Package: survMisc (via r-universe)

October 26, 2024

Type Package

Version 0.5.1

Date 2016-05-11

Depends survival

Imports graphics, grDevices, stats, utils, knitr, KMsurv, ggplot2, data.table, zoo, grid, gridExtra, km.ci

Author Chris Dardis

Maintainer Chris Dardis <christopherdardis@gmail.com>

License GPL-2

Title Miscellaneous Functions for Survival Data

Description A collection of functions to help in the analysis of right-censored survival data. These extend the methods available in package:survival.

BugReports https://github.com/dardisco/survMisc/issues

LazyData true

VignetteBuilder knitr

Collate 'sf.R' 'nc.R' 'ci.R' 'autoplotTAP.R' 'autoplotTen.R' 'print.R' 'asWide.R' 'ten.R' 'COV.R' 'predict.R' 'comp.R' 'cutp.R' 'gastric.R' 'gof.R' 'onAttach.R' 'plotSurv.R' 'profLik.R' 'rsq.R' 'survMisc_package.R'

Repository https://dardisco.r-universe.dev

RemoteUrl https://github.com/dardisco/survmisc

RemoteRef HEAD

RemoteSha d6cefe378d52a6233e75c47828d3d2f2b138794c

Contents

`asWide` *Convert an object to "wide" or "long" form.*

Description

Convert an object to "wide" or "long" form.

Usage

```
asWide(x, ...)

## S3 method for class 'ten'
asWide(x, ...)

asLong(x, ...)

## S3 method for class 'ten'
asLong(x, ...)
```

Arguments

`x`	An object of class `ten` or `pred`.
`...`	Additional arguments (not implemented).

Value

A new `data.table` is returned, with the data in 'wide' or 'long' format.
There is one row for each time point.
For a `ten` object generated from a `numeric` or `Surv` object, this has columns:

t	**t**ime.
e	number of **e**vents.
n	**n**umber at risk.

If derived from a survfit, coxph or formula object, there are additional columns for e and n for *each* covariate group.

Note

Most methods for ten objects are designed for the 'long' form.

Examples

```
data("bmt", package="KMsurv")
require("survival")
t1 <- ten(c1 <- coxph(Surv(t2, d3) ~ z3*z10, data=bmt))
asWide(t1)
asLong(asWide(t1))
stopifnot(asLong(asWide(t1)) == ten(ten(t1)))
```

---

autoplotTableAndPlot *Arrange a survival plot with corresponding table and legend.*

Description

Arrange a survival plot with corresponding table and legend.

Usage

```
## S3 method for class 'tableAndPlot'
autoplot(object, ..., hideTabLeg = TRUE,
  tabHeight = 0.25)
```

Arguments

object	An object of class "tableAndPlot", as returned by ggplot.Ten.
...	Additional arguments (not implemented).
hideTabLeg	Hide table legend. If hideTabLeg = TRUE (the default), the table legend will not appear.
tabHeight	Table height, as a fraction/ proportion of the whole. tabHeight=0.25 (the default) makes the table $0.25 = 25\%$ of the whole plot height.

Details

Arguments to plotHeigth and tabHeight are best specified as fractions adding to 1,

Value

A graph, plotted with `gridExtra::grid.arrange`.

Note

This method is called by `print.tableAndPlot` and by `print.stratTableAndPlot`.

Author(s)

Chris Dardis. Based on existing work by R. Saccilotto, Abhijit Dasgupta, Gil Tomas and Mark Cowley.

Examples

```
data("kidney", package="KMsurv")
autoplot(survfit(Surv(time, delta) ~ type, data=kidney), type="fill")
autoplot(ten(survfit(Surv(time, delta) ~ type, data=kidney)), type="fill")
data("bmt", package="KMsurv")
s2 <- survfit(Surv(time=t2, event=d3) ~ group, data=bmt)
autoplot(s2)
```

---

autoplotTen *Generate a* ggplot *for a* survfit *or* ten *object*

Description

Generate a ggplot for a survfit or ten object

Usage

```
autoplot(object, ...)

## S3 method for class 'ten'
autoplot(object, ..., title = "Marks show times with censoring",
  type = c("single", "CI", "fill"), alpha = 0.05, ciLine = 10,
  censShape = 3, palette = c("Dark2", "Set2", "Accent", "Paired", "Pastel1",
  "Pastel2", "Set1", "Set3"), jitter = c("none", "noEvents", "all"),
  tabTitle = "Number at risk by time", xLab = "Time",
  timeTicks = c("major", "minor", "days", "months", "custom"), times = NULL,
  yLab = "Survival", yScale = c("perc", "frac"), legend = TRUE,
  legTitle = "Group", legLabs = NULL, legOrd = NULL, titleSize = 15,
  axisTitleSize = 15, axisLabSize = 10, survLineSize = 0.5,
  censSize = 5, legTitleSize = 10, legLabSize = 10, fillLineSize = 0.05,
  tabTitleSize = 15, tabLabSize = 5, nRiskSize = 5)

## S3 method for class 'stratTen'
autoplot(object, ..., title = NULL, type = c("single",
```

```
  "CI", "fill"), alpha = 0.05, ciLine = 10, censShape = 3,
  palette = c("Dark2", "Set2", "Accent", "Paired", "Pastel1", "Pastel2",
  "Set1", "Set3"), jitter = c("none", "noEvents", "all"),
  tabTitle = "Number at risk by time", xLab = "Time",
  timeTicks = c("major", "minor", "days", "months", "custom"), times = NULL,
  yLab = "Survival", yScale = c("perc", "frac"), legend = TRUE,
  legTitle = "Group", legLabs = NULL, legOrd = NULL, titleSize = 15,
  axisTitleSize = 15, axisLabSize = 10, survLineSize = 0.5,
  censSize = 5, legTitleSize = 10, legLabSize = 10, fillLineSize = 0.05,
  tabTitleSize = 15, tabLabSize = 5, nRiskSize = 5)

## S3 method for class 'survfit'
autoplot(object, ...,
  title = "Marks show times with censoring", type = c("single", "CI",
  "fill"), alpha = 0.05, ciLine = 10, censShape = 3,
  palette = c("Dark2", "Set2", "Accent", "Paired", "Pastel1", "Pastel2",
  "Set1", "Set3"), jitter = c("none", "noEvents", "all"),
  tabTitle = "Number at risk by time", xLab = "Time",
  timeTicks = c("major", "minor", "weeks", "months", "custom"),
  times = NULL, yLab = "Survival", yScale = c("perc", "frac"),
  legend = TRUE, legLabs = NULL, legOrd = NULL, legTitle = "Group",
  titleSize = 15, axisTitleSize = 15, axisLabSize = 10,
  survLineSize = 0.5, censSize = 5, legTitleSize = 10, legLabSize = 10,
  fillLineSize = 0.05, tabTitleSize = 15, tabLabSize = 5, nRiskSize = 5,
  pVal = FALSE, sigP = 1, pX = 0.1, pY = 0.1)
```

Arguments

`object`	An object of class `survfit`, `ten` or `stratTen`.
`...`	Additional arguments (not implemented).
`title`	Title for survival plot.
`type`	`type="single"` (the default) plots single lines. `type="CI"` Adds lines indicating confidence intervals (taken from `upper` and `lower` values of `survfit` object). Higher values of `alpha` (opacity) are recommended for this, e.g. `alpha=0.8`. `type="fill"` Adds filled rectangles from the survival lines to the confidence intervals above.
`alpha`	Opacity of lines indicating confidence intervals or filled rectangles. Should be in range $0 - 1$. Lower = more transparent. Larger values e.g. `alpha=0.7` are recommended for confidence intervals.
`ciLine`	Confidence interval line type. See 'line type specification' in ?graphics::par
`censShape`	Shape of marks to indicate censored onservations. Default is `3` which gives vertical ticks. Use `censShape=10` for circular marks. See ?graphics::points

`palette` Options are taken from color_brewer.

- `palette="Dark2"` (the default) is recommended for `single` or `CI` plots.
- `palette="Set2"` is recommended for `type="fill"` plots.

`jitter` By default, `jitter="none"`.

- If `jitter="noEvents"`, adds some random, positive noise to survival lines with no events (i.e. all observations censored). This will bring them just above 1 on the y-axis, making them easier to see separately.
- If `jitter="all"` add some vertical and horizontal noise to all survival lines. This can prevent overlapping of lines for censoring.

`tabTitle` Table title.

–Axis arguments:

`xLab` Label for x axis on survival plot.

`timeTicks` Numbers to mark on the x axis of the survival plot and the table.

`"major"` (the default) only the major x-axis (time) marks from the survival plot are are labelled on the plot and table.

`"minor"` minor axis marks are labelled instead.

`"days"` scale is $0, 7, 14, ..., t_{max}$

`"months"` scale is $0, 12, , 24, ..., t_{max}$

`"custom"` scale is given by `times` below

`times` Vector of custom times to use for x axis.

`yLab` Label for y axis on survival plot.

`yScale` Display for point on y axis:

`"perc"` Displays as percentages.

`"frac"` Displays as fractions e.g. $0, 0.1, 0.2, ..., 1.0$.

–Legend arguments:

`legend` If `legend=FALSE`, no legends will be produced for the plot or table.

`legTitle` Legend title.

`legLabs` Legend labels. These can be used to replace the names of the covariate groups ('strata' in the case of a `survfit` object).
Should be given in the same order as those strata.

`legOrd` Legend order.

–Size arguments:
Size arguments are passed to `ggplot2::element_text(size=)`.

`titleSize` Title size for survival plot.

`axisTitleSize` Title size for axes.

`axisLabSize` Title size for labels on axes.

`survLineSize` Survival line size.

`legTitleSize` Title size for legend.

legLabSize	Legend labels width and height.
censSize	Size of marks to indicate censored onservations.
fillLineSize	Line size surrouding filled boxes.
tabTitleSize	Table title text size.
tabLabSize	Table legend text size.
nRiskSize	Number at risk - text size.
	–Arguments for autoplot.survfit only:
pVal	If pVal=TRUE, adds p value from log-rank test to plot
sigP	No. of significant digits to display in p value. Typically 1 to 3.
pX	Location of p value on x axis. Should be in the range of $0 - 1$, where value is to be placed relative to the maximum observed time. E.g. pX = 0.5 will place it half-way along x-axis
pY	Location of p value on y axis. Should be in the range of $0 - 1$, as above.

Note

autoplot.survfit may be deprecated after packageVersion 0.6. Please try to use autoplot.ten instead.

Author(s)

Chris Dardis. autoplot.survfit based on existing work by R. Saccilotto, Abhijit Dasgupta, Gil Tomas and Mark Cowley.

See Also

?ggplot2::ggplot_build

Examples

```
## examples are slow to run; see vignette for output from these
## Not run:
### autoplot.ten
data("kidney", package="KMsurv")
t1 <- ten(survfit(Surv(time, delta) ~ type, data=kidney))
autoplot(t1)
autoplot(t1, type="fill", survLineSize=2, jitter="all")
autoplot(t1, timeTicks="months",
 type="CI", jitter="all",
 legLabs=c("surgical", "percutaneous"),
 title="Time to infection following catheter placement \n
   by type of catheter, for dialysis patients",
 titleSize=10, censSize=2)$plot
t2 <- ten(survfit(Surv(time=time, event=delta) ~ 1, data=kidney))
```

```
autoplot(t2, legLabs="")$plot
autoplot(t2, legend=FALSE)
data("rectum.dat", package="km.ci")
t3 <- ten(survfit(Surv(time, status) ~ 1, data=rectum.dat))
## change confidence intervals to log Equal-Precision confidence bands
ci(t3, how="nair", tL=1, tU=40)
autoplot(t3, type="fill", legend=FALSE)$plot
## manually changing the output
t4 <- ten(survfit(Surv(time, delta) ~ type, data=kidney))
(a4 <- autoplot(t4, type="CI", alpha=0.8, survLineSize=2)$plot)
## change default colors
a4 + list(ggplot2::scale_color_manual(values=c("red", "blue")),
          ggplot2::scale_fill_manual(values=c("red", "blue")))
## change limits of y-axis
suppressMessages(a4 + ggplot2::scale_y_continuous(limits=c(0, 1)))

## End(Not run)
## Not run:
data("pbc", package="survival")
t1 <- ten(Surv(time, status==2) ~ trt + strata(edema), data=pbc, abbNames=FALSE)
autoplot(t1)

## End(Not run)
### autoplot.survfit
## Not run:
data(kidney, package="KMsurv")
s1 <- survfit(Surv(time, delta) ~ type, data=kidney)
autoplot(s1, type="fill", survLineSize=2)
autoplot(s1, type="CI", pVal=TRUE, pX=0.3,
 legLabs=c("surgical", "percutaneous"),
 title="Time to infection following catheter placement \n
   by type of catheter, for dialysis patients")$plot
s1 <- survfit(Surv(time=time, event=delta) ~ 1, data=kidney)
autoplot(s1, legLabs="")$plot
autoplot(s1, legend=FALSE)$plot
data(rectum.dat, package="km.ci")
s1 <- survfit(Surv(time, status) ~ 1, data=rectum.dat)
## change confidence intervals to log Equal-Precision confidence bands
if (require("km.ci")) {
 km.ci::km.ci(s1, method="logep")
 autoplot(s1, type="fill", legend=FALSE)$plot
}
## manually changing the output
s1 <- survfit(Surv(time, delta) ~ type, data=kidney)
g1 <- autoplot(s1, type="CI", alpha=0.8, survLineSize=2)$plot
## change default colors
g1 + ggplot2::scale_colour_manual(values=c("red", "blue")) +
    ggplot2::scale_fill_manual(values=c("red", "blue"))
## change limits of y-axis
g1 + ggplot2::scale_y_continuous(limits=c(0, 1))

## End(Not run)
```

ci *confidence intervals for survival curves.*

Description

confidence **i**ntervals for survival curves.

Usage

```
ci(x, ...)

## S3 method for class 'ten'
ci(x, ..., CI = c("0.95", "0.9", "0.99"), how = c("point",
  "nair", "hall"), trans = c("log", "lin", "asi"), tL = NULL, tU = NULL,
  reCalc = FALSE)

## S3 method for class 'stratTen'
ci(x, ..., CI = c("0.95", "0.9", "0.99"),
  how = c("point", "nair", "hall"), trans = c("log", "lin", "asi"),
  tL = NULL, tU = NULL)
```

Arguments

x	An object of class ten.
CI	Confidence intervals. As the function currently relies on lookup tables, currently only 90%, 95% (the default) and 99% are supported.
how	Method to use for confidence interval. point (the default) uses pointwise confirence intervals. The alternatives use confidence *bands* (see details).
trans	Transformation to use. The default is trans="log". Also supported are linear and arcsine-square root transformations.
tL	**L**ower time point. Used in construction of confidence bands.
tU	**U**pper time point. Used in construction of confidence bands.
...	Additional arguments (not implemented).
reCalc	Recalcuate the values? If reCalc=FALSE (the default) and the ten object already has the calculated values stored as an attribute, the value of the attribute is returned directly.

Details

In the equations below

$$\sigma_s^2(t) = \frac{\hat{V}[\hat{S}(t)]}{\hat{S}^2(t)}$$

Where $\hat{S}(t)$ is the Kaplan-Meier survival estimate and $\hat{V}[\hat{S}(t)]$ is Greenwood's estimate of its variance.
The **pointwise** confidence intervals are valid for *individual* times, e.g. `median` and `quantile` values. When plotted and joined for multiple points they tend to be narrower than the *bands* described below. Thus they tend to exaggerate the impression of certainty when used to plot confidence intervals for a time range. They should not be interpreted as giving the intervals within which the *entire* survival function lies.
For a given significance level α, they are calculated using the standard normal distribution Z as follows:

- linear
$$\hat{S}(t) \pm Z_{1-\alpha}\sigma(t)\hat{S}(t)$$
- log transform
$$[\hat{S}(t)^{\frac{1}{\theta}}, \hat{S}(t)^{\theta}]$$
where
$$\theta = \exp\frac{Z_{1-\alpha}\sigma(t)}{\log \hat{S}(t)}$$
- arcsine-square root transform
upper:
$$\sin^2(\max[0, \arcsin\sqrt{\hat{S}(t)} - \frac{Z_{1-\alpha}\sigma(t)}{2}\sqrt{\frac{\hat{S}(t)}{1-\hat{S}(t)}}])$$
lower:
$$\sin^2(\min[\frac{\pi}{2}, \arcsin\sqrt{\hat{S}(t)} + \frac{Z_{1-\alpha}\sigma(t)}{2}\sqrt{\frac{\hat{S}(t)}{1-\hat{S}(t)}}])$$

Confidence **bands** give the values within which the survival function falls within a *range* of timepoints.

The time range under consideration is given so that $t_l \geq t_{min}$, the minimum or lowest event time and $t_u \leq t_{max}$, the maximum or largest event time.
For a sample size n and $0 < a_l < a_u < 1$:

$$a_l = \frac{n\sigma_s^2(t_l)}{1 + n\sigma_s^2(t_l)}$$

$$a_u = \frac{n\sigma_s^2(t_u)}{1 + n\sigma_s^2(t_u)}$$

For the **Nair** or **equal precision** (**EP**) confidence bands, we begin by obtaining the relevant confidence coefficient c_α. This is obtained from the upper α-th fractile of the random variable

$$U = \sup|W^o(x)\sqrt{[x(1-x)]}|, \quad a_l \leq x \leq a_u$$

Where W^o is a standard Brownian bridge.
The intervals are:

- linear
$$\hat{S}(t) \pm c_\alpha \sigma_s(t) \hat{S}(t)$$
- log transform (the default)
This uses θ as below:
$$\theta = \exp \frac{c_\alpha \sigma_s(t)}{\log \hat{S}(t)}$$
And is given by:
$$[\hat{S}(t)^{\frac{1}{\theta}}, \hat{S}(t)^\theta]$$
- arcsine-square root transform
upper:
$$\sin^2(\max[0, \arcsin \sqrt{\hat{S}(t)} - \frac{c_\alpha \sigma_s(t)}{2} \sqrt{\frac{\hat{S}(t)}{1-\hat{S}(t)}}])$$
lower:
$$\sin^2(\min[\frac{\pi}{2}, \arcsin \sqrt{\hat{S}(t)} + \frac{c_\alpha \sigma_s(t)}{2} \sqrt{\frac{\hat{S}(t)}{1-\hat{S}(t)}}])$$

For the **Hall-Wellner** bands the confidence coefficient k_α is obtained from the upper α-th fractile of a Brownian bridge.
In this case t_l can be $= 0$.
The intervals are:

- linear
$$\hat{S}(t) \pm k_\alpha \frac{1 + n\sigma_s^2(t)}{\sqrt{n}} \hat{S}(t)$$
- log transform
$$[\hat{S}(t)^{\frac{1}{\theta}}, \hat{S}(t)^\theta]$$
where
$$\theta = \exp \frac{k_\alpha [1 + n\sigma_s^2(t)]}{\sqrt{n} \log \hat{S}(t)}$$
- arcsine-square root transform
upper:
$$\sin^2(\max[0, \arcsin \sqrt{\hat{S}(t)} - \frac{k_\alpha [1 + n\sigma_s(t)]}{2\sqrt{n}} \sqrt{\frac{\hat{S}(t)}{1-\hat{S}(t)}}])$$
lower:
$$\sin^2(\min[\frac{\pi}{2}, \arcsin \sqrt{\hat{S}(t)} + \frac{k_\alpha [1 + n\sigma_s^2(t)]}{2\sqrt{n}} \sqrt{\frac{\hat{S}(t)}{1-\hat{S}(t)}}])$$

Value

The `ten` object is modified in place by the additional of a `data.table` as an `attribute`.
`attr(x, "ci")` is printed.
This A `survfit` object. The `upper` and `lower` elements in the list (representing confidence intervals) are modified from the original.
Other elements will also be shortened if the time range under consideration has been reduced from the original.

Note

- For the Nair and Hall-Wellner bands, the function currently relies on the lookup tables in `package:km.ci`.
- Generally, the arcsin-square root transform has the best coverage properties.
- All bands have good coverage properties for samples as small as $n = 20$, except for the **Nair / EP** bands with a linear transformation, which perform poorly when $n < 200$.

Source

The function is loosely based on `km.ci::km.ci`.

References

Nair V, 1984. Confidence bands for survival functions with censored data: a comparative study. *Technometrics*. **26**(3):265-75. JSTOR.

Hall WJ, Wellner JA, 1980. Confidence bands for a survival curve from censored data. *Biometrika*. **67**(1):133-43. JSTOR.

See Also

`sf`

`quantile`

Examples

```
## K&M 2nd ed. Section 4.3. Example 4.2, pg 105.
data("bmt", package="KMsurv")
b1 <- bmt[bmt$group==1, ] # ALL patients
## K&M 2nd ed. Section 4.4. Example 4.2 (cont.), pg 111.
## patients with ALL
t1 <- ten(Surv(t2, d3) ~ 1, data=bmt[bmt$group==1, ])
ci(t1, how="nair", trans="lin", tL=100, tU=600)
## Table 4.5, pg. 111.
lapply(list("lin", "log", "asi"),
       function(x) ci(t1, how="nair", trans=x, tL=100, tU=600))
## Table 4.6, pg. 111.
lapply(list("lin", "log", "asi"),
       function(x) ci(t1, how="hall", trans=x, tL=100, tU=600))
t1 <- ten(Surv(t2, d3) ~ group, data=bmt)
ci(t1, CI="0.95", how="nair", trans="lin", tL=100, tU=600)
## stratified model
data("pbc", package="survival")
t1 <- ten(coxph(Surv(time, status==2) ~ log(bili) + age + strata(edema), data=pbc))
ci(t1)
```

comp *compare survival curves*

Description

compare survival curves

Usage

```
comp(x, ...)

## S3 method for class 'ten'
comp(x, ..., p = 1, q = 1, scores = seq.int(attr(x, "ncg")),
  reCalc = FALSE)
```

Arguments

x	A tne object
p	p for Fleming-Harrington test
q	q for Fleming-Harrington test
scores	scores for tests for trend
...	Additional arguments (not implemented).
reCalc	Recalcuate the values? If reCalc=FALSE (the default) and the ten object already has the calculated values stored as an attribute, the value of the attribute is returned directly.

Details

The **log-rank** tests are formed from the following elements, with values for each time where there is at least one event:

- W_i, the weights, given below.
- e_i, the number of events (per time).
- $\hat{e_i}$, the number of *predicted* events, given by predict.
- COV_i, the covariance matrix for time i, given by COV.

It is calculated as:

$$Q_i = \sum W_i(e_i - \hat{e}_i)^T \sum W_i C\hat{O}V_i W_i^{-1} \sum W_i(e_i - \hat{e}_i)$$

If there are K groups, then $K-1$ are selected (arbitrary).
Likewise the corresponding variance-covariance matrix is reduced to the appropriate $K-1 \times K-1$ dimensions.
Q is distributed as chi-square with $K-1$ degrees of freedom.

For 2 covariate groups, we can use:

- e_i the number of events (per time).
- n_i the number at risk overall.
- $e1_i$ the number of events in group 1.
- $n1_i$ the number at risk in group 1.

Then:

$$Q = \frac{\sum W_i[e1_i - n1_i(\frac{e_i}{n_i})]}{\sqrt{\sum W_i^2 \frac{n1_i}{n_i}(1 - \frac{n1_i}{n_i})(\frac{n_i - e_i}{n_i - 1})e_i}}$$

Below, for the Fleming-Harrington weights, $\hat{S}(t)$ is the Kaplan-Meier (product-limit) estimator.
Note that both p and q need to be ≥ 0.

The weights are given as follows:

1	log-rank	
n_i	Gehan-Breslow generalized Wilcoxon	
$\sqrt{n_i}$	Tarone-Ware	
$S1_i$	Peto-Peto's modified survival estimate	$\bar{S}(t) = \prod 1 - \frac{e_i}{n_i+1}$
$S2_i$	modified Peto-Peto (by Andersen)	$\tilde{S}(t) = \bar{S} - \frac{n_i}{n_i+1}$
FH_i	Fleming-Harrington	The weight at $t_0 = 1$ and thereafter is: $\hat{S}(t_{i-1})^p[1 - \hat{S}(t_{i-1})^q]$

The **supremum (Renyi)** family of tests are designed to detect differences in survival curves which *cross*.
That is, an early difference in survival in favor of one group is balanced by a later reversal.
The same weights as above are used.
They are calculated by finding

$$Z(t_i) = \sum_{t_k \leq t_i} W(t_k)[e1_k - n1_k \frac{e_k}{n_k}], \quad i = 1, 2, ..., k$$

(which is similar to the numerator used to find Q in the log-rank test for 2 groups above).
and it's variance:

$$\sigma^2(\tau) = \sum_{t_k \leq \tau} W(t_k)^2 \frac{n1_k n2_k (n_k - e_k)e_k}{n_k^2(n_k - 1)}$$

where τ is the largest t where both groups have at least one subject at risk.

Then calculate:

$$Q = \frac{\sup |Z(t)|}{\sigma(\tau)}, \quad t < \tau$$

When the null hypothesis is true, the distribution of Q is approximately

$$Q \sim \sup |B(x)|, \quad 0 \leq x \leq 1$$

And for a standard Brownian motion (Wiener) process:

$$Pr[\sup |B(t)| > x] = 1 - \frac{4}{\pi} \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} \exp \frac{-\pi^2(2k+1)^2}{8x^2}$$

Tests for trend are designed to detect ordered differences in survival curves.
That is, for at least one group:

$$S_1(t) \geq S_2(t) \geq ... \geq S_K(t) \quad t \leq \tau$$

where τ is the largest t where all groups have at least one subject at risk. The null hypothesis is that

$$S_1(t) = S_2(t) = ... = S_K(t) \quad t \leq \tau$$

Scores used to construct the test are typically $s = 1, 2, ..., K$, but may be given as a vector representing a numeric characteristic of the group.
They are calculated by finding:

$$Z_j(t_i) = \sum_{t_i \leq \tau} W(t_i)[e_{ji} - n_{ji}\frac{e_i}{n_i}], \quad j = 1, 2, ..., K$$

The test statistic is:

$$Z = \frac{\sum_{j=1}^{K} s_j Z_j(\tau)}{\sqrt{\sum_{j=1}^{K}\sum_{g=1}^{K} s_j s_g \sigma_{jg}}}$$

where σ is the the appropriate element in the variance-covariance matrix (see COV).
If ordering is present, the statistic Z will be greater than the upper α-th percentile of a standard normal distribution.

Value

The `tne` object is given additional `attributes`.
The following are always added:

`lrt`	The **l**og-**r**ank family of **t**ests
`lrw`	The **l**og-**r**ank **w**eights (used in calculating the tests).

An additional item depends on the number of covariate groups.
If this is $= 2$:

`sup`	The **sup**remum or Renyi family of tests

and if this is > 2:

`tft`	Tests for trend. This is given as a `list`, with the statistics and the scores used.

Note

Regarding the Fleming-Harrington weights:

- $p = q = 0$ gives the log-rank test, i.e. $W = 1$
- $p = 1, q = 0$ gives a version of the Mann-Whitney-Wilcoxon test (tests if populations distributions are identical)
- $p = 0, q > 0$ gives more weight to differences later on
- $p > 0, q = 0$ gives more weight to differences early on

The example using `alloauto` data illustrates this. Here the log-rank statistic has a p-value of around 0.5 as the late advantage of allogenic transplants is offset by the high early mortality. However using Fleming-Harrington weights of $p = 0, q = 0.5$, emphasising differences later in time, gives a p-value of 0.04.
Stratified models (`stratTen`) are *not* yet supported.

References

Gehan A. A Generalized Wilcoxon Test for Comparing Arbitrarily Singly-Censored Samples. Biometrika 1965 Jun. 52(1/2):203–23. JSTOR

Tarone RE, Ware J 1977 On Distribution-Free Tests for Equality of Survival Distributions. *Biometrika*;**64**(1):156–60. JSTOR

Peto R, Peto J 1972 Asymptotically Efficient Rank Invariant Test Procedures. *J Royal Statistical Society* **135**(2):186–207. JSTOR

Fleming TR, Harrington DP, O'Sullivan M 1987 Supremum Versions of the Log-Rank and Generalized Wilcoxon Statistics. *J American Statistical Association* **82**(397):312–20. JSTOR

Billingsly P 1999 *Convergence of Probability Measures.* New York: John Wiley & Sons. Wiley (paywall)

Examples

```
## Two covariate groups
## K&M 2nd ed. Example 7.2, Table 7.2, pp 209--210.
data("kidney", package="KMsurv")
t1 <- ten(Surv(time=time, event=delta) ~ type, data=kidney)
comp(t1, p=c(0, 1, 1, 0.5, 0.5), q=c(1, 0, 1, 0.5, 2))
## see the weights used
attributes(t1)$lrw
## supremum (Renyi) test; two-sided; two covariate groups
## K&M 2nd ed. Example 7.9, pp 223--226.
data("gastric", package="survMisc")
g1 <- ten(Surv(time, event) ~ group, data=gastric)
comp(g1)
## Three covariate groups
## K&M 2nd ed. Example 7.4, pp 212-214.
data("bmt", package="KMsurv")
b1 <- ten(Surv(time=t2, event=d3) ~ group, data=bmt)
comp(b1, p=c(1, 0, 1), q=c(0, 1, 1))
## Tests for trend
## K&M 2nd ed. Example 7.6, pp 217-218.
data("larynx", package="KMsurv")
l1 <- ten(Surv(time, delta) ~ stage, data=larynx)
comp(l1)
attr(l1, "tft")
### see effect of F-H test
data("alloauto", package="KMsurv")
a1 <- ten(Surv(time, delta) ~ type, data=alloauto)
comp(a1, p=c(0, 1), q=c(1, 1))
```

COV ***cov**ariance matrix for survival data*

Description

covariance matrix for survival data

Usage

```
COV(x, ...)

## S3 method for class 'ten'
COV(x, ..., reCalc = FALSE)

## S3 method for class 'stratTen'
COV(x, ..., reCalc = FALSE)

## S3 method for class 'numeric'
COV(x, ..., n, ncg)
```

Arguments

x	A `numeric` vector of *number of events*, e_t. These are assumed to be ordered by discrete times. A method is available for objects of `class` `ten`.
...	Additional arguments (not implemented).
reCalc	Recalcuate the values? If `reCalc=FALSE` (the default) and the `ten` object already has the calculated values stored as an `attribute`, the value of the `attribute` is returned directly.
	–Arguments for the numeric method:
n	**n**umber at risk (total).
ncg	**n**umber at risk, per **c**ovariate **g**roup. If there are 2 groups, this can be given as a `vector` with the number at risk for group 1. If there are ≥ 2 groups, it is a `matrix` with one column for each group.

Details

Gives variance-covariance matrix for comparing survival data for two or more groups.
Inputs are vectors corresponding to observations at a set of discrete time points for right censored data, except for $n1$, the no. at risk by predictor.
This should be specified as a vector for one group, otherwise as a matrix with each column corresponding to a group.

Value

An `array`.
The first two dimensions = the number of covariate groups K, $k = 1, 2, \ldots K$. This is the square matrix below.
The third dimension is the number of observations (discrete time points).

To calculate this, we use `x` (= e_t below) and n_1, the number at risk in covariate group 1.
Where there are 2 groups, the resulting sparse square matrix (i.e. the non-diagonal elements are 0) at time t has diagonal elements:

$$cov_t = -\frac{n_{0t} n_{1t} e_t (n_t - e_t)}{n_t^2 (n_t - 1)}$$

For ≥ 2 groups, the resulting square matrix has diagonal elements given by:

$$cov_{kkt} = \frac{n_{kt}(n_t - n_{kt}) e_t (n_t - e_t)}{n_t^2 (n_t - 1)}$$

The off diagonal elements are:

$$cov_{klt} = \frac{-n_{kt} n_{lt} e_t (n_t - e_t)}{n_t^2 (n_t - 1)}$$

Note

Where the is just one subject at risk $n = 1$ at the final timepoint, the equations above may produce `NaN` due to division by zero. This is converted to `0` for simplicity.

See Also

Called by `comp`

The name of the function is capitalized to distinguish it from:
?stats::cov

Examples

```
## Two covariate groups
## K&M. Example 7.2, pg 210, table 7.2 (last column).
data("kidney", package="KMsurv")
k1 <- with(kidney,
           ten(Surv(time=time, event=delta) ~ type))
COV(k1)[COV(k1) > 0]
## Four covariate groups
## K&M. Example 7.6, pg 217.
data("larynx", package="KMsurv")
l1 <- ten(Surv(time, delta) ~ stage, data=larynx)
rowSums(COV(l1), dims=2)
## example of numeric method
## Three covariate groups
## K&M. Example 7.4, pg 212.
data("bmt", package="KMsurv")
b1 <- asWide(ten(Surv(time=t2, event=d3) ~ group, data=bmt))
rowSums(b1[, COV(x=e, n=n, ncg=matrix(data=c(n_1, n_2, n_3), ncol=3))], dims=2)
```

cutp — ***cut p**oint for a continuous variable in a model fit with* `coxph` *or* `survfit`.

Description

Determine the optimal cut point for a continuous variable in a `coxph` or `survfit` model.

Usage

```
cutp(x, ...)

## S3 method for class 'coxph'
cutp(x, ..., defCont = 3)

## S3 method for class 'survfit'
cutp(x, ..., defCont = 3)
```

Arguments

`x`	A `survfit` or `coxph` object
`defCont`	**def**inition of a **cont**inuous variable. If the variable has $>$ `defCont` unique values, it is treated as continuous and a cut point is determined.
`...`	Additional arguments (not implemented).

Details

For a cut point μ, of a predictor K, the variable is split into two groups, those $\geq \mu$ and those $< \mu$. The score (or log-rank) statistic, sc, is calculated for each unique element k in K and uses

- e_i^+ the number of events
- n_i^+ the number at risk

in those above the cut point, respectively.
The basic statistic is

$$sc_k = \sum_{i=1}^{D} (e_i^+ - n_i^+ \frac{e_i}{n_i})$$

The sum is taken across times with observed events, to D, the largest of these.
It is normalized (standardized), in the case of censoring, by finding σ^2 which is:

$$\sigma^2 = \frac{1}{D-1} \sum_{i}^{D} (1 - \sum_{j=1}^{i} \frac{1}{D+1-j})^2$$

The test statistic is then

$$Q = \frac{\max |sc_k|}{\sigma \sqrt{D-1}}$$

Under the null hypothesis that the chosen cut point does *not* predict survival, the distribution of Q has a limiting distibution which is the supremum of the absolute value of a Brownian bridge:

$$p = Pr(\sup Q \geq q) = 2 \sum_{i=1}^{\infty} (-1)^{i+1} \exp(-2i^2 q^2)$$

Value

A `list` of `data.tables`.
There is one list element per continuous variable.
Each has a column with possible values of the cut point (i.e. unique values of the variable), and the additional columns:

`U`	The score (log-rank) test for a model with the variable 'cut' into into those $\geq$ the cutpoint and those below.
`Q`	The test statistic.
`p`	The p-value.

The tables are ordered by p-value, lowest first.

References

Contal C, O'Quigley J, 1999. An application of changepoint methods in studying the effect of age on survival in breast cancer. *Computational Statistics & Data Analysis* **30**(3):253–70. ScienceDirect (paywall)

Mandrekar JN, Mandrekar, SJ, Cha SS, 2003. Cutpoint Determination Methods in Survival Analysis using SAS. *Proceedings of the 28th SAS Users Group International Conference (SUGI)*. Paper 261-28. SAS (free)

Examples

```
## Mandrekar et al. above
data("bmt", package="KMsurv")
b1 <- bmt[bmt$group==1, ] # ALL patients
c1 <- coxph(Surv(t2, d3) ~ z1, data=b1) # z1=age
c1 <- cutp(c1)$z1
data.table::setorder(c1, "z1")
## [] below is used to print data.table to console
c1[]

## Not run:
## compare to output from survival::coxph
matrix(
    unlist(
        lapply(26:30,
               function(i) c(i, summary(coxph(Surv(t2, d3) ~ z1 >= i, data=b1))$sctest))),
    ncol=5,
    dimnames=list(c("age", "score_test", "df", "p")))
cutp(coxph(Surv(t2, d3) ~ z1, data=bmt[bmt$group==2, ]))$z1[]
cutp(coxph(Surv(t2, d3) ~ z1, data=bmt[bmt$group==3, ]))[[1]][]
```

```
## K&M. Example 8.3, pg 273-274.
data("kidtran", package="KMsurv")
k1 <- kidtran
## patients who are male and black
k2 <- k1[k1$gender==1 & k1$race==2, ]
c2 <- coxph(Surv(time, delta) ~ age, data=k2)
print(cutp(c2))
## check significance of computed value
summary(coxph(Surv(time, delta) ~ age >= 58, data=k2))
k3 <- k1[k1$gender==2 & k1$race==2, ]
c3 <- coxph(Surv(time, delta) ~ age, data=k3)
print(cutp(c3))
## doesn't apply to binary variables e.g. gender
print(cutp(coxph(Surv(time, delta) ~ age + gender, data=k1)))

## End(Not run)
```

`gastric` *gastric cancer trial data*

Description

gastric cancer trial data

Format

A `data.frame` with 90 rows (observations) and 3 columns (variables).

Details

Data from a trial of locally unresectable gastic cancer.
Patients ($n = 45$ in each group) were randomized to one of two groups: chemotheapy vs. chemotherapy + radiotherapy.
Columns are:

time Time, in days

event Death

group Treatment

- **0** chemotherapy
- **1** chemotherapy + radiotherapy

Source

Klein J, Moeschberger. Survival Analysis, 2nd edition. Springer 2003. Example 7.9, pg 224.

References

Gastrointestinal Tumor Study Group, 1982. A comparison of combination chemotherapy and combined modality therapy for locally advanced gastric carcinoma. *Cancer*. **49**(9):1771-7. Wiley (free).

Stablein DM, Koutrouvelis IA, 1985. A two-sample test sensitive to crossing hazards in uncensored and singly censored data. *Biometrics*. **41**(3):643-52. JSTOR.

See Also

Examples in comp

Examples

```
data("gastric", package="survMisc", verbose=TRUE)
head(gastric)
```

---

gof *goodness of fit test for a* coxph *object*

Description

goodness **o**f **f**it test for a coxph object

Usage

```
gof(x, ...)

## S3 method for class 'coxph'
gof(x, ..., G = NULL)
```

Arguments

x	An object of class coxph
...	Additional arguments (not implemented)
G	Number of **g**roups into which to divide risk score. If G=NULL (the default), uses closest integer to $$G = \max(2, \quad \min(10, \quad \frac{ne}{40}))$$ where ne is the number of events overall.

Details

In order to verify the overall goodness of fit, the risk score r_i for each observation i is given by

$$r_i = \hat{\beta} X_i$$

where $\hat{\beta}$ is the vector of fitted coefficients and X_i is the vector of predictor variables for observation i.
This risk score is then sorted and 'lumped' into a grouping variable with G groups, (containing approximately equal numbers of observations).
The number of observed (e) and expected (exp) events in each group are used to generate a Z statistic for each group, which is assumed to follow a normal distribution with $Z \sim N(0, 1)$.
The indicator variable `indicG` is added to the original model and the two models are compared to determine the improvement in fit via the likelihood ratio test.

Value

A `list` with elements:

`groups` A `data.table` with one row per group G. The columns are

n Number of observations

e Number of events

exp Number of events expected. This is

$$exp = \sum e_i - M_i$$

where e_i are the events and M_i are the martingale residuals for each observation i

z Z score, calculated as

$$Z = \frac{e - exp}{\sqrt{exp}}$$

p p-value for Z, which is

$$p = 2.\texttt{pnorm}(-|z|)$$

where `pnorm` is the normal distribution function with mean $\mu = 0$ and standard deviation $\sigma = 1$ and $|z|$ is the absolute value.

`lrTest` Likelihood-ratio test. Tests the improvement in log-likelihood with addition of an indicator variable with $G-1$ groups. This is done with `survival:::anova.coxph`. The test is distributed as chi-square with $G - 1$ degrees of freedom

Note

The choice of G is somewhat arbitrary but rarely should be > 10.
As illustrated in the example, a larger value for G makes the Z test for each group more likely to be significant. This does *not* affect the significance of adding the indicator variable `indicG` to the original model.

The Z score is chosen for simplicity, as for large sample sizes the Poisson distribution approaches the normal. Strictly speaking, the Poisson would be more appropriate for e and exp as per Counting Theory.

The Z score may be somewhat conservative as the expected events are calculated using the martingale residuals from the overall model, rather than by group. This is likely to bring the expected events closer to the observed events.

This test is similar to the Hosmer-Lemeshow test for logistic regression.

Source

Method and example are from:
May S, Hosmer DW 1998. A simplified method of calculating an overall goodness-of-fit test for the Cox proportional hazards model. *Lifetime Data Analysis* **4**(2):109–20. Springer (paywall)

References

Default value for G as per:
May S, Hosmer DW 2004. A cautionary note on the use of the Gronnesby and Borgan goodness-of-fit test for the Cox proportional hazards model. *Lifetime Data Analysis* **10**(3):283–91. Springer (paywall)

Changes to the pbc dataset in the example are as detailed in:
Fleming T, Harrington D 2005. *Counting Processes and Survival Analysis*. New Jersey: Wiley and Sons. Chapter 4, section 4.6, pp 188. Wiley (paywall)

Examples

```
data("pbc", package="survival")
pbc <- pbc[!is.na(pbc$trt), ]
## make corrections as per Fleming
pbc[pbc$id==253, "age"] <-  54.4
pbc[pbc$id==107, "protime"] <-  10.7
### misspecified; should be log(bili) and log(protime) instead
c1 <- coxph(Surv(time, status==2) ~
            age + log(albumin) + bili + edema + protime,
            data=pbc)
gof(c1, G=10)
gof(c1)
```

---

nc *Add **n**umber **c**ensored.*

Description

Add **n**umber **c**ensored.

Usage

```
nc(x, ...)

## S3 method for class 'ten'
```

```
nc(x, ...)

## S3 method for class 'stratTen'
nc(x, ...)
```

Arguments

x	An object of class ten or stratTen.
...	Additional arguments (not implemented).

Value

The original object, with new column(s) added indicating the number censored at each time point, depending on attr(x, "shape"):

"long"	the new column, c, gives the number censored at each timepoint, by covariate group.
"wide"	new columns, beginning with c_, give the number censored at each timepoint, by covariate group. There is an additional nc column giving the *total* number censored at each timepoint.

A stratTen object has each ten element in the list modified as above.

Examples

```
data("kidney", package="KMsurv")
t1 <- ten(survfit(Surv(time, delta) ~ type, data=kidney))
nc(t1)
nc(asWide(t1))
## stratified model
data("pbc", package="survival")
t1 <- ten(coxph(Surv(time, status==2) ~ log(bili) + age + strata(edema), data=pbc))
nc(t1)
```

---

plotSurv *Plot a* Surv *object.*

Description

Plot a Surv object.

Usage

```
## S3 method for class 'Surv'
plot(x, l = 3, ...)
```

Arguments

x	A Surv object
l	length of arrow. Length is l / nrow(x)
...	Additional arguments. These are passed to as ... (an ellipsis) to the following functions, respectively: graphics::arrows for plotting right- or left-censored observations graphics::segments for plotting interval-censored observations

Value

A graph (base graphics). The type of graph depends on the type of the Surv object. This is given by attr(s, which="type") :

counting	Lines with an arrow pointing right if right censored.
right	Lines with an arrow pointing right if right censored.
left	Lines with an arrow pointing left if left censored.
interval	If censored: **arrow points right** right censored **arrow points left** left censored If not censored: **lines** observations of more than one time point **points** observation of one time only (i.e. start and end times are the same)

See Also

?graphics::arrows

?graphics::segments

Examples

```
df0 <- data.frame(t1=c(0, 2, 4, 6, NA, NA, 12, 14),
                  t2=c(NA, NA, 4, 6, 8, 10, 16, 18))
s5 <- Surv(df0$t1, df0$t2, type="interval2")
plot(s5)
```

---

predict *predicted events*

Description

predicted events

Usage

```
## S3 method for class 'ten'
predict(object, ..., eMP = TRUE, reCalc = FALSE)
```

Arguments

object	An object of class ten.
eMP	Add column(s) indicating **e**vents **m**inus **p**redicted.
...	Additional arguments (not implemented).
reCalc	Recalcuate the values? If reCalc=FALSE (the default) and the ten object already has the calculated values stored as an attribute, the value of the attribute is returned directly.

Details

With K covariate groups, We use ncg_{ik}, the number at risk for group k, to calculate the number of expected events:

$$P_{ik} = \frac{e_i(ncg_{ik})}{n_i} \quad k = 1, 2 \ldots K$$

Value

An attribute, pred is added to object:

t	Times with at least one observation
P_	**p**redicted number of events

And if eMP==TRUE (the default):

eMP_	**e**vents **m**inus **p**redicted

The names of the object's covariate groups are used to make the suffixes of the column names (i.e. after the _ character).

Note

There is a predicted value for each unique time, for each covariate group.

See Also

?survival::predict.coxph methods("predict")

Examples

```
## K&M. Example 7.2, Table 7.2, pp 209-210.
data("kidney", package="KMsurv")
k1 <- ten(Surv(time=time, event=delta) ~ type, data=kidney)
predict(k1)
predict(asWide(k1))
stopifnot(predict(asWide(k1))[, sum(eMP_1 + eMP_2)] <=
```

```
          .Machine$double.neg.eps)
## Three covariate groups
## K&M. Example 7.4, pp 212-214.
data("bmt", package="KMsurv")
b1 <- ten(Surv(time=t2, event=d3) ~ group, data=bmt)
predict(b1)
## one group only
predict(ten(Surv(time=t2, event=d3) ~ 1, data=bmt))
```

---

print *print methods*

Description

print methods

Usage

```
## S3 method for class 'ten'
print(x, ..., maxRow = getOption("datatable.print.nrows", 50L),
  nRowP = getOption("datatable.print.topn", 5L), pRowNames = TRUE,
  maxCol = getOption("survMisc.maxCol", 8L),
  nColSP = getOption("survMisc.nColSP", 7L),
  sigDig = getOption("survMisc.sigDig", 2L))

## S3 method for class 'COV'
print(x, ..., n = 2L)

## S3 method for class 'lrt'
print(x, ..., dist = c("n", "c"))

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

## S3 method for class 'tableAndPlot'
print(x, ..., hideTabLeg = TRUE, tabHeight = 0.25)

## S3 method for class 'stratTableAndPlot'
print(x, ..., hideTabLeg = TRUE,
  tabHeight = 0.25)
```

Arguments

x	An object of class ten.
...	Additional arguments (not implemented).

maxRow	Maximum number of rows to print. If nrow(x) > maxRow, just the first and last nRowP (below) are printed. The default value is that used by data.table.
nRowP	**N**umber of rows to **p**rint from the start and end of the object. Used if nrow(x) > maxRow.
pRowNames	Print row names? Default is TRUE.
maxCol	Maximum number of columns to print. If ncol(x) > maxCol, just the first nColSP and last maxCol - nColSP columns are printed.
nColSP	**N**umber of **col**umns to **p**rint from the **s**tart of the object. Used if Used if ncol(x) > maxCol.
sigDig	**Sig**nificant **dig**its. This is passed as an argument to ?signif when preparing the object for printing.
	–print.tableAndPlot and **print.tableAndPlot**
hideTabLeg	Hide table legend.
tabHeight	Table height (relative to whole plot).
	–print.COV
n	Similar to n from e.g. ?utils::head
	–print.lrt
dist	Which distribution to use for the statistics when printing. Default (dist="n") prints Z and p values based on the normal distribution. If dist="c", gives values based on the χ^2 distribution. The results are the same. The default value is typically easier to read. Both options are given for completeness.

Details

Prints a ten object with 'nice' formatting.
Options may be set for a session using e.g.
options(survMisc.nColSP=4L)
It is similar to the behavior of print.data.table but has additional arguments controlling the number of columns sent to the terminal.

Value

A printed representation of the object is send to the terminal as a *side effect* of calling the function. The return value cannot be assigned.

Note

All numeric arguments to the function must be supplied as integers.

Author(s)

Chris Dardis. Based on existing work by Brian Diggs.

See Also

For `print.ten`:

data.table:::print.data.table

?stats::printCoefmat

options()$datatable.print.nrows

sapply(c("datatable.print.nrows", "datatable.print.topn"), getOption)

Examples

```
set.seed(1)
(x <- data.table::data.table(matrix(rnorm(1800), ncol=15, nrow=120)))
data.table::setattr(x, "class", c("ten", class(x)))
p1 <- print(x)
stopifnot(is.null(p1))
x[1:80, ]
x[0, ]
(data.table::set(x, j=seq.int(ncol(x)), value=NULL))
```

`profLik` *Profile likelihood for coefficients in a* `coxph` *model*

Description

Profile likelihood for coefficients in a coxph model

Usage

```
profLik(x, CI = 0.95, interval = 50, mult = c(0.1, 2), devNew = TRUE,
  ...)
```

Arguments

`x`	A coxph model.
`CI`	**C**onfidence **I**nterval.
`interval`	Number of points over which to evaluate coefficient.
`mult`	**Mult**iplier. Coefficent will be multiplied by lower and upper value and evaluated across this range.
`devNew`	Open a new device for each plot. See ?grDevices::dev.new
`...`	Additional parameters passed to `graphics::plot.default`.

Details

Plots of range of values for coefficient in model with log-likelihoods for the model with the coefficient fixed at these values.

For each coefficient a range of possible values is chosen, given by $\hat{B} * mult_{lower} - \hat{B} * mult_{upper}$. A series of models are fit (given by `interval`). The coefficient is included in the model as a *fixed* term and the partial log-likelihood for the model is calculated.

A curve is plotted which gives the partial log-likelihood for each of these candidate values. An appropriate confidence interval (CI) is given by subtracting 1/2 the value of the appropriate quantile of a chi-squared distribution with 1 degree of freedom.

Two circles are also plotted giving the 95

Value

One plot for each coefficient in the model.

References

Example is from: **T&G**. Section 3.4.1, pg 57.

Examples

```
data("pbc", package="survival")
c1 <- coxph(formula = Surv(time, status == 2) ~ age + edema + log(bili) +
                      log(albumin) + log(protime), data = pbc)
profLik(c1, col="red")
```

---

rsq *r^2 measures for a a* coxph *or* survfit *model*

Description

r^2 measures for a a coxph or survfit model

Usage

```
rsq(x, ...)

## S3 method for class 'coxph'
rsq(x, ..., sigD = 2)

## S3 method for class 'survfit'
rsq(x, ..., sigD = 2)
```

Arguments

x	A survfit or coxph object.
sigD	**sig**nificant **d**igits (for ease of display). If sigD=NULL, will return the original numbers.
...	Additional arguments (not implemented).

Value

A list with the following elements:

cod — The **c**oefficient **o**f **d**etermination, which is

$$R^2 = 1 - \exp(\frac{2}{n}L_0 - L_1)$$

where L_0 and L_1 are the log partial likelihoods for the *null* and *full* models respectively and n is the number of observations in the data set.

mer — The **m**easure of **e**xplained **r**andomness, which is:

$$R^2_{mer} = 1 - \exp(\frac{2}{m}L_0 - L_1)$$

where m is the number of observed *events*.

mev — The **m**easure of **e**xplained **v**ariation (similar to that for linear regression), which is:

$$R^2 = \frac{R^2_{mer}}{R^2_{mer} + \frac{\pi}{6}(1 - R^2_{mer})}$$

References

Nagelkerke NJD, 1991. A Note on a General Definition of the Coefficient of Determination. *Biometrika* **78**(3):691–92. JSTOR

O'Quigley J, Xu R, Stare J, 2005. Explained randomness in proportional hazards models. *Stat Med* **24**(3):479–89. Wiley (paywall) Available at UCSD

Royston P, 2006. Explained variation for survival models. *The Stata Journal* **6**(1):83–96. The Stata Journal

Examples

```
data("kidney", package="KMsurv")
c1 <- coxph(Surv(time=time, event=delta) ~ type, data=kidney)
cbind(rsq(c1), rsq(c1, sigD=NULL))
```

sf *survival (or hazard) function based on e and n.*

Description

survival (or hazard) function based on e and n.

Usage

```
sf(x, ...)

## Default S3 method:
sf(x, ..., what = c("S", "H"), SCV = FALSE,
  times = NULL)

## S3 method for class 'ten'
sf(x, ..., what = c("S", "H"), SCV = FALSE, times = NULL,
  reCalc = FALSE)

## S3 method for class 'stratTen'
sf(x, ..., what = c("S", "H"), SCV = FALSE,
  times = NULL, reCalc = FALSE)

## S3 method for class 'numeric'
sf(x, ..., n = NULL, what = c("all", "S", "Sv", "H",
  "Hv"), SCV = FALSE, times = NULL)
```

Arguments

x	One of the following: **default** A numeric vector of events status (assumed sorted by time). **numeric** Vectors of events and numbers at risk (assumed sorted by time). A ten object. A stratTen object.
...	Additional arguments (not implemented).
n	Number at risk.
what	See return, below.
SCV	Include the **S**quared **C**oefficient of **V**ariation, which is calcluated using the mean $\bar{x}$ and the variance σ_x^2: $$SCV_x = \frac{\sigma_x^2}{\bar{x}^2}$$ This measure of *dispersion* is also referred to as the 'standardized variance' or the 'noise'.

`times`	Times for which to calculate the function. If `times=NULL` (the default), times are used for which at least one event occurred in at least one covariate group.
`reCalc`	Recalcuate the values? If `reCalc=FALSE` (the default) and the `ten` object already has the calculated values stored as an `attribute`, the value of the `attribute` is returned directly.

Value

A data.table which is stored as an attribute of the `ten` object.
If `what="s"`, the **s**urvival is returned, based on the Kaplan-Meier or product-limit estimator. This is 1 at $t = 0$ and thereafter is given by:

$$\hat{S}(t) = \prod_{t \leq t_i} (1 - \frac{e_i}{n_i})$$

If `what="sv"`, the **s**urvival **v**ariance is returned.
Greenwoods estimtor of the variance of the Kaplan-Meier (product-limit) estimator is:

$$Var[\hat{S}(t)] = [\hat{S}(t)]^2 \sum_{t_i \leq t} \frac{e_i}{n_i(n_i - e_i)}$$

If `what="h"`, the **h**azard is returned, based on the the Nelson-Aalen estimator. This has a value of $\hat{H} = 0$ at $t = 0$ and thereafter is given by:

$$\hat{H}(t) = \sum_{t \leq t_i} \frac{e_i}{n_i}$$

If `what="hv"`, the **h**azard **v**ariance is returned.
The variance of the Nelson-Aalen estimator is given by:

$$Var[\hat{H}(t)] = \sum_{t_i \leq t} \frac{e_i}{n_i^2}$$

If `what="all"` (the default), *all* of the above are returned in a `data.table`, along with:
Survival, based on the Nelson-Aalen hazard estimator H, which is:

$$\hat{S_{na}} = e^H$$

Hazard, based on the Kaplan-Meier survival estimator S, which is:

$$\hat{H_{km}} = -\log S$$

Examples

```
data("kidney", package="KMsurv")
k1 <- ten(Surv(time=time, event=delta) ~ type, data=kidney)
sf(k1)
sf(k1, times=1:10, reCalc=TRUE)
k2 <- ten(with(kidney, Surv(time=time, event=delta)))
```

```
sf(k2)
## K&M. Table 4.1A, pg 93.
## 6MP patients
data("drug6mp", package="KMsurv")
d1 <- with(drug6mp, Surv(time=t2, event=relapse))
(d1 <- ten(d1))
sf(x=d1$e, n=d1$n, what="S")
data("pbc", package="survival")
t1 <- ten(Surv(time, status==2) ~ log(bili) + age + strata(edema), data=pbc)
sf(t1)
## K&M. Table 4.2, pg 94.
data("bmt", package="KMsurv")
b1 <- bmt[bmt$group==1, ] # ALL patients
t2 <- ten(Surv(time=b1$t2, event=b1$d3))
with(t2, sf(x=e, n=n, what="Hv"))
## K&M. Table 4.3, pg 97.
sf(x=t2$e, n=t2$n, what="all")
```

`survMisc_package` *Miscellaneous Functions for Survival Analysis*

Description

Package:	survMisc
Type:	Package
Version:	0.5
Date:	2015-07-15
License:	GPL (>= 2)
LazyLoad:	yes

A collection of functions for the analysis of survival data. These extend the methods already available in `package:survival`.
The intent is to generate a workspace for some of the common tasks arising in survival analysis.

There are references in many of the functions to the textbooks:

K&M Klein J, Moeschberger M (2003). *Survival Analysis*, 2nd edition. New York: Springer. Springer (paywall).

T&G Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. New York: Springer. Springer (paywall).

Notes for developers:

- This package should be regarded as 'in development' until release 1.0, meaning that there may be changes to certain function names and parameters, although I will try to keep this to a minimum. As such it is recommended that other packages do *not* depend on or import from this one until at least version 1.0.

- Naming tends to follow the **camelCase** convention; variables within functions are typically alphanumeric e.g. `a1 <- 1`.

For bug reports, feature requests or suggestions for improvement, please try to submit to github. Otherwise email me at the address below.

Author(s)

Chris Dardis <christopherdardis@gmail.com>

`ten` *time, event(s) and number at risk.*

Description

time, **e**vent(s) and **n**umber at risk.

Usage

```
ten(x, ...)

## S3 method for class 'numeric'
ten(x, ...)

## S3 method for class 'Surv'
ten(x, ..., call = NULL)

## S3 method for class 'coxph'
ten(x, ..., abbNames = TRUE, contrasts.arg = NULL)

## S3 method for class 'survfit'
ten(x, ..., abbNames = TRUE, contrasts.arg = NULL)

## S3 method for class 'formula'
ten(x, ..., abbNames = TRUE, contrasts.arg = NULL)

## S3 method for class 'data.frame'
ten(x, ..., abbNames = TRUE, contrasts.arg = NULL,
  call = NULL)

## S3 method for class 'data.table'
ten(x, ..., abbNames = TRUE, mm = NULL, call = NULL)

## S3 method for class 'ten'
ten(x, ..., abbNames = NULL, call = NULL)
```

Arguments

x	For the default method, a numeric vector indicating an *event* (or status). Each element indicates whether an event occurred (1) or not (0) for an observation. These are assumed to be ordered by discrete times. This is similar to the event argument for Surv objects. Methods are available for objects of class Surv, survfit, coxph and formula.
abbNames	**Abb**reviate names? If abbNames="TRUE" (the default), the covariate groups are referred to by number. As the names for each covariate group are made by concatenating the predictor names, the full names can become unwieldly. If abbNames="FALSE", the full names are given. In either case, the longNames are given as an attribute of the returned ten object.
contrasts.arg	Methods for handling factors. A list. The names are the names of columns of the model.frame containing factors. The *values* are used as replacement values for the stats::contrasts replacement function. These should be functions (given as character strings) or numeric matrices. This can be passed from survfit, coxph and formula objects to: ?stats::model.matrix
call	Used to pass the call from a formula to the final ten.data.table method.
mm	Used to pass the model.matrix from a formula to the final ten.data.table method.
...	Additional arguments (not implemented).

Value

A data.table with the additional class ten.
By default, the shape returned is 'long' i.e. there is one row for each unique timepoint per covariate group.
The basic form, for a numeric or Surv object, has columns:

t	**t**ime.
e	number of **e**vents.
n	**n**umber at risk.

A survfit, coxph or formula object will have additional columns:

cg	**c**ovariate **g**roup. This is formed by combining the variables; these are separated by a comma ','.
ncg	**n**umber at risk, by **c**ovariate **g**roup

Special terms.

The following are considered 'special' terms in a survival model:

`strata`	For a stratified model, `ten` returns a `list` with one element per strata, which is a `ten` object. This has the class `stratTen`. The name of the list elements are those of the strata in the model.
`cluster`	These terms are dropped.
`tt`	The variable is unchanged. That is, time-transform terms are handled as if the the function `tt(x)` was `identity(x)`.

Attribures.
The returned object will also have the following `attributes`:

`shape`	The default is `"long"` but is changed to `"wide"` when `asWide` is called on the object.
`abbNames`	Abbreviate names?
`longNames`	A `data.table` with two columns, showing the abbrevbiated and full names.
`ncg`	Number of covariate groups
`call`	The call used to generate the object
`mm`	The `model.matrix` used to generate to generate the object, if applicable.

Additional attributes will be added by the following functions:
`sf` `ci`

Note

The methods for `data.frame` (for a model frame) and `data.table` are not typically intended for interactive use.

Currently only binary status and right-censoring are supported.

In stratified models, only one level of stratification is supported (i.e. strata cannot be 'nested' currently).

Partial matching is available for the following arguments, based on the characters in bold:

- **abb**Names
- **con**trasts.arg

See Also

`asWide`

`print`

Examples

```
require("survival")
## binary vector
ten(c(1, 0, 1, 0, 1))
## Surv object
df0 <- data.frame(t=c(1, 1, 2, 3, 5, 8, 13, 21),
                  e=rep(c(0, 1), 4))
s1 <- with(df0, Surv(t, e, type="right"))
ten(s1)
## some awkward values
suppressWarnings(
    s1 <- Surv(time=c(Inf, -1, NaN, NA, 10, 12),
               event=c(c(NA, 1, 1, NaN, Inf, 0.75))))
ten(s1)
## coxph object
## K&M. Section 1.2. Table 1.1, page 2.
data("hodg", package="KMsurv")
hodg <- data.table::data.table(hodg)
data.table::setnames(hodg,
                     c(names(hodg)[!names(hodg) %in%
                                   c("score", "wtime")],
                       "Z1", "Z2"))
c1 <- coxph(Surv(time=time, event=delta) ~ Z1 + Z2,
            data=hodg[gtype==1 && dtype==1, ])
ten(c1)
data("bmt", package="KMsurv")
ten(c1 <- coxph(Surv(t2, d3) ~ z3*z10, data=bmt))
## T&G. Section 3.2, pg 47.
## stratified model
data("pbc", package="survival")
c1 <- coxph(Surv(time, status==2) ~ log(bili) + age + strata(edema), data=pbc)
ten(c1)
## K&M. Example 7.2, pg 210.
data("kidney", package="KMsurv")
with(kidney[kidney$type==2, ], ten(Surv(time=time, event=delta)))
s1 <- survfit(Surv(time=time, event=delta) ~ type, data=kidney)
ten(s1)[e > 0, ]
## A null model is passed to ten.Surv
(t1 <- with(kidney, ten(Surv(time=time, event=delta) ~ 0)))
## but the original call is preserved
attr(t1, "call")
## survival::survfit doesn't accept interaction terms...
## Not run:
    s1 <- survfit(Surv(t2, d3) ~ z3*z10, data=bmt)
## End(Not run)
## but ten.formula does:
ten(Surv(time=t2, event=d3) ~ z3*z10, data=bmt)
## the same is true for the '.' (dot operator) in formulas
(t1 <- ten(Surv(time=t2, event=d3) ~ ., data=bmt))
## impractical long names stored as an attribute
attr(t1, "longNames")
## not typically intended to be called directly
```

```
mf1 <- model.frame(Surv(time, status==2) ~ age + strata(edema) + strata(spiders), pbc,
                   drop.unused.levels = TRUE)
ten(mf1)
```

Index