

# Package: SurvivalClusteringTree (via r-universe)

September 6, 2024

**Type** Package

**Title** Clustering Analysis Using Survival Tree and Forest Algorithms

**Version** 1.0

**Date** 2023-07-10

**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.

**License** GPL (>= 2)

**Suggests** knitr, rmarkdown, tinytest

**Encoding** UTF-8

**RoxygenNote** 7.1.2

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

**LinkingTo** Rcpp, RcppArmadillo

**VignetteBuilder** knitr

**Repository** <https://luyouepiusf.r-universe.dev>

**RemoteUrl** <https://github.com/luyouepiusf/survivalclusteringtree>

**RemoteRef** HEAD

**RemoteSha** 621fb090ed78983d9193a70835bed10038462aeb

## Contents

SurvivalClusteringTree-package . . . . .	2
plot_survival_tree . . . . .	3
predict_distance_forest . . . . .	3
predict_distance_forest_matrix . . . . .	4

predict_distance_tree . . . . .	5
predict_distance_tree_matrix . . . . .	6
predict_weights . . . . .	7
predict_weights_matrix . . . . .	8
survival_forest . . . . .	9
survival_forest_matrix . . . . .	10
survival_tree . . . . .	11
survival_tree_matrix . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

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.

## Package Content

Index: This package was not yet installed at build time.

## Maintainer

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

## Author(s)

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.), 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.)

---

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.

---

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  $\sim x_1 + x_2 + x_3$ , the three numeric variables  $x_1$ ,  $x_2$ , and  $x_3$  are included as numeric predictors.  $x_1[i]$ ,  $x_2[i]$ , and  $x_3[i]$  are the predictors of the  $i$ th 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 $\sim z_1+z_2+z_3$ , the three character variables $z_1$ , $z_2$ , and $z_3$ are included as factor predictors. $z_1[i]$ , $z_2[i]$ , and $z_3[i]$ are the predictors of the $i$ th 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 <code>missing=="omit"</code> , samples with missing values in the splitting variables will be discarded. If <code>missing=="majority"</code> , samples with missing values in the splitting variables will be assigned to the majority node. If <code>missing=="weighted"</code> , 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)

---

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  $j$ th numeric predictor of the  $i$ th sample. The best practice is to have the same column names in the training and testing dataset.

matrix_factor	factor predictors, a character matrix. <code>matrix_factor[i, j]</code> is the <i>j</i> th predictor of the <i>i</i> th 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 <code>missing=="omit"</code> , samples with missing values in the splitting variables will be discarded. If <code>missing=="majority"</code> , samples with missing values in the splitting variables will be assigned to the majority node. If <code>missing=="weighted"</code> , 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)

---

`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 *i*th 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 *i*th sample. The best practice is to use the same variables names in the training and testing dataset.

<code>data</code>	the dataframe (test data) that stores the outcome and predictor variables. Variables in the global environment will be used if <code>data</code> is missing.
<code>missing</code>	a character value that specifies the handling of missing data. If <code>missing=="omit"</code> , samples with missing values in the splitting variables will be discarded. If <code>missing=="majority"</code> , samples with missing values in the splitting variables will be assigned to the majority node. If <code>missing=="weighted"</code> , 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)

---

`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

<code>survival_tree</code>	a fitted survival tree
<code>matrix_numeric</code>	numeric predictors, a numeric matrix. <code>matrix_numeric[i, j]</code> is the <i>j</i> th numeric predictor of the <i>i</i> th sample. The best practice is to have the same column names in the training and testing dataset.
<code>matrix_factor</code>	factor predictors, a character matrix. <code>matrix_factor[i, j]</code> is the <i>j</i> th predictor of the <i>i</i> th sample. The best practice is to have the same column names in the training and testing dataset.
<code>missing</code>	a character value that specifies the handling of missing data. If <code>missing=="omit"</code> , samples with missing values in the splitting variables will be discarded. If <code>missing=="majority"</code> , samples with missing values in the splitting variables will be assigned to the majority node. If <code>missing=="weighted"</code> , 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)

---

predict_weights	<i>Predict Weights of Samples in Terminal Nodes Based on a Survival Tree Fit (Data Supplied as a Dataframe)</i>
-----------------	---

---

## 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 $\sim x_1 + x_2 + x_3$ , the three numeric variables $x_1$ , $x_2$ , and $x_3$ are included as numeric predictors. $x_1[i]$ , $x_2[i]$ , and $x_3[i]$ are the predictors of the $i$ th 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 $\sim z_1 + z_2 + z_3$ , the three character variables $z_1$ , $z_2$ , and $z_3$ are included as factor predictors. $z_1[i]$ , $z_2[i]$ , and $z_3[i]$ are the predictors of the $i$ th 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 <code>missing=="omit"</code> , samples with missing values in the splitting variables will be discarded. If <code>missing=="majority"</code> , 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)

---

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

<code>survival_tree</code>	a fitted survival tree
<code>matrix_numeric</code>	numeric predictors, a numeric matrix. <code>matrix_numeric[i, j]</code> is the <code>j</code> th numeric predictor of the <code>i</code> th sample. The best practice is to have the same column names in the training and testing dataset.
<code>matrix_factor</code>	factor predictors, a character matrix. <code>matrix_factor[i, j]</code> is the <code>j</code> th predictor of the <code>i</code> th sample. The best practice is to have the same column names in the training and testing dataset.
<code>missing</code>	a character value that specifies the handling of missing data. If <code>missing=="omit"</code> , samples with missing values in the splitting variables will be discarded. If <code>missing=="majority"</code> , samples with missing values in the splitting variables will be assigned to the majority node. If <code>missing=="weighted"</code> , 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)

---

survival_forest	<i>Build a Survival Forest (Data Supplied as a Dataframe)</i>
-----------------	---

---

**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",
  nboot = 100,
  seed = 0
)
```

**Arguments**

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

`weights` sample weights, a numeric vector. `weights[i]` is the weight of the *i*th 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.
nboot	an integer value that specifies the number of bootstrap replications.
seed	an integer value that specifies the seed.

### Details

Build a Survival Forest (Data Supplied as a Dataframe)

---

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",
  nboot = 100,
  seed = 0
)
```

**Arguments**

time	survival times, a numeric vector. <code>time[i]</code> is the survival time of the <i>i</i> th sample.
event	survival events, a logical vector. <code>event[i]</code> is the survival event of the <i>i</i> th sample.
matrix_numeric	numeric predictors, a numeric matrix. <code>matrix_numeric[i, j]</code> is the <i>j</i> th numeric predictor of the <i>i</i> th sample.
matrix_factor	factor predictors, a character matrix. <code>matrix_factor[i, j]</code> is the <i>j</i> th predictor of the <i>i</i> th sample.
weights	sample weights, a numeric vector. <code>weights[i]</code> is the weight of the <i>i</i> th sample.
significance	significance threshold, a numeric value. Stop the splitting algorithm when no splits give a p-value smaller than <code>significance</code> .
min_weights	minimum weight threshold, a numeric value. The weights in a node are greater than <code>min_weights</code> .
missing	a character value that specifies the handling of missing data. If <code>missing=="omit"</code> , samples with missing values in the splitting variables will be discarded. If <code>missing=="majority"</code> , samples with missing values in the splitting variables will be assigned to the majority node. If <code>missing=="weighted"</code> , 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 <code>test_type=="univariate"</code> , then it performs a log-rank test without p-value adjustments. If <code>test_type</code> is in <code>p.adjust.methods</code> , i.e., one of <code>holm</code> , <code>hochberg</code> , <code>hommel</code> , <code>bonferroni</code> , <code>BH</code> , <code>BY</code> , or <code>fdr</code> , then the p-values will be adjusted using the corresponding method.
nboot	an integer value that specifies the number of bootstrap replications.
seed	an integer value that specifies the seed.

**Details**

Build a Survival Forest (Data Supplied as Matrices)

---

survival_tree	<i>Build a Survival Tree (Data Supplied as a Dataframe)</i>
---------------	---

---

**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"
)

```

## 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.

## Details

Build a Survival Tree (Data Supplied as a Dataframe)

---

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"
)
```

### Arguments

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

**Details**

Build a Survival Tree (Data Supplied as Matrices)

# Index

## \* package

SurvivalClusteringTree-package, [2](#)

plot\_survival\_tree, [3](#)

predict\_distance\_forest, [3](#)

predict\_distance\_forest\_matrix, [4](#)

predict\_distance\_tree, [5](#)

predict\_distance\_tree\_matrix, [6](#)

predict\_weights, [7](#)

predict\_weights\_matrix, [8](#)

survival\_forest, [9](#)

survival\_forest\_matrix, [10](#)

survival\_tree, [11](#)

survival\_tree\_matrix, [13](#)

SurvivalClusteringTree

(SurvivalClusteringTree-package),

[2](#)

SurvivalClusteringTree-package, [2](#)