

# Package ‘EquiSurv’

October 12, 2022

**Type** Package

**Title** Modeling, Confidence Intervals and Equivalence of Survival Curves

**Version** 0.1.0

**Author** Kathrin Moellenhoff

**Maintainer** Kathrin Moellenhoff <kathrin.moellenhoff@rub.de>

## Description

We provide a non-parametric and a parametric approach to investigate the equivalence (or non-inferiority) of two survival curves, obtained from two given datasets. The test is based on the creation of confidence intervals at pre-specified time points.

For the non-parametric approach, the curves are given by Kaplan-Meier curves and the variance for calculating the confidence intervals is obtained by Greenwood's formula.

The parametric approach is based on estimating the underlying distribution, where the user can choose between a Weibull, Exponential, Gaussian, Logistic, Log-normal or a Log-logistic distribution. Estimates for the variance for calculating the confidence bands are obtained by a (parametric) bootstrap approach. For this bootstrap censoring is assumed to be exponentially distributed and estimates are obtained from the datasets under consideration.

All details can be found in K.Moellenhoff and A.Tresch: Survival analysis under non-proportional hazards: investigating non-inferiority or equivalence in time-to-event data <[arXiv:2009.06699](https://arxiv.org/abs/2009.06699)>.

**Depends** survival, eha, graphics

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-09-23 13:30:02 UTC

## R topics documented:

boot_exponential . . . . .	2
boot_gaussian . . . . .	3
boot_logistic . . . . .	3
boot_loglogistic . . . . .	4
boot_lognormal . . . . .	5
boot_weibull . . . . .	6
confint_diff . . . . .	6
confint_km_diff . . . . .	7
test_diff . . . . .	8
test_nonpar . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

boot_exponential	<i>Parametric Bootstrap of time-to-event data following an exponential distribution</i>
------------------	---

---

### Description

Function generating bootstrap data according to an exponential distribution (specified by a model parameter  $\theta$ ), assuming exponentially distributed right-censoring (specified by a rate  $C$ ). After data generation again a model is fitted and evaluated at a pre-specified time point  $t_0$  yielding the response vector.

### Usage

```
boot_exponential(t0, B = 1000, theta, C, N)
```

### Arguments

t0	time point of interest
B	number of bootstrap repetitions. The default is B=1000
theta	parameter of the exponential distribution, theta=rate
C	rate of the exponential distribution specifying the censoring
N	size of the dataset = number of observations

### Value

A vector of length B containing the estimated survival at t0

### Examples

```
t0<-2
N<-30
C<-1
boot_exponential(t0=t0,theta=1,C=C,N=N)
```

---

boot_gaussian	<i>Parametric Bootstrap of time-to-event data following a gaussian distribution</i>
---------------	---

---

### Description

Function generating bootstrap data according to a gaussian distribution (specified by a model parameter  $\theta$ ), assuming exponentially distributed right-censoring (specified by a rate C). After data generation again a model is fitted and evaluated at a pre-specified time point  $t_0$  yielding the response vector.

### Usage

```
boot_gaussian(t0, B = 1000, theta, C, N)
```

### Arguments

t0	time point of interest
B	number of bootstrap repetitions. The default is B=1000
theta	parameter of the gaussian distribution, theta=(mean,sd)
C	rate of the exponential distribution specifying the censoring
N	size of the dataset = number of observations

### Value

A vector of length B containing the estimated survival at t0

### Examples

```
t0<-2
N<-30
C<-1
boot_gaussian(t0=t0,theta=c(1.7,1),C=C,N=N)
```

---

boot_logistic	<i>Parametric Bootstrap of time-to-event data following a logistic distribution</i>
---------------	---

---

### Description

Function generating bootstrap data according to a logistic distribution (specified by a model parameter  $\theta$ ), assuming exponentially distributed right-censoring (specified by a rate C). After data generation again a model is fitted and evaluated at a pre-specified time point  $t_0$  yielding the response vector.

**Usage**

```
boot_logistic(t0, B = 1000, theta, C, N)
```

**Arguments**

t0	time point of interest
B	number of bootstrap repetitions. The default is B=1000
theta	parameter of the logistic distribution, theta=(location,scale)
C	rate of the exponential distribution specifying the censoring
N	size of the dataset = number of observations

**Value**

A vector of length B containing the estimated survival at t0

**Examples**

```
t0<-2
N<-30
C<-1
boot_logistic(t0=t0, theta=c(1,0.4), C=C, N=N)
```

---

boot_loglogistic	<i>Parametric Bootstrap of time-to-event data following a loglogistic distribution</i>
------------------	--

---

**Description**

Function generating bootstrap data according to a loglogistic distribution (specified by a model parameter  $\theta$ ), assuming exponentially distributed right-censoring (specified by a rate C). After data generation again a model is fitted and evaluated at a pre-specified time point  $t_0$  yielding the response vector.

**Usage**

```
boot_loglogistic(t0, B = 1000, theta, C, N)
```

**Arguments**

t0	time point of interest
B	number of bootstrap repetitions. The default is B=1000
theta	parameter of the loglogistic distribution, theta=(shape,scale)
C	rate of the exponential distribution specifying the censoring
N	size of the dataset = number of observations

**Value**

A vector of length B containing the estimated survival at t0

**Examples**

```
alpha<-0.05
t0<-2
N<-30
C<-1
boot_loglogistic(t0=t0,theta=c(1,3),C=C,N=N)
```

---

boot_lognormal	<i>Parametric Bootstrap of time-to-event data following a lognormal distribution</i>
----------------	--

---

**Description**

Function generating bootstrap data according to a lognormal distribution (specified by a model parameter  $\theta$ ), assuming exponentially distributed right-censoring (specified by a rate C). After data generation again a model is fitted and evaluated at a pre-specified time point  $t_0$  yielding the response vector.

**Usage**

```
boot_lognormal(t0, B = 1000, theta, C, N)
```

**Arguments**

t0	time point of interest
B	number of bootstrap repetitions. The default is B=1000
theta	parameter of the lognormal distribution, theta=(meanlog,sdlog)
C	rate of the exponential distribution specifying the censoring
N	size of the dataset = number of observations

**Value**

A vector of length B containing the estimated survival at t0

**Examples**

```
t0<-2
N<-30
C<-1
boot_lognormal(t0=t0,theta=c(0.6,1),C=C,N=N)
```

---

boot_weibull	<i>Parametric Bootstrap of time-to-event data following a Weibull distribution</i>
--------------	--

---

### Description

Function generating bootstrap data according to a Weibull distribution (specified by a model parameter  $\theta$ ), assuming exponentially distributed right-censoring (specified by a rate  $C$ ). After data generation again a model is fitted and evaluated at a pre-specified time point  $t_0$  yielding the response vector.

### Usage

```
boot_weibull(t0, B = 1000, theta, C, N)
```

### Arguments

t0	time point of interest
B	number of bootstrap repetitions. The default is B=1000
theta	parameter of the Weibull distribution, theta=(shape,scale)
C	rate of the exponential distribution specifying the censoring
N	size of the dataset = number of observations

### Value

A vector of length B containing the estimated survival at t0

### Examples

```
t0<-2
N<-30
C<-1
boot_weibull(t0=t0,theta=c(1,3),C=C,N=N)
```

---

confint_diff	<i>Lower and upper confidence bounds for the difference of two parametric survival curves</i>
--------------	---

---

### Description

Function fitting parametric survival curves  $S_1, S_2$  to two groups and yielding lower and upper  $(1-\alpha)$ -confidence bounds for the difference  $S_1 - S_2$  of these two curves at a specific time point, based on approximating the variance via bootstrap. For the bootstrap exponentially distributed random censoring is assumed and the parameters estimated from the datasets.  $m_1$  and  $m_2$  are parametric survival models following a Weibull, exponential, gaussian, logistic, log-normal or log-logistic distribution. For the generation of the bootstrap data exponentially distributed right-censoring is assumed and the rates estimated from the datasets. See Moellenhoff and Tresch <arXiv:2009.06699> for details.

**Usage**

```
confint_diff(alpha, t0, m1, m2, B = 1000, data_r, data_t, plot = TRUE)
```

**Arguments**

alpha	confidence level
t0	time point of interest
m1, m2	type of parametric model. Possible model types are "weibull", "exponential", "gaussian", "logistic", "lognormal" and "loglogistic"
B	number of bootstrap repetitions. The default is B=1000
data_r, data_t	datasets containing time and status for each individual (have to be referenced as this)
plot	if TRUE, a plot of the two survival curves will be given

**Value**

A list containing the difference  $S_1(t_0) - S_2(t_0)$ , the lower and upper  $(1-\alpha)$ -confidence bounds and a summary of the two model fits. Further a plot of the curves is given.

**References**

K.Moellenhoff and A.Tresch: Survival analysis under non-proportional hazards: investigating non-inferiority or equivalence in time-to-event data <arXiv:2009.06699>

**Examples**

```
data(veteran)
veteran_r <- veteran[veteran$strtr==1,]
veteran_t <- veteran[veteran$strtr==2,]
alpha<-0.05
t0<-80
confint_diff(alpha=alpha,t0=t0,m1="weibull",m2="weibull",data_r=veteran_r,data_t=veteran_t)
```

---

confint_km_diff	<i>Lower and upper confidence bounds for the difference of two Kaplan-Meier curves</i>
-----------------	--

---

**Description**

Function fitting Kaplan-Meier curves  $S_1, S_2$  to two groups and yielding lower and upper  $(1-\alpha)$ -confidence bounds for the difference  $S_1 - S_2$  of these two curves at a specific time point by using Greenwood's formula.

**Usage**

```
confint_km_diff(alpha, t0, data_r, data_t, plot = TRUE)
```

**Arguments**

alpha            confidence level  
 t0                time point of interest  
 data\_r, data\_t   datasets containing time and status for each individual  
 plot             if TRUE, a plot of the two Kaplan Meier curves will be given

**Value**

A list containing the difference  $S_1(t_0) - S_2(t_0)$  and the lower and upper  $(1-\alpha)$ -confidence bounds. Further a plot of the curves is given.

**Examples**

```
data(veteran)
veteran_r <- veteran[veteran$strtr==1,]
veteran_t <- veteran[veteran$strtr==2,]
alpha<-0.05
t0<-80
confint_km_diff(alpha=alpha,t0=t0,data_r=veteran_r,data_t=veteran_t)
```

---

test_diff	<i>Non-inferiority and equivalence test for the difference of two parametric survival curves</i>
-----------	--

---

**Description**

Function for fitting and testing two parametric survival curves  $S_1, S_2$  at  $t_0$  concerning the hypotheses of non-inferiority

$$H_0 : S_1(t_0) - S_2(t_0) \geq \epsilon \text{ vs. } H_1 : S_1(t_0) - S_2(t_0) < \epsilon$$

or equivalence

$$H_0 : |S_1(t_0) - S_2(t_0)| \geq \epsilon \text{ vs. } H_1 : |S_1(t_0) - S_2(t_0)| < \epsilon.$$

$m_1$  and  $m_2$  are parametric survival models following a Weibull, exponential, gaussian, logistic, log-normal or log-logistic distribution. The test procedure is based on confidence intervals obtained via bootstrap. For the generation of the bootstrap data exponentially distributed random censoring is assumed and the rates estimated from the datasets. See Moellenhoff and Tresch <arXiv:2009.06699> for details.

**Usage**

```
test_diff(
  epsilon,
  alpha,
  t0,
  type,
  m1,
```



```

    m2,
    B = 1000,
    plot = TRUE,
    data_r,
    data_t
  )

```

### Arguments

epsilon	non-inferiority/equivalence margin
alpha	significance level
t0	time point of interest
type	type of the test. "ni" for non-inferiority, "eq" for equivalence test
m1, m2	type of parametric model. Possible model types are "weibull", "exponential", "gaussian", "logistic", "lognormal" and "loglogistic"
B	number of bootstrap repetitions. The default is B=1000
plot	if TRUE, a plot of the two survival curves will be given
data_r, data_t	datasets containing time and status for each individual (have to be referenced as this)

### Value

A list containing the difference  $S_1(t_0) - S_2(t_0)$ , the lower and upper  $(1-\alpha)$ -confidence bounds, the summary of the two model fits, the chosen margin and significance level and the test decision. Further a plot of the curves is given.

### References

K.Moellenhoff and A.Tresch: Survival analysis under non-proportional hazards: investigating non-inferiority or equivalence in time-to-event data <arXiv:2009.06699>

### Examples

```

data(veteran)
veteran_r <- veteran[veteran$strtr==1,]
veteran_t <- veteran[veteran$strtr==2,]
alpha<-0.05
t0<-80
epsilon<-0.15
test_diff(epsilon=epsilon,alpha=alpha,t0=t0,type="eq",m1="weibull",m2="weibull",
data_r=veteran_r,data_t=veteran_t)

```

---

test_nonpar	<i>Non-inferiority and equivalence test for the difference of two Kaplan-Meier curves</i>
-------------	---

---

**Description**

Function for fitting and testing two Kaplan Meier curves  $S_1, S_2$  at  $t_0$  concerning the hypotheses of non-inferiority

$$H_0 : S_1(t_0) - S_2(t_0) \geq \epsilon \text{ vs. } H_1 : S_1(t_0) - S_2(t_0) < \epsilon$$

or equivalence

$$H_0 : |S_1(t_0) - S_2(t_0)| \geq \epsilon \text{ vs. } H_1 : |S_1(t_0) - S_2(t_0)| < \epsilon.$$

**Usage**

```
test_nonpar(epsilon, alpha, t0, type, data_r, data_t, plot = TRUE)
```

**Arguments**

epsilon	non-inferiority/equivalence margin
alpha	significance level
t0	time point of interest
type	type of the test. "ni" for non-inferiority, "eq" for equivalence test
data_r, data_t	datasets containing time and status for each individual
plot	if TRUE, a plot of the two Kaplan Meier curves will be given

**Value**

A list containing the difference  $S_1(t_0) - S_2(t_0)$ , the lower and upper  $(1-\alpha)$ -confidence bounds, the chosen margin and significance level and the test decision. Further a plot of the curves is given.

**Examples**

```
data(veteran)
veteran_r <- veteran[veteran$strtr==1,]
veteran_t <- veteran[veteran$strtr==2,]
alpha<-0.05
t0<-80
epsilon<-0.15
test_nonpar(epsilon=epsilon,alpha=alpha,t0=t0,type="eq",data_r=veteran_r,data_t=veteran_t)
```

# Index

boot\_exponential, 2  
boot\_gaussian, 3  
boot\_logistic, 3  
boot\_loglogistic, 4  
boot\_lognormal, 5  
boot\_weibull, 6  
  
confint\_diff, 6  
confint\_km\_diff, 7  
  
test\_diff, 8  
test\_nonpar, 10