Package 'SurvivalClusteringTree'

February 25, 2026

Type Package

Title Clustering Analysis Using Survival Tree and Forest Algorithms

Version 1.1.3

Date 2026-02-23

Maintainer Lu You <lu.you@epi.usf.edu>

Description

An outcome-guided algorithm is developed to identify clusters of samples with similar characteristics and survival rate. The algorithm first builds a random forest and then defines distances between samples based on the fitted random forest. Given the distances, we can apply hierarchical clustering algorithms to define clusters. Details about this method is described in <https://github.com/luyouepiusf/SurvivalClusteringTree>.

License GPL (>= 2)

Suggests knitr, rmarkdown, tinytest, miceRanger

Encoding UTF-8

RoxygenNote 7.3.2

Imports Rcpp, survival, dplyr, grid, formula.tools

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

NeedsCompilation yes

Author Lu You [aut, cre] (Created the package. Maintains the package.),
Lauric Ferrat [aut] (Added functionality. Revised the package. Wrote the vignette.),
Hemang Parikh [aut] (Checked and revised the package.),
Yanan Huo [aut] (Revised plotting functions of the package.),
Yuting Yang [aut] (Added some data frame features.),
Jeffrey Krischer [ctb] (Supervisor the medical research. Coauthor of the medical manuscript.),
Maria Redondo [ctb] (Principal investigators of the medical research. Coauthor of the medical manuscript.),
Richard Oram [ctb] (Coauthor of the medical manuscript.),
Andrea Steck [ctb] (Coauthor of the medical manuscript.)

Repository CRAN

Date/Publication 2026-02-25 06:10:02 UTC

Contents

`SurvivalClusteringTree-package`
Clustering Analysis Using Survival Tree and Forest Algorithms

---

Description

An outcome-guided algorithm is developed to identify clusters of samples with similar characteristics and survival rate. The algorithm first builds a random forest and then defines distances between samples based on the fitted random forest. Given the distances, we can apply hierarchical clustering algorithms to define clusters. Details about this method is described in <https://github.com/luyouepiusf/SurvivalClusteringTre

Package Content

Index of help topics:

```
SurvivalClusteringTree-package
                        Clustering Analysis Using Survival Tree and
                        Forest Algorithms
plot_survival_tree      Visualize the Fitted Survival Tree
predict_distance_forest
                        Predict Distances Between Samples Based on a
                        Survival Forest Fit (Data Supplied as a
                        Dataframe)
predict_distance_forest_matrix
                        Predict Distances Between Samples Based on a
                        Survival Forest Fit (Data Supplied as Matrices)
predict_distance_tree   Predict Distances Between Samples Based on a
                        Survival Tree Fit (Data Supplied as a
                        Dataframe)
predict_distance_tree_matrix
                        Predict Distances Between Samples Based on a
```

	Survival Tree Fit (Data Supplied as Matrices)
predict_weights	Predict Weights of Samples in Terminal Nodes Based on a Survival Tree Fit (Data Supplied as a Dataframe)
predict_weights_matrix	Predict Weights of Samples in Terminal Nodes Based on a Survival Tree Fit (Data Supplied as Matrices)
survival_forest	Build a Survival Forest (Data Supplied as a Dataframe)
survival_forest_matrix	Build a Survival Forest (Data Supplied as Matrices)
survival_tree	Build a Survival Tree (Data Supplied as a Dataframe)
survival_tree_matrix	Build a Survival Tree (Data Supplied as Matrices)

Maintainer

Lu You <lu.you@epi.usf.edu>

Author(s)

NA

`plot_survival_tree` *Visualize the Fitted Survival Tree*

Description

Visualize the Fitted Survival Tree

Usage

```
plot_survival_tree(survival_tree, cex = 0.75)
```

Arguments

survival_tree	a fitted survival tree object.
cex	numeric character expansion factor.

Value

No return value, called for generating graphical outputs.

Examples

```
library(survival)
a_survival_tree<-
  survival_tree(
  survival_outcome=Surv(time,status==2)~1,
  numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
  factor_predictor=~as.factor(sex),
  data=lung)
plot_survival_tree(a_survival_tree)
```

`predict_distance_forest` *Predict Distances Between Samples Based on a Survival Forest Fit (Data Supplied as a Dataframe)*

Description

The function `predict_distance_forest` predicts distances between samples based on a survival forest fit.

Usage

```
predict_distance_forest(
  survival_forest,
  numeric_predictor,
  factor_predictor,
  data,
  missing = "omit"
)
```

Arguments

`survival_forest`	a fitted survival forest
`numeric_predictor`	a formula specifying the numeric predictors. As in `~x1+x2+x3`, the three numeric variables `x1`, `x2`, and `x3` are included as numeric predictors. `x1[i]`, `x2[i]`, and `x3[i]` are the predictors of the ith sample. The best practice is to use the same variables names in the training and testing dataset.
`factor_predictor`	a formula specifying the numeric predictors. As in `~z1+z2+z3`, the three character variables `z1`, `z2`, and `z3` are included as factor predictors. `z1[i]`, `z2[i]`, and `z3[i]` are the predictors of the ith sample. The best practice is to use the same variables names in the training and testing dataset.
`data`	the dataframe (test data) that stores the outcome and predictor variables. Variables in the global environment will be used if `data` is missing.

`missing` a character value that specifies the handling of missing data. If `missing=="omit"`, samples with missing values in the splitting variables will be discarded. If `missing=="majority"`, samples with missing values in the splitting variables will be assigned to the majority node. If `missing=="weighted"`, samples with missing values in the splitting variables will be weighted by the weights of branch nodes. The best practice is to use the same method as the trained random forest.

Details

Predict Distances Between Samples Based on a Survival Forest Fit (Data Supplied as a Dataframe)

Value

A list. `mean_distance` is the mean distance matrix. `sum_distance` is the matrix that sums the distances between samples. `sum_non_na` is the matrix of the number of non NA distances being averaged.

Examples

```
library(survival)
a_survival_forest<-
  survival_forest(
    survival_outcome=Surv(time,status==2)~1,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
    factor_predictor=~as.factor(sex),
    data=lung,nboot=20)
a_distance<-
  predict_distance_forest(
    a_survival_forest,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
    factor_predictor=~as.factor(sex),
    data=lung)
```

---

predict_distance_forest_matrix

Predict Distances Between Samples Based on a Survival Forest Fit (Data Supplied as Matrices)

Description

The function `predict_distance_forest_matrix` predicts distances between samples based on a survival forest fit.

Usage

```
predict_distance_forest_matrix(
  survival_forest,
  matrix_numeric,
  matrix_factor,
  missing = "omit"
)
```

Arguments

survival_forest	a fitted survival forest
matrix_numeric	numeric predictors, a numeric matrix. matrix_numeric[i,j] is the jth numeric predictor of the ith sample. The best practice is to have the same column names in the training and testing dataset.
matrix_factor	factor predictors, a character matrix. matrix_factor[i,j] is the jth predictor of the ith sample. The best practice is to have the same column names in the training and testing dataset.
missing	a character value that specifies the handling of missing data. If missing=="omit", samples with missing values in the splitting variables will be discarded. If missing=="majority", samples with missing values in the splitting variables will be assigned to the majority node. If missing=="weighted", samples with missing values in the splitting variables will be weighted by the weights of branch nodes. The best practice is to use the same method as the trained random forest.

Details

Predict Distances Between Samples Based on a Survival Forest Fit (Data Supplied as Matrices) (Works for raw matrices)

Value

A list. mean_distance is the mean distance matrix. sum_distance is the matrix that sums the distances between samples. sum_non_na is the matrix of the number of non NA distances being averaged.

Examples

```
library(survival)
a_survival_forest<-
  survival_forest_matrix(
    time=lung$time,
    event=lung$status==2,
    matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
    matrix_factor=data.matrix(lung[,5,drop=FALSE]),
    nboot=20)
a_distance<-
  predict_distance_forest_matrix(
```

```
  a_survival_forest,
  matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
  matrix_factor=data.matrix(lung[,5,drop=FALSE]))
```

`predict_distance_tree` *Predict Distances Between Samples Based on a Survival Tree Fit (Data Supplied as a Dataframe)*

Description

The function `predict_distance_tree` predicts distances between samples based on a survival tree fit.

Usage

```
predict_distance_tree(
  survival_tree,
  numeric_predictor,
  factor_predictor,
  data,
  missing = "omit"
)
```

Arguments

`survival_tree`	a fitted survival tree
`numeric_predictor`	a formula specifying the numeric predictors. As in `~x1+x2+x3`, the three numeric variables `x1`, `x2`, and `x3` are included as numeric predictors. `x1[i]`, `x2[i]`, and `x3[i]` are the predictors of the ith sample. The best practice is to use the same variables names in the training and testing dataset.
`factor_predictor`	a formula specifying the numeric predictors. As in `~z1+z2+z3`, the three character variables `z1`, `z2`, and `z3` are included as factor predictors. `z1[i]`, `z2[i]`, and `z3[i]` are the predictors of the ith sample. The best practice is to use the same variables names in the training and testing dataset.
`data`	the dataframe (test data) that stores the outcome and predictor variables. Variables in the global environment will be used if `data` is missing.
`missing`	a character value that specifies the handling of missing data. If `missing=="omit"`, samples with missing values in the splitting variables will be discarded. If `missing=="majority"`, samples with missing values in the splitting variables will be assigned to the majority node. If `missing=="weighted"`, samples with missing values in the splitting variables will be weighted by the weights of branch nodes. The best practice is to use the same method as the trained random tree.

Details

Predict Distances Between Samples Based on a Survival Tree Fit (Data Supplied as a Dataframe)

Value

A list. `node_distance` gives the distance matrix between nodes. `ind_distance` gives the distance matrix between samples. `ind_weights` gives the weights of samples in each node.

Examples

```
library(survival)
a_survival_tree<-
  survival_tree(
    survival_outcome=Surv(time,status==2)~1,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
    factor_predictor=~as.factor(sex),
    data=lung)
a_distance<-
  predict_distance_tree(
    a_survival_tree,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
    factor_predictor=~as.factor(sex),
    data=lung)
```

---

predict_distance_tree_matrix

Predict Distances Between Samples Based on a Survival Tree Fit (Data Supplied as Matrices)

Description

The function `predict_distance_tree_matrix` predicts distances between samples based on a survival tree fit.

Usage

```
predict_distance_tree_matrix(
  survival_tree,
  matrix_numeric,
  matrix_factor,
  missing = "omit"
)
```

Arguments

`survival_tree`	a fitted survival tree

matrix_numeric	numeric predictors, a numeric matrix. matrix_numeric[i,j] is the jth numeric predictor of the ith sample. The best practice is to have the same column names in the training and testing dataset.
matrix_factor	factor predictors, a character matrix. matrix_factor[i,j] is the jth predictor of the ith sample. The best practice is to have the same column names in the training and testing dataset.
missing	a character value that specifies the handling of missing data. If missing=="omit", samples with missing values in the splitting variables will be discarded. If missing=="majority", samples with missing values in the splitting variables will be assigned to the majority node. If missing=="weighted", samples with missing values in the splitting variables will be weighted by the weights of branch nodes. The best practice is to use the same method as the trained random tree.

Details

Predict Distances Between Samples Based on a Survival Tree Fit (Data Supplied as Matrices) (Works for raw matrices)

Value

A list. node_distance gives the distance matrix between nodes. ind_distance gives the distance matrix between samples. ind_weights gives the weights of samples in each node.

Examples

```
library(survival)
a_survival_tree<-
  survival_tree_matrix(
    time=lung$time,
    event=lung$status==2,
    matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
    matrix_factor=data.matrix(lung[,5,drop=FALSE]))
a_distance<-
  predict_distance_tree_matrix(
    a_survival_tree,
    matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
    matrix_factor=data.matrix(lung[,5,drop=FALSE]))
```

---

predict_weights *Predict Weights of Samples in Terminal Nodes Based on a Survival Tree Fit (Data Supplied as a Dataframe)*

Description

The function predict_weights predicts weights of samples in terminal nodes based on a survival tree fit.

Usage

```
predict_weights(
  survival_tree,
  numeric_predictor,
  factor_predictor,
  data,
  missing = "omit"
)
```

Arguments

survival_tree	a fitted survival tree
numeric_predictor	a formula specifying the numeric predictors. As in ~x1+x2+x3, the three numeric variables x1, x2, and x3 are included as numeric predictors. x1[i], x2[i], and x3[i] are the predictors of the ith sample. The best practice is to use the same variables names in the training and testing dataset.
factor_predictor	a formula specifying the numeric predictors. As in ~z1+z2+z3, the three character variables z1, z2, and z3 are included as factor predictors. z1[i], z2[i], and z3[i] are the predictors of the ith sample. The best practice is to use the same variables names in the training and testing dataset.
data	the dataframe (test data) that stores the outcome and predictor variables. Variables in the global environment will be used if data is missing.
missing	a character value that specifies the handling of missing data. If missing=="omit", samples with missing values in the splitting variables will be discarded. If missing=="majority", samples with missing values in the splitting variables will be assigned to the majority node. If missing=="weighted", samples with missing values in the splitting variables will be weighted by the weights of branch nodes. The best practice is to use the same method as the trained random tree.

Details

Predict Weights of Samples in Terminal Nodes Based on a Survival Tree Fit (Data Supplied as a Dataframe)

Value

A weight matrix representing the weights of samples in each node.

Examples

```
library(survival)
a_survival_tree<-
  survival_tree(
    survival_outcome=Surv(time,status==2)~1,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
```

```
    factor_predictor=~as.factor(sex),
    data=lung)
a_weight<-
  predict_weights(
    a_survival_tree,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
    factor_predictor=~as.factor(sex),
    data=lung)
```

`predict_weights_matrix` *Predict Weights of Samples in Terminal Nodes Based on a Survival Tree Fit (Data Supplied as Matrices)*

Description

The function `predict_weights_matrix` predicts weights of samples in terminal nodes based on a survival tree fit.

Usage

```
predict_weights_matrix(
  survival_tree,
  matrix_numeric,
  matrix_factor,
  missing = "majority"
)
```

Arguments

`survival_tree`	a fitted survival tree
`matrix_numeric`	numeric predictors, a numeric matrix. `matrix_numeric[i,j]` is the jth numeric predictor of the ith sample. The best practice is to have the same column names in the training and testing dataset.
`matrix_factor`	factor predictors, a character matrix. `matrix_factor[i,j]` is the jth predictor of the ith sample. The best practice is to have the same column names in the training and testing dataset.
`missing`	a character value that specifies the handling of missing data. If `missing=="omit"`, samples with missing values in the splitting variables will be discarded. If `missing=="majority"`, samples with missing values in the splitting variables will be assigned to the majority node. If `missing=="weighted"`, samples with missing values in the splitting variables will be weighted by the weights of branch nodes. The best practice is to use the same method as the trained tree.

Details

Predict Weights of Samples in Terminal Nodes Based on a Survival Tree Fit (Data Supplied as Matrices)

Value

A weight matrix representing the weights of samples in each node.

Examples

```
library(survival)
a_survival_tree<-
  survival_tree_matrix(
    time=lung$time,
    event=lung$status==2,
    matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
    matrix_factor=data.matrix(lung[,5,drop=FALSE]))
a_weight<-
  predict_weights_matrix(
    a_survival_tree,
    matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
    matrix_factor=data.matrix(lung[,5,drop=FALSE]))
```

---

survival_forest *Build a Survival Forest (Data Supplied as a Dataframe)*

Description

The function `survival_forest` build a survival forest given the survival outcomes and predictors of numeric and factor variables.

Usage

```
survival_forest(
  survival_outcome,
  numeric_predictor,
  factor_predictor,
  weights = NULL,
  data,
  significance = 0.05,
  min_weights = 50,
  missing = "omit",
  test_type = "univariate",
  cut_type = 0,
  nboot = 100,
  seed = 0,
  args_miceRanger = NULL
)
```

Arguments

`survival_outcome`
: a `Surv` object of right-censored outcomes. In `Surv(time,event)`, `time[i]` is the survival time of the ith sample. `event[i]` is the survival event of the ith sample.

`numeric_predictor`
: a formula specifying the numeric predictors. As in `~x1+x2+x3`, the three numeric variables `x1`, `x2`, and `x3` are included as numeric predictors. `x1[i]`, `x2[i]`, and `x3[i]` are the predictors of the ith sample.

`factor_predictor`
: a formula specifying the numeric predictors. As in `~z1+z2+z3`, the three character variables `z1`, `z2`, and `z3` are included as factor predictors. `z1[i]`, `z2[i]`, and `z3[i]` are the predictors of the ith sample.

`weights`
: sample weights, a numeric vector. `weights[i]` is the weight of the ith sample.

`data`
: the dataframe that stores the outcome and predictor variables. Variables in the global environment will be used if `data` is missing.

`significance`
: significance threshold, a numeric value. Stop the splitting algorithm when no splits give a p-value smaller than `significance`.

`min_weights`
: minimum weight threshold, a numeric value. The weights in a node are greater than `min_weights`.

`missing`
: a character value that specifies the handling of missing data. If `missing=="omit"`, samples with missing values in the splitting variables will be discarded. If `missing=="majority"`, samples with missing values in the splitting variables will be assigned to the majority node. If `missing=="weighted"`, samples with missing values in the splitting variables will be weighted by the weights of branch nodes. If `missing=="miceRanger"`, samples with missing values will be imputed using the package "miceRanger".

`test_type`
: a character value that specifies the type of statistical tests. If `test_type=="univariate"`, then it performs a log-rank test without p-value adjustments. If `test_type` is in `p.adjust.methods`, i.e., one of holm, hochberg, hommel, bonferroni, BH, BY, or fdr, then the p-values will be adjusted using the corresponding method.

`cut_type`
: an integer value that specifies how to cut between two numeric values. If `cut_type==0`, then cut at the ends. If `cut_type==1`, then cut from the middle. If `cut_type==2`, then cut randomly between the two values.

`nboot`
: an integer value that specifies the number of bootstrap replications.

`seed`
: an integer value that specifies the seed.

`args_miceRanger`
: a list specifying additional arguments to be used to impute missing data using `miceRanger()`. Only applies when `missing=="miceRanger"`.

Details

Build a Survival Forest (Data Supplied as a Dataframe)

Value

A list containing the information of the survival forest fit.

Examples

```
library(survival)
a_survival_forest<-
  survival_forest(
    survival_outcome=Surv(time,status==2)~1,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
    factor_predictor=~as.factor(sex),
    data=lung,nboot=20)
```

---

survival_forest_matrix

Build a Survival Forest (Data Supplied as Matrices)

Description

The function `survival_forest_matrix` build a survival forest given the survival outcomes and predictors of numeric and factor variables.

Usage

```
survival_forest_matrix(
  time,
  event,
  matrix_numeric,
  matrix_factor,
  weights = rep(1, length(time)),
  significance = 0.05,
  min_weights = 50,
  missing = "omit",
  test_type = "univariate",
  cut_type = 0,
  nboot = 100,
  seed = 0,
  args_miceRanger = NULL
)
```

Arguments

`time`	survival times, a numeric vector. `time[i]` is the survival time of the ith sample.
`event`	survival events, a logical vector. `event[i]` is the survival event of the ith sample.
`matrix_numeric`	numeric predictors, a numeric matrix. `matrix_numeric[i,j]` is the jth numeric predictor of the ith sample.

matrix_factor	factor predictors, a character matrix. matrix_factor[i,j] is the jth predictor of the ith sample.
weights	sample weights, a numeric vector. weights[i] is the weight of the ith sample.
significance	significance threshold, a numeric value. Stop the splitting algorithm when no splits give a p-value smaller than significance.
min_weights	minimum weight threshold, a numeric value. The weights in a node are greater than min_weights.
missing	a character value that specifies the handling of missing data. If missing=="omit", samples with missing values in the splitting variables will be discarded. If missing=="majority", samples with missing values in the splitting variables will be assigned to the majority node. If missing=="weighted", samples with missing values in the splitting variables will be weighted by the weights of branch nodes.
test_type	a character value that specifies the type of statistical tests. If test_type=="univariate", then it performs a log-rank test without p-value adjustments. If test_type is in p.adjust.methods, i.e., one of holm, hochberg, hommel, bonferroni, BH, BY, or fdr, then the p-values will be adjusted using the corresponding method.
cut_type	an integer value that specifies how to cut between two numeric values. If cut_type==0, then cut at the ends. If cut_type==1, then cut from the middle. If cut_type==2, then cut randomly between the two values.
nboot	an integer value that specifies the number of bootstrap replications.
seed	an integer value that specifies the seed.
args_miceRanger	a list specifying additional arguments to be used to impute missing data using miceRanger(). Only applies when missing=="miceRanger".

Details

Build a Survival Forest (Data Supplied as Matrices)

Value

A list containing the information of the survival forest fit.

Examples

```
library(survival)
a_survival_forest<-
  survival_forest_matrix(
    time=lung$time,
    event=lung$status==2,
    matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
    matrix_factor=data.matrix(lung[,5,drop=FALSE]),
    nboot=20)
```

survival_tree *Build a Survival Tree (Data Supplied as a Dataframe)*

Description

The function survival_tree build a survival tree given the survival outcomes and predictors of numeric and factor variables.

Usage

```
survival_tree(
  survival_outcome,
  numeric_predictor,
  factor_predictor,
  weights = NULL,
  data,
  significance = 0.05,
  min_weights = 50,
  missing = "omit",
  test_type = "univariate",
  cut_type = 0
)
```

Arguments

survival_outcome	a Surv object of right-censored outcomes. In Surv(time,event), time[i] is the survival time of the ith sample. event[i] is the survival event of the ith sample.
numeric_predictor	a formula specifying the numeric predictors. As in ~x1+x2+x3, the three numeric variables x1, x2, and x3 are included as numeric predictors. x1[i], x2[i], and x3[i] are the predictors of the ith sample.
factor_predictor	a formula specifying the numeric predictors. As in ~z1+z2+z3, the three character variables z1, z2, and z3 are included as factor predictors. z1[i], z2[i], and z3[i] are the predictors of the ith sample.
weights	sample weights, a numeric vector. weights[i] is the weight of the ith sample.
data	the dataframe that stores the outcome and predictor variables. Variables in the global environment will be used if data is missing.
significance	significance threshold, a numeric value. Stop the splitting algorithm when no splits give a p-value smaller than significance.
min_weights	minimum weight threshold, a numeric value. The weights in a node are greater than min_weights.

missing	a character value that specifies the handling of missing data. If missing=="omit", samples with missing values in the splitting variables will be discarded. If missing=="majority", samples with missing values in the splitting variables will be assigned to the majority node. If missing=="weighted", samples with missing values in the splitting variables will be weighted by the weights of branch nodes.
test_type	a character value that specifies the type of statistical tests. If test_type=="univariate", then it performs a log-rank test without p-value adjustments. If test_type is in p.adjust.methods, i.e., one of holm, hochberg, hommel, bonferroni, BH, BY, or fdr, then the p-values will be adjusted using the corresponding method.
cut_type	an integer value that specifies how to cut between two numeric values. If cut_type==0, then cut at the ends. If cut_type==1, then cut from the middle. If cut_type==2, then cut randomly between the two values.

Details

Build a Survival Tree (Data Supplied as a Dataframe)

Value

A list containing the information of the survival tree fit.

Examples

```
library(survival)
a_survival_tree<-
  survival_tree(
    survival_outcome=Surv(time,status==2)~1,
    numeric_predictor=~age+ph.ecog+ph.karno+pat.karno+meal.cal,
    factor_predictor=~as.factor(sex),
    data=lung)
```

---

survival_tree_matrix *Build a Survival Tree (Data Supplied as Matrices)*

Description

The function survival_tree_matrix build a survival tree given the survival outcomes and predictors of numeric and factor variables.

Usage

```
survival_tree_matrix(
  time,
  event,
  matrix_numeric,
  matrix_factor,
  weights = rep(1, length(time)),
```

```
  significance = 0.05,
  min_weights = 50,
  missing = "omit",
  test_type = "univariate",
  cut_type = 0
)
```

Arguments

time	survival times, a numeric vector. time[i] is the survival time of the ith sample.
event	survival events, a logical vector. event[i] is the survival event of the ith sample.
matrix_numeric	numeric predictors, a numeric matrix. matrix_numeric[i,j] is the jth numeric predictor of the ith sample.
matrix_factor	factor predictors, a character matrix. matrix_factor[i,j] is the jth predictor of the ith sample.
weights	sample weights, a numeric vector. weights[i] is the weight of the ith sample.
significance	significance threshold, a numeric value. Stop the splitting algorithm when no splits give a p-value smaller than significance.
min_weights	minimum weight threshold, a numeric value. The weights in a node are greater than min_weights.
missing	a character value that specifies the handling of missing data. If missing=="omit", samples with missing values in the splitting variables will be discarded. If missing=="majority", samples with missing values in the splitting variables will be assigned to the majority node. If missing=="weighted", samples with missing values in the splitting variables will be weighted by the weights of branch nodes.
test_type	a character value that specifies the type of statistical tests. If test_type=="univariate", then it performs a log-rank test without p-value adjustments. If test_type is in p.adjust.methods, i.e., one of holm, hochberg, hommel, bonferroni, BH, BY, or fdr, then the p-values will be adjusted using the corresponding method.
cut_type	an integer value that specifies how to cut between two numeric values. If cut_type==0, then cut at the ends. If cut_type==1, then cut from the middle. If cut_type==2, then cut randomly between the two values.

Details

Build a Survival Tree (Data Supplied as Matrices)

Value

A list containing the information of the survival tree fit.

Examples

```
library(survival)
a_survival_tree<-
  survival_tree_matrix(
```

```
        time=lung$time,
        event=lung$status==2,
        matrix_numeric=data.matrix(lung[,c(4,6:9),drop=FALSE]),
        matrix_factor=data.matrix(lung[,5,drop=FALSE]))
```

Index