Package ‘cutpoint’

July 22, 2025

Type Package

Title Estimate Cutpoints of Metric Variables in the Context of Cox Regression

Version 1.0.0

Description Estimate one or two cutpoints of a metric or ordinal-scaled variable in the multivariable context of survival data or time-to-event data. Visualise the cutpoint estimation process using contour plots, index plots, and spline plots. It is also possible to estimate cutpoints based on the assumption of a U-shaped or inverted U-shaped relationship between the predictor and the hazard ratio. Govindarajulu, U., and Tarpey, T. (2022) <doi:10.1080/02664763.2020.1846690>.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

LazyData true

Imports graphics, magrittr, plotly, RcppAlgos, stats, survival, utils

RoxygenNote 7.3.2

URL `https://github.com/jan-por/cutpoint`

BugReports `https://github.com/jan-por/cutpoint/issues`

Depends R (>= 3.5)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Jan Porthun [aut, cre, cph] (ORCID: <`https://orcid.org/0000-0002-4100-4361`>)

Maintainer Jan Porthun <`jan.porthun@ntnu.no`>

Repository CRAN

Date/Publication 2025-05-09 15:20:09 UTC

Contents

cp_est *Estimate cutpoints in a multivariable setting for survival data*

Description

One or two cutpoints of a metric variable are estimated using either the AIC (Akaike Information Criterion) or the LRT (Likelihood-Ratio Test statistic) within a multivariable Cox proportional hazards model. These cutpoints are used to create two or three groups with different survival probabilities.

The cutpoints are estimated by dichotomising the variable of interest, which is then incorporated into the Cox regression model. The cutpoint of this variable is the value at which the AIC reaches its lowest value or the LRT statistic achieves its maximum for the corresponding Cox-regression model.

This process occurs within a multivariable framework, as other covariates and/or factors are considered during the search for the cutpoints. Cutpoints can also be estimated when the variable of interest shows a U-shaped or inverse U-shaped relationship to the hazard ratio of time-to-event data. The argument `symtail` facilitates the estimation of two cutpoints, ensuring that the two outer tails represent groups of equal size.

Usage

```
cp_est(
  cpvarname,
  time = "time",
  event = "event",
  covariates = NULL,
  data = data,
  nb_of_cp = 1,
  bandwith = 0.1,
  est_type = "AIC",
  cpvar_strata = FALSE,
  ushape = FALSE,
  symtails = FALSE,
  dp = 2,
  plot_splines = TRUE,
  all_splines = TRUE,
  print_res = TRUE,
  verbose = TRUE
)
```

Arguments

cpvarname character, the name of the variable for which the cutpoints are estimated.

time character, this is the follow-up time.

event character, the status indicator, normally 0=no event, 1=event

covariates character vector with the names of the covariates and/ or factors. If no covariates are used, set covariates = NULL.

data a data.frame, contains the following variables:

- variable which is dichotomized
- follow-up time
- event (status indicator)
- covariates and/or cofactors

nb_of_cp numeric, number of cutpoints to be estimated (1 or 2). The default is: nb_of_cp = 1. The other option is nb_of_cp = 2.

bandwith numeric, minimum group size per group in percent of the total sample size, bandwith must be between 0.05 and 0.30, default is 0.1 If ushape = TRUE, bandwidth must be at least 0.1.

est_type character, the method used to estimate the cutpoints. The default is 'AIC' (Akaike information criterion). The other options is 'LRT' (likelihood ratio test statistic)

cpvar_strata logical value: if FALSE, The dichotomised variable serves as covariate in the Cox-regression model for cutpoint determination. If TRUE, the dichotomised variable is included as a strata in the Cox-regression model to determine the cutpoint rather than as a covariate. Default is FALSE.

ushape logical value: if TRUE, the cutpoints are estimated under the assumtion that the spline plot shows a U-shaped form or a inverted U-shaped curve. Default is FALSE.

symtails logical value: if TRUE, the cutpoints are estimated with symmetric tails. If nb_of_cp = 1, symtails is set to FALSE. Default is FALSE.

dp numeric, number of decimal places the cutpoints are rounded to. Default is dp = 2.

plot_splines logical value: if TRUE, a penalized spline plot is created. Default is TRUE.

all_splines logical value: if TRUE, The plot shows splines with different degrees of freedom. This may help determine whether misspecification or overfitting occurs. Default is TRUE.

print_res logical value: if TRUE the function prints the summary of the cutpoint estimation to the console. Default is TRUE.

verbose logical value: if TRUE the function prints the approximate remaining process-time and other information to the console. If FALSE, no information will be printed to the console, including the summary of the cutpoint estimation. Default is TRUE.

Value

Returns the `cpobj` object with cutpoints and the characteristics of the formed groups.

References

Govindarajulu, U., & Tarpey, T. (2020). Optimal partitioning for the proportional hazards model. Journal of Applied Statistics, 49(4), 968–987. https://doi.org/10.1080/02664763.2020.1846690

See Also

`cp_splines_plot()` for penalized spline plots, `cp_value_plot()` for Value plots and Index plots

Examples

```
# Example 1:
# Estimate two cutpoints of the variable biomarker.
# The dataset data1 is included in this package and contains
# the variables time, event, biomarker, covariate_1, and covariate_2.
cpobj <- cp_est(
  cpvarname  = "biomarker",
  covariates = c("covariate_1", "covariate_2"),
  data       = data1,
  nb_of_cp   = 2,
  plot_splines = FALSE
  )

# Example 2:
# Searching for cutpoints, if the variable shows a U-shaped or
# inverted U-shaped relationship to the hazard ratio.
# The dataset data2_ushape is included in this package and contains
# the variables time, event, biomarker, and cutpoint_1.
cpobj <- cp_est(
  cpvarname  = "biomarker",
  covariates = c("covariate_1"),
  data       = data2_ushape,
  nb_of_cp   = 2,
  bandwith   = 0.2,
  ushape     = TRUE,
  plot_splines = FALSE
  )
```

`cp_estsum` *Summarise cutpoint estimation*

Description

Writes the summary of the cutpoint estimation to the console.

Usage

```
cp_estsum(cpobj, verbose = TRUE)
```

Arguments

cpobj	list, contains variables for cp_estsum function
verbose	logical value: if TRUE the summary of the cutpoint estimation is writing to the console. Default is TRUE.

Value

Summary of the cutpoint estimation.

See Also

cp_est() for main function of the package.

Examples

```
# Example
# Writes the summary to the console
# The data set data1 is included in this package
cpobj <- cp_est(
  cpvarname    = "biomarker",
  covariates   = c("covariate_1", "covariate_2"),
  data         = data1,
  nb_of_cp     = 2,
  plot_splines = FALSE,
  print_res    = FALSE
)
cp_estsum(cpobj, verbose = TRUE)
```

cp_splines_plot *Plot penalized smoothing splines from* cpobj *object*

Description

Create penalized smoothing splines plot with different degrees of freedom and shows the cutpoints of the dichotomised variable.

Usage

```
cp_splines_plot(cpobj, show_splines = TRUE, adj_splines = TRUE)
```

Arguments

cpobj
: list, contains variables for pspline plot:

- nb_of_cp (number of cutpoints)
- cp (contain one or two cutpoint/s)
- dp (digits for plot)
- cpvarname (name of the variable for that the cutpoints are estimated)
- cpdata a data frame, contains the following variables: a variable that is dichotomized, time (follow-up time), event (status indicator), covariates (a vector with the names of the covariates and/or factors))

show_splines
: logical, if TRUE, The plot shows splines with different degrees of freedom. This may help determine whether misspecification or overfitting occurs.

adj_splines
: logical, if TRUE, the splines are adjusted for the covariates. Default is TRUE.

Value

Plots penalized smoothing splines and shows the cutpoints.

See Also

cp_est() for main function of the package, cp_value_plot() for Value plots and Index plots

Examples

```
cpvar <- rnorm(100, mean = 100, sd = 10)
time <- seq(1, 100, 1)
event <- rbinom(100, 1, 0.5)
datf <- data.frame(time, event, cpvar)
plot_splines_list <- list(cpdata = datf, nb_of_cp = 1, cp = 95, dp = 2,
    cpvarname = "Biomarker")
cp_splines_plot(plot_splines_list)
```

---

cp_value_plot *Plot AIC and LRT-statistics values from* cpobj *object*

Description

Create a plot of AIC or Likelihood ratio test statistic values for the estimation procedure. If there are two cutpoints, a Contour-plot and an Index-plot can be generated.

Usage

```
cp_value_plot(
  cpobj,
  plotvalues = "AIC",
  dp.plot = 2,
  show_limit = TRUE,
  plottype2cp = "contour"
)
```

Arguments

`cpobj` list, contains a vector of AIC values (AIC_values) and Likelihood ratio test statistic values (LRT_values) of the estimating procedure

`plotvalues` character, either `AIC` or `LRT`. Either the AIC or LRT values are displayed. Default is `AIC`.

`dp.plot` numeric, digits for the AIC values and LRT values. Default is `2`.

`show_limit` logical, if `TRUE` the minimum AIC value is shown in the plot if `plotvalues = "AIC"`, and the maximum LRT value is shown if `plotvalues = "LRT"`

`plottype2cp` character, either `contour` or `index`. Default is `contour`. This option is available only when searching for two cutpoints. Index plots and contour plots can be selected. Index plots display all AIC or LRT values from the estimation process as a scatter plot. Contour plots are shown in the RStudio viewer and illustrate the two potential cutpoints along with the corresponding AIC or LRT values. Index plots that do not show extreme values suggest that there may not be any actual cutpoints in the data. Contour plots provide an opportunity to explore whether there might be other potential cutpoints with similar AIC or LRT values. The smaller the `bandwidth` (minimum group size per group), the more precise and meaningful the contour plots can be interpreted.

Value

Plots the AIC- or LRT-values, derived from the estimation procedure.

See Also

`cp_est()` for main function of the package, `cp_splines_plot()` for penalized spline plots

Examples

```
# Example 1
# Plot AIC-values and potential cutpoints of the estimation process

# Create AIC values:
AIC_values <- c(1950:1910, 1910:1920, 1920:1880, 1880:1920)
AIC_values <- round(AIC_values + rnorm(length(AIC_values),
                   mean = 0, sd = 5), digits = 2)

# Create a cutpoint variable:
cpvariable_values <- matrix(NA, nrow = length(AIC_values), ncol = 2)
cpvariable_values[ ,1] <- c(1:length(AIC_values))

# Create a cutpoint object (cpobj):
cpobj <- list(AIC_values        = AIC_values,
              nb_of_cp          = 1,
              cpvariable_values = cpvariable_values,
              cpvarname         = "Cutpoint variable"
              )

cp_value_plot(cpobj, plotvalues = "AIC", dp.plot = 2, show_limit = TRUE)
```

```
# Example 2
# Splines plot based on data1
# The data set data1 is included in this package
cpobj <- cp_est(
  cpvarname    = "biomarker",
  covariates   = c("covariate_1", "covariate_2"),
  data         = data1,
  nb_of_cp     = 2,
  plot_splines = TRUE,
)
# Example 3
# Contour plot based on data1
# The data set data1 is included in this package
cpobj <- cp_est(
   cpvarname    = "biomarker",
   covariates   = c("covariate_1", "covariate_2"),
   data         = data1,
   nb_of_cp     = 2,
   plot_splines = FALSE,
)
cp_value_plot(cpobj, plotvalues = "AIC", plottype2cp = "contour")
```

---

data1 *Dataset for testing the cutpoint estimating function: cp_est*

Description

A dataset containing data for testing the estimating of one or two cutpoints

Usage

```
data(data1)
```

Format

"data1":
A data frame with 100 rows and 5 variables:

biomarker numeric from 1 to 257

covariate_1 numeric, from 4.25 to 12.33, with effect of the cutpoint of the biomarker

covariate_2 numeric, from 465 to 1205, with no or small effect of the cutpoint of the biomarker

time numeric, from 3 to 328

event numeric, 0 or 1

Author(s)

Jan Porthun

Source

Self-generated example data

Examples

```
data(data1)
```

data2_ushape *Dataset for testing the ushape argument of cp_est function*

Description

A dataset containing data for testing the ushape argument of cp_est function.

Usage

```
data(data2_ushape)
```

Format

"data2_ushape":
A data frame with 200 rows and 4 variables:

biomarker numeric from 1e-04 to 4.7
covariate_1 numeric, from 8.07e-05 to 1.90
time numeric, from 0.002 to 5.09
event numeric, 0 or 1

Author(s)

Jan Porthun

Source

Self-generated example data

Examples

```
data(data2_ushape)
```

Index