Package ‘contsurvplot’

January 29, 2026

Title Visualize the Effect of a Continuous Variable on a Time-to-Event Outcome

Version 0.2.3

Maintainer Robin Denz <robin.denz@rub.de>

Description

Graphically display the (causal) effect of a continuous variable on a time-to-event outcome using multiple different types of plots based on g-computation. Those functions include, among others, survival area plots, survival contour plots, survival quantile plots and 3D surface plots. Due to the use of g-computation, all plot allow confounder-adjustment naturally. For details, see Robin Denz, Nina Timmesfeld (2023) <doi:10.1097/EDE.0000000000001630>.

License GPL (>= 3)

URL https://github.com/RobinDenz1/contsurvplot, https://robindenz1.github.io/contsurvplot/

BugReports https://github.com/RobinDenz1/contsurvplot/issues

Imports ggplot2 (>= 3.4.0), dplyr, rlang, riskRegression, foreach

Suggests survival, pammtools, gganimate, transformr, plotly, reshape2, doParallel, knitr, rmarkdown, testthat (>= 3.0.0), vdiffr (>= 1.0.0), covr

VignetteBuilder knitr

Config/testthat/edition 3

Contact <robin.denz@rub.de>

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation no

Author Robin Denz [aut, cre]

Repository CRAN

Date/Publication 2026-01-29 09:00:02 UTC

Contents

`contsurvplot-package` *Visualize the Effect of a Continuous Variable on a Time-To-Event Outcome*

Description

What is this package about?

This package provides different plotting routines to visualize the (causal) effect of a continuous variable on a time-to-event outcome using a previously fit model and g-computation. Unlike simpler alternatives, such as plotting survival curves for some categories, these plots always correspond to the results obtained by the time-to-event model and give an accurate depiction of the causal effect, if all assumptions are met.

What features are included in this package?

The package includes 11 different plot functions, most based on the **ggplot2** package. Those 11 plotting routines include *value specific survival curves*, *landmark survival probability plots*, *survival time quantile plots*, *survival probability heatmaps*, and *survival area plots*, among others. A description and comparison of these plots can be found in the article associated with this R-package (Denz & Timmesfeld 2022).

What does a typical workflow using this package look like?

All the user has to do is fit a time-to-event model, such as the cox-model, including the continuous variable of interest (and possibly confounders) and plug it into one of the plot functions included in this package. Many different kind of models are supported. See `curve_cont` for more details.

What type of plot should I use?

There is no general answer to this question, but we would usually suggest using a plot method that is able to visualize the causal survival probability both as a function of time and as a function of the continuous variable. The `plot_surv_area`, `plot_surv_heatmap` and `plot_surv_contour` functions do just that. More discussion about this topic can be found in the vignette and the associated paper.

What is the difference between displaying causal effects and associations?

The plots generated by this package offer different ways to depict the association between a continuous variable and a time-to-event outcome. Under certain causal identifiability assumptions, which are described in detail in our article on this topic (see references), this association can be endowed with a causal interpretation. Under these assumptions, the estimates can be interpreted as the survival probability that would have been observed if all individuals in the target population had received a specific level of the continuous variable. If these assumptions are not met, this interpretation is invalid.

Where can I get more information?

The documentation pages contain a lot of information, relevant examples and literature references. Additional examples can be found in the vignette of this package, which can be accessed using `vignette(topic="introduction", package="contsurvplot")`. We also published a preprint of the article about this package on arXiv (see references), which contains an in-depth discussion about the plots and how to interpret them.

I want to suggest a new feature / I want to report a bug. Where can I do this?

Bug reports, suggestions and feature requests are highly welcome. Please file an issue on the official github page (<https://github.com/RobinDenz1/contsurvplot>) or contact the author directly using the supplied e-mail address.

Author(s)

Robin Denz (robin.denz@rub.de)

References

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

James Robins. A New Approach to Causal Inference in Mortality Studies with a Sustained Exposure Period: Application to Control of the Healthy Worker Survivor Effect. Mathematical Modelling (1986) 7, pages 1393-1512.

`curve_cont` *Estimate Counterfactual Survival or Failure Probabilities for Levels of a Continuous Variable*

Description

This function can be utilized to estimate counterfactual survival curves or cumulative incidence functions (CIF) for specific values of a continuous covariate.

Usage

```
curve_cont(data, variable, model, horizon,
           times, group=NULL, cause=1, cif=FALSE,
           contrast="none", reference="km", ref_value=NULL,
           event_time=NULL, event_status=NULL,
           conf_int=FALSE, conf_level=0.95,
           n_boot=300, n_cores=1,
           na.action=options()$na.action,
           return_boot=FALSE, ...)
```

Arguments

`data` A `data.frame` containing all required variables.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated.

`times` A numeric vector containing points in time at which the survival probabilities should be calculated.

`group` An optional single character string specifying a factor variable in `data`. When used, the regression standardization is performed conditional on this factor variable, meaning that one estimate is returned for each level of the factor variable. See details for a better description. Set to `NULL` (default) to use no grouping variable.

`cause` The cause of interest. In standard survival data with only one event type, this should be kept at 1. For data with multiple failure types, this argument should be specified. In addition, the `cif` argument should be set to `TRUE` in those cases.

`cif` Whether to calculate the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. In those cases, this argument should always be set to `TRUE`.

`contrast` Defines what kind of estimate should be returned. Can be either `"none"` (default), `"diff"` or `"ratio"`. When `"none"` is used, it simply returns the counterfactual survival probabilities (or CIF if `cif=TRUE`) without any further calculations. If `"diff"` or `"ratio"` is used instead, the difference or ratio to some reference value will be returned. See argument below.

`reference` Defines what kind of reference value to use when estimating causal contrasts. Only used if `contrast!="none"`, ignored otherwise. Can be either `"km"` (using standard Kaplan-Meier estimates as reference) or `"value"` (using the g-computation estimates of a specific value of the `variable` as reference). To specify which value to use when using `reference="value"`, the `ref_value` argument should be used. See details for more information.

`ref_value` A single number corresponding to the reference value used when estimating causal contrasts using `reference="value"`. See details.

`event_time` A single character string specifying the time until the occurrence of the event of interest or `NULL` (default). Only used when estimating causal contrasts with `reference="km"`, ignored otherwise. If specified, this variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`event_status` A single character string specifying the status of the event of interest or `NULL` (default). Only used when estimating causal contrasts with `reference="km"`, ignored otherwise. If specified, this variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`conf_int` Whether to calculate point-wise confidence intervals or not. If `TRUE`, `n_boot` bootstrap samples are drawn, the g-computation step is performed on each bootstrap sample and the confidence intervals are calculated using the percentile method from these results. Can get very slow if the dataset is large or there are many values in `horizon` or `times`. Using `n_cores` can speed up the process a lot.

`conf_level` A number specifying the confidence level of the bootstrap confidence intervals. Ignored if `conf_int=FALSE`.

`n_boot` A single integer specifying how many bootstrap repetitions should be performed. Ignored if `conf_int=FALSE`.

`n_cores` The number of processor cores to use when performing the calculations. If `n_cores=1` (default), single threaded processing is used. If `n_cores` > `1` the **foreach** package and the **doParallel** package are used to run the calculations on `n_cores` in parallel. This might speed up the runtime considerably when it is initially slow.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`return_boot` Either `TRUE` or `FALSE` (default). If `TRUE` (and `conf_int=TRUE`) this function will return the individual bootstrap estimates instead of the usual estimates. This may be useful to perform tests or calculate other type of bootstrap confidence intervals manually.

`...` Further arguments passed to `predictRisk`.

Details

This function is used internally in all plot functions included in this R-package and generally does not need to be called directly by the user. It can however be used to get specific values or as a basis to create custom plots not included in this package. Below we give a small introduction to what this function does. A more detailed description of the underlying methodology can be found in our article on this subject (Denz & Timmesfeld 2022).

Target Estimand (default)

By default (`contrast="none"`) this function tries to estimate the survival probability at `times` that would have been observed if every individual in the sample had received a value of `horizon` in the variable. Let Z be the continuous variable we are interested in. Let T be the time until the occurrence of the event of interest. Under the potential outcome framework, there is an uncountably infinite amount of potential survival times $T^{(Z=z)}$, one for each possible value of Z. The target estimand is then defined as:

$$S_z(t) = E(I(T^{(Z=z)} > t))$$

If we additionally consider a categorical group variable D, the target estimand is similarly defined as:

$$S_{zd}(t) = E(I(T^{(Z=z,D=d)} > t))$$

where $T^{(Z=z,D=d)}$ is the survival time that would have been observed if the individual had received both $Z = z$ and $D = d$.

Target Estimand (using contrasts)

If contrasts are used (contrast!="none"), target estimands based on $S_z(t)$ or $S_{zd}(t)$ are used instead. When using contrast="diff" and reference="km", the target estimand is simply the difference between the observed survival probability in the entire sample (denoted by $S(t)$, estimated using a Kaplan-Meier estimator) and the counterfactual survival probability:

$$\Delta_{KM}(t, z) = S(t) - S_z(t)$$

If contrast="ratio" is used instead, the substraction sign is simply replace by a division. If group was specified, $S(t)$ is replaced by $S_d(t)$ (a stratified Kaplan-Meier estimator) and $S_z(t)$ is replaced by $S_{zd}(t)$. Instead of using a Kaplan-Meier estimator as reference, one may also use a specific $S_z(t)$ as reference using reference="value" and setting ref_value to the Z that should be used. All of these causal contrasts and their implications are described in detail in the appendix of our article on this topic (Denz & Timmesfeld 2022).

Estimation Methodology

G-Computation, also known as the *Corrected Group Prognosis* method, *Direct-Standardization* or *G-Formula*, is used internally to estimate the counterfactual survival probability or CIF for values of a continuous variable. This is done by setting the variable to a specific value for all rows in data first. Afterwards, the model is used to predict the survival probability of each individual at all times, given the value of variable and their other observed covariates (which are included in the model as independent variables). These estimates are then averaged for each time point. This procedure is repeated for every value in horizon. If a group is supplied, these calculations are repeated for every possible value in group, with the estimated individual survival probabilities also being conditional on that value.

To obtain valid estimates of the target estimand using this function, the fundamental causal identifiability assumptions have to be met. Those are described in detail in our article on this topic (Denz & Timmesfeld 2022). If those assumptions are not met, the estimates may only be used to showcase simple associations. They cannot be endowed with a counterfactual interpretation in this case.

Supported Models

This function relies on the predictRisk function from the **riskRegression** package to create the covariate and time specific estimates of the probabilities. All models with an associated predictRisk method may be used in this function. This includes a variety of models, such as the Cox proportional hazards regression model and the aalen additive hazards model. If the model that should be used has no predictRisk method, the user either needs to write their own predictRisk method or contact the maintainers of the **riskRegression** package.

Bootstrap Confidence Intervals

By using conf_int=TRUE, bootstrap confidence intervals may be estimated. This will draw n_boot samples of size nrow(data) with replacement from data in the first step. Afterwards, the model is fit to each of those bootstrap samples and the curve_cont function is recursively called to obtain the estimates of interests for each sample. Using the percentile approach, the bootstrap confidence intervals are calculated. This requires that the model object contains a call parameter, because the update() function is used internally.

Computational Complexity

If many values are included in times, horizon or both and/or data has a lot of rows, this function may become very slow. Since bootstrapping relies on sampling with replacement from the original data and repeating the entire procedure n_boot times, this also dramatically increases the runtime. Parallel processing may be used to speed up the computations. This can be done by simply setting n_cores to values higher than 1.

Missing Data

Currently, this function does not support advanced handling of missing data. The data.frame supplied to data should contain no missing data in the relevant columns. To achieve this, the na.action argument can be set to "na.omit", which will remove all rows that do contain missing data.

Competing Events

By supplying a model that directly takes into account competing events and using the cause argument, the user may also use the functionality offered in this package to create plots in this setting. Internally, the predictRisk method will then be used to estimate the conditional cause-specific cumulative incidence function, which will then be used to carry out the g-computation step explained above. **However** the underlying target estimand of this procedure is dependent on which kind of model was supplied. It is therefore not possible to define it here concisely. Future research is necessary to clarify this point. This feature should only be used with great caution.

Value

Returns a data.frame containing the columns time (the point in time), est (the estimated survival probability or CIF) and cont (the specific value of variable used). If group was used, it includes the additional group column, specifying the level of the grouping variable. If conf_int=TRUE was used, it additionally includes the columns se (bootstrap standard error of the estimate), ci_lower (lower limit of the bootstrap confidence interval) and ci_upper (upper limit of the bootstrap confidence interval).

Author(s)

Robin Denz

References

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

Brice Ozenne, Anne Lyngholm Sorensen, Thomas Scheike, Christian Torp-Pedersen and Thomas Alexander Gerds. riskRegression: Predicting the Risk of an Event using Cox Regression Models. The R Journal (2017) 9:2, pages 440-460.

I-Ming Chang, Rebecca Gelman, and Marcello Pagano. Corrected Group Prognostic Curves and Summary Statistics. Journal of Chronic Diseases (1982) 35, pages 669-674

James Robins. A New Approach to Causal Inference in Mortality Studies with a Sustained Exposure Period: Application to Control of the Healthy Worker Survivor Effect. Mathematical Modelling (1986) 7, pages 1393-1512.

See Also

`predictRisk`

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# estimate survival probability at some points in time, for
# a range of age values
plotdata <- curve_cont(data=nafld1,
                       variable="age",
                       model=model,
                       horizon=c(50, 60, 70, 80),
                       times=c(1000, 2000, 3000, 4000))

# estimate cumulative incidences instead
plotdata <- curve_cont(data=nafld1,
                       variable="age",
                       model=model,
                       horizon=c(50, 60, 70, 80),
                       times=c(1000, 2000, 3000, 4000),
                       cif=TRUE)
```

plot_surv_3Dsurface — *Plot the Survival Curve or CIF Dependent on a Continuous Variable as a 3D Surface*

Description

Using a previously fit time-to-event model, this function plots the survival curve as a function of a continuous variable using a 3D surface representation. This function can produce interactive plots using the **plotly** package, or a static image using the persp function of the **graphics** package.

Usage

```
plot_surv_3Dsurface(time, status, variable, data, model,
                    cif=FALSE, na.action=options()$na.action,
                    horizon=NULL, fixed_t=NULL, max_t=Inf,
                    interactive=FALSE,
                    xlab="Time", ylab="Survival Probability",
                    zlab=variable, ticktype="detailed",
                    theta=120, phi=20, col="green", shade=0.5,
                    range_x=NULL, range_y=NULL, range_z=NULL,
                    ...)
```

Arguments

time
: A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in data.

status
: A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in data.

variable
: A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the data.frame that is supplied to the data argument.

data
: A data.frame containing all required variables.

model
: A model describing the time-to-event process (such as an coxph model). Needs to include variable as an independent variable. It also has to have an associated predictRisk method. See ?predictRisk for more details.

cif
: Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

na.action
: How missing values should be handled. Can be one of: na.fail, na.omit, na.pass, na.exclude or a user-defined custom function. Also accepts strings of the function names. See ?na.action for more details. By default it uses the na.action which is set in the global options by the respective user.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 40 equally spaced steps.

`fixed_t` A numeric vector containing points in time at which the survival probabilities should be calculated or `NULL` (default). If `NULL`, the survival probability is estimated at every point in time at which an event occurred.

`max_t` A number indicating the latest survival time which is to be plotted.

`interactive` Whether to draw the 3D surface as a static plot (`interactive=FALSE`, the default) or as an interactive plot. When `interactive=TRUE` is used, this function relies on the **plotly** package. In this case, the only aesthetic arguments that still work are the `xlab`, `ylab` and `zlab` arguments. Everything else has to be set manually using the functionality of the **plotly** package directly.

`xlab` A character string used as the x-axis label of the plot, representing the survival time.

`ylab` A character string used as the y-axis label of the plot, representing the survival probability or CIF.

`zlab` A character string used as the z-axis of the plot, representing the continuous covariate of interest.

`ticktype` A single character: `"simple"` draws just an arrow parallel to the axis to indicate direction of increase; `"detailed"` draws normal ticks as per 2D plots. Passed to the `ticktype` argument in the `persp` function. Ignored if `interactive=TRUE`.

`theta` Angles defining the viewing direction. `theta` gives the azimuthal direction and `phi` the colatitude. Passed to the `theta` argument in the `persp` function. Ignored if `interactive=TRUE`.

`phi` See argument `theta`. Passed to the `phi` argument in the `persp` function. Ignored if `interactive=TRUE`.

`col` The color(s) of the surface facets. Transparent colours are ignored. This is recycled to the `(nx-1)(ny-1)` facets. Passed to the `col` argument in the `persp` function. Ignored if `interactive=TRUE`.

`shade` The shade at a surface facet is computed as `((1+d)/2)^shade`, where d is the dot product of a unit vector normal to the facet and a unit vector in the direction of a light source. Values of shade close to one yield shading similar to a point light source model and values close to zero produce no shading. Values in the range 0.5 to 0.75 provide an approximation to daylight illumination. Passed to the `shade` argument in the `persp` function. Ignored if `interactive=TRUE`.

`range_x` Either `NULL` or a numeric vector of length 2, specifying the range of the x-axis. Set to `NULL` by default, which is equivalent to not changing the values from their default values. Only used when `interactive=TRUE`, ignored otherwise.

`range_y` Same as `range_x`, but for the y-axis.

`range_z` Same as `range_x`, but for the z-axis.

`...` Further arguments passed to `curve_cont`.

Details

The survival curve or CIF dependent on a continuous variable can be viewed as a 3D surface. All other plots in this package try to visualize this surface by reducing it to two dimensions using color scales or summary statistics. This function on the other hand directly plots the 3D surface as such.

Although 3D plots are frowned upon by many scientists, it might be a good visualisation choice from time to time. Using the `interactive=TRUE` option makes the plot interactive, which lessens some of the valid criticism of 3D graphics. Similar looking plots for the same purpose but using a likelihood-ratio approach have been proposed in Smith et al. (2019).

Value

Returns a `plotly` object if `interactive=TRUE` is used and a matrix when `interactive=FALSE` is used (still drawing the plot).

Author(s)

Robin Denz

References

Smith, A. M.; Christodouleas, J. P. & Hwang, W.-T. Understanding the Predictive Value of Continuous Markers for Censored Survival Data Using a Likelihood Ratio Approach BMC Medical Research Methodology, 2019, 19

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(splines)
library(ggplot2)
library(plotly)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(41)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on survival for ages 50 to 80
plot_surv_3Dsurface(time="futime",
                    status="status",
                    variable="age",
                    data=nafld1,
                    model=model,
                    horizon=seq(50, 80, 0.5),
                    interactive=FALSE)
```

```
# plot effect of age on survival using an interactive plot for ages 50 to 80
plot_surv_3Dsurface(time="futime",
                    status="status",
                    variable="age",
                    data=nafld1,
                    model=model,
                    horizon=seq(50, 80, 0.5),
                    interactive=TRUE)

## showing non-linear effects

# fit cox-model with bmi modeled using B-Splines,
# adjusting for age
model2 <- coxph(Surv(futime, status) ~ age + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival using normal plot
plot_surv_3Dsurface(time="futime",
                    status="status",
                    variable="bmi",
                    data=nafld1,
                    model=model2,
                    interactive=FALSE)

# plot effect of bmi on survival using interactive plot
plot_surv_3Dsurface(time="futime",
                    status="status",
                    variable="bmi",
                    data=nafld1,
                    model=model2,
                    interactive=TRUE)
```

plot_surv_animated — *Create an Animated Plot of Survival Curves of CIFs as a Function of a Continuous Variable*

Description

Using a previously fit time-to-event model, this function plots a single covariate-specific survival curve or CIF. Using either the **plotly** or the **gganimate** package, an animated plot is created that is either simply showing how the curve evolves over values of the continuous covariate, or includes a slide to let the user interact with it.

Usage

```
plot_surv_animated(time, status, variable, group=NULL,
                   data, model, cif=FALSE, conf_int=FALSE,
                   conf_level=0.95, n_boot=300,
                   na.action=options()$na.action,
                   horizon=NULL, fixed_t=NULL, max_t=Inf,
                   slider=TRUE, size=1, color="black",
                   linetype="solid", alpha=1,
                   xlab="Time", ylab="Survival Probability",
                   title=NULL, subtitle=NULL,
                   gg_theme=ggplot2::theme_bw(),
                   facet_args=list(), ci_alpha=0.4,
                   kaplan_meier=FALSE, km_size=0.5,
                   km_linetype="solid", km_alpha=1,
                   km_color="black", km_ci=FALSE,
                   km_ci_type="plain", km_ci_level=0.95,
                   km_ci_alpha=0.4, ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`cif` Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

`conf_int` Whether to plot point-wise bootstrap confidence intervals or not. Currently only supported when using `slider=FALSE`.

`conf_level` A number specifying the confidence level of the bootstrap confidence intervals. Ignored if `conf_int=FALSE`.

`n_boot` A single integer specifying how many bootstrap repetitions should be performed. Ignored if `conf_int=FALSE`.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

horizon A numeric vector containing a range of values of variable for which the survival curves should be calculated or NULL (default). If NULL, the horizon is constructed as a sequence from the lowest to the highest value observed in variable with 40 equally spaced steps.

fixed_t A numeric vector containing points in time at which the survival probabilities should be calculated or NULL (default). If NULL, the survival probability is estimated at every point in time at which an event occurred.

max_t A number indicating the latest survival time which is to be plotted.

slider Whether to include a slider that controls the value of the continuous covariate or not. If not, the plot simply cycles through the values in horizon at a constant speed.

size A single number specifying the size of the drawn curve.

color A single character string specifying the color of the curve.

linetype A single character string specifying the linetype of the curve.

alpha The transparency level of the plot.

xlab A character string used as the x-axis label of the plot.

ylab A character string used as the y-axis label of the plot.

title A character string used as the title of the plot. When slider=FALSE is used, the title is also used in a dynamic way to show which value the continuous variable currently has. The user-defined title is still shown, but the covariate level is also appended to it in that case.

subtitle A character string used as the subtitle of the plot.

gg_theme A **ggplot2** theme which is applied to the plot.

facet_args A named list of arguments that are passed to the facet_wrap function call when creating a plot separated by groups. Ignored if group=NULL. Any argument except the facets argument of the facet_wrap function can be used. For example, if the user wants to allow free y-scales, this argument could be set to list(scales="free_y").

ci_alpha A single number defining the transparency level of the confidence interval bands.

kaplan_meier Whether to add a standard Kaplan-Meier estimator to the plot or not. If group is defined, the Kaplan-Meier estimator will be stratified by the grouping variable. If cif=TRUE was used, the cumulative incidence will be displayed instead of the survival curve.

km_size The size of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

km_linetype The linetype of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

km_alpha The transparency level of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

km_color The color of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

km_ci Whether to draw a confidence interval around the Kaplan-Meier estimates. Ignored if kaplan_meier=FALSE.

km_ci_type Which type of confidence interval to calculate for the Kaplan-Meier estimates. Corresponds to the conf.type argument in the survfit function. Ignored if kaplan_meier=FALSE or km_ci=FALSE.

km_ci_level
: Which confidence level to use for the confidence interval of the Kaplan-Meier estimates. Ignored if `kaplan_meier=FALSE` or `km_ci=FALSE`.

km_ci_alpha
: The transparency level of the confidence interval of the Kaplan-Meier estimates. Ignored if `kaplan_meier=FALSE` or `km_ci=FALSE`.

...
: Further arguments passed to `curve_cont`.

Details

When the plot doesn't have to be on paper, it is also possible to use animation to represent how the survival curve or CIF changes when different values of a continuous covariate are used. There are two options included in this function. The first one is creating a plot with a slider that can be interactively manipulated by the user (argument `slider=TRUE`). The second option creates a .gif file in which the continuous variable changes constantly over time (argument `slider=FALSE`).

This type of plot is only useful when visualizing the effect on a computer. It is also important to keep in mind that these plots might look slightly different to users with different hardware and or software.

An additional interactive plot can be created using the `plot_surv_3Dsurface` function with `interactive=TRUE`.

Value

Returns a `plotly` object if `slider=TRUE` and a `gganim` object if `slider=FALSE`.

Author(s)

Robin Denz

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(splines)
library(ggplot2)
library(gganimate)
library(transformr)
library(plotly)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

if (interactive()) {
# plot effect of age on survival for ages 50 to 80
plot_surv_animated(time="futime",
                    status="status",
                    variable="age",
                    data=nafld1,
                    model=model,
                    horizon=seq(50, 80, 1),
                    slider=TRUE)

# plot effect of age on survival using an interactive plot for ages 50 to 80
plot_surv_animated(time="futime",
                   status="status",
                   variable="age",
                   data=nafld1,
                   model=model,
                   horizon=seq(50, 80, 1),
                   slider=FALSE)
}

## showing non-linear effects

# fit cox-model with bmi modeled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

if (interactive()) {
# plot effect of bmi on survival using slider
plot_surv_animated(time="futime",
                   status="status",
                   variable="bmi",
                   data=nafld1,
                   model=model2,
                   slider=TRUE)

# plot effect of bmi on survival using a .gif
plot_surv_animated(time="futime",
                   status="status",
                   variable="bmi",
                   data=nafld1,
                   model=model2,
                   slider=FALSE)
}
```

plot_surv_area *Plot a Survival Area Plot for the Effect of a Continuous Variable on a Time-To-Event Outcome*

Description

Using a previously fit time-to-event model, this function plots a survival curve or CIF area plot. Instead of plotting value-specific curves, the probability of interest is represented as an area as a function of time, where the color changes according to the continuous variable.

Usage

```
plot_surv_area(time, status, variable, group=NULL,
               data, model, cif=FALSE,
               na.action=options()$na.action,
               horizon=NULL, fixed_t=NULL, max_t=Inf,
               start_color="blue", end_color="red", alpha=1,
               discrete=FALSE, bins=ifelse(discrete, 10, 40),
               sep_lines=FALSE, sep_color="black",
               sep_size=0.1, sep_linetype="solid",
               sep_alpha=alpha, xlab="Time",
               ylab="Survival Probability", title=NULL,
               subtitle=NULL, legend.title=variable,
               legend.position="right",
               gg_theme=ggplot2::theme_bw(),
               facet_args=list(), label_digits=NULL,
               transition_size=0.01, kaplan_meier=FALSE,
               km_size=0.5, km_linetype="solid", km_alpha=1,
               km_color="black", monotonic=TRUE, ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`cif` Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 40 equally spaced steps.

`fixed_t` A numeric vector containing points in time at which the survival probabilities should be calculated or `NULL` (default). If `NULL`, the survival probability is estimated at every point in time at which an event occurred.

`max_t` A number indicating the latest survival time which is to be plotted.

`start_color` The color used for the lowest value in `horizon`. This and the `end_color` argument can be used to specify custom continuous color scales used in the plot. For example, if a black and white plot is desired, the user can set `start_color="white"` and `end_color="black"`. See `?scale_color_gradient` for more information.

`end_color` The color used for the highest value in `horizon`. See argument `start_color`.

`alpha` The transparency level of the main plot.

`discrete` Whether to plot the area as a single continuously shaded area (default) or as multiple discrete blocks. The blocks correspond to ranges (such as 5-10). This only really changes the style of the legend and decreases the default value of `bins` to make the discrete blocks visable.

`bins` The number of bins the survival area should be divided into. In a nutshell, this function simply estimates a lot of covariate specific curves and colors the area between them on a graded scale. If many bins are used, this gives the appearance of a single shaded area, changing continuously. By using only a few bins however, the specific covariate values might be easier to read off the plot. See also the `discrete` argument.

`sep_lines` Whether to draw lines between the individual area segments or not. When `discrete=TRUE` is used, this might be a good option to help visually separate the segments.

`sep_color` The color of the separator lines. Ignored if `sep_lines=FALSE`.

`sep_size` The size of the separator lines. Ignored if `sep_lines=FALSE`.

`sep_linetype` The linetype of the separator lines. Ignored if `sep_lines=FALSE`.

`sep_alpha` The transparency level of the separator lines. Ignored if `sep_lines=FALSE`.

`xlab` A character string used as the x-axis label of the plot.

`ylab` A character string used as the y-axis label of the plot.

`title` A character string used as the title of the plot.

`subtitle` A character string used as the subtitle of the plot.

`legend.title` A character string used as the legend title of the plot.

`legend.position`
Where to put the legend. See `?theme` for more details.

`gg_theme` A **ggplot2** theme which is applied to the plot.

`facet_args` A named list of arguments that are passed to the `facet_wrap` function call when creating a plot separated by groups. Ignored if `group=NULL`. Any argument except the `facets` argument of the `facet_wrap` function can be used. For example, if the user wants to allow free y-scales, this argument could be set to `list(scales="free_y")`.

label_digits A single number specifying to how many digits the labels of the plot should be rounded or NULL (default). If NULL, no rounding is performed and the exact values are presented. This argument is only used when discrete=TRUE is used and ignored otherwise.

transition_size A single number > 0 specifying the size parameter in the pammtools::geom_stepribbon() calls whenever discrete=FALSE is used. This size parameter defines the size of the outline of each individual element of the plot. Can usually be left as is. Sometimes when saving these graphics to .png or .jpg files it may be useful to increase this value to avoid weird looking lines between the areas.

kaplan_meier Whether to add a standard Kaplan-Meier estimator to the plot or not. If group is defined, the Kaplan-Meier estimator will be stratified by the grouping variable. If cif=TRUE was used, the cumulative incidence will be displayed instead of the survival curve.

km_size The size of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

km_linetype The linetype of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

km_alpha The transparency level of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

km_color The color of the Kaplan-Meier line. Ignored if kaplan_meier=FALSE.

monotonic A single logical value specifying whether the relationship between the variable and the survival probability is monotonic or not. If there are non-monotonic effects, the areas will be drawn on top of each other when using monotonic=TRUE (default). By setting this argument to FALSE, the plot will be divided into multiple facets. Each facet contains the plot for a certain range of values of the variable, in which the effect is monotonic. The group argument may not be used with monotonic=FALSE. This also only works for continuously shaded ares, e.g. with horizons containing at least 40 distinct values.

... Further arguments passed to curve_cont.

Details

This function is very similar to a contour plot (see plot_surv_contour), but with the probability of interest on the y-axis and the color representing the values of the continuous variable of interest. The main advantage of this type of plot is, that it has the same structure as a usual Kaplan-Meier plot. The only difference is that instead of single curves stratified by some variable, the curves are plotted as an area instead.

This is achieved by estimating value-specific curves over the whole range of the continuous covariate using direct standardization and a previously fitted time-to-event model (see curve_cont). The area between those curves is then simply filled in (using the geom_stepribbon function from the **pammtools** package). By using a big number of individual curves, the appearance of a continuously shaded area is created. This is done by default when using the argument discrete=FALSE. By using discrete=TRUE, only a small number of value-specific curves are estimated, resulting in only a few discrete bins. When using only a few bins, the output is therefore even closer to a contour plot.

The major downside of this plotting method is that it has problems handling non-monotone effects. Curved relationships between the variable and the survival time result in the areas being plotted on top of each other in a standard survival area plot. By using monotonic=FALSE, the user may still use this function for nonlinear effects. This strategy is not optimal when the relationship is very complex, resulting in lots of facets in the plot. We recommend using either the `plot_surv_contour`, `plot_surv_heatmap` or `plot_surv_matrix` functions in this case.

Value

Returns a `ggplot2` object.

Note

When saving this graphic to `.png` or `.jpg` files while using `discrete=FALSE`, they may sometimes contain some barely visible (but very annoying) lines. Those are the divisions between the areas the plot is made off. This only happens when saving this plot to non-vector graphics, which is why we recommend saving it to `.pdf` or similar file types only. If other file types are necessary, one can set the `transition_size` argument to higher values (such as 0.5 or 1) to remove this problem as well.

Author(s)

Robin Denz

References

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(pammtools)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 100), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on survival using defaults
plot_surv_area(time="futime",
               status="status",
               variable="age",
               data=nafld1,
               model=model)

# plot it only for 60 to 80 year old people
plot_surv_area(time="futime",
                 status="status",
                 variable="age",
                 data=nafld1,
                 model=model,
                 horizon=seq(60, 80, 0.5))

# plot it only for 60 to 60 year old people, using discrete bins
# and a black and white color scale
plot_surv_area(time="futime",
               status="status",
               variable="age",
               data=nafld1,
               model=model,
               horizon=seq(60, 80, 5),
               discrete=TRUE,
               start_color="grey",
               end_color="black")
```

plot_surv_at_t — *Plot the Survival Probability or CIF at a Fixed Point in Time as a Function of a Continuous Variable*

Description

Using a previously fit time-to-event model, this function plots the survival probability or CIF at one or multiple user defined points at time as a function of a continuous variable.

Usage

```
plot_surv_at_t(time, status, variable, group=NULL,
               data, model, cif=FALSE, conf_int=FALSE,
               conf_level=0.95, n_boot=300,
               na.action=options()$na.action, t, horizon=NULL,
               size=1, linetype="solid", alpha=1,
               xlab=variable, ylab="Survival Probability at t",
               title=NULL, subtitle=NULL,
               legend.title="t", legend.position="right",
               gg_theme=ggplot2::theme_bw(),
               facet_args=list(), ci_alpha=0.4, ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`cif` Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

`conf_int` Whether to plot point-wise bootstrap confidence intervals or not.

`conf_level` A number specifying the confidence level of the bootstrap confidence intervals. Ignored if `conf_int=FALSE`.

`n_boot` A single integer specifying how many bootstrap repetitions should be performed. Ignored if `conf_int=FALSE`.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`t` The point in time at which the survival probability should be calculated. For example, by setting this parameter to 10, this function will display the survival probability at t = 10 over values of `variable`. If multiple values are supplied, one curve is drawn for each of them.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 100 equally spaced steps.

`size` A single number specifying how thick the lines should be drawn.

`linetype` The linetype of the drawn lines. See documentation of **ggplot2** for more details on allowed values.

`alpha` The transparency level of the lines.

`xlab` A character string used as the x-axis label of the plot.

`ylab` A character string used as the y-axis label of the plot.

`title` A character string used as the title of the plot.

`subtitle` A character string used as the subtitle of the plot.

`legend.title` A character string used as the legend title of the plot.

`legend.position`
Where to put the legend. See `?theme` for more details.

`gg_theme`
A **ggplot2** theme which is applied to the plot.

`facet_args`
A named list of arguments that are passed to the `facet_wrap` function call when creating a plot separated by groups. Ignored if `group=NULL`. Any argument except the `facets` argument of the `facet_wrap` function can be used. For example, if the user wants to allow free y-scales, this argument could be set to `list(scales="free_y")`.

`ci_alpha`
A single number defining the transparency level of the confidence interval bands.

`...`
Further arguments passed to `curve_cont`.

Details

By picking a single value of time, we can effectively reduce the dimensions of the plot by one, resulting in a simple curve plot of the survival probability or CIF at the specific point in time as it changes over values of the continuous covariate. By supplying a vector of times to the `t` argument, multiple such curves can be produced in the same plot.

This can be a very effective plotting strategy if there are specific points in time of interest already (even better if those were pre-defined). It is however a poor strategy if a representation of the effect on the survival over time is desired. An empirical example of this strategy using the 5 year risk is given in Shen et al. (2017).

As all plot functions in this package, this plot relies on the `curve_cont` function to calculate the needed probability estimates.

Value

Returns a `ggplot2` object.

Author(s)

Robin Denz

References

Shen, Y.-M.; Le, L. D.; Wilson, R. & Mansmann, U. Graphical Presentation of Patient-Treatment Interaction Elucidated by Continuous Biomarkers: Current Practice and Scope for Improvement Methods of Information in Medicine, 2017, 56, 13-27

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)

# using data from the survival package
```

```
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on survival at t=2000
plot_surv_at_t(time="futime",
               status="status",
               variable="age",
               data=nafld1,
               model=model,
               t=2000)

# plot it for arbitrary multiple values of t
plot_surv_at_t(time="futime",
               status="status",
               variable="age",
               data=nafld1,
               model=model,
               t=c(1000, 2000, 3200, 5643))
```

plot_surv_contour *Create a Contour Plot of the Effect of a Continuous Covariate on the Survival Probability*

Description

Using a previously fit time-to-event model, this function creates a contour plot with the continuous covariate on the y-axis and the time-to-event on the x-axis. The color is made in accordance with the corresponding survival probability at that point, binned into separate categories. This is very similar to the plot_surv_heatmap plot, but using a categorical representation.

Usage

```
plot_surv_contour(time, status, variable, group=NULL,
                  data, model, cif=FALSE,
                  na.action=options()$na.action,
                  horizon=NULL, fixed_t=NULL, max_t=Inf,
                  size=0.1, linetype="solid", alpha=1,
                  bins=NULL, binwidth=NULL, breaks=NULL,
                  custom_colors=NULL,
                  xlab="Time", ylab=variable,
                  title=NULL, subtitle=NULL,
                  legend.title="S(t)", legend.position="right",
                  gg_theme=ggplot2::theme_bw(),
                  facet_args=list(),
                  panel_border=FALSE, axis_dist=0, ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`cif` Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 40 equally spaced steps.

`fixed_t` A numeric vector containing points in time at which the survival probabilities should be calculated or `NULL` (default). If `NULL`, the survival probability is estimated at 100 equally spaced steps from 0 to the maximum observed event time.

`max_t` A number indicating the latest survival time which is to be plotted.

`size` The size of the individual lines plotted by the `geom_contour_filled` function. Usually this option can be kept at 0.1 (default).

`linetype` The linetype of the contour lines.

`alpha` The transparency level of the plot.

`bins` A single number specifying how many categories the survival probability or CIF should be divided in, or `NULL` (default). If `NULL`, the defaults in `geom_contour_filled` are used.

binwidth	A single number specifying the width of the bins of the survival probability or CIF categories, or NULL (default). If NULL, the defaults in geom_contour_filled are used.
breaks	A vector of specific breaks to create the survival probability or CIF bins, or NULL (default). If NULL, the defaults in geom_contour_filled are used.
custom_colors	A character vector of custom colors that should be used for each contour area.
xlab	A character string used as the x-axis label of the plot.
ylab	A character string used as the y-axis label of the plot.
title	A character string used as the title of the plot.
subtitle	A character string used as the subtitle of the plot.
legend.title	A character string used as the legend title of the plot.
legend.position	Where to put the legend. See ?theme for more details.
gg_theme	A **ggplot2** theme which is applied to the plot.
facet_args	A named list of arguments that are passed to the facet_wrap function call when creating a plot separated by groups. Ignored if group=NULL. Any argument except the facets argument of the facet_wrap function can be used. For example, if the user wants to allow free y-scales, this argument could be set to list(scales="free_y").
panel_border	Whether to draw a border around the heatmap or not. Is set to FALSE by default to mimic standard heatmaps.
axis_dist	The distance of the axis ticks to the colored heatmap. Is set to 0 by default to mimic standard heatmaps.
...	Further arguments passed to curve_cont.

Details

Contour plots are a great way to reduce three dimensions to a two-dimensional plot and therefore lend themselves easily to plot the effect of continuous variable on the survival probability or CIF over time. Heatmaps (as available in the plot_surv_heatmap) function are very similar, but are often less clear then contour plots. The major downside of these plots is that they do not have the survival probability (or CIF) on the y-axis, which is of course the standard in standard Kaplan-Meier plots. A very similar plot but with the probability of interest on the y-axis can be created using the plot_surv_area function.

This type of plot has been described in Jackson & Cox (2021). However, the authors of that article propose a different method to estimate the needed data for the plot.

Internally, this function uses the geom_contour_filled function of the **ggplot2** package to create the plot, after estimating the required probabilities using the curve_cont function.

Value

Returns a ggplot2 object.

Author(s)

Robin Denz

References

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

Jackson, R. J. & Cox, T. F. Kernel Hazard Estimation for Visualisation of the Effect of a Continuous Covariate on Time-To-Event Endpoints Pharamaceutical Statistics, 2021

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(splines)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on survival using defaults
plot_surv_contour(time="futime",
                  status="status",
                  variable="age",
                  data=nafld1,
                  model=model)

# plot it only for 60 to 80 year old people
plot_surv_contour(time="futime",
                  status="status",
                  variable="age",
                  data=nafld1,
                  model=model,
                  horizon=seq(60, 80, 0.5))

## showing non-linear effects

# fit cox-model with bmi modelled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival using defaults
plot_surv_contour(time="futime",
                  status="status",
                  variable="bmi",
                  data=nafld1,
                    model=model2)
```

plot_surv_heatmap *Plot a Heatmap of the Effect of a Continuous Covariate on a Time-To-Event Outcome*

Description

Using a previously fit time-to-event model, this function plots a heatmap with the continuous covariate on the y-axis and the time-to-event on the x-axis. The color is made in accordance with the corresponding survival probability or CIF at that point.

Usage

```
plot_surv_heatmap(time, status, variable, group=NULL,
                  data, model, cif=FALSE,
                  na.action=options()$na.action,
                  horizon=NULL, fixed_t=NULL, max_t=Inf,
                  start_color=NULL, end_color=NULL,
                  alpha=1, xlab="Time", ylab=variable,
                  title=NULL, subtitle=NULL,
                  legend.title="S(t)", legend.position="right",
                  gg_theme=ggplot2::theme_bw(),
                  facet_args=list(), panel_border=FALSE,
                  axis_dist=0, interpolate=TRUE,
                  contour_lines=FALSE, contour_color="white",
                  contour_size=0.3, contour_linetype="dashed",
                  ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`cif` Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 40 equally spaced steps.

`fixed_t` A numeric vector containing points in time at which the survival probabilities should be calculated or `NULL` (default). If `NULL`, the survival probability is estimated at 100 equally spaced steps from 0 to the maximum observed event time.

`max_t` A number indicating the latest survival time which is to be plotted.

`start_color` The color used for the lowest value in `horizon`. This and the `end_color` argument can be used to specify custom continuous color scales used in the plot. For example, if a black and white plot is desired, the user can set `start_color="white"` and `end_color="black"`. See `?scale_color_gradient` for more information.

`end_color` The color used for the highest value in `horizon`. See argument `start_color`.

`alpha` The transparency level of the plot.

`xlab` A character string used as the x-axis label of the plot.

`ylab` A character string used as the y-axis label of the plot.

`title` A character string used as the title of the plot.

`subtitle` A character string used as the subtitle of the plot.

`legend.title` A character string used as the legend title of the plot.

`legend.position`
Where to put the legend. See `?theme` for more details.

`gg_theme` A **ggplot2** theme which is applied to the plot.

`facet_args` A named list of arguments that are passed to the `facet_wrap` function call when creating a plot separated by groups. Ignored if `group=NULL`. Any argument except the `facets` argument of the `facet_wrap` function can be used. For example, if the user wants to allow free y-scales, this argument could be set to `list(scales="free_y")`.

`panel_border` Whether to draw a border around the heatmap or not. Is set to FALSE by default to mimic standard heatmaps.

`axis_dist` The distance of the axis ticks to the colored heatmap. Is set to 0 by default to mimic standard heatmaps.

`interpolate` Whether to linearly interpolate the colors or not. Set to `TRUE` by default, which results in a smooth surface being plotted. Corresponds to the `interpolate` argument in the `geom_raster` function, which is used internally.

`contour_lines` Whether to add some contour lines to the heatmap. To get a proper contour plot, use the `plot_surv_contour` function instead.

`contour_color` The color of the contour lines. Defaults to `"white"`. Ignored if `contour_lines=FALSE`.

`contour_size` The size of the contour lines. Ignored if `contour_lines=FALSE`.

`contour_linetype`
The linetype of the contour lines. Defaults to `"dashed"`. Ignored if `contour_lines=FALSE`.

`...` Further arguments passed to `curve_cont`.

Details

Heatmaps are a great tool to visualize a three dimensional surface in a two-dimensional plot. A continuous color scale is used to represent the probability of interest. Although this is fine theoretically, it is often hard to read specific information off these plots. Contour lines can be added to the plot in order to make this easier by using `contour_lines=TRUE` in the function call.

In most cases, however, it is probably better to use a proper contour plot instead, which can be produced using the `plot_surv_contour` function. This is mostly a matter of taste, which is why both types of plots are included in this package. Another alternative is the to use the `plot_surv_matrix` function, which is basically a discretized version of a survival heatmap.

The main advantage of the heatmap and the contour plots is that they can visualize the effect of a continuous covariate regardless of how it was modeled. Non-linear relationships can be visualized just as well as linear ones. The major downside is, that the structure of the plot is not the same as that of a standard Kaplan-Meier plot. An alternative that is closer to the standard plot can be created using the `plot_surv_area` function.

Value

Returns a `ggplot2` object.

Author(s)

Robin Denz

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(splines)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
```

```
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on survival using defaults
plot_surv_heatmap(time="futime",
                  status="status",
                  variable="age",
                  data=nafld1,
                  model=model)

# plot it only for 60 to 80 year old people
plot_surv_heatmap(time="futime",
                  status="status",
                  variable="age",
                  data=nafld1,
                  model=model,
                  horizon=seq(60, 80, 0.5))

## showing non-linear effects

# fit cox-model with bmi modelled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival using defaults
plot_surv_heatmap(time="futime",
                  status="status",
                  variable="bmi",
                  data=nafld1,
                  model=model2)

# plot effect of bmi on survival with contour lines
plot_surv_heatmap(time="futime",
                  status="status",
                  variable="bmi",
                  data=nafld1,
                  model=model2,
                  contour_lines=TRUE)
```

`plot_surv_lines` *Plot Individual Survival Curves or CIFs for Specific Values of a Continuous Covariate*

Description

Using a previously fit time-to-event model, this function plots survival curves or CIFs that would have been observed if every individual in the dataset had been set to specific values of a continuous covariate.

Usage

```
plot_surv_lines(time, status, variable, group=NULL,
                data, model, cif=FALSE, conf_int=FALSE,
                conf_level=0.95, n_boot=300,
                na.action=options()$na.action,
                horizon=NULL, fixed_t=NULL, max_t=Inf,
                discrete=TRUE, custom_colors=NULL,
                start_color="blue", end_color="red",
                size=1, linetype="solid", alpha=1,
                xlab="Time", ylab="Survival Probability",
                title=NULL, subtitle=NULL,
                legend.title=variable, legend.position="right",
                gg_theme=ggplot2::theme_bw(),
                facet_args=list(), ci_alpha=0.4,
                kaplan_meier=FALSE, km_size=0.5,
                km_linetype="solid", km_alpha=1,
                km_color="black", km_ci=FALSE,
                km_ci_type="plain", km_ci_level=0.95,
                km_ci_alpha=0.4, ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`cif` Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

`conf_int` Whether to plot point-wise bootstrap confidence intervals or not.

`conf_level` A number specifying the confidence level of the bootstrap confidence intervals. Ignored if `conf_int=FALSE`.

`n_boot` A single integer specifying how many bootstrap repetitions should be performed. Ignored if `conf_int=FALSE`.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 12 equally spaced steps.

`fixed_t` A numeric vector containing points in time at which the survival probabilities should be calculated or `NULL` (default). If `NULL`, the survival probability is estimated at every point in time at which an event occurred.

`max_t` A number indicating the latest survival time which is to be plotted.

`discrete` Whether to use a continuous color scale or a discrete one (default). If `FALSE`, the default **ggplot2** colors are used.

`custom_colors` An optional character vector of colors to use when `discrete=FALSE`. Ignored if `discrete=TRUE`, in which case the color gradient can be defined using the `start_color` and `end_color` arguments.

`start_color` The color used for the lowest value in `horizon`. This and the `end_color` argument can be used to specify custom continuous color scales used in the plot. For example, if a black and white plot is desired, the user can set `start_color="white"` and `end_color="black"`. See `?scale_color_gradient` for more information.

`end_color` The color used for the highest value in `horizon`. See argument `start_color`.

`size` A single number specifying the size of the drawn curves.

`linetype` A single character string specifying the linetype of the curves.

`alpha` The transparency level of the plot.

`xlab` A character string used as the x-axis label of the plot.

`ylab` A character string used as the y-axis label of the plot.

`title` A character string used as the title of the plot.

`subtitle` A character string used as the subtitle of the plot.

`legend.title` A character string used as the legend title of the plot.

`legend.position`
Where to put the legend. See `?theme` for more details.

`gg_theme` A **ggplot2** theme which is applied to the plot.

`facet_args` A named list of arguments that are passed to the `facet_wrap` function call when creating a plot separated by groups. Ignored if `group=NULL`. Any argument except the `facets` argument of the `facet_wrap` function can be used. For example, if the user wants to allow free y-scales, this argument could be set to `list(scales="free_y")`.

`ci_alpha` A single number defining the transparency level of the confidence interval bands.

`kaplan_meier` Whether to add a standard Kaplan-Meier estimator to the plot or not. If `group` is defined, the Kaplan-Meier estimator will be stratified by the grouping variable. If `cif=TRUE` was used, the cumulative incidence will be displayed instead of the survival curve.

`km_size` The size of the Kaplan-Meier line. Ignored if `kaplan_meier=FALSE`.

`km_linetype` The linetype of the Kaplan-Meier line. Ignored if `kaplan_meier=FALSE`.

`km_alpha` The transparency level of the Kaplan-Meier line. Ignored if `kaplan_meier=FALSE`.

`km_color` The color of the Kaplan-Meier line. Ignored if `kaplan_meier=FALSE`.

`km_ci` Whether to draw a confidence interval around the Kaplan-Meier estimates. Ignored if `kaplan_meier=FALSE`.

`km_ci_type` Which type of confidence interval to calculate for the Kaplan-Meier estimates. Corresponds to the `conf.type` argument in the `survfit` function. Ignored if `kaplan_meier=FALSE` or `km_ci=FALSE`.

`km_ci_level` Which confidence level to use for the confidence interval of the Kaplan-Meier estimates. Ignored if `kaplan_meier=FALSE` or `km_ci=FALSE`.

`km_ci_alpha` The transparency level of the confidence interval of the Kaplan-Meier estimates. Ignored if `kaplan_meier=FALSE` or `km_ci=FALSE`.

`...` Further arguments passed to `curve_cont`.

Details

A simple plot of multiple covariate-specific survival curves. Internally, it uses the `curve_cont` function to calculate the survival curves.

Value

Returns a `ggplot2` object.

Author(s)

Robin Denz

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(splines)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
```

```
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on survival using defaults
plot_surv_lines(time="futime",
                status="status",
                variable="age",
                data=nafld1,
                model=model)

# plot it only for some specific user-defined values
plot_surv_lines(time="futime",
                status="status",
                variable="age",
                data=nafld1,
                model=model,
                horizon=c(40, 52, 63, 81))

## showing non-linear effects

# fit cox-model with bmi modelled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival
plot_surv_lines(time="futime",
                status="status",
                variable="bmi",
                data=nafld1,
                model=model2,
                horizon=c(20, 30, 40))
```

`plot_surv_matrix` *Plot a Discretized Heatmap of the Effect of a Continuous Covariate on a Time-To-Event Outcome*

Description

Using a previously fit time-to-event model, this function plots a discretized heatmap with the continuous covariate on the y-axis and the time-to-event on the x-axis. This is essentially a discretized version of the `plot_surv_heatmap` plot, which makes it look very similar to a correlation matrix.

Usage

```
plot_surv_matrix(time, status, variable, group=NULL, data, model,
                 cif=FALSE, na.action=options()$na.action,
                 horizon=NULL, fixed_t=NULL, max_t=Inf,
                 n_col=10, n_row=10,
                 start_color="red", end_color="blue",
                 alpha=1, xlab="Time", ylab=variable,
                 title=NULL, subtitle=NULL,
                 legend.title="S(t)", legend.position="none",
                 gg_theme=ggplot2::theme_bw(),
                 facet_args=list(),
                 panel_border=FALSE, axis_dist=0,
                 border_color="white", border_size=0.5,
                 numbers=TRUE, number_color="white",
                 number_size=3, number_family="sans",
                 number_fontface="plain", number_digits=2,
                 ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`cif` Whether to plot the cumulative incidence (CIF) instead of the survival probability. If multiple failure types are present, the survival probability cannot be estimated in an unbiased way. This function will always return CIF estimates in that case.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 100 equally spaced steps. In this function, this needs to be a equally spaced vector.

`fixed_t` A numeric vector containing points in time at which the survival probabilities should be calculated or `NULL` (default). If `NULL`, the survival probability is estimated at 100 equally spaced steps from 0 to the maximum observed event time. In this function, this needs to be a equally spaced vector.

max_t A number indicating the latest survival time which is to be plotted.

n_col The number of columns to use in the matrix style heatmap. This parameter only controls how many tiles are shown, not how many are estimated. The amount of estimated probabilities is controlled using the fixed_t and horizon arguments. See details.

n_row The number of rows to use in the matrix style heatmap. See n_col.

start_color The color used for the lowest value in horizon. This and the end_color argument can be used to specify custom continuous color scales used in the plot. For example, if a black and white plot is desired, the user can set start_color="white" and end_color="black". See ?scale_color_gradient for more information. Defaults to "red", set to NULL to use the default **ggplot2** palette.

end_color The color used for the highest value in horizon. Defaults to "blue". See argument start_color.

alpha The transparency level of the plot.

xlab A character string used as the x-axis label of the plot.

ylab A character string used as the y-axis label of the plot.

title A character string used as the title of the plot.

subtitle A character string used as the subtitle of the plot.

legend.title A character string used as the legend title of the plot.

legend.position

Where to put the legend. See ?theme for more details.

gg_theme A **ggplot2** theme which is applied to the plot.

facet_args A named list of arguments that are passed to the facet_wrap function call when creating a plot separated by groups. Ignored if group=NULL. Any argument except the facets argument of the facet_wrap function can be used. For example, if the user wants to allow free y-scales, this argument could be set to list(scales="free_y").

panel_border Whether to draw a border around the heatmap or not. Is set to FALSE by default to mimic standard heatmaps.

axis_dist The distance of the axis ticks to the colored heatmap. Is set to 0 by default to mimic standard heatmaps.

border_color The color of the individual rectangles borders. Defaults to "white".

border_size The size of the individual rectangles borders. Defaults to 0.5.

numbers Whether to put the numbers of the average estimated probabilities into the rectangles or not. Defaults to TRUE.

number_color The color of the numbers inside the rectangles. Ignored if numbers=FALSE.

number_size The size of the numbers inside the rectangles. Ignored if numbers=FALSE.

number_family The font family of the numbers inside the rectangles. Ignored if numbers=FALSE.

number_fontface

The fontface of the numbers inside the rectangles. Ignored if numbers=FALSE.

number_digits The amount of digits the numbers inside the rectangles should be rounded to. Ignored if numbers=FALSE.

... Further arguments passed to curve_cont.

Details

Heatmaps are a great tool to visualize a three dimensional surface in a two-dimensional plot. Continuously changing colors over a single area can, however, be difficult to interpret correctly if the color does not change a lot. This version makes the heatmap easier to understand by discretizing the space into equally spaced rectangles. The survival or failure probability is still estimated at a very fine grid of points in time (controlled using the `fixed_t` and `horizon` arguments), but is then aggregated into average probabilities afterwards. The number inside each rectangle then shows the *average* probability inside the region defined by the rectangle. This makes the plot look a lot like a correlation matrix.

The dimensions of the plot can be controlled using the `n_col` and `n_row` arguments. Using high numbers in these parameters makes the plot look more similar to a standard `plot_surv_heatmap` plot. It is recommended to create the `plot_surv_heatmap` plot first to pick appropriate dimensions for the discretized version here.

Value

Returns a `ggplot2` object.

Author(s)

Robin Denz

References

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(splines)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on survival using defaults
plot_surv_matrix(time="futime",
                 status="status",
                 variable="age",
                 data=nafld1,
                 model=model)

# plot it only for 60 to 80 year old people
plot_surv_matrix(time="futime",
                 status="status",
                 variable="age",
                 data=nafld1,
                 model=model,
                 horizon=seq(60, 80, 0.5))

## showing non-linear effects

# fit cox-model with bmi modelled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival using defaults
plot_surv_matrix(time="futime",
                 status="status",
                 variable="bmi",
                 data=nafld1,
                 model=model2)
```

plot_surv_quantiles *Plot Survival Time Quantiles as a Function of a Continuous Variable*

Description

Using a previously fit time-to-event model, this function plots one or multiple survival time quantiles (such as the median survival time) as a function of a continuous variable of interest.

Usage

```
plot_surv_quantiles(time, status, variable, group=NULL,
                    data, model, na.action=options()$na.action,
                    p=0.5, horizon=NULL,
                    size=1, linetype="solid", alpha=1,
                    custom_colors=NULL, single_color=NULL,
                    xlab=variable, ylab="Survival Time Quantile",
                    title=NULL, subtitle=NULL,
                    legend.title="p", legend.position="right",
                    gg_theme=ggplot2::theme_bw(),
                    facet_args=list(), ...)
```

Arguments

`time`	A single character string specifying the time-to-event variable. Needs to be a valid column name of a numeric variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a numeric or logical variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`p` A numeric vector containing the survival time quantiles of interest. For example, if the user is interested in plotting only the median survival time `p=0.5` should be used. When multiple values are supplied, one curve is drawn for each quantile.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 100 equally spaced steps.

`size` A single number specifying how thick the lines should be drawn.

`linetype` The linetype of the drawn lines. See documentation of **ggplot2** for more details on allowed values.

`alpha` The transparency level of the lines.

`custom_colors` An optional character vector specifying the colors that should be used when multiple quantiles were supplied to the `p` argument. To set the whole plot or a single curve to one color only, use the `single_color` argument instead.

`single_color` A single character string specifying the color of all drawn lines.

`xlab` A character string used as the x-axis label of the plot.

`ylab` A character string used as the y-axis label of the plot.

`title` A character string used as the title of the plot.

`subtitle` A character string used as the subtitle of the plot.

`legend.title` A character string used as the legend title of the plot.

`legend.position`
Where to put the legend. See `?theme` for more details.

`gg_theme` A **ggplot2** theme which is applied to the plot.

`facet_args` A named list of arguments that are passed to the `facet_wrap` function call when creating a plot separated by groups. Ignored if `group=NULL`. Any argument except the `facets` argument of the `facet_wrap` function can be used. For example, if the user wants to allow free y-scales, this argument could be set to `list(scales="free_y")`.

`...` Further arguments passed to `curve_cont`.

Details

Survival Time Quantiles are a single value summarizing the entire survival curve. For example, the most prominently used survival time quantile is the median survival time, which can be interpreted as the time at which half of the people in the sample are expected to have experienced the event of interest. This plot shows one or more of these quantiles as a function of a continuous variable of interest.

To calculate the survival time quantiles, it first calls the `curve_cont` function to get estimates of the value-specific survival curves. Afterwards, it uses step function interpolation to read off the survival time quantile from the estimates.

Although this is a simple way to plot the effect of a continuous covariate on the survival, it can give a misleading visualization of the relationship in some situations. Plots that do not use summary statistics, such as the `plot_surv_contour` and `plot_surv_area` plots, may be preferable.

Value

Returns a `ggplot2` object.

Author(s)

Robin Denz

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(splines)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on the median survival time
plot_surv_quantiles(time="futime",
                    status="status",
                    variable="age",
                    data=nafld1,
                    model=model)

# plot multiple survival time quantiles
plot_surv_quantiles(time="futime",
                    status="status",
                    variable="age",
                    data=nafld1,
                    model=model,
                    p=c(0.1, 0.25, 0.5, 0.75, 0.9))

## showing non-linear effects

# fit cox-model with bmi modelled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival
plot_surv_quantiles(time="futime",
                    status="status",
                    variable="bmi",
                    data=nafld1,
                    model=model2,
                    p=c(0.1, 0.25, 0.5, 0.75, 0.9))
```

`plot_surv_rmst` *Plot the Effect of a Continuous Variable on the Restricted Mean Survival Time*

Description

Using a previously fit time-to-event model, this function plots the restricted mean survival time (RMST) as a function of a continuous variable.

Usage

```
plot_surv_rmst(time, status, variable, group=NULL,
               data, model, conf_int=FALSE, conf_level=0.95,
               n_boot=300, na.action=options()$na.action,
               tau, horizon=NULL, custom_colors=NULL,
               size=1, linetype="solid", alpha=1, color="black",
               xlab=variable, ylab="Restricted Mean Survival Time",
               title=NULL, subtitle=NULL,
               legend.title="tau", legend.position="right",
               gg_theme=ggplot2::theme_bw(),
               facet_args=list(), ci_alpha=0.4, ...)
```

Arguments

`time` A single character string specifying the time-to-event variable. Needs to be a valid column name of a variable in `data`.

`status` A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a variable in `data`.

`variable` A single character string specifying the continuous variable of interest, for which the survival curves should be estimated. This variable has to be contained in the `data.frame` that is supplied to the `data` argument.

`group` An optional single character string specifying a factor variable in `data`. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See `curve_cont` for a detailed description of the estimation strategy. Set to `NULL` (default) to use no grouping variable.

`data` A `data.frame` containing all required variables.

`model` A model describing the time-to-event process (such as an `coxph` model). Needs to include `variable` as an independent variable. It also has to have an associated `predictRisk` method. See `?predictRisk` for more details.

`conf_int` Whether to plot point-wise bootstrap confidence intervals or not.

`conf_level` A number specifying the confidence level of the bootstrap confidence intervals. Ignored if `conf_int=FALSE`.

`n_boot` A single integer specifying how many bootstrap repetitions should be performed. Ignored if `conf_int=FALSE`.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`tau` The point in time to which the RMST should be calculated. Can be a vector of numbers. If multiple values are supplied, one curve is drawn for each of them.

`horizon` A numeric vector containing a range of values of `variable` for which the survival curves should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 100 equally spaced steps.

`custom_colors` An optional character vector of colors to use when there are multiple values in `tau`. Ignored if `length(tau)==1`, in which case the `color` argument can be used to specify the color of the single line.

`size` A single number specifying how thick the lines should be drawn.

`linetype` The linetype of the drawn lines. See documentation of **ggplot2** for more details on allowed values.

`alpha` The transparency level of the lines.

`color` The color of the curve if `tau` is a single value. If a numeric vector was supplied to `tau`, use the `custom_colors` argument to specify them instead.

`xlab` A character string used as the x-axis label of the plot.

`ylab` A character string used as the y-axis label of the plot.

`title` A character string used as the title of the plot.

`subtitle` A character string used as the subtitle of the plot.

`legend.title` A character string used as the legend title of the plot.

`legend.position` Where to put the legend. See `?theme` for more details.

`gg_theme` A **ggplot2** theme which is applied to the plot.

`facet_args` A named list of arguments that are passed to the `facet_wrap` function call when creating a plot separated by groups. Ignored if `group=NULL`. Any argument except the `facets` argument of the `facet_wrap` function can be used. For example, if the user wants to allow free y-scales, this argument could be set to `list(scales="free_y")`.

`ci_alpha` A single number defining the transparency level of the confidence interval bands.

`...` Further arguments passed to `curve_cont`.

Details

Similar to the `plot_surv_at_t` and `plot_surv_quantiles` plots, this function produces a plot of a survival curve summary statistic as a function of a continuous variable. The summary statistic in question is the restricted mean survival time (RMST), which is the area under a survival curve up to a specific point in time `tau`. It can be interpreted as the mean survival time of the population up to `tau`.

First, a range of value-specific survival curves are estimated. The RMST is then calculated for each one using step function interpolation. This only works for survival curves, not for CIFs. If the area under the CIF should be used instead, the `plot_surv_rmtl` function can be used.

An advantage of this method over the `plot_surv_at_t` and `plot_surv_quantiles` function is, that it does take the whole survival curve into account, kind of. Although it is the area under it, it is only calculated up to `tau` which makes the output dependent on the choice of `tau`. This, again, can result in deceiving plots in some cases. Plots visualizing the entire survival curves such as `plot_surv_contour` and `plot_surv_area` might be preferable.

Value

Returns a `ggplot2` object.

Author(s)

Robin Denz

References

Eng, K. H.; Schiller, E. & Morrell, K. On Representing the Prognostic Value of Continuous Gene Expression Biomarkers with the Restricted Mean Survival Curve. In: Oncotarget, 2015, 6, 36308-36318

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(splines)

# using data from the survival package
data(nafld, package="survival")

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on the RMST for ages 50 to 80
plot_surv_rmst(time="futime",
               status="status",
               variable="age",
               data=nafld1,
               model=model,
               horizon=seq(50, 80, 1),
               tau=2500)

# plot RMST for multiple tau values for ages 50 to 80
plot_surv_rmst(time="futime",
               status="status",
               variable="age",
               data=nafld1,
               model=model,
               horizon=seq(50, 80, 1),
               tau=c(2000, 3000, 5000))

## showing non-linear effects

# fit cox-model with bmi modeled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival
plot_surv_rmst(time="futime",
               status="status",
               variable="bmi",
               data=nafld1,
               model=model2,
               tau=c(2000, 3000, 5000))
```

plot_surv_rmtl *Plot the Effect of a Continuous Variable on the Restricted Mean Time Lost*

Description

Using a previously fit time-to-event model, this function plots the restricted mean time lost (RMTL) as a function of a continuous variable. Unlike the plot_surv_rmst function, this function can deal with multiple event types in the status variable.

Usage

```
plot_surv_rmtl(time, status, variable, group=NULL, cause=1,
               data, model, conf_int=FALSE, conf_level=0.95,
               n_boot=300, na.action=options()$na.action,
               tau, horizon=NULL, custom_colors=NULL,
               size=1, linetype="solid", alpha=1, color="black",
               xlab=variable, ylab="Restricted Mean Time Lost",
               title=NULL, subtitle=NULL,
               legend.title="tau", legend.position="right",
               gg_theme=ggplot2::theme_bw(),
               facet_args=list(), ci_alpha=0.4, ...)
```

Arguments

time	A single character string specifying the time-to-event variable. Needs to be a valid column name of a variable in data.
status	A single character string specifying the status variable, indicating if a person has experienced an event or not. Needs to be a valid column name of a variable in data.
variable	A single character string specifying the continuous variable of interest, for which the CIFs should be estimated. This variable has to be contained in the data.frame that is supplied to the data argument.
group	An optional single character string specifying a factor variable in data. When used, the plot is created conditional on this factor variable, meaning that a facetted plot is produced with one facet for each level of the factor variable. See curve_cont for a detailed description of the estimation strategy. Set to NULL (default) to use no grouping variable.
cause	The cause of interest. In standard survival data with only one event type, this should be kept at 1. For data with multiple failure types, this argument should be specified.
data	A data.frame containing all required variables.
model	A model describing the time-to-event process (such as an coxph model). Needs to include variable as an independent variable. It also has to have an associated predictRisk method. See ?predictRisk for more details.

`conf_int` Whether to plot point-wise bootstrap confidence intervals or not.

`conf_level` A number specifying the confidence level of the bootstrap confidence intervals. Ignored if `conf_int=FALSE`.

`n_boot` A single integer specifying how many bootstrap repetitions should be performed. Ignored if `conf_int=FALSE`.

`na.action` How missing values should be handled. Can be one of: `na.fail`, `na.omit`, `na.pass`, `na.exclude` or a user-defined custom function. Also accepts strings of the function names. See `?na.action` for more details. By default it uses the na.action which is set in the global options by the respective user.

`tau` The point in time to which the RMTL should be calculated. Can be a vector of numbers. If multiple values are supplied, one curve is drawn for each of them.

`horizon` A numeric vector containing a range of values of `variable` for which the CIFs should be calculated or `NULL` (default). If `NULL`, the horizon is constructed as a sequence from the lowest to the highest value observed in `variable` with 100 equally spaced steps.

`custom_colors` An optional character vector of colors to use when there are multiple values in `tau`. Ignored if `length(tau)==1`, in which case the `color` argument can be used to specify the color of the single line.

`size` A single number specifying how thick the lines should be drawn.

`linetype` The linetype of the drawn lines. See documentation of **ggplot2** for more details on allowed values.

`alpha` The transparency level of the lines.

`color` The color of the curve if `tau` is a single value. If a numeric vector was supplied to `tau`, use the `custom_colors` argument to specify them instead.

`xlab` A character string used as the x-axis label of the plot.

`ylab` A character string used as the y-axis label of the plot.

`title` A character string used as the title of the plot.

`subtitle` A character string used as the subtitle of the plot.

`legend.title` A character string used as the legend title of the plot.

`legend.position`
Where to put the legend. See `?theme` for more details.

`gg_theme` A **ggplot2** theme which is applied to the plot.

`facet_args` A named list of arguments that are passed to the `facet_wrap` function call when creating a plot separated by groups. Ignored if `group=NULL`. Any argument except the `facets` argument of the `facet_wrap` function can be used. For example, if the user wants to allow free y-scales, this argument could be set to `list(scales="free_y")`.

`ci_alpha` A single number defining the transparency level of the confidence interval bands.

`...` Further arguments passed to `curve_cont`.

Details

This function is essentially equal to the plot_surv_rmtl function. The only difference is that instead of using the restricted mean survival time (RMST), which is the area under the survival curve up to a specific point in time tau, this function uses the restricted mean time lost (RMTL), which is the area under the (cause-specific) CIF up to a specific point in time tau. It can be interpreted as the mean time it takes an individual to succumb to the event of interest before tau.

The reason this is split into two functions is, that the RMTL can be calculated when mutually exclusive competing events are present, while the RMST cannot. The basic pros and cons of the plot_surv_rmst function still apply here, however. For more information we suggest consulting the documentation page of the plot_surv_rmst function.

Value

Returns a ggplot2 object.

Author(s)

Robin Denz

References

Eng, K. H.; Schiller, E. & Morrell, K. On Representing the Prognostic Value of Continuous Gene Expression Biomarkers with the Restricted Mean Survival Curve. In: Oncotarget, 2015, 6, 36308-36318

Robin Denz, Nina Timmesfeld (2023). "Visualizing the (Causal) Effect of a Continuous Variable on a Time-To-Event Outcome". In: Epidemiology 34.5

Examples

```
library(contsurvplot)
library(riskRegression)
library(survival)
library(ggplot2)
library(splines)

# using data from the survival package
data(nafld, package=”survival”)

# take a random sample to keep example fast
set.seed(42)
nafld1 <- nafld1[sample(nrow(nafld1), 150), ]

# fit cox-model with age
model <- coxph(Surv(futime, status) ~ age, data=nafld1, x=TRUE)

# plot effect of age on the RMST for ages 50 to 80
plot_surv_rmtl(time=”futime”,
               status=”status”,
               variable=”age”,
               data=nafld1,
               model=model,
               horizon=seq(50, 80, 1),
               tau=2500)

# plot RMST for multiple tau values for ages 50 to 80
plot_surv_rmtl(time="futime",
               status="status",
               variable="age",
               data=nafld1,
               model=model,
               horizon=seq(50, 80, 1),
               tau=c(2000, 3000, 5000))

## showing non-linear effects

# fit cox-model with bmi modeled using B-Splines,
# adjusting for age and sex
model2 <- coxph(Surv(futime, status) ~ age + male + bs(bmi, df=3),
                data=nafld1, x=TRUE)

# plot effect of bmi on survival
plot_surv_rmtl(time="futime",
               status="status",
               variable="bmi",
               data=nafld1,
               model=model2,
               tau=c(2000, 3000, 5000))
```

Index