

# Package ‘dsSurvival’

May 11, 2021

**Title** DataSHIELD server site base functions for survival functions

**Description** DataSHIELD server site base functions fro survival functions.

**Version** 6.2.0-1

**Author** Soumya Banerjee, Tom Bishop, Demetris Avraam, Paul Burton and DataSHIELD technical team <datashield@newcastle.ac.uk>

**Maintainer** Soumya Banerjee, Tom Bishop, Demetris Avraam, Paul Burton and DataSHIELD technical team <datashield@newcastle.ac.uk>

**License** GPL-3

**Depends** R (≥ 3.5.0)

**Imports** RANN,

stringr,  
survival,  
ggplot2,  
dplyr,  
reshape2,  
dsBase

**AggregateMethods** coxphSLMADS,

coxphSummaryDS,  
cox.zphSLMADS,  
summarySurvDS,  
plotsurvfitDS

**AssignMethods** coxphSLMAassignDS,

SurvDS,  
survfitDS

**Options** datashield.privacyLevel=5,

default.nfilter.glm=0.33,  
default.nfilter.kNN=3,  
default.nfilter.string=80,  
default.nfilter.subset=3,  
default.nfilter.stringShort=20,  
default.nfilter.tab=3,  
default.nfilter.noise=0.25,  
default.nfilter.levels=0.33

**RoxygenNote** 7.1.1

## R topics documented:

cox.zphSLMADS	2
coxphSLMAassignDS	3
coxphSLMADS	4
coxphSummaryDS	5
listDisclosureSettingsDS	6
plotsurvfitDS	7
summarySurvDS	7
SurvDS	8
survfitDS	9

---

cox.zphSLMADS	<i>Tests the proportional hazards assumption of a Cox proportional hazards model that has been fit and saved serverside.</i>
---------------	--

---

### Description

Tests the proportional hazards assumption of a Cox proportional hazards that has been fit and saved on the server side environment.

### Usage

```
cox.zphSLMADS(
  fit = NULL,
  transform = "km",
  terms = TRUE,
  singledf = FALSE,
  global = TRUE
)
```

### Arguments

<b>fit</b>	character string specifying name of fit Cox proportional hazards model saved in the server-side.
<b>transform</b>	character string specifying how the survival times should be transformed before the test is performed. Possible values are "km", "rank", "identity" or a function of one argument.
<b>terms</b>	logical if TRUE, do a test for each term in the model rather than for each separate covariate. For a factor variable with k levels, for instance, this would lead to a k-1 degree of freedom test. The plot for such variables will be a single curve evaluating the linear predictor over time.
<b>singledf</b>	logical use a single degree of freedom test for terms that have multiple coefficients, i.e., the test that corresponds most closely to the plot. If terms=FALSE this argument has no effect.
<b>global</b>	logical should a global chi-square test be done, in addition to the per-variable or per-term tests tests.

## Details

Serverside aggregate function `cox.zphSLMADS` called by clientside function. `ds.cox.zphSLMA`. returns diagnostics for the test of proportional hazards assumptions from a Cox proportional hazards model. This request is not disclosive as it only returns summary statistics. For further details see help for `ds.cox.zphSLMA` function.

## Value

diagnostics for the Cox proportional hazards from the server side environment.

## Author(s)

Soumya Banerjee and Tom Bishop (2020).

---

<code>coxphSLMAassignDS</code>	<i>Performs survival analysis using the Cox proportional hazards model at the serverside environment.</i>
--------------------------------	---

---

## Description

Performs survival analysis using the Cox proportional hazards models and stores the model on the server side environment.

## Usage

```
coxphSLMAassignDS(
  formula = NULL,
  dataName = NULL,
  weights = NULL,
  init = NULL,
  ties = "efron",
  singular.ok = TRUE,
  model = FALSE,
  x = FALSE,
  y = TRUE,
  control = NULL
)
```

## Arguments

<code>formula</code>	either NULL or a character string (potentially including '*' wildcards) specifying a formula.
<code>dataName</code>	character string of name of data frame
<code>weights</code>	vector of case weights
<code>init</code>	vector of initial values of the iteration
<code>ties</code>	character string specifying the method for tie handling. The Efron approximation is used as the default. Other options are 'breslow' and 'exact'.

<code>singular.ok</code>	Logical value indicating how to handle collinearity in the model matrix. Default is TRUE. If TRUE, the program will automatically skip over columns of the X matrix that are linear combinations of earlier columns. In this case the coefficients of such columns will be NA and the variance matrix will contain zeros.
<code>model</code>	logical value. If TRUE, the model frame is returned in component <code>model</code> .
<code>x</code>	logical value. If TRUE, the x matrix is returned in component <code>x</code> .
<code>y</code>	logical value. If TRUE, the response vector is returned in component <code>y</code> .
<code>control</code>	object of type <code>survival::coxph.control()</code> specifying iteration limit and other control options. Default is <code>survival::coxph.control()</code>

### Details

Serverside assign function `coxphSLMAassignDS` called by clientside function. `ds.coxphSLMAassign` stores the Cox proportional hazards in the server side environment This request is not disclosure as it only returns a string. For further details see help for `ds.coxphSLMAassign` function.

### Value

the Cox proportional hazards from the server side environment from the server side environment.

### Author(s)

Soumya Banerjee and Tom Bishop (2020).

---

<code>coxphSLMADS</code>	<i>Performs survival analysis using the Cox proportional hazards model at the serverside environment.</i>
--------------------------	---

---

### Description

returns a summary of the Cox proportional hazards from the server side environment.

### Usage

```
coxphSLMADS(
  formula = NULL,
  dataName = NULL,
  weights = NULL,
  init = NULL,
  ties = "efron",
  singular.ok = TRUE,
  model = FALSE,
  x = FALSE,
  y = TRUE,
  control = NULL
)
```

**Arguments**

<code>formula</code>	either NULL or a character string (potentially including '*' wildcards) specifying a formula.
<code>dataName</code>	character string of name of data frame
<code>weights</code>	vector of case weights
<code>init</code>	vector of initial values of the iteration
<code>ties</code>	character string specifying the method for tie handling. The Efron approximation is used as the default. Other options are 'breslow' and 'exact'.
<code>singular.ok</code>	Logical value indicating how to handle collinearity in the model matrix. Default is TRUE. If TRUE, the program will automatically skip over columns of the X matrix that are linear combinations of earlier columns. In this case the coefficients of such columns will be NA and the variance matrix will contain zeros.
<code>model</code>	logical value. If TRUE, the model frame is returned in component model.
<code>x</code>	logical value. If TRUE, the x matrix is returned in component x.
<code>y</code>	logical value. If TRUE, the response vector is returned in component y.
<code>control</code>	object of type <code>survival::coxph.control()</code> specifying iteration limit and other control options. Default is <code>survival::coxph.control()</code>

**Details**

Serverside aggregate function `coxphSLMADS` called by clientside function. `ds.coxphSLMA` returns a summary of the Cox proportional hazards from the server side environment from the server side environment. This request is not disclosive as it only returns a string. For further details see help for `ds.coxphSLMA` function.

**Value**

a summary of the Cox proportional hazards from the server side environment from the server side environment.

**Author(s)**

Soumya Banerjee and Tom Bishop (2020).

---

<code>coxphSummaryDS</code>	<i>Returns the summary of a Cox proportional hazards model that has been fit and saved serverside.</i>
-----------------------------	--

---

**Description**

This function returns the summary of a Cox proportional hazards that has been fit and saved on the server side environment.

**Usage**

```
coxphSummaryDS(x = NULL)
```

**Arguments**

**x** character string specifying name of fit Cox proportional hazards model saved in the server-side.

**Details**

Serverside aggregate function `coxphSummaryDS` called by clientside function. `ds.coxphSummary` returns the summary from a Cox proportional hazards model. This request is not disclosive as it only returns summary statistics. For further details see help for `ds.coxphSummary` function.

**Value**

summary of the Cox proportional hazards from the server side environment.

**Author(s)**

Soumya Banerjee and Tom Bishop (2020).

---

```
listDisclosureSettingsDS
      listDisclosureSettingsDS
```

---

**Description**

This serverside function is an aggregate function that is called by the `ds.listDisclosureSettings`

**Usage**

```
listDisclosureSettingsDS()
```

**Details**

For more details see the extensive header for `ds.listDisclosureSettings`

**Author(s)**

Paul Burton, Demetris Avraam for DataSHIELD Development Team

---

plotsurvfitDS	<i>Performs plotting of survival analysis curves.</i>
---------------	---

---

**Description**

returns a privacy preserving survival curve.

**Usage**

```
plotsurvfitDS(formula = NULL, dataName = NULL)
```

**Arguments**

formula	a character string which has the name of server-side survfit() object. This should be created using ds.survfit()
dataName	character string of name of data frame

**Details**

Serverside aggregate function plotsurvfitDS called by clientside function. ds.plotsurvfit. returns a privacy preserving survival curve from the server side environment. This request is not disclosive as it is randomized. For further details see help for ds.coxphSLMA function.

**Value**

a privacy preserving survival curve from the server side environment.

**Author(s)**

Soumya Banerjee, Tom Bishop, Demetris Avraam, Paul Burton and DataSHIELD technical team (2021).

---

summarySurvDS	<i>Returns summary of survival object.</i>
---------------	--

---

**Description**

returns a summary of the survival Surv() object from the server side environment.

**Usage**

```
summarySurvDS(object = NULL)
```

**Arguments**

object	name of server-side survival object.
--------	--------------------------------------

## Details

Serverside aggregate function `coxphSLMADS` called by clientside function `ds.summary`. returns a list which is summary of the survival `Surv()` object. The list has the summary of the time and event parameter in the survival object. This request is not disclosive. For further details see help for `ds.summary` function.

## Value

a list which is a summary of server-side survival model.

## Author(s)

Soumya Banerjee and Tom Bishop (2020).

---

SurvDS	<i>Creates a survival object for survival analysis using the Cox proportional hazards model at the serverside environment</i>
--------	---

---

## Description

returns a summary of the Cox proportional hazards from the server side environment.

## Usage

```
SurvDS(time = NULL, time2 = NULL, event = NULL, type = NULL, origin = NULL)
```

## Arguments

<code>time</code>	name of start time or follow-up time parameter to be passed to <code>Surv()</code> . Should be a character string.
<code>time2</code>	name of stop time parameter to be passed to <code>Surv()</code> . Should be a character string.
<code>event</code>	name of event parameter to be passed to <code>Surv()</code> Should be character string.
<code>type</code>	character string specifying the type of censoring. Possible values are "right", "left", "counting", "interval", "interval2", or "mstate"
<code>origin</code>	numeric, used for counting process data and is the hazard function origin. The origin parameter is used with time-dependent strata in order to align the subjects properly when they cross over from one strata to another. This parameter has rarely proven useful.

## Details

Serverside assign function `SurvDS` called by clientside function. `ds.Surv`. returns a Survival object for use in Cox proportional hazards from the server side environment from the server side environment. This request is not disclosive as it only returns a string. For further details see help for `ds.Surv` function.



**Value**

a survival::Surv() object from the server side environment.

**Author(s)**

Soumya Banerjee and Tom Bishop (2020).

---

survfitDS

*Creates a survival survfit object for survival analysis at the serverside environment. This is to be used for eventually plotting survival models. A survival curve is based on a tabulation of the number at risk and number of events at each unique death time.*

---

**Description**

creates a survfit survival object in the server side environment.

**Usage**

```
survfitDS(formula = NULL)
```

**Arguments**

formula            this is the formula to be passed to survfit(). Should be a character string.

**Details**

Serverside assign function survfitDS called by clientside function. ds.survfit. creates a survfit survival object in the server side environment This request is not disclosive. For further details see help for ds.survfit function.

**Value**

creates a survfit survival object in the server side environment.

**Author(s)**

Soumya Banerjee and Tom Bishop (2020).