# Package ‘CPsurv’

July 21, 2025

**Type** Package

**Title** Nonparametric Change Point Estimation for Survival Data

**Version** 1.0.0

**Date** 2017-03-03

**Author** Stefanie Krügel [aut, cre],
Alessandra R. Brazzale, [aut],
Helmut Kuechenhoff [aut]

**Maintainer** Stefanie Krügel <`stefanie.kruegel@gmail.com`>

**Description** Nonparametric change point estimation for survival data based on p-values of exact binomial tests.

**License** GPL-2

**LazyData** TRUE

**Encoding** UTF-8

**Depends** R (>= 3.1.0)

**Imports** survival (>= 2.40-1), muhaz (>= 1.2.6), parallel (>= 3.2.3)

**Suggests** testthat (>= 0.11.0)

**RoxygenNote** 6.0.1

**Collate** 'CPsurv-package.R' 'bootbiascorrect.R' 'cpest.R'
'cpsurv-data.R' 'cpsurv.R' 'km.sim.survtimes.R' 'methods.R'
'neg.loglik.WeibExp.R' 'sim.survdata.R'

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-03-04 08:21:36

## Contents

---

## `bootbiascorrect` *Implements Bootstrap Bias Correction*

### Description

Implements Bootstrap Bias Correction

### Usage

```
bootbiascorrect(changeP, time, event, censoring, censpoint, intwd, cpmax, cpmin,
  norm.riskset, B.correct, parametric, times.int, opt.start)
```

### Arguments

| | |
|---|---|
| `changeP` | Estimated change point. |
| `time` | Numeric vector with survival times. |
| `event` | Numeric vector indicating censoring status; 0 = alive (censored), 1 = dead (uncensored). If missing, all observations are assumed to be uncensored. |
| `censoring` | Type of right-censoring for simulated data on which the bootstrap bias correction is based. Possible types are "random" for *random censoring* (default), "type1" for *Type I censoring* or "no" for data without censored observations. Because simulated data should be similar to given data, the censoring type is adapted from vector 'events' if given and argument 'censoring' is ignored than. |
| `censpoint` | Point of *Type I censoring*; if missing, minimum time after which all events are equal to 0 is used. Censpoint is only needed for bootstrap bias correction. |
| `intwd` | Width of intervals into which the time period is split; default is `ceiling(cpmax/20)`. Has to be an integer value. |
| `cpmax` | Upper bound for estimated change point. Time period is split into intervals up to this point. Has to be an integer value. |
| `cpmin` | Lower bound for estimated change point; default is `cpmin=0`. Has to be an integer value. |
| `norm.riskset` | Logical; if `TRUE` normalized number of units at risk is used within an interval. |
| `B.correct` | Number of bootstrap samples for bias correction; defaults to 49. |
| `parametric` | Logical; if `TRUE` parametric bootstrap bias correction is used (simulation of boostrap samples is based on estimated Weibull parameters); otherwise Kaplan-Meier is used for a nonparametric bootstrap bias correction. |
| `times.int` | Logical; if `TRUE` simulated survival times are integers. |

| | |
|---|---|
| `opt.start` | Numeric vector of length two; initial values for the Weibull parameters (shape and scale parameters) to be optimized if parametric bootstrap bias correction is used. |

### Value

A list with bias-corrected change point and optional estimated shape and scale parameters of the Weibull distribution.

---

## `cpest` *Estimates change point using shifted intervals*

### Description

Shifts intervals iteratively and estimates change point at each step. Final change point is calculated by optimization over all estimations.

### Usage

```
cpest(time, event, cpmax, intwd, cpmin, norm.riskset)
```

### Arguments

| | |
|---|---|
| `time` | Numeric vector with survival times. |
| `event` | Numeric vector indicating censoring status; 0 = alive (censored), 1 = dead (uncensored). If missing, all observations are assumed to be uncensored. |
| `cpmax` | Upper bound for estimated change point. Time period is split into intervals up to this point. Has to be an integer value. |
| `intwd` | Width of intervals into which the time period is split; default is `ceiling(cpmax/20)`. Has to be an integer value. |
| `cpmin` | Lower bound for estimated change point; default is `cpmin=0`. Has to be an integer value. |
| `norm.riskset` | Logical; if `TRUE` normalized number of units at risk is used within an interval. |

### Value

A list with estimated change point, p-values of exact binomial test, mean of p-values above estimated change point (part of regression function), lower and upper bounds of confidence intervals.

### See Also

`cpsurv`

---

## `cpsurv` *Nonparametric Change Point Estimation*

---

### Description

Change point estimation for survival data based on exact binomial test.

### Usage

```
cpsurv(time, event, cpmax, intwd, cpmin = 0, censoring = c("random",
  "type1", "no"), censpoint = NULL, biascorrect = FALSE,
  parametric = FALSE, B.correct = 49, opt.start = c(0.1, 50),
  boot.ci = FALSE, B = 999, conf.level = 0.95, norm.riskset = TRUE,
  seed = NULL, parallel = TRUE, cores = 4L)
```

### Arguments

| | |
|---|---|
| `time` | Numeric vector with survival times. |
| `event` | Numeric vector indicating censoring status; 0 = alive (censored), 1 = dead (uncensored). If missing, all observations are assumed to be uncensored. |
| `cpmax` | Upper bound for estimated change point. Time period is split into intervals up to this point. Has to be an integer value. |
| `intwd` | Width of intervals into which the time period is split; default is `ceiling(cpmax/20)`. Has to be an integer value. |
| `cpmin` | Lower bound for estimated change point; default is `cpmin=0`. Has to be an integer value. |
| `censoring` | Type of right-censoring for simulated data on which the bootstrap bias correction is based. Possible types are "random" for *random censoring* (default), "type1" for *Type I censoring* or "no" for data without censored observations. Because simulated data should be similar to given data, the censoring type is adapted from vector 'events' if given and argument 'censoring' is ignored than. |
| `censpoint` | Point of *Type I censoring*; if missing, minimum time after which all events are equal to 0 is used. Censpoint is only needed for bootstrap bias correction. |
| `biascorrect` | Logical; if `TRUE`, a bootstrap bias correction is performed; see 'Details'. |
| `parametric` | Indicator for parametric bias-correction (see Details for more information). |
| `B.correct` | Number of bootstrap samples for bias-correction; defaults to 49. |
| `opt.start` | Numeric vector of length two; initial values for the Weibull parameters (shape and scale parameters) to be optimized if parametric bootstrap bias correction is used. |
| `boot.ci` | Indicator if confidence intervals (and thereby standard deviation) should be calculated by bootstrap sampling. Please note the extended runtime (see details for examples). |
| `B` | Number of bootstrap samples for confidence intervals; defaults to 999. |

| | |
|---|---|
| `conf.level` | Confidence level for bootstrap confidence intervals. |
| `norm.riskset` | Logical; if `TRUE` normalized number of units at risk is used within an interval. |
| `seed` | Seed for random number generator (optional). |
| `parallel` | Indicator if bootstrap-sampling is executed parallelized (based on package 'parallel'); operating system is identified automatically. |
| `cores` | Number of CPU-cores that are used for parallelization; maximum possible value is the detected number of logical CPU cores. |

## Details

Change point is a point in time, from which on the hazard rate is supposed to be constant. For its estimation the timeline up to `cpmax` is split into equidistant intervals of width `intwd` and exact binomial tests are executed for each interval. The change point is estimated by fitting a regression model on the resulting p-values. See Brazzale *et al* (2017) for details.

For bootstrap bias correction the change point is estimated for a given number (`B.correct`) of bootstrap samples whereupon the bias is built by subtracting their median from primary estimation. Depending on argument `parametric` the data for bootstrapping are simulated either parametric (Weibull distributed with estimated shape and scale parameters) or nonparametric (based on Kaplan-Meier estimation).

## Value

| | |
|---|---|
| `cp` | estimated change point |
| `p.values` | p-values resulting from exact binomial test |
| `pv.mean` | mean of p-values for intervals above the estimated change point |
| `lower.lim` | lower interval limits |
| `upper.lim` | upper interval limits |
| `cp.bc` | bias corrected change point |
| `ml.shape` | ML estimator of shape parameter for Weibull distribution |
| `ml.scale` | ML estimator of scale parameter for Weibull distribution |
| `cp.boot` | estimated change points for bootstrap samples |
| `sd` | standard deviation estimated by bootstrap sampling |
| `ci.normal` | confidence interval with normal approximation |
| `ci.percent` | bootstrap percentile interval |
| `conf.level` | the `conf.level` argument passed to `cpsurv` |
| `B` | the `B` argument passed to `cpsurv` |
| `time` | the `time` argument passed to `cpsurv` |
| `event` | the `event` argument passed to `cpsurv` |
| `cpmax` | the `cpmax` argument passed to `cpsurv` |
| `intwd` | the `intwd` argument passed to `cpsurv` |
| `call` | matched call |

## Author(s)

Stefanie Krügel <`stefanie.kruegel@gmail.com`>

## References

Brazzale, A. R. and Küchenhoff, H. and Krügel, S. and Hartl, W. (2017) *Nonparametric change point estimation for survival distributions with a partially constant hazard rate.*

## Examples

```
data(survdata)
# estimate change point for survdata (random censored)
cp <- cpsurv(survdata$time, survdata$event, cpmax = 360, intwd = 20)
summary(cp)

## Not run:
# estimation with parametric bootstrap bias correction
cp_param <- cpsurv(survdata$time, survdata$event, cpmax = 360, intwd = 20,
            biascorrect = TRUE, parametric = TRUE)
summary(cp_param)

# with bootstrap confidence intervals and parametric bootstrap bias
cp_ci <- cpsurv(survdata$time, survdata$event, cpmax = 360, intwd = 20,
biascorrect = TRUE, parametric = FALSE, boot.ci = TRUE, cores = 4, seed = 36020)
# runtime: approx. 180 min (with Intel(R) Core(TM) i7 CPU 950 @ 3.07GHz, 4 logical CPUs used)

## End(Not run)
```

---

# `km.sim.survtimes` *Simulates Survival Times using Kaplan-Meier*

## Description

Simulates Survival Times using Kaplan-Meier

## Usage

```
km.sim.survtimes(nobs, time, event, weibexp, changeP = NULL)
```

## Arguments

| | |
|---|---|
| `nobs` | Number of observations. |
| `time` | Numeric vector with survival times. |
| `event` | Numeric vector indicating censoring status; 0 = alive (censored), 1 = dead (uncensored). If missing, all observations are assumed to be uncensored. |
| `weibexp` | Logical; if `TRUE`, survival times above change point have constant hazard; if `FALSE` all survival times are generated by using the estimated survival curve (relevant for generation of censoring times). |
| `changeP` | Change point |

---

## `neg.loglik.WeibExp` *Negative Log-Likelihood for Weibull-Exponential Distribution*

---

### Description

Negative Log-Likelihood for Weibull-Exponential Distribution

### Usage

```
neg.loglik.WeibExp(param, changeP, time, event)
```

### Arguments

| | |
|---|---|
| `param` | Shape and scale parameter for Weibull distribution. |
| `changeP` | Changepoint. |
| `time` | Vector of survival times. |
| `event` | Vector indicating censoring status; 0 = alive (censored), 1 = dead (uncensored). |

### Value

Value of the negative log-likelihood.

---

## `plot.cpsurv` *Plot method for objects of class cpsurv*

---

### Description

Plot method for objects of class ’cpsurv’ inheriting from a call to `cpsurv`.

### Usage

```
## S3 method for class 'cpsurv'
plot(x, type = "all", ci = TRUE, ci.type = c("perc",
  "norm"), const.haz = TRUE, regline = TRUE, legend = TRUE, xlim = NULL,
  ylim = NULL, main = NULL, xlab = NULL, ylab = NULL, min.time,
  max.time, n.est.grid = 101, ask = TRUE, ...)
```

## Arguments

| | |
|---|---|
| `x` | An object of class 'cpsurv' (estimated with `cpsurv`). |
| `type` | A vector of character strings to select the plots for printing. The value should be any subset of the values c("pvals", "events", "hazard") or simply "all", where all possible plots are shown. |
| `ci` | Logical; if `TRUE`, a bootstrap confidence interval is plotted (if existing). |
| `ci.type` | Character representing the type of confidence interval to plot (if existing); "perc" for percentile interval and "norm" for CI with normal approximation (default is "perc"). |
| `const.haz` | Logical; if `TRUE`, the estimated constant hazardrate is plotted. |
| `regline` | Logical; if `TRUE`, the regression line is plotted. |
| `legend` | Logical; if `TRUE`, the plots contain legends. |
| `xlim` | Vector with x limits (timeline) for each plot if supplied; default is c(0, x$cpmax). |
| `ylim` | Vector with y limits for plots of type "events" and "hazard". For changing ylim for only one of them, plot them separately by use of argument 'type'. |
| `main` | Main title for each plot if supplied. |
| `xlab` | Character vector used as x label for all plots if supplied. |
| `ylab` | Character vector used as y label for all plots if supplied. |
| `min.time` | Left bound of time domain used for `muhaz`. If missing, min.time is considered 0. |
| `max.time` | Right bound of time domain used for `muhaz`. If missing, value 'cpmax' of object x is used. |
| `n.est.grid` | Number of points in the estimation grid, where hazard estimates are computed (used for `muhaz`). Default value is 101. |
| `ask` | If `TRUE`, the user is asked for input, before a new figure is drawn. |
| `...` | Additional arguments passed through to plotting functions. |

## Details

The value `type = "pvals"` produces a plot with p-values used to estimate the stump regression model with superimposed least squares regression line. For `type = "events"` a barplot is produced with frequency of events per unit at risk for each interval (with length `intwd`. For `type = "hazard"` the estimated hazard rate (based on `muhaz`) is plotted with optional (normal- or percentile-) confidence intervals and the estimated constant hazard rate.

## See Also

`muhaz`

## Examples

```
data(survdata)
cp <- cpsurv(survdata$time, survdata$event, cpmax = 360, intwd = 10)
plot(cp, ask = FALSE)

## Not run:
cp <- cpsurv(survdata$time, survdata$event, cpmax = 360, intwd = 10,
boot.ci = TRUE)
plot(cp, type = "pvals", ask = FALSE)

## End(Not run)
```

---

# `sim.survdata` *Simulate Survival Data with Change Point*

---

## Description

Simulates Weibull distributed survival data from a given data set with change point above which hazard rate is constant.

## Usage

```
sim.survdata(time, event, changeP, shape, scale, censoring, censpoint,
  times.int, parametric)
```

## Arguments

| | |
|---|---|
| `time` | Numeric vector with survival times. |
| `event` | Numeric vector indicating censoring status; 0 = alive (censored), 1 = dead (uncensored). If missing, all observations are assumed to be uncensored. |
| `changeP` | Change point. |
| `shape` | Shape parameter of Weibull distribution. |
| `scale` | Scale parameter of Weibull distribution. |
| `censoring` | Logical; if `TRUE`, censored data are generated. |
| `censpoint` | Censoring point for Type I censoring. |
| `times.int` | Logical; if `TRUE`, returned survival times are integers. |
| `parametric` | Logical; if `TRUE`, survival times are generated parametrically by inverse transform sampling; otherwise Kaplan-Meier is used for simulation. |

## Value

A dataset with survival times and corresponding censoring status (’event’).

---

## `summarize.cpsurv` *Summarize and print cpsurv objects*

---

### Description

Summary and print methods for objects inheriting from a call to `cpsurv`.

### Usage

```
## S3 method for class 'cpsurv'
print(x, ...)

## S3 method for class 'cpsurv'
summary(object, ...)

## S3 method for class 'summary.cpsurv'
print(x, ...)
```

### Arguments

| | |
|---|---|
| `x` | An object of class `cpsurv` or `summary.cpsurv` to be printed out. |
| `...` | not used |
| `object` | An object of class `cpsurv`. |

### Details

The main results from `cpsurv` are printed out in a well-arranged format. If the estimated change point is bias corrected, both estimates (the original, and the corrected one) are shown in the summary. If a bootstrap-sampling was executed, the output contains a summary of the resultant bootstrap-estimates.

### See Also

`cpsurv`

### Examples

```
data(survdata)
cpest <- cpsurv(survdata$time, survdata$event, cpmax = 360)
summary(cpest)
```

---

`survdata` *Simulated Survival Data*

---

## Description

A simulated dataset with 1500 fake right-censored survival times with a change point at `time = 90`. The survival times are Weibull distributed with parameters `shape = 0.44` and `scale = 100` below the change point and have a constant hazard rate above.

## Usage

```
survdata
```

## Format

| | |
|---|---|
| `time` | survival or censoring time |
| `event` | censoring status (0 = alive, 1 = dead) |

# Index