

Package ‘survminer’

March 21, 2017

Type Package

Title Drawing Survival Curves using 'ggplot2'

Version 0.3.1

Date 2017-03-21

Description Contains the function 'ggsurvplot()' for drawing easily beautiful and 'ready-to-publish' survival curves with the 'number at risk' table and 'censoring count plot'. Other functions are also available to plot adjusted curves for `Cox` model and to visually examine 'Cox' model assumptions.

License GPL-2

LazyData TRUE

Depends ggplot2, ggpubr

Imports grid, gridExtra(>= 2.0), magrittr, maxstat, methods, scales, survival, stats, broom, dplyr, tidyr, survMisc, cmprsk

Suggests knitr, KMsurv

VignetteBuilder knitr

URL <http://www.sthda.com/english/rpkgs/survminer/>

BugReports <https://github.com/kassambara/survminer/issues>

RoxygenNote 6.0.1

Collate 'BMT.R' 'BRCAOV.survInfo.R' 'add_ggsurvplot.R' 'utilities.R' 'surv_summary.R' 'theme_classic2.R' 'ggsurvplot.R' 'arrange_ggsurvplots.R' 'ggcompetingrisks.R' 'ggcoxadjustedcurves.R' 'ggcoxdiagnostics.R' 'ggcoxfunctional.R' 'ggcoxzph.R' 'ggcumcensor.R' 'ggcumevents.R' 'ggforest.R' 'ggrisktable.R' 'ggsurvevents.R' 'ggsurvtheme.R' 'myeloma.R' 'pairwise_survdiff.R' 'surv_cutpoint.R'

NeedsCompilation no

Author Alboukadel Kassambara [aut, cre],
Marcin Kosinski [aut],
Przemysław Biecek [ctb]

Maintainer Alboukadel Kassambara <alboukadel.kassambara@gmail.com>
Repository CRAN
Date/Publication 2017-03-21 17:33:15 UTC

R topics documented:

add_ggsurvplot	2
arrange_ggsurvplots	3
BMT	5
BRCAOV.survInfo	6
ggcompetingrisks	7
ggcoxadjustedcurves	8
ggcoxdiagnostics	9
ggcoxfunctional	11
ggcoxzph	13
ggcumcensor	14
ggcumevents	16
ggforest	17
ggrisktable	18
ggsurvevents	20
ggsurvplot	21
myeloma	28
pairwise_survdif	29
surv_cutpoint	30
surv_summary	32
theme_classic2	33
theme_survminer	34
Index	36

add_ggsurvplot	<i>Add Components to a ggsurvplot</i>
----------------	---------------------------------------

Description

Allows to add ggplot components - theme(), labs(), ... - to an object of class ggsurv, which is a list of ggplots.

Usage

```
## S3 method for class 'ggsurv'  
e1 + e2  
  
e1 %+% e2
```

Arguments

- e1 an object of class ggsurv.
e2 a plot component such as theme and labs.

See Also

[theme_survminer](#) and [ggsurvplot](#)

Examples

```
# Fit survival curves
require("survival")
fit<- survfit(Surv(time, status) ~ sex, data = lung)

# Basic survival curves
p <- ggsurvplot(fit, data = lung, risk.table = TRUE,
  main = "Survival curve",
  submain = "Based on Kaplan-Meier estimates",
  caption = "created with survminer"
)
p

# Customizing the plots
p + theme_survminer(
  font.main = c(16, "bold", "darkblue"),
  font.submain = c(15, "bold.italic", "purple"),
  font.caption = c(14, "plain", "orange"),
  font.x = c(14, "bold.italic", "red"),
  font.y = c(14, "bold.italic", "darkred"),
  font.tickslab = c(12, "plain", "darkgreen")
)
```

arrange_ggsurvplots *Arranging Multiple ggsurvplots*

Description

Arranging multiple ggsurvplots on the same page.

Usage

```
arrange_ggsurvplots(x, print = TRUE, title = NA, ncol = 2, nrow = 1,
  surv.plot.height = NULL, risk.table.height = NULL,
  ncensor.plot.height = NULL, ...)
```

Arguments

<code>x</code>	a list of ggsurvplots.
<code>print</code>	logical value. If TRUE, the arranged plots are displayed.
<code>title</code>	character vector specifying page title. Default is NA.
<code>ncol, nrow</code>	the number of columns and rows, respectively.
<code>surv.plot.height</code>	the height of the survival plot on the grid. Default is 0.75. Ignored when <code>risk.table = FALSE</code> . <code>1-risk.table.height -ncensor.plot.height</code> when <code>risk.table = TRUE</code> and <code>ncensor.plot = TRUE</code>
<code>risk.table.height</code>	the height of the risk table on the grid. Increase the value when you have many strata. Default is 0.25. Ignored when <code>risk.table = FALSE</code> .
<code>ncensor.plot.height</code>	The height of the censor plot. Used when <code>ncensor.plot = TRUE</code> .
<code>...</code>	not used

Value

returns an invisible object of class `arrangelist` (see [marrangeGrob](#)), which can be saved into a pdf file using the function [ggsave](#).

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
# Fit survival curves
require("survival")
fit<- survfit(Surv(time, status) ~ sex, data = lung)

# List of ggsurvplots
require("survminer")
splots <- list()
splots[[1]] <- ggsurvplot(fit, data = lung, risk.table = TRUE, ggtheme = theme_minimal())
splots[[2]] <- ggsurvplot(fit, data = lung, risk.table = TRUE, ggtheme = theme_grey())

# Arrange multiple ggsurvplots and print the output
arrange_ggsurvplots(splots, print = TRUE,
  ncol = 2, nrow = 1, risk.table.height = 0.4)

## Not run:
# Arrange and save into pdf file
res <- arrange_ggsurvplots(splots, print = FALSE)
ggsave("myfile.pdf", res)

## End(Not run)
```

BMT

*Bone Marrow Transplant***Description**

Bone marrow transplant data from L. Scrucca et al., Bone Marrow Transplantation (2007). Data from 35 patients with acute leukaemia who underwent HSCT. Used for competing risk analysis.

Usage

```
data("BMT")
```

Format

A data frame with 35 rows and 3 columns.

- dis: disease; 0 = ALL; 1 = AML
- ftime: follow-up time
- status: 0 = censored (survival); 1 = Transplant-related mortality; 2 = relapse

References

Scrucca L, Santucci A, Aversa F. Competing risk analysis using R: an easy guide for clinicians. Bone Marrow Transplant. 2007 Aug;40(4):381-7.

Examples

```
data(BMT)

# Data preparation
#+++++
# Label diseases
BMT$dis <- factor(BMT$dis, levels = c(0,1),
  labels = c("ALL", "AML"))
# Label status
BMT$status <- factor(BMT$status, levels = c(0,1,2),
  labels = c("Censored", "Mortality", "Relapse"))

# Cumulative Incidence Function
# +++++
fit <- cmprsk::cuminc(
  ftime = BMT$ftime,      # Failure time variable
  fstatus = BMT$status,   # Codes for different causes of failure
  group = BMT$dis         # Estimates will be calculated within groups
)

# Visualize
```

```
# ++++++
ggcompetingrisks(fit)
ggcompetingrisks(fit, multiple_panels = FALSE,
  legend = "right")
```

BRCAOV.survInfo

Breast and Ovarian Cancers Survival Information

Description

Breast and Ovarian cancers survival information from the RTCGA.clinical R/Bioconductor package. <http://rtcg.github.io/RTCGA/>.

Usage

```
data("BRCAOV.survInfo")
```

Format

A data frame with 1674 rows and 4 columns.

- times: follow-up time;
- bcr_patient_barcode: Patient bar code;
- patient.vital_status = survival status. 0 = alive, 1 = dead;
- admin.disease_code: disease code. brca = breast cancer, ov = ovarian cancer.

Source

From the RTCGA.clinical R/Bioconductor package. The data is generated as follow:

```
# Installing RTCGA.clinical
source("https://bioconductor.org/biocLite.R")
biocLite("RTCGA.clinical")

# Generating the BRCAOV survival information
library(RTCGA.clinical)
survivalTCGA(BRCA.clinical, OV.clinical,
  extract.cols = "admin.disease_code") -> BRCAOV.survInfo
```

Examples

```
data(BRCAOV.survInfo)
library(survival)
fit <- survfit(Surv(times, patient.vital_status) ~ admin.disease_code,
  data = BRCAOV.survInfo)
ggsurvplot(fit, data = BRCAOV.survInfo, risk.table = TRUE)
```

Description

This function plots Cumulative Incidence Curves. For `cuminc` objects it's a `ggplot2` version of `plot.cuminc`. For `survfitms` objects a different geometry is used, as suggested by @teigentler.

Usage

```
ggcompetingrisks(fit, gnames = NULL, gsep = " ", multiple_panels = TRUE,
  ggtheme = theme_survminer(), ...)
```

Arguments

<code>fit</code>	an object of a class <code>cuminc</code> - created with <code>cuminc</code> function or <code>survfitms</code> created with <code>survfit</code> function.
<code>gnames</code>	a vector with group names. If not supplied then will be extracted from <code>fit</code> object (<code>cuminc</code> only).
<code>gsep</code>	a separator that extracts group names and event names from <code>gnames</code> object (<code>cuminc</code> only).
<code>multiple_panels</code>	if <code>TRUE</code> then groups will be plotted in different panels (<code>cuminc</code> only).
<code>ggtheme</code>	function, <code>ggplot2</code> theme name. Default value is <code>theme_survminer</code> . Allowed values include <code>ggplot2</code> official themes: see theme .
<code>...</code>	further arguments passed to the function <code>ggpar</code> for customizing the plot.

Value

Returns an object of class `gg`.

Author(s)

Przemysław Biecek, <przemyslaw.biecek@gmail.com>

Examples

```
set.seed(2)
ss <- rexp(100)
gg <- factor(sample(1:3,100,replace=TRUE),1:3,c('BRCA','LUNG','OV'))
cc <- factor(sample(0:2,100,replace=TRUE),0:2,c('no event','death','progression'))
strt <- sample(1:2,100,replace=TRUE)

# handles cuminc objects
print(fit <- cmprsk::cuminc(ss,cc,gg,strt))
ggcompetingrisks(fit)
ggcompetingrisks(fit, multiple_panels = FALSE)
```

```
# handles survfitms objects
library(survival)
df <- data.frame(time = ss, group = gg, status = cc, strt)
fit2 <- survfit(Surv(time, status, type="mstate") ~ 1, data=df)
ggcompetingrisks(fit2)
fit3 <- survfit(Surv(time, status, type="mstate") ~ group, data=df)
ggcompetingrisks(fit3)
## Not run:
  library(ggsci)
  library(cowplot)
  ggcompetingrisks(fit3) + theme_cowplot() + scale_fill_jco()

## End(Not run)
```

ggcoxadjustedcurves *Adjusted Survival Curves for Cox Proportional Hazards Model*

Description

This function plots adjusted survival curves for coxph model. The idea behind this function is described in <https://cran.r-project.org/web/packages/survival/vignettes/adjcurve.pdf>. For every observation in the dataset a prediction for survival curve is made. Then the predictions are averaged with respect to a selected variable.

Usage

```
ggcoxadjustedcurves(fit, variable = NULL, individual.curves = FALSE,
  data = NULL, palette = "hue", curve.size = 2, curve.alpha = 0.2,
  ylab = "Survival rate", ggtheme = theme_survminer(), ...)
```

Arguments

fit	an object of class <code>coxph.object</code> - created with <code>coxph</code> function.
variable	a variable (vector) with values corresponding to groups to be plotted
individual.curves	if TRUE then all individual predicted survival curves will be plotted
data	a dataset for predictions. If not supplied then data will be extracted from 'fit' object.
palette	the color palette to be used for coloring or filling by groups. Allowed values include "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; or custom color palette e.g. c("blue", "red"); and scientific journal palettes from ggsci R package, e.g.: "npg", "aaas", "lancet", "jco", "ucscgb", "uchicago", "simpsons" and "rickandmarty".
curve.size, curve.alpha	size and alpha for individual survival curves
ylab	y axis label.

ggtheme function, ggplot2 theme name. Default value is [theme_classic2](#). Allowed values include ggplot2 official themes: see [theme](#).

... further arguments passed to the function [ggpar](#) for customizing the plot.

Value

Returns an object of class gg.

Author(s)

Przemyslaw Biecek, <przemyslaw.biecek@gmail.com>

Examples

```
library(survival)
fit2 <- coxph( Surv(stop, event) ~ rx + size, data = bladder )
ggcoxadjustedcurves(fit2, data = bladder)
ggcoxadjustedcurves(fit2, individual.curves = TRUE, data = bladder, curve.alpha=0.01)
ggcoxadjustedcurves(fit2, data = bladder, variable= bladder[, "rx"])
ggcoxadjustedcurves(fit2, data = bladder, variable= bladder[, "rx"],
  individual.curves=TRUE, curve.alpha=0.01)
```

ggcoxdiagnostics

Diagnostic Plots for Cox Proportional Hazards Model with ggplot2

Description

Displays diagnostics graphs presenting goodness of Cox Proportional Hazards Model fit, that can be calculated with [coxph](#) function.

Usage

```
ggcoxdiagnostics(fit, type = c("martingale", "deviance", "score",
  "schoenfeld", "dfbeta", "dfbetas", "scaledsch", "partial"), ...,
  linear.predictions = type %in% c("martingale", "deviance"),
  ox.scale = ifelse(linear.predictions, "linear.predictions",
    "observation.id"), hline = TRUE, sline = TRUE, sline.se = TRUE,
  hline.col = "red", hline.size = 1, hline.alpha = 1,
  hline.yintercept = 0, hline.lty = "dashed", sline.col = "blue",
  sline.size = 1, sline.alpha = 0.3, sline.lty = "dashed",
  point.col = "black", point.size = 1, point.shape = 19,
  point.alpha = 1, title = NULL, subtitle = NULL, caption = NULL,
  ggtheme = ggplot2::theme_bw())
```

Arguments

<code>fit</code>	an object of class <code>coxph.object</code> - created with <code>coxph</code> function.
<code>type</code>	the type of residuals to present on Y axis of a diagnostic plot. The same as in <code>residuals.coxph</code> : character string indicating the type of residual desired. Possible values are "martingale", "deviance", "score", "schoenfeld", "dfbeta", "dfbetas" and "scaledsch". Only enough of the string to determine a unique match is required.
<code>...</code>	further arguments passed to <code>residuals.coxph</code> or to the function <code>ggpar</code> for customizing the plot.
<code>linear.predictions</code>	(deprecated, see <code>ox.scale</code>) a logical value indicating whether to show linear predictions for observations (TRUE) or just indexed of observations (FALSE) on X axis.
<code>ox.scale</code>	one value from <code>c("linear.predictions", "observation.id", "time")</code> . It defines what will be presented on OX scale. Possible values: \hat{y} for "linear.predictions", Id of an observation for "observation.id" or Time for "time".
<code>hline</code>	a logical - should the horizontal line be added to highlight the $Y=0$ level.
<code>sline, sline.se</code>	a logical - should the smooth line be added to highlight the local average for residuals.
<code>hline.col, hline.size, hline.lty, hline.alpha, hline.yintercept</code>	color, size, linetype, visibility and Y-axis coordinate to be used for <code>geom_hline</code> . Used only when <code>hline = TRUE</code> .
<code>sline.col, sline.size, sline.lty, sline.alpha</code>	color, size, linetype and visibility to be used for <code>geom_smooth</code> . Used only when <code>sline = TRUE</code> .
<code>point.col, point.size, point.shape, point.alpha</code>	color, size, shape and visibility to be used for points.
<code>title, subtitle, caption</code>	main title, subtitle and caption.
<code>ggtheme</code>	function, ggplot2 theme name. Default value is <code>ggplot2::theme_bw()</code> . Allowed values include ggplot2 official themes: see theme .

Value

Returns an object of class `ggplot`.

Functions

- `ggcoxdiagnostics`: Diagnostic Plots for Cox Proportional Hazards Model with **ggplot2**

Author(s)

Marcin Kosinski, <m.p.kosinski@gmail.com>

Examples

```
library(survival)
coxph.fit2 <- coxph(Surv(futime, fustat) ~ age + ecog.ps, data=ovarian)
ggcoxdiagnostics(coxph.fit2, type = "deviance")

ggcoxdiagnostics(coxph.fit2, type = "schoenfeld", title = "Diagnostic plot")
ggcoxdiagnostics(coxph.fit2, type = "deviance", ox.scale = "time")
ggcoxdiagnostics(coxph.fit2, type = "schoenfeld", ox.scale = "time",
                  title = "Diagnostic plot", subtitle = "Data comes from survey XYZ",
                  font.subtitle = 9)
ggcoxdiagnostics(coxph.fit2, type = "deviance", ox.scale = "linear.predictions",
                  caption = "Code is available here - link", font.caption = 10)
ggcoxdiagnostics(coxph.fit2, type = "schoenfeld", ox.scale = "observation.id")
ggcoxdiagnostics(coxph.fit2, type = "scaledsch", ox.scale = "time")
```

ggcoxfunctional	<i>Functional Form of Continuous Variable in Cox Proportional Hazards Model</i>
-----------------	---

Description

Displays graphs of continuous explanatory variable against martingale residuals of null cox proportional hazards model, for each term in of the right side of formula. This might help to properly choose the functional form of continuous variable in cox model ([coxph](#)). Fitted lines with [lowess](#) function should be linear to satisfy cox proportional hazards model assumptions.

Usage

```
ggcoxfunctional(formula, data = NULL, fit, iter = 0, f = 0.6,
                point.col = "red", point.size = 1, point.shape = 19, point.alpha = 1,
                xlim = NULL, ylim = NULL,
                ylab = "Martingale Residuals \nof Null Cox Model", title = NULL,
                caption = NULL, ggtheme = theme_survminer(), ...)
```

```
## S3 method for class 'ggcoxfunctional'
print(x, ..., newpage = TRUE)
```

Arguments

formula	a formula object, with the response on the left of a ~ operator, and the terms on the right. The response must be a survival object as returned by the Surv function.
data	a data.frame in which to interpret the variables named in the formula,
fit	an object of class coxph.object - created with coxph function.
iter	parameter of lowess .

<code>f</code>	parameter of lowess .
<code>point.col</code> , <code>point.size</code> , <code>point.shape</code> , <code>point.alpha</code>	color, size, shape and visibility to be used for points.
<code>xlim</code> , <code>ylim</code>	x and y axis limits e.g. <code>xlim = c(0, 1000)</code> , <code>ylim = c(0, 1)</code> .
<code>ylab</code>	y axis label.
<code>title</code>	the title of the final grob (top in arrangeGrob)
<code>caption</code>	the caption of the final grob (bottom in arrangeGrob)
<code>ggtheme</code>	function, ggplot2 theme name. Default value is theme_classic2 . Allowed values include ggplot2 official themes: see theme .
<code>...</code>	further arguments passed to the function ggpar for customizing the plot.
<code>x</code>	an object of class <code>ggcoxfunctional</code>
<code>newpage</code>	open a new page. See grid.arrange .

Value

Returns an object of class `ggcoxfunctional` which is a list of ggplots.

Functions

- `ggcoxfunctional`: Functional Form of Continuous Variable in Cox Proportional Hazards Model.

Author(s)

Marcin Kosinski, <m.p.kosinski@gmail.com>

Examples

```
library(survival)
data(mgus)
res.cox <- coxph(Surv(futime, death) ~ mspike + log(mspike) + I(mspike^2) +
  age + I(log(age)^2) + I(sqrt(age)), data = mgus)
ggcoxfunctional(res.cox, data = mgus, point.col = "blue", point.alpha = 0.5)
ggcoxfunctional(res.cox, data = mgus, point.col = "blue", point.alpha = 0.5,
  title = "Pass the title", caption = "Pass the caption")
```

Description

Displays a graph of the scaled Schoenfeld residuals, along with a smooth curve using **ggplot2**. Wrapper around [plot.cox.zph](#).

Usage

```
ggcoxzph(fit, resid = TRUE, se = TRUE, df = 4, nsmo = 40, var,
  point.col = "red", point.size = 1, point.shape = 19, point.alpha = 1,
  caption = NULL, ggtheme = theme_survminer(), ...)
```

```
## S3 method for class 'ggcoxzph'
print(x, ..., newpage = TRUE)
```

Arguments

<code>fit</code>	an object of class cox.zph .
<code>resid</code>	a logical value, if TRUE the residuals are included on the plot, as well as the smooth fit.
<code>se</code>	a logical value, if TRUE, confidence bands at two standard errors will be added.
<code>df</code>	the degrees of freedom for the fitted natural spline, <code>df=2</code> leads to a linear fit.
<code>nsmo</code>	number of points used to plot the fitted spline.
<code>var</code>	the set of variables for which plots are desired. By default, plots are produced in turn for each variable of a model.
<code>point.col</code> , <code>point.size</code> , <code>point.shape</code> , <code>point.alpha</code>	color, size, shape and visibility to be used for points.
<code>caption</code>	the caption of the final grob (bottom in arrangeGrob)
<code>ggtheme</code>	function, ggplot2 theme name. Default value is theme_classic2 . Allowed values include ggplot2 official themes: see theme .
<code>...</code>	further arguments passed to either the <code>print()</code> function or to the ggpar function for customizing the plot (see Details section).
<code>x</code>	an object of class <code>ggcoxzph</code>
<code>newpage</code>	open a new page. See grid.arrange .

Details

Customizing the plots: The plot can be easily customized using additional arguments to be passed to the function `ggpar()`. Read `?ggpubr::ggpar`. These arguments include `font.main`, `font.submain`, `font.caption`, `font.x`, `font.y`, `font`, a vector of length 3 indicating respectively the size (e.g.: 14), the style (e.g.: "plain", "bold", "italic", "bold.italic") and the color (e.g.: "red") of main title, subtitle, caption, xlab and ylab and axis tick labels, respectively. For example `font.x = c(14, "bold", "red")`. Use `font.x = 14`, to change only font size; or use `font.x = "bold"`, to change only font face.

Value

Returns an object of class `ggcoxzph` which is a list of `ggplots`.

Functions

- `ggcoxzph`: Graphical Test of Proportional Hazards using `ggplot2`.

Author(s)

Marcin Kosinski, <m.p.kosinski@gmail.com>

Examples

```
library(survival)
fit <- coxph(Surv(futime, fustat) ~ age + ecog.ps + rx, data=ovarian)
cox.zph.fit <- cox.zph(fit)
# plot all variables
ggcoxzph(cox.zph.fit)
# plot all variables in specified order
ggcoxzph(cox.zph.fit, var = c("ecog.ps", "rx", "age"), font.main = 12)
# plot specified variables in specified order
ggcoxzph(cox.zph.fit, var = c("ecog.ps", "rx"), font.main = 12, caption = "Caption goes here")
```

ggcumcensor

Cumulative Number of Censored Subjects

Description

Plot the cumulative number of censored subjects, the number of subjects who exit the risk set, without an event, at time t . Normally, users don't need to use this function directly. Internally used by the function [ggsurvplot](#).

Usage

```
ggcumcensor(fit, data = NULL, color = "black", palette = NULL,
  break.time.by = NULL, xlim = NULL, title = "Number of censored subjects",
  xlab = "Time", ylab = "Strata", legend = "top",
  legend.title = "Strata", legend.labs = NULL, y.text = TRUE,
  y.text.col = TRUE, fontsize = 4.5, ggtheme = theme_survminer(), ...)
```

Arguments

<code>fit</code>	an object of class <code>survfit</code> .
<code>data</code>	a dataset used to fit survival curves. If not supplied then data will be extracted from 'fit' object.

color	color to be used for the survival curves. This argument is ignored when the number of strata (groups > 1). In this case, use the argument palette.
palette	the color palette to be used. Allowed values include "hue" for the default hue color scale; "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; or custom color palette e.g. c("blue", "red"). See details section for more information.
break.time.by	numeric value controlling time axis breaks. Default value is NULL.
xlim	x and y axis limits e.g. xlim = c(0, 1000), ylim = c(0, 1).
title	the title of the plot.
xlab	main title and axis labels
ylab	main title and axis labels
legend	character specifying legend position. Allowed values are one of c("top", "bottom", "left", "right", "none"). Default is "top" side position. to remove the legend use legend = "none". Legend position can be also specified using a numeric vector c(x, y); see details section.
legend.title	legend title.
legend.labs	character vector specifying legend labels. Used to replace the names of the strata from the fit. Should be given in the same order as those strata.
y.text	logical. Default is TRUE. If FALSE, the table y axis. tick labels will be hidden.
y.text.col	logical. Default value is FALSE. If TRUE, the table tick labels will be colored by strata.
fontsize	text font size.
ggtheme	function, ggplot2 theme name. Default value is theme_survminer . Allowed values include ggplot2 official themes: see theme .
...	other arguments passed to the function ggpar .

Value

a ggplot.

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
# Fit survival curves
require("survival")
fit<- survfit(Surv(time, status) ~ sex, data = lung)

# Basic plot
ggcumcensor(fit, data = lung)

# Change color by strata
# Remove y tick labels
ggcumcensor(fit, data = lung, color = "strata",
```

```
y.text = FALSE)
```

ggcumevents

Cumulative Number of Events Table

Description

Cumulative number of events table. Normally, users don't need to use this function directly. Internally used by the function [ggsurvplot](#).

Usage

```
ggcumevents(fit, data = NULL, color = "black", palette = NULL,
  break.time.by = NULL, xlim = NULL,
  title = "Cumulative number of events", xlab = "Time", ylab = "Strata",
  legend = "top", legend.title = "Strata", legend.labs = NULL,
  y.text = TRUE, y.text.col = TRUE, fontsize = 4.5,
  ggtheme = theme_survminer(), ...)
```

Arguments

fit	an object of class <code>survfit</code> .
data	a dataset used to fit survival curves. If not supplied then data will be extracted from 'fit' object.
color	color to be used for the survival curves. This argument is ignored when the number of strata (groups > 1). In this case, use the argument <code>palette</code> .
palette	the color palette to be used. Allowed values include "hue" for the default hue color scale; "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; or custom color palette e.g. <code>c("blue", "red")</code> . See details section for more information.
break.time.by	numeric value controlling time axis breaks. Default value is <code>NULL</code> .
xlim	x and y axis limits e.g. <code>xlim = c(0, 1000)</code> , <code>ylim = c(0, 1)</code> .
title	the title of the plot.
xlab	main title and axis labels
ylab	main title and axis labels
legend	character specifying legend position. Allowed values are one of <code>c("top", "bottom", "left", "right", "none")</code> . Default is "top" side position. to remove the legend use <code>legend = "none"</code> . Legend position can be also specified using a numeric vector <code>c(x, y)</code> ; see details section.
legend.title	legend title.

<code>legend.labs</code>	character vector specifying legend labels. Used to replace the names of the strata from the fit. Should be given in the same order as those strata.
<code>y.text</code>	logical. Default is TRUE. If FALSE, the table y axis. tick labels will be hidden.
<code>y.text.col</code>	logical. Default value is FALSE. If TRUE, the table tick labels will be colored by strata.
<code>fontsize</code>	text font size.
<code>ggtheme</code>	function, ggplot2 theme name. Default value is theme_survminer . Allowed values include ggplot2 official themes: see theme .
<code>...</code>	other arguments passed to the function ggpar .

Value

a ggplot.

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
# Fit survival curves
require("survival")
fit<- survfit(Surv(time, status) ~ sex, data = lung)

# Basic plot
ggcumevents(fit, data = lung)

# Change color by strata
# Remove y tick labels
ggcumevents(fit, data = lung, color = "strata",
  y.text = FALSE)
```

ggforest

Drawing Forest Plot for CoxPH model

Description

Drawing Forest Plot for CoxPH model

Usage

```
ggforest(model, alpha = 0.05, plot.title = "Forest plot for coxph model",
  ggtheme = theme_survminer(), palette = c("black", "red4"),
  xlab = "Hazard ratio", ...)
```

Arguments

<code>model</code>	an object of class <code>coxph</code> .
<code>alpha</code>	significance level for coloring.
<code>plot.title</code>	legend title.
<code>ggtheme</code>	function, <code>ggplot2</code> theme name. Default value is <code>theme_classic2</code> . Allowed values include <code>ggplot2</code> official themes: see <code>theme</code> .
<code>palette</code>	the color palette to be used for coloring of significant variables.
<code>xlab</code>	Label in OX axis.
<code>...</code>	further arguments passed to the function <code>ggpar</code> for customizing the plot.

Value

return an object of class `ggplot`

Author(s)

Przemyslaw Biecek, <przemyslaw.biecek@gmail.com>

Examples

```
require("survival")
model <- coxph( Surv(time, status) ~ sex + rx + adhere,
               data = colon )
ggforest(model)
```

<code>ggrisktable</code>	<i>Number at Risk Table</i>
--------------------------	-----------------------------

Description

Number at risk table. Normally, users don't need to use this function directly. Internally used by the function `ggsurvplot`.

Usage

```
ggrisktable(fit, data = NULL, type = c("absolute", "percentage", "abs_pct",
    "nrisk_cumcensor", "nrisk_cumevents"), color = "black", palette = NULL,
    break.time.by = NULL, xlim = NULL, title = NULL, xlab = "Time",
    ylab = "Strata", legend = "top", legend.title = "Strata",
    legend.labs = NULL, y.text = TRUE, y.text.col = TRUE, fontsize = 4.5,
    ggtheme = theme_light(), ...)
```

Arguments

<code>fit</code>	an object of class <code>survfit</code> .
<code>data</code>	a dataset used to fit survival curves. If not supplied then data will be extracted from 'fit' object.
<code>type</code>	risk table type. Allowed values include: "absolute" or "percentage": to show the absolute number and the percentage of subjects at risk by time, respectively. Use "abs_pct" to show both absolute number and percentage.
<code>color</code>	color to be used for the survival curves. This argument is ignored when the number of strata (groups > 1). In this case, use the argument <code>palette</code> .
<code>palette</code>	the color palette to be used. Allowed values include "hue" for the default hue color scale; "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; or custom color palette e.g. <code>c("blue", "red")</code> . See details section for more information.
<code>break.time.by</code>	numeric value controlling time axis breaks. Default value is <code>NULL</code> .
<code>xlim</code>	x and y axis limits e.g. <code>xlim = c(0, 1000)</code> , <code>ylim = c(0, 1)</code> .
<code>title</code>	the title of the plot.
<code>xlab</code>	main title and axis labels
<code>ylab</code>	main title and axis labels
<code>legend</code>	character specifying legend position. Allowed values are one of <code>c("top", "bottom", "left", "right", "none")</code> . Default is "top" side position. to remove the legend use <code>legend = "none"</code> . Legend position can be also specified using a numeric vector <code>c(x, y)</code> ; see details section.
<code>legend.title</code>	legend title.
<code>legend.labs</code>	character vector specifying legend labels. Used to replace the names of the strata from the fit. Should be given in the same order as those strata.
<code>y.text</code>	logical. Default is <code>TRUE</code> . If <code>FALSE</code> , the table y axis. tick labels will be hidden.
<code>y.text.col</code>	logical. Default value is <code>FALSE</code> . If <code>TRUE</code> , the table tick labels will be colored by strata.
<code>fontsize</code>	text font size.
<code>ggtheme</code>	function, ggplot2 theme name. Default value is theme_survminer . Allowed values include ggplot2 official themes: see theme .
<code>...</code>	other arguments passed to the function ggpar .

Value

a ggplot.

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
# Fit survival curves
require("survival")
fit<- survfit(Surv(time, status) ~ sex, data = lung)

# Basic plot
ggrisktable(fit, data = lung)

# Change color by strata
# Remove y tick labels
ggrisktable(fit, data = lung, color = "strata",
  y.text = FALSE)
```

ggsurvevents

Distribution of Event's Times

Description

Distribution of Event's Times

Usage

```
ggsurvevents(surv = NULL, fit = NULL, data = NULL, type = "fraction",
  normalized = TRUE, censored.on.top = TRUE, ggtheme = theme_survminer(),
  palette = c("grey75", "grey25"), ...)
```

Arguments

surv	an object of Surv . If not supplied, the censoring variable is extracted from the model.
fit	an object of class survfit .
data	a dataset for predictions. If not supplied then data will be extracted from 'fit' object.
type	one of <code>c("cumulative", "radius", "fraction")</code> . "cumulative" stands for cumulative number of events, "radius" stands for number of events within a given radius,
normalized	if TRUE relative number of events is presented,
censored.on.top	if TRUE then censored events are on the top
ggtheme	function, ggplot2 theme name. Default value is <code>theme_classic2</code> . Allowed values include ggplot2 official themes: see theme .
palette	the color palette to be used for coloring of significant variables.
...	other graphical parameters to be passed to the function ggpar .

Value

return an object of class ggplot

Author(s)

Przemyslaw Biecek, <przemyslaw.biecek@gmail.com>

Examples

```
require("survival")
# from Surv
surv <- Surv(lung$time, lung$status)
ggsurvevents(surv)

surv2 <- Surv(colon$time, colon$status)
ggsurvevents(surv2)
ggsurvevents(surv2, normalized = TRUE)

# from survfit
fit <- survfit(Surv(time, status) ~ sex, data = lung)
ggsurvevents(fit = fit, data = lung)

# from coxph
model <- coxph( Surv(time, status) ~ sex + rx + adhere, data = colon )
ggsurvevents(fit = model, data = colon)
ggsurvevents(surv2, normalized = TRUE, type = "radius")
ggsurvevents(surv2, normalized = TRUE, type = "fraction")
```

ggsurvplot

Drawing Survival Curves Using ggplot2

Description

Drawing survival curves using ggplot2

Usage

```
ggsurvplot(fit, data = NULL, fun = NULL, color = NULL, palette = NULL,
  linetype = 1, break.x.by = NULL, break.y.by = NULL,
  break.time.by = NULL, surv.scale = c("default", "percent"),
  conf.int = FALSE, conf.int.fill = "gray", conf.int.style = "ribbon",
  censor = TRUE, pval = FALSE, pval.size = 5, pval.coord = c(NULL,
  NULL), pval.method = FALSE, pval.method.size = pval.size,
  pval.method.coord = c(NULL, NULL), log.rank.weights = c("survdif", "1",
  "n", "sqrtN", "S1", "S2", "FH_p=1_q=1"), title = NULL, xlab = "Time",
  ylab = "Survival probability", xlim = NULL, ylim = NULL,
  legend = c("top", "bottom", "left", "right", "none"),
```

```

legend.title = "Strata", legend.labs = NULL, tables.height = 0.25,
tables.y.text = TRUE, tables.col = "black", risk.table = FALSE,
risk.table.pos = c("out", "in"), risk.table.title = NULL,
risk.table.col = tables.col, risk.table.fontsize = 4.5, fontsize = 4.5,
risk.table.y.text = tables.y.text, risk.table.y.text.col = TRUE,
risk.table.height = tables.height, surv.plot.height = 0.75,
ncensor.plot.height = tables.height, cumevents.height = tables.height,
cumcensor.height = tables.height, ncensor.plot = FALSE,
ncensor.plot.title = NULL, cumevents = FALSE,
cumevents.col = tables.col, cumevents.title = NULL,
cumevents.y.text = tables.y.text, cumevents.y.text.col = TRUE,
cumcensor = FALSE, cumcensor.col = tables.col, cumcensor.title = NULL,
cumcensor.y.text = tables.y.text, cumcensor.y.text.col = TRUE,
surv.median.line = c("none", "hv", "h", "v"), ggtheme = theme_survminer(),
tables.theme = ggtheme, ...)

## S3 method for class 'ggsurvplot'
print(x, surv.plot.height = NULL,
      risk.table.height = NULL, ncensor.plot.height = NULL, newpage = TRUE,
      ...)

```

Arguments

<code>fit</code>	an object of class <code>survfit</code> .
<code>data</code>	a dataset used to fit survival curves. If not supplied then data will be extracted from 'fit' object.
<code>fun</code>	an arbitrary function defining a transformation of the survival curve. Often used transformations can be specified with a character argument: "event" plots cumulative events ($f(y) = 1 - y$), "cumhaz" plots the cumulative hazard function ($f(y) = -\log(y)$), and "pct" for survival probability in percentage.
<code>color</code>	color to be used for the survival curves. This argument is ignored when the number of strata (groups > 1). In this case, use the argument <code>palette</code> .
<code>palette</code>	the color palette to be used. Allowed values include "hue" for the default hue color scale; "grey" for grey color palettes; brewer palettes e.g. "RdBu", "Blues", ...; or custom color palette e.g. <code>c("blue", "red")</code> . See details section for more information.
<code>linetype</code>	line types. Allowed values includes i) "strata" for changing linetypes by strata (i.e. groups); ii) a numeric vector (e.g., <code>c(1, 2)</code>) or a character vector <code>c("solid", "dashed")</code> .
<code>break.x.by</code>	alias of <code>break.time.by</code> . Numeric value controlling x axis breaks. Default value is NULL.
<code>break.y.by</code>	same as <code>break.x.by</code> but for y axis.
<code>break.time.by</code>	numeric value controlling time axis breaks. Default value is NULL.
<code>surv.scale</code>	scale transformation of survival curves. Allowed values are "default" or "percent".
<code>conf.int</code>	logical value. If TRUE, plots confidence interval.

<code>conf.int.fill</code>	fill color to be used for confidence interval.
<code>conf.int.style</code>	confidence interval style. Allowed values include <code>c("ribbon", "step")</code> .
<code>censor</code>	logical value. If TRUE, censors will be drawn.
<code>pval</code>	logical value. If TRUE, the p-value is added on the plot.
<code>pval.size</code>	numeric value specifying the p-value text size. Default is 5.
<code>pval.coord</code>	numeric vector, of length 2, specifying the x and y coordinates of the p-value. Default values are NULL.
<code>pval.method</code>	whether to add a text with the test name used for calculating the pvalue, that corresponds to survival curves' comparison - used only when <code>pval=TRUE</code>
<code>pval.method.size</code>	the same as <code>pval.size</code> but for displaying <code>log.rank.weights</code> name
<code>pval.method.coord</code>	the same as <code>pval.coord</code> but for displaying <code>log.rank.weights</code> name
<code>log.rank.weights</code>	The name for the type of weights to be used in computing the p-value for log-rank test. By default <code>survdifff</code> is used to calculate regular log-rank test (with <code>weights == 1</code>). A user can specify <code>"1"</code> , <code>"n"</code> , <code>"sqrtN"</code> , <code>"S1"</code> , <code>"S2"</code> , <code>"FH"</code> to use weights specified in comp , so that weight correspond to the test as : 1 - log-rank, n - Gehan-Breslow (generalized Wilcoxon), <code>sqrtN</code> - Tarone-Ware, <code>S1</code> - Peto-Peto's modified survival estimate, <code>S2</code> - modified Peto-Peto (by Andersen), <code>FH</code> - Fleming-Harrington(<code>p=1</code> , <code>q=1</code>).
<code>title, xlab, ylab</code>	main title and axis labels
<code>xlim, ylim</code>	x and y axis limits e.g. <code>xlim = c(0, 1000)</code> , <code>ylim = c(0, 1)</code> .
<code>legend</code>	character specifying legend position. Allowed values are one of <code>c("top", "bottom", "left", "right", "none")</code> . Default is "top" side position. to remove the legend use <code>legend = "none"</code> . Legend position can be also specified using a numeric vector <code>c(x, y)</code> ; see details section.
<code>legend.title</code>	legend title.
<code>legend.labs</code>	character vector specifying legend labels. Used to replace the names of the strata from the fit. Should be given in the same order as those strata.
<code>tables.height</code>	numeric value (in <code>[0 - 1]</code>) specifying the general height of all tables under the main survival plot.
<code>tables.y.text</code>	logical. Default is TRUE. If FALSE, the y axis tick labels of tables will be hidden.
<code>tables.col</code>	color to be used for all tables under the main plot. Default value is "black". If you want to color by strata (i.e. groups), use <code>tables.col = "strata"</code> .
<code>risk.table</code>	Allowed values include: <ul style="list-style-type: none"> • TRUE or FALSE specifying whether to show or not the risk table. Default is FALSE. • "absolute" or "percentage": to show the absolute number and the percentage of subjects at risk by time, respectively. Use i) <code>"abs_pct"</code> to show both absolute number and percentage. ii) <code>"nrisk_cumcensor"</code> and <code>"nrisk_cumevents"</code> to show the number at risk and, the cumulative number of censoring and events, respectively.

`risk.table.pos` character vector specifying the risk table position. Allowed options are one of `c("out", "in")` indicating 'outside' or 'inside' the main plot, respectively. Default value is "out".

`risk.table.title`
The title to be used for the risk table.

`risk.table.col` same as `tables.col` but for risk table only.

`risk.table.fontsize, fontsize`
font size to be used for the risk table and the cumulative events table.

`risk.table.y.text`
logical. Default is TRUE. If FALSE, risk table y axis tick labels will be hidden.

`risk.table.y.text.col`
logical. Default value is FALSE. If TRUE, risk table tick labels will be colored by strata.

`risk.table.height`
the height of the risk table on the grid. Increase the value when you have many strata. Default is 0.25. Ignored when `risk.table = FALSE`.

`surv.plot.height`
the height of the survival plot on the grid. Default is 0.75. Ignored when `risk.table = FALSE`. `1-risk.table.height -ncensor.plot.height` when `risk.table = TRUE` and `ncensor.plot = TRUE`

`ncensor.plot.height`
The height of the censor plot. Used when `ncensor.plot = TRUE`.

`cumevents.height`
the height of the cumulative events table on the grid. Default is 0.25. Ignored when `cumevents = FALSE`.

`cumcensor.height`
the height of the cumcensor table on the grid. Default is 0.25. Ignored when `cumcensor = FALSE`.

`ncensor.plot` logical value. If TRUE, the number of censored subjects at time t is plotted. Default is FALSE. Ignored when `cumcensor = TRUE`.

`ncensor.plot.title`
The title to be used for the censor plot. Used when `ncensor.plot = TRUE`.

`cumevents` logical value specifying whether to show or not the table of the cumulative number of events. Default is FALSE.

`cumevents.col` same as `tables.col` but for the cumulative events table only.

`cumevents.title`
The title to be used for the cumulative events table.

`cumevents.y.text`
logical. Default is TRUE. If FALSE, the y axis tick labels of the cumulative events table will be hidden.

`cumevents.y.text.col`
logical. Default value is FALSE. If TRUE, the y tick labels of the cumulative events will be colored by strata.

`cumcensor` logical value specifying whether to show or not the table of the cumulative number of censoring. Default is FALSE.

<code>cumcensor.col</code>	same as <code>tables.col</code> but for <code>cumcensor</code> table only.
<code>cumcensor.title</code>	The title to be used for the <code>cumcensor</code> table.
<code>cumcensor.y.text</code>	logical. Default is TRUE. If FALSE, the y axis tick labels of the <code>cumcensor</code> table will be hidden.
<code>cumcensor.y.text.col</code>	logical. Default value is FALSE. If TRUE, the y tick labels of the <code>cumcensor</code> will be colored by strata.
<code>surv.median.line</code>	character vector for drawing a horizontal/vertical line at median survival. Allowed values include one of <code>c("none", "hv", "h", "v")</code> . v: vertical, h:horizontal.
<code>ggtheme</code>	function, ggplot2 theme name. Default value is theme_survminer . Allowed values include ggplot2 official themes: see theme .
<code>tables.theme</code>	function, ggplot2 theme name. Default value is theme_survminer . Allowed values include ggplot2 official themes: see theme .
<code>...</code>	other arguments to be passed i) to ggplot2 <code>geom_*()</code> functions such as <code>linetype</code> , <code>size</code> , ii) or to the function <code>ggpubr::ggpar()</code> for customizing the plots. See details section.
<code>x</code>	an object of class <code>ggsurvplot</code>
<code>newpage</code>	open a new page. See grid.arrange

Details

- **legend position:** The argument **legend** can be also a numeric vector `c(x,y)`. In this case it is possible to position the legend inside the plotting area. `x` and `y` are the coordinates of the legend box. Their values should be between 0 and 1. `c(0,0)` corresponds to the "bottom left" and `c(1,1)` corresponds to the "top right" position. For instance use `legend = c(0.8, 0.2)`.
- **Color palettes:** The argument **palette** can be used to specify the color to be used for each group. By default, the first color in the palette is used to color the first level of the factor variable. This default behavior can be changed by assigning correctly a named vector. That is, the names of colors should match the strata names as generated by the `ggsurvplot()` function in the legend.
- **Customizing the plots:** The plot can be easily customized using additional arguments to be passed to the function `ggpar()`. Read `?ggpubr::ggpar`. These arguments include *font.title*, *font.subtitle*, *font.caption*, *font.x*, *font.y*, *font.tickslab* and *font.legend*: a vector of length 3 indicating respectively the size (e.g.: 14), the style (e.g.: "plain", "bold", "italic", "bold.italic") and the color (e.g.: "red") of main title, subtitle, caption, xlab and ylab, axis tick labels and legend, respectively. For example `font.x = c(14, "bold", "red")`. Use `font.x = 14`, to change only font size; or use `font.x = "bold"`, to change only font face.

Value

return an object of class `ggsurvplot` which is list containing the following components:

- plot: the survival plot (ggplot object)
- table: the number of subjects at risk table per time (ggplot object).
- cumevents: the cumulative number of events table (ggplot object).
- ncensor.plot: the number of censoring (ggplot object).
- data.survplot: the data used to plot the survival curves (data.frame).
- data.survtable: the data used to plot the tables under the main survival curves (data.frame).

Functions

- ggsurvplot: Draws survival curves using ggplot2.

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
#####
# Example 1: Survival curves with two groups
#####

# Fit survival curves
#+++++
require("survival")
fit<- survfit(Surv(time, status) ~ sex, data = lung)

# Basic survival curves
ggsurvplot(fit, data = lung)

# Customized survival curves
ggsurvplot(fit, data = lung,
  surv.median.line = "hv", # Add medians survival

# Change legends: title & labels
legend.title = "Sex",
legend.labs = c("Male", "Female"),
# Add p-value and confidence intervals
pval = TRUE,

conf.int = TRUE,
# Add risk table
risk.table = TRUE,
tables.height = 0.2,
tables.theme = theme_cleantable(),

# Color palettes. Use custom color: c("#E7B800", "#2E9FDF"),
# or brewer color (e.g.: "Dark2"), or ggsci color (e.g.: "jco")
palette = c("#E7B800", "#2E9FDF"),
ggtheme = theme_bw() # Change ggplot2 theme
```

```

)

# Change font size, style and color
#####
## Not run:
# Change font size, style and color at the same time
ggsurvplot(fit, data = lung, main = "Survival curve",
  font.main = c(16, "bold", "darkblue"),
  font.x = c(14, "bold.italic", "red"),
  font.y = c(14, "bold.italic", "darkred"),
  font.tickslab = c(12, "plain", "darkgreen"))

## End(Not run)

#####
# Example 2: Facet ggsurvplot() output by
# a combination of factors
#####

# Fit (complexe) survival curves
#####
## Not run:
require("survival")
fit3 <- survfit( Surv(time, status) ~ sex + rx + adhere,
  data = colon )

# Visualize
#####
ggsurv <- ggsurvplot(fit3, data = colon,
  fun = "cumhaz", conf.int = TRUE,
  risk.table = TRUE, risk.table.col="strata",
  ggtheme = theme_bw())

# Faceting survival curves
curv_facet <- ggsurv$plot + facet_grid(rx ~ adhere)
curv_facet

# Faceting risk tables:
# Generate risk table for each facet plot item
ggsurv$table + facet_grid(rx ~ adhere, scales = "free")+
  theme(legend.position = "none")

# Generate risk table for each facet columns
tbl_facet <- ggsurv$table + facet_grid(.~ adhere, scales = "free")
tbl_facet + theme(legend.position = "none")

# Arrange faceted survival curves and risk tables
g2 <- ggplotGrob(curv_facet)
g3 <- ggplotGrob(tbl_facet)
min_ncol <- min(ncol(g2), ncol(g3))
g <- gridExtra::rbind.gtable(g2[, 1:min_ncol], g3[, 1:min_ncol], size="last")

```

```
g$widths <- grid::unit.pmax(g2$widths, g3$widths)
grid::grid.newpage()
grid::grid.draw(g)
```

```
## End(Not run)
```

myeloma

Multiple Myeloma Data

Description

Multiple Myeloma data extracted from publicly available gene expression data (GEO Id: GSE4581).

Usage

```
data("myeloma")
```

Format

A data frame with 256 rows and 12 columns.

molecular_group Patients' molecular subgroups

chr1q21_status Amplification status of the chromosome 1q21

treatment treatment

event survival status 0 = alive, 1 = dead

time Survival time in months

CCND1 Gene expression

CRIM1 Gene expression

DEPDC1 Gene expression

IRF4 Gene expression

TP53 Gene expression

WHSC1 Gene expression

The remaining columns (CCND1, CRIM1, DEPDC1, IRF4, TP53, WHSC1) correspond to the gene expression level of specified genes.

Examples

```
data(myeloma)
head(myeloma)
```

pairwise_survdiff	<i>Multiple Comparisons of Survival Curves</i>
-------------------	--

Description

Calculate pairwise comparisons between group levels with corrections for multiple testing.

Usage

```
pairwise_survdiff(formula, data, p.adjust.method = "BH", na.action, rho = 0)
```

Arguments

formula	a formula expression as for other survival models, of the form <code>Surv(time, status) ~ predictors</code> .
data	a data frame in which to interpret the variables occurring in the formula.
p.adjust.method	method for adjusting p values (see p.adjust). Allowed values include <code>c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")</code> . If you don't want to adjust the p value (not recommended), use <code>p.adjust.method = "none"</code> .
na.action	a missing-data filter function. Default is <code>options()\$na.action</code> .
rho	a scalar parameter that controls the type of test. Allowed values include 0 (for Log-Rank test) and 1 (for peto & peto test).

Value

Returns an object of class "pairwise.htest", which is a list containing the p values.

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

See Also

`survival::survdiff`

Examples

```
library(survival)
library(survminer)
data(myeloma)

# Pairwise survdiff
res <- pairwise_survdiff(Surv(time, event) ~ molecular_group,
  data = myeloma)
res
```

```
# Symbolic number coding
symnum(res$p.value, cutpoints = c(0, 0.0001, 0.001, 0.01, 0.05, 0.1, 1),
       symbols = c("****", "***", "**", "*", "+", " "),
       abbr.colnames = FALSE, na = "")
```

surv_cutpoint

Determine the Optimal Cutpoint for Continuous Variables

Description

Determine the optimal cutpoint for one or multiple continuous variables at once, using the maximally selected rank statistics from the 'maxstat' R package. This is an outcome-oriented methods providing a value of a cutpoint that correspond to the most significant relation with outcome (here, survival).

Usage

```
surv_cutpoint(data, time = "time", event = "event", variables,
              minprop = 0.1, progressbar = TRUE)

surv_categorize(x, variables = NULL, labels = c("low", "high"))

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

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

## S3 method for class 'surv_cutpoint'
plot(x, variables = NULL,
     ggtheme = theme_classic2(), bins = 30, ...)

## S3 method for class 'plot_surv_cutpoint'
print(x, ..., newpage = TRUE)
```

Arguments

data	a data frame containing survival information (time, event) and continuous variables (e.g.: gene expression data).
time, event	column names containing time and event data, respectively. Event values should be 0 or 1.
variables	a character vector containing the names of variables of interest, for which we want to estimate the optimal cutpoint.
minprop	the minimal proportion of observations per group.

progressbar	logical value. If TRUE, show progress bar. Progressbar is shown only, when the number of variables > 5.
x, object	an object of class surv_cutpoint
labels	labels for the levels of the resulting category.
...	other arguments. For plots, see ?ggpubr::ggpar
ggtheme	function, ggplot2 theme name. Default value is theme_classic2 . Allowed values include ggplot2 official themes. see ?ggplot2::ggtheme.
bins	Number of bins for histogram. Defaults to 30.
newpage	open a new page. See grid.arrange .

Value

- **surv_cutpoint()**: returns an object of class 'surv_cutpoint', which is a list with the following components:
 - maxstat results for each variable (see ?maxstat::maxstat)
 - cutpoint: a data frame containing the optimal cutpoint of each variable. Rows are variable names and columns are c("cutpoint", "statistic").
 - data: a data frame containing the survival data and the original data for the specified variables.
 - minprop: the minimal proportion of observations per group.
 - not_numeric: contains data for non-numeric variables, in the context where the user provided categorical variable names in the argument variables.

Methods defined for surv_cutpoint object are summary, print and plot.

- **surv_categorize()**: returns an object of class 'surv_categorize', which is a data frame containing the survival data and the categorized variables.

Functions

- surv_cutpoint: Determine the optimal cutpoint for each variable using 'maxstat'
- surv_categorize: Divide each variable values based on the cutpoint returned by surv_cutpoint().

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
# 0. Load some data
data(myeloma)
head(myeloma)

# 1. Determine the optimal cutpoint of variables
res.cut <- surv_cutpoint(myeloma, time = "time", event = "event",
  variables = c("DEPDC1", "WHSC1", "CRIM1"))

summary(res.cut)
```

```
# 2. Plot cutpoint for DEPD1
# palette = "npg" (nature publishing group), see ?ggpubr::ggpar
plot(res.cut, "DEPD1", palette = "npg")

# 3. Categorize variables
res.cat <- surv_categorize(res.cut)
head(res.cat)

# 4. Fit survival curves and visualize
library("survival")
fit <- survfit(Surv(time, event) ~DEPD1, data = res.cat)
ggsurvplot(fit, data = res.cat, risk.table = TRUE, conf.int = TRUE)
```

surv_summary

Nice Summary of a Survival Curve

Description

Compared to the default `summary()` function, `surv_summary()` creates a data frame containing a nice summary from `survfit` results.

Usage

```
surv_summary(x, data = NULL)
```

Arguments

<code>x</code>	an object of class <code>survfit</code> .
<code>data</code>	a dataset used to fit survival curves. If not supplied then data will be extracted from 'fit' object.

Value

An object of class **'surv_summary'**, which is a data frame with the following columns:

- `time`: the time points at which the curve has a step.
- `n.risk`: the number of subjects at risk at `t`.
- `n.event`: the number of events that occur at time `t`.
- `n.censor`: number of censored events.
- `surv`: estimate of survival.
- `std.err`: standard error of survival.
- `upper`: upper end of confidence interval.
- `lower`: lower end of confidence interval.
- `strata`: stratification of survival curves.

In a situation, where survival curves have been fitted with one or more variables, `surv_summary` object contains **extra columns** representing the variables. This makes it possible to facet the output of `ggsurvplot` by strata or by some combinations of factors.

`surv_summary` object has also an attribute named **'table'** containing information about the survival curves, including medians of survival with confidence intervals, as well as, the total number of subjects and the number of event in each curve.

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
# Fit survival curves
require("survival")
fit <- survfit(Surv(time, status) ~ rx + adhere, data = colon)

# Summarize
res.sum <- surv_summary(fit, data = colon)
head(res.sum)

# Information about the survival curves
attr(res.sum, "table")
```

theme_classic2	<i>ggplot2 classic theme with axis lines</i>
----------------	--

Description

Create a `ggplot2` classic theme with axis lines.

Usage

```
theme_classic2(base_size = 12, base_family = "")
```

Arguments

<code>base_size</code>	base font size
<code>base_family</code>	base font family

Examples

```
p <- ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point(aes(color = gear))

# Default plot
p

# Use theme_classic()
p + theme_classic()

# Use theme_classic2()
p + theme_classic2()
```

theme_survminer	<i>Theme for Survminer Plots</i>
-----------------	----------------------------------

Description

Default theme for plots generated with survminer.

Usage

```
theme_survminer(base_size = 12, base_family = "", font.main = c(16,
  "plain", "black"), font.submain = c(15, "plain", "black"), font.x = c(14,
  "plain", "black"), font.y = c(14, "plain", "black"), font.caption = c(15,
  "plain", "black"), font.tickslab = c(12, "plain", "black"),
  legend = c("top", "bottom", "left", "right", "none"), font.legend = c(10,
  "plain", "black"), ...)

theme_cleantable(base_size = 12, base_family = "", ...)
```

Arguments

base_size	base font size
base_family	base font family
font.main, font.submain, font.caption, font.x, font.y, font.tickslab, font.legend	a vector of length 3 indicating respectively the size (e.g.: 14), the style (e.g.: "plain", "bold", "italic", "bold.italic") and the color (e.g.: "red") of main title, subtitle, caption, xlab and ylab, axis tick labels and legend, respectively. For example <i>font.x = c(14, "bold", "red")</i> . Use font.x = 14, to change only font size; or use font.x = "bold", to change only font face.
legend	character specifying legend position. Allowed values are one of c("top", "bottom", "left", "right", "none"). Default is "top" side position. to remove the legend use legend = "none". Legend position can be also specified using a numeric vector c(x, y); see details section.
...	additional arguments passed to the function theme_survminer().

Functions

- `theme_survminer`: Default theme for survminer plots. A theme similar to `theme_classic()` with large font size.
- `theme_cleantable`: theme for drawing a clean risk table and cumulative number of events table. A theme similar to `theme_survminer()` without i) axis lines and, ii) x axis ticks and title.

Author(s)

Alboukadel Kassambara, <alboukadel.kassambara@gmail.com>

Examples

```
# Fit survival curves
#+++++
require("survival")
fit<- survfit(Surv(time, status) ~ sex, data = lung)

# Basic survival curves
#+++++
ggsurv <- ggsurvplot(fit, data = lung, risk.table = TRUE,
  main = "Survival curves",
  submain = "Based on Kaplan-Meier estimates",
  caption = "created with survminer",
)

# Change font size, style and color
#+++++
# Change font size, style and color at the same time
# Use font.x = 14, to change only font size; or use
# font.x = "bold", to change only font face.
ggsurv %>% theme_survminer(
  font.main = c(16, "bold", "darkblue"),
  font.submain = c(15, "bold.italic", "purple"),
  font.caption = c(14, "plain", "orange"),
  font.x = c(14, "bold.italic", "red"),
  font.y = c(14, "bold.italic", "darkred"),
  font.tickslab = c(12, "plain", "darkgreen")
)

# Clean risk table
# +++++
ggsurv$table <- ggsurv$table + theme_cleantable()
ggsurv
```

Index

`+.ggsurv (add_ggsurvplot), 2`
`%%(add_ggsurvplot), 2`

`add_ggsurvplot, 2`
`arrange_ggsurvplots, 3`
`arrangeGrob, 12, 13`

`BMT, 5`
`BRCAOV.survInfo, 6`

`comp, 23`
`cox.zph, 13`
`coxph, 8–11`
`coxph.object, 8, 10, 11`
`cuminc, 7`

`geom_hline, 10`
`geom_smooth, 10`
`ggcompetingrisks, 7`
`ggcoxadjustedcurves, 8`
`ggcoxdiagnostics, 9`
`ggcoxfunctional, 11`
`ggcoxzph, 13`
`ggcumcensor, 14`
`ggcumevents, 16`
`ggforest, 17`
`ggpar, 7, 9, 10, 12, 13, 15, 17–20`
`ggrisktable, 18`
`ggsave, 4`
`ggsurvevents, 20`
`ggsurvplot, 3, 14, 16, 18, 21, 33`
`grid.arrange, 12, 13, 25, 31`
`grob, 12, 13`

`lowess, 11, 12`

`marrangeGrob, 4`
`myeloma, 28`

`p.adjust, 29`
`pairwise_survdiff, 29`

`plot.cox.zph, 13`
`plot.surv_cutpoint (surv_cutpoint), 30`
`print.ggcoxfunctional
 (ggcoxfunctional), 11`
`print.ggcoxzph (ggcoxzph), 13`
`print.ggsurvplot (ggsurvplot), 21`
`print.plot_surv_cutpoint
 (surv_cutpoint), 30`
`print.surv_cutpoint (surv_cutpoint), 30`

`residuals.coxph, 10`

`summary.surv_cutpoint (surv_cutpoint),
 30`
`Surv, 11, 20`
`surv_categorize (surv_cutpoint), 30`
`surv_cutpoint, 30`
`surv_summary, 32`
`survfit, 7, 20, 32`

`theme, 7, 9, 10, 12, 13, 15, 17, 19, 25`
`theme_classic2, 9, 12, 13, 31, 33`
`theme_cleantable (theme_survminer), 34`
`theme_survminer, 3, 7, 15, 17, 19, 25, 34`