("hla", "retransplant"), cv=3)

# Individual prediction
pred <- predict(s11, newdata=data.frame(age=c(52,52), hla=c(0,1),
retransplant=c(1,1), ecd=c(0,1)))

plot(y=pred$predictions$s1[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$predictions$s1[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.cox

*Summaries of a Cox-like Model***Description**

Return goodness-of-fit indicators of a cox-like model obtained by the function `cox`.

Usage

```

## S3 method for class 'cox'
summary(object, ..., digits, pro.time, newdata, times, failures)

```

Arguments

<code>object</code>	An object returned by the function <code>cox</code> .
<code>...</code>	Additional arguments affecting the plot.
<code>digits</code>	An optional integer for the number of digits to print when printing numeric values.
<code>pro.time</code>	The prognostic time up to which the time-dependent indicators are estimated.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is <code>NULL</code> , the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> . If <code>NULL</code> , the default is the name used in the learning sample.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> . If <code>NULL</code> , the default is the name used in the learning sample.

Details

The following metrics are returned: "brier" for the Brier score at the prognostic time `pro.time`, "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

No return value for this S3 method.

See Also

[cox.all](#), [cox.en](#), [cox.lasso](#), [cox.ridge](#).

Examples

```
data(dataDIVAT2)

# The training of the cox ridge model with the first 400 patients
model<-cox.ridge(times="times", failures="failures", data=dataDIVAT2[1:400,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"), lambda=1.2)

# The prognostic capacities from the same training sample
# (up to 4 years for several indicators)
summary(model, pro.time=4)

# The prognostic capacities from a validation of the next 150 patients
# (up to 4 years for several indicators)
summary(model, pro.time=4, newdata=dataDIVAT2[401:550,], times="times",
  failures="failures")
```

summary.flexsurv

Summaries of a Parametric Survival Model

Description

Return goodness-of-fit indicators of a parametric survival model obtained by the function `flexsurv`.

Usage

```
## S3 method for class 'flexsurv'
summary(object, ..., digits, pro.time, newdata, times, failures)
```

Arguments

object	An object returned by the function flexsurv.
...	Additional arguments affecting the plot.
digits	An optional integer for the number of digits to print when printing numeric values.
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, follow-up times, and event status. The default value is NULL, 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. If NULL, the default is the name used in the learning sample.
failures	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in newdata. If NULL, the default is the name used in the learning sample.

Details

The following metrics are returned: "brier" for the Brier score at the prognostic time `pro.time`, "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 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

No return value for this S3 method.

See Also

[aft.gamma](#), [aft.ggamma](#), [aft.llogis](#), [aft.weibull](#), [ph.exponential](#), [ph.gompertz](#).

Examples

```
data(dataDIVAT2)

# The training of the Weibull model with the first 400 patients
model<-ph.gompertz(times="times", failures="failures", data=dataDIVAT2[1:400,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"))

# The prognostic capacities from the same training sample
# (up to 4 years for several indicators)
summary(model, pro.time=4)

# The prognostic capacities from a validation of the next 150 patients
# (up to 4 years for several indicators)
summary(model, pro.time=4, newdata=dataDIVAT2[401:550,], times="times",
  failures="failures")
```

Description

Return goodness-of-fit indicators of survival neural network obtained by the function `nn.time`.

Usage

```
## S3 method for class 'nn.time'  
summary(object, ..., digits, pro.time, newdata, times, failures)
```

Arguments

<code>object</code>	An object returned by the function <code>nn.time</code> .
<code>...</code>	Additional arguments affecting the plot.
<code>digits</code>	An optional integer for the number of digits to print when printing numeric values.
<code>pro.time</code>	The prognostic time up to which the time-dependent indicators are estimated.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is <code>NULL</code> , the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> . If <code>NULL</code> , the default is the name used in the learning sample.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> . If <code>NULL</code> , the default is the name used in the learning sample.

Details

The following metrics are returned: "brier" for the Brier score at the prognostic time `pro.time`, "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 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

No return value for this S3 method.

summary.rf.time

*Summaries of Survival Random Forest***Description**

Return goodness-of-fit indicators of Survival Random Forest Tree obtained by the function `rf.time`.

Usage

```
## S3 method for class 'rf.time'
summary(object, ..., digits, pro.time, newdata, times, failures)
```

Arguments

<code>object</code>	An object returned by the function <code>rf.time</code> .
<code>...</code>	Additional arguments affecting the plot.
<code>digits</code>	An optional integer for the number of digits to print when printing numeric values.
<code>pro.time</code>	The prognostic time up to which the time-dependent indicators are estimated.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is <code>NULL</code> , the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> . If <code>NULL</code> , the default is the name used in the learning sample.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> . If <code>NULL</code> , the default is the name used in the learning sample.

Details

The following metrics are returned: "brier" for the Brier score at the prognostic time `pro.time`, "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

No return value for this S3 method.

See Also

[rf.time](#).

Examples

```

data(dataDIVAT2)

# The training of the random forest with the first 150 patients
model<-rf.time(times="times", failures="failures", data=dataDIVAT2[1:150,],
  cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"), nodesize=10,
  mtry=2, ntree=100)

# The prognostic capacities from the same training sample
# (up to 4 years for several indicators)
summary(model, pro.time=4)

# The prognostic capacities from a validation of the next 150 patients
# (up to 4 years for several indicators)
summary(model, pro.time=4, newdata=dataDIVAT2[151:300,], times="times",
  failures="failures")

```

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, ..., digits, method, pro.time, newdata, times, failures)

```

Arguments

<code>object</code>	An object returned by the function <code>sl.time</code> .
<code>...</code>	Additional arguments affecting the plot.
<code>digits</code>	An optional integer for the number of digits to print when printing numeric values.
<code>method</code>	A character string with the name of the algorithm included in the SL for which the calibration plot is performed. The default is "sl" for the Super Learner.
<code>pro.time</code>	The prognostic time up to which the time-dependent indicators are estimated.
<code>newdata</code>	An optional data frame containing the new sample for validation with covariate values, follow-up times, and event status. The default value is NULL, the calibration plot is performed from the same subjects of the training sample.
<code>times</code>	The name of the variable related the numeric vector with the follow-up times in <code>newdata</code> . If NULL, the default is the name used in the learning sample.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event) in <code>newdata</code> . If NULL, the default is the name used in the learning sample.

Details

The following metrics are returned: "brier" for the Brier score at the prognostic time `pro.time`, "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

No return value for this S3 method.

See Also

[sl.time](#).

Examples

```
data(dataDIVAT2)
dataDIVAT2$train <- 1*rbinom(n=dim(dataDIVAT2)[1], size = 1, prob=1/2)

# The training of the super learner with 2 algorithms from the
# first 100 patients of the training sample
sl<-sl.time(method=c("aft.gamma", "ph.gompertz"), metric="ibs",
  data=dataDIVAT2[dataDIVAT2$train==1,][1:100,], times="times", failures="failures",
  pro.time = 12, cov.quant=c("age"), cov.quali=c("hla", "retransplant", "ecd"),
  cv=3)

# The prognostic capacities from the same training sample
summary(sl)
```

tune.cox.aic

Tune a Cox Model with a Forward Selection Based on the AIC

Description

This function finds the model which minimize the AIC of a Cox PH model

Usage

```
tune.cox.aic(times, failures, group, cov.quant,
  cov.quali, data, model.min, model.max)
```

Arguments

<code>times</code>	The name of the variable related the numeric vector with the follow-up times.
<code>failures</code>	The name of the variable related the numeric vector with the event indicators (0=right censored, 1=event).
<code>group</code>	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.
<code>cov.quant</code>	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
<code>cov.qual</code>	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.
<code>data</code>	A data frame for training the model in which to look for the variables related to the status of the follow-up time (<code>times</code>), the event (<code>failures</code>), the optional treatment/exposure (<code>group</code>) and the covariables included in the previous model (<code>cov.quant</code> and <code>cov.qual</code>).
<code>model.min</code>	An optional argument with the minimal set of covariates.
<code>model.max</code>	An optional argument with the maximal set of covariates.

Details

The function runs the `stepAIC` function of the MASS package for covariates' selection.

Value

<code>optimal</code>	The names of covariate to adjuste the fit.
<code>results</code>	The result of the <code>stepAIC</code> process.

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.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"))

tune.model$optimal$final.model # the covariate to include in the model with the best AIC

# The estimation of the training model with the corresponding lambda value
model<-cox.aic(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=c("hla", "retransplant", "ecd"),
```

```

final.model=tune.model$optimal$final.model)

# 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.quant,
            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.quant	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.quant 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.

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.quant=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 model with the corresponding lambda value
model<-cox.ridge(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=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	<i>Tune Lasso Cox Regression</i>
----------------	----------------------------------

Description

This function finds the optimal lambda parameter for a Lasso Cox regression.

Usage

```
tune.cox.lasso(times, failures, group, cov.quant,
  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.quant	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.qual	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.quant and cov.qual).
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.

References

Simon et al. (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.quant=c("age"), cov.qual=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 model with the corresponding lambda value
model<-cox.lasso(times="times", failures="failures", data=dataDIVAT2,
  cov.quant=c("age"), cov.qual=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	<i>Tune Ridge Cox Regression</i>
----------------	----------------------------------

Description

This function finds the optimal lambda parameter for a ridge Cox regression.

Usage

```
tune.cox.ridge(times, failures, group, cov.quant,
               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.quant	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.quant 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.

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 model with 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.nn.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.nn.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.quant i	The name(s) of the variable(s) related to the possible quantitative covariates. These variables must be numeric.
cov.qual i	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.quant i and cov.qual i).
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.

References

Katzman et al. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1), 24. 1018. <https://doi.org/10.1186/s12874-018-0482-1>

tune.rf.time

Tune a Survival Random Forest

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.quant,
             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.quant	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.quant 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.

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 model with 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

- * **Dataset**
 - dataDIVAT2, 18
 - dataDIVAT3, 19
 - dataOFSEP, 20
- * **Metrics**
 - auc, 9
 - metric, 22
 - roc.time, 47
- * **Plotting**
 - lines.rocrisca, 21
 - plot.cox, 28
 - plot.flexsurv, 30
 - plot.nn.time, 31
 - plot.rf.time, 33
 - plot.rocrisca, 34
 - plot.sl.time, 35
- * **Predicting**
 - predict.cox, 36
 - predict.flexsurv, 38
 - predict.nn.time, 39
 - predict.rf.time, 40
 - predict.sl.time, 41
- * **Printing**
 - print.cox, 42
 - print.flexsurv, 43
 - print.nn.time, 43
 - print.rf.time, 44
 - print.sl.time, 45
- * **Summarizing**
 - summary.cox, 52
 - summary.flexsurv, 53
 - summary.nn.time, 55
 - summary.rf.time, 56
 - summary.sl.time, 57
- * **Super Learner Library**
 - aft.gamma, 3
 - aft.ggamma, 4
 - aft.llogis, 6
 - aft.weibull, 7
 - cox.aic, 10
 - cox.all, 11
 - cox.en, 13
 - cox.lasso, 14
 - cox.ridge, 16
 - nn.time, 24
 - ph.exponential, 25
 - ph.gompertz, 27
 - rf.time, 45
- * **Super Learner**
 - sl.time, 49
- * **Tunning**
 - tune.cox.aic, 58
 - tune.cox.en, 60
 - tune.cox.lasso, 61
 - tune.cox.ridge, 63
 - tune.nn.time, 64
 - tune.rf.time, 65
- aft.gamma, 3, 31, 43, 54
- aft.ggamma, 4, 31, 43, 54
- aft.llogis, 6, 31, 43, 54
- aft.weibull, 7, 31, 43, 54
- auc, 9
- cox.aic, 10
- cox.all, 11, 29, 37, 42, 53
- cox.en, 13, 29, 37, 42, 53
- cox.lasso, 14, 29, 37, 42, 53
- cox.ridge, 16, 29, 37, 42, 53
- dataDIVAT2, 18
- dataDIVAT3, 19
- dataOFSEP, 20
- differentiation, 21
- lines.rocrisca, 21
- metric, 22
- nn.time, 24, 32, 39, 44

ph.exponential, [25](#), [31](#), [43](#), [54](#)
ph.gompertz, [27](#), [31](#), [43](#), [54](#)
plot.cox, [28](#)
plot.flexsurv, [30](#)
plot.nn.time, [31](#)
plot.rf.time, [33](#)
plot.rocrisca, [34](#)
plot.sl.time, [35](#)
predict.cox, [36](#)
predict.flexsurv, [38](#)
predict.nn.time, [39](#)
predict.rf.time, [40](#)
predict.sl.time, [41](#)
print.cox, [42](#)
print.flexsurv, [43](#)
print.nn.time, [43](#)
print.rf.time, [44](#)
print.sl.time, [45](#)

rf.time, [34](#), [40](#), [44](#), [45](#), [56](#)
roc.time, [47](#)

sl.time, [41](#), [49](#), [58](#)
summary.cox, [52](#)
summary.flexsurv, [53](#)
summary.nn.time, [55](#)
summary.rf.time, [56](#)
summary.sl.time, [57](#)

tune.cox.aic, [58](#)
tune.cox.en, [60](#)
tune.cox.lasso, [61](#)
tune.cox.ridge, [63](#)
tune.nn.time, [64](#)
tune.rf.time, [65](#)