

Package ‘survBootOutliers’

July 23, 2025

Title Concordance Based Bootstrap Methods for Outlier Detection in Survival Analysis

Maintainer Joao Pinto <joao.pinto@tecnico.ulisboa.pt>

Version 1.0

URL <https://github.com/jonydog/survBootOutliers>

Depends R (>= 3.4.0), survival, stats

Imports methods, utils

Suggests BiocParallel

Author Joao Pinto <joao.pinto@tecnico.ulisboa.pt>, Andre Verissimo <andre.verissimo@tecnico.ulisboa.pt>, Alexandra Carvalho <alexandra.carvalho@tecnico.ulisboa.pt>, Susana Vinga <susanavinga@tecnico.ulisboa.pt>

Description Three new methods to perform outlier detection in a survival context. In total there are six methods provided, the first three methods are traditional residual-based outlier detection methods, the second three are the concordance-based. Package developed during the work on the two following publications: Pinto J., Carvalho A. and Vinga S. (2015) <[doi:10.5220/0005225300750082](https://doi.org/10.5220/0005225300750082)>; Pinto J.D., Carvalho A.M., Vinga S. (2015) <[doi:10.1007/978-3-319-27926-8_22](https://doi.org/10.1007/978-3-319-27926-8_22)>.

License GPL-2

RoxygenNote 6.0.1

biocViews Survival

NeedsCompilation no

Repository CRAN

Date/Publication 2018-05-28 11:10:03 UTC

Contents

display.obs.histogram	2
get.whas100.dataset	3
survBootOutliers	3

Index	6
--------------	----------

`display.obs.histogram` *Auxiliar function that displays the concordance histogram associated with the observation.*

Description

Auxiliar function that displays the concordance histogram associated with the observation.

Usage

```
display.obs.histogram(histograms, type, obs.index)
```

Arguments

<code>histograms</code>	The histograms object returned by the <code>survBootOutliers</code> function when the method selected is "bht" or "dbht".
<code>type</code>	The type of histogram that is given as input, possible choices are again "bht" or "dbht".
<code>obs.index</code>	The original index of the observation of the concordance histograms to be displayed

Value

No value is returned

Examples

```
## Not run:
whas <- get.whas100.dataset()
outliers_bht <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "bht",
  B = 2000, B.N = 100 ,
  parallel.param = BiocParallel::MulticoreParam()
)
display.obs.histogram(outliers_bht$histograms, "bht", 67)

## End(Not run)
```

get.whas100.dataset	<i>Function that retrieves the WHAS100 dataset.</i>
---------------------	---

Description

This function retrieves the well known Worcester Heart Attack dataset with 100 individuals (WHAS100). This dataset is taken from the book by Hosmer D, Lemeshow S, May S. Applied Survival Analysis: Regression Modeling of Time to Event Data, 2nd edition. John Wiley and Sons Inc., New York, NY, 2008.

Usage

```
get.whas100.dataset()
```

Value

A data.frame containing the WHAS100 dataset

Examples

```
whas100_data <- get.whas100.dataset()
```

survBootOutliers	<i>Extract the most outlying observations following a criteria based on the bootstrapped concordance with parallel processing</i>
------------------	---

Description

Extract the most outlying observations following a criteria based on the bootstrapped concordance with parallel processing

Usage

```
survBootOutliers(surv.object, covariate.data, sod.method, B, B.N = NULL,
  max.outliers, parallel.param = NULL)
```

Arguments

surv.object	An object of type survival::Surv containing lifetimes and right-censoring status
covariate.data	A data frame containing the data with covariate values for each individual
sod.method	One of c("osd", "bht", "dbht", "ld", "martingale", "deviance")
B	The number of bootstrap samples generated only applicable for "bht" and "dbht" methods. Typically at least 10x the size of the dataset, ideally should be increased until convergence.
B.N	the number of observations in each bootstrap sample
max.outliers	This parameter is only used for the "osd" method
parallel.param	(Optional) A BiocParallel object, examples: SerialParam(), MulticoreParam()

Value

For all methods except for "bht" and "dbht" the value returned is a data.frame containing the most outlying observations sorted by outlying score. For the "bht" method the value returned is a list of two members: "outlier_set": the most outlying observations sorted by p-values; "histograms": histogram of concordance variation for each observation. For the "dbht" method the value returned is a list of two members: "outlier_set": the most outlying observations sorted by p-values; "histograms": histograms of concordance for each observations for the two types of bootstrap: "poison" and "antidote".

Examples

```
## One Step Deletion "osd" method
## Not run:
whas <- get.whas100.dataset()
print( getwd() )
outliers_osd <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "osd",
  max.outliers = 5
)

## End(Not run)

## Bootstrap Hypothesis Test "bht" with 1000 bootstrap samples,
## each with 100 individuals, running without parallelism.
## Not run:
whas <- get.whas100.dataset()
outliers_bht <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "bht",
  B = 1000,
  B.N = 100,
  parallel.param = BiocParallel::MulticoreParam()
)

## End(Not run)

## Dual Bootstrap Hypothesis Test "dbht" with 1000 bootstrap samples,
## each with 50 individuals and running on all available cores.
## Not run: whas <- get.whas100.dataset()
outliers_dbht <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "dbht",
  B = 1000,
  B.N = 50,
  parallel.param = BiocParallel::MulticoreParam()
)
```

```
## End(Not run)
## One Step Deletion "osd" with an amount of 10 for maximum outlier count
whas <- get.whas100.dataset()
outliers_osd <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "osd",
  max.outliers = 10
)

## Likelihood displacement criterion for outlier ranking
whas <- get.whas100.dataset()
outliers_ld <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "ld"
)

## Cox regression deviance residuals criterion for outlier ranking
whas <- get.whas100.dataset()
outliers_deviance <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "deviance"
)

## Cox regression Martingale residuals criterion for outlier ranking
whas <- get.whas100.dataset()
outliers_martingale <- survBootOutliers(
  surv.object=Surv(time = whas$times,event = whas$status ),
  covariate.data = whas[,2:5],
  sod.method = "martingale"
)
```

Index

`display.obs.histogram`, [2](#)

`get.whas100.dataset`, [3](#)

`survBootOutliers`, [3](#)