Package 'simPH'

October 4, 2014

Title Tools for Simulating and Plotting Quantities of Interest Estimated From Cox Proportional Hazards Models

Description The package simulates and plots quantities of interest (relative hazards, first differences, and hazard ratios) for linear coefficients,multiplicative interactions, polynomials, penalised splines, and non-proportional hazards, as well as stratified survival curves from Cox Proportional Hazard models. It also simulates and plots marginal effects for multiplicative interactions.

Version 1.2.3

Date 2014-10-01

Author Christopher Gandrud

Maintainer Christopher Gandrud <christopher.gandrud@gmail.com>

URL http://christophergandrud.github.io/simPH/

BugReports https://github.com/christophergandrud/simPH/issues

Depends R (>= 3.0.2)

License GPL-3

Imports car, DataCombine, data.table (>= 1.9.2), dplyr (>= 0.3),ggplot2, gridExtra, lazyeval, MASS, mgcv, reshape2, stringr,survival, quadprog

Suggests knitr, stats

VignetteBuilder knitr

BuildVignettes true

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2014-10-04 09:15:46

R topics documented:

CarpenterFdaData . . . 2
coxsimInteract . . . 3
coxsimLinear . . . 5
coxsimPoly . . . 7
coxsimSpline . . . 9
coxsimtvc . . . 12
ggfitStrata . . . 14
GolubEUPData . . . 16
MinMaxLines . . . 16
setXl . . . 17
simGG . . . 18
simGG.siminteract . . . 19
simGG.simlinear . . . 21
simGG.simpoly . . . 24
simGG.simspline . . . 26
simGG.simtvc . . . 29
simPH . . . 32
SurvExpand . . . 33
tvc . . . 34

Index **36**

`CarpenterFdaData` *A data set from Carpenter (2002).*

Description

A data set from Carpenter (2002).

Format

A data set with 408 observations and 32 variables.

Source

Carpenter, Daniel P. 2002. "Groups, the Media, Agency Waiting Costs, and FDA Drug Approval." American Journal of Political Science 46(3): 490-505.

Luke Keele, "Replication data for: Proportionally Difficult: Testing for Nonproportional Hazards In Cox Models". http://hdl.handle.net/1902.1/17068. V1 [Version].

coxsimInteract *Simulate quantities of interest for linear multiplicative interactions */ from Cox Proportional Hazards models*

Description

coxsimInteract simulates quantities of interest for linear multiplicative interactions using multivariate normal distributions. These can be plotted with simGG.

Usage

```
coxsimInteract(obj, b1, b2, qi = "Marginal Effect", X1 = NULL, X2 = NULL,
  means = FALSE, expMarg = TRUE, nsim = 1000, ci = 0.95, spin = FALSE,
  extremesDrop = TRUE)
```

Arguments

obj	a coxph class fitted model object with a linear multiplicative interaction.
b1	character string of the first constitutive variable's name. Note b1 and b2 must be entered in the order in which they are entered into the coxph model.
b2	character string of the second constitutive variable's name.
qi	quantity of interest to simulate. Values can be "Marginal Effect", "First Difference", "Hazard Ratio", and "Hazard Rate". The default is qi = "Hazard Ratio". If qi = "Hazard Rate" and the coxph model has strata, then hazard rates for each strata will also be calculated.
X1	numeric vector of fitted values of b1 to simulate for. If qi = "Marginal Effect" then only X2 can be set. If you want to plot the results, X1 should have more than one value.
X2	numeric vector of fitted values of b2 to simulate for.
means	logical, whether or not to use the mean values to fit the hazard rate for covaraiates other than b1 b2 and b1*b2. Note: it does not currently support models that include polynomials created by I. Note: EXPERIMENTAL. lines are not currently supported in simGG if means = TRUE.
expMarg	logical. Whether or not to exponentiate the marginal effect.
nsim	the number of simulations to run per value of X. Default is nsim = 1000.
ci	the proportion of middle simulations to keep. The default is ci = 0.95, i.e. keep the middle 95 percent. If spin = TRUE then ci is the confidence level of the shortest probability interval. Any value from 0 through 1 may be used.
spin	logical, whether or not to keep only the shortest probability interval rather than the middle simulations. Currently not supported for hazard rates.
extremesDrop	logical whether or not to drop simulated quantity of interest values that are Inf, NA, NaN and > 1000000 for spin = FALSE or > 800 for spin = TRUE. These values are difficult to plot simGG and may prevent spin from finding the central interval.

Details

Simulates marginal effects, first differences, hazard ratios, and hazard rates for linear multiplicative interactions. Marginal effects are calculated as in Brambor et al. (2006) with the addition that we take the exponent, so that it resembles a hazard ratio. You can choose not to take the exponent by setting the argument `expMarg = FALSE`. For an interaction between variables X and Z the marginal effect for X is:

$$ME_X = e^(\beta_X + \beta_{XZ}Z)$$

Note that for First Differences the comparison is not between two values of the same variable but two values of the constitute variable and 0 for the two variables.

Value

a `siminteract` class object

References

Brambor, Thomas, William Roberts Clark, and Matt Golder. 2006. "Understanding Interaction Models: Improving Empirical Analyses." Political Analysis 14(1): 63-82.

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." American Journal of Political Science 44(2): 347-61.

Liu, Ying, Andrew Gelman, and Tian Zheng. 2013. "Simulation-Efficient Shortest Probability Intervals." Arvix. http://arxiv.org/pdf/1302.2142v1.pdf.

See Also

simGG, survival, strata, and coxph,

Examples

```
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
M1 <- coxph(Surv(acttime, censor) ~ lethal*prevgenx,
            data = CarpenterFdaData)

# Simulate Marginal Effect of lethal for multiple
# values of prevgenx
Sim1 <- coxsimInteract(M1, b1 = "lethal", b2 = "prevgenx",
                       X2 = seq(2, 115, by = 5), spin = TRUE)

## Not run:
# Change the order of the covariates to make a more easily
# interpretable relative hazard graph.
M2 <- coxph(Surv(acttime, censor) ~ prevgenx*lethal +
             orphdum, data = CarpenterFdaData)
```

```
# Simulate Hazard Ratio of lethal for multiple
# values of prevgenx
Sim2 <- coxsimInteract(M2, b1 = "prevgenx", b2 = "lethal",
                   X1 = seq(2, 115, by = 2),
                   X2 = c(0, 1),
                   qi = "Hazard Ratio", ci = 0.9)

# Simulate First Difference
Sim3 <- coxsimInteract(M2, b1 = "prevgenx", b2 = "lethal",
                       X1 = seq(2, 115, by = 2),
                       X2 = c(0, 1),
                       qi = "First Difference", spin = TRUE)

# Simulate Hazard Rate
Sim4 <- coxsimInteract(M2, b1 = "prevgenx", b2 = "lethal",
                       X1 = 90, X2 = 1, qi = "Hazard Rate",
                       means = TRUE)

## End(Not run)
```

`coxsimLinear` *Simulate quantities of interest for covariates from Cox Proportional Hazards models that are not interacted with time or nonlinearly transformed*

Description

Simulates relative hazards, first differences, hazard ratios, and hazard rates for linear, non-time interacted covariates from Cox Proportional Hazard models. These can be plotted with `simGG`.

Usage

```
coxsimLinear(obj, b, qi = "Relative Hazard", Xj = NULL, Xl = NULL,
  means = FALSE, nsim = 1000, ci = 0.95, spin = FALSE,
  extremesDrop = TRUE)
```

Arguments

`obj`	a `coxph` class fitted model object.
`b`	character string name of the coefficient you would like to simulate.
`qi`	quantity of interest to simulate. Values can be `"Relative Hazard"`, `"First Difference"`, `"Hazard Ratio"`, and `"Hazard Rate"`. The default is `qi = "Relative Hazard"`. If `qi = "Hazard Rate"` and the coxph model has strata, then hazard rates for each strata will also be calculated.
`Xj`	numeric vector of fitted values for `b` to simulate for.
`Xl`	numeric vector of values to compare `Xj` to. Note if `code = "Relative Hazard"` only `Xj` is relevant.

means	logical, whether or not to use the mean values to fit the hazard rate for covariates other than b. Note: EXPERIMENTAL. lines are not currently supported in simGG if means = TRUE.
nsim	the number of simulations to run per value of X. Default is nsim = 1000. Note: it does not currently support models that include polynomials created by I.
ci	the proportion of simulations to keep. The default is ci = 0.95, i.e. keep the middle 95 percent. If spin = TRUE then ci is the confidence level of the shortest probability interval. Any value from 0 through 1 may be used.
spin	logical, whether or not to keep only the shortest probability interval rather than the middle simulations. Currently not supported for Hazard Rates.
extremesDrop	logical whether or not to drop simulated quantity of interest values that are Inf, NA, NaN and $>$ 1000000 for spin = FALSE or $>$ 800 for spin = TRUE. These values are difficult to plot simGG and may prevent spin from finding the central interval.

Details

coxsimLinear simulates relative hazards, first differences, and hazard ratios for linear covariates that are not interacted with time or nonlinearly transformed from models estimated with coxph using the multivariate normal distribution. These can be plotted with simGG.

Value

a simlinear object

References

Licht, Amanda A. 2011. "Change Comes with Time: Substantive Interpretation of Nonproportional Hazards in Event History Analysis." Political Analysis 19: 227-43.

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." American Journal of Political Science 44(2): 347-61.

Liu, Ying, Andrew Gelman, and Tian Zheng. 2013. "Simulation-Efficient Shortest Probability Intervals." Arvix. http://arxiv.org/pdf/1302.2142v1.pdf.

See Also

simGG, survival, strata, and coxph

Examples

```
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
M1 <- coxph(Surv(acttime, censor) ~ prevgenx + lethal +
```

```
            deathrt1 + acutediz + hosp01  + hhosleng +
            mandiz01 + femdiz01 + peddiz01 + orphdum +
            vandavg3 + wpnoavg3 + condavg3 + orderent +
            stafcder, data = CarpenterFdaData)

# Simulate Hazard Ratios
Sim1 <- coxsimLinear(M1, b = "stafcder",
                     Xj = c(1237, 1600),
                     Xl = c(1000, 1000),
                     qi = "Hazard Ratio",
                     spin = TRUE, ci = 0.99)

## Not run:
# Simulate Hazard Rates
Sim2 <- coxsimLinear(M1, b = "stafcder",
                      Xj = 1237,
                      ci = 0.99)

## End(Not run)
```

`coxsimPoly` *Simulate quantities of interest for a range of values for a polynomial nonlinear effect from Cox Proportional Hazards models*

Description

`coxsimPoly` simulates quantities of interest for polynomial covariate effects estimated from Cox Proportional Hazards models. These can be plotted with `simGG`.

Usage

```
coxsimPoly(obj, b = NULL, qi = "Relative Hazard", pow = 2, Xj = NULL,
  Xl = NULL, nsim = 1000, ci = 0.95, spin = FALSE,
  extremesDrop = TRUE)
```

Arguments

`obj`	a `coxph` class fitted model object with a polynomial coefficient. These can be plotted with `simGG`.
`b`	character string name of the coefficient you would like to simulate. To find the quantity of interest using only the polynomial and not the polynomial + the linear terms enter the polynomial created using `I`, e.g. `I(natreg^2)` as a string.
`qi`	quantity of interest to simulate. Values can be `"Relative Hazard"`, `"First Difference"`, `"Hazard Ratio"`, and `"Hazard Rate"`. The default is `qi = "Relative Hazard"`. If `qi = "Hazard Rate"` and the `coxph` model has strata, then hazard rates for each strata will also be calculated.
`pow`	numeric polynomial used in `coxph`.
`Xj`	numeric vector of fitted values for `b` to simulate for.

Xl	numeric vector of values to compare Xj to. If NULL, then it is authomatically set to 0.
nsim	the number of simulations to run per value of Xj. Default is nsim = 1000.
ci	the proportion of simulations to keep. The default is ci = 0.95, i.e. keep the middle 95 percent. If spin = TRUE then ci is the confidence level of the shortest probability interval. Any value from 0 through 1 may be used.
spin	logical, whether or not to keep only the shortest probability interval rather than the middle simulations. Currently not supported for hazard rates.
extremesDrop	logical whether or not to drop simulated quantity of interest values that are Inf, NA, NaN and > 1000000 for spin = FALSE or > 800 for spin = TRUE. These values are difficult to plot simGG and may prevent spin from finding the central interval.

Details

Simulates quantities of interest for polynomial covariate effects. For example if a nonlinear effect is modeled with a second order polynomial–i.e. $\beta_1 x_i + \beta_2 x_i^2$–we can once again draw n simulations from the multivariate normal distribution for both β_1 and β_2. Then we simply calculate quantities of interest for a range of values and plot the results as before. For example, we find the first difference for a second order polynomial with:

$$\%\triangle h_i(t) = (\mathrm{e}^{\beta_1 x_{j-1} + \beta_2 x_{j-l}^2} - 1) * 100$$

where $x_{j-l} = x_j - x_l$.

Note, you must use I to create the polynomials.

Value

a simpoly class object.

References

Keele, Luke. 2010. "Proportionally Difficult: Testing for Nonproportional Hazards in Cox Models." Political Analysis 18(2): 189-205.

Carpenter, Daniel P. 2002. "Groups, the Media, Agency Waiting Costs, and FDA Drug Approval." American Journal of Political Science 46(3): 490-505.

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." American Journal of Political Science 44(2): 347-61.

Liu, Ying, Andrew Gelman, and Tian Zheng. 2013. "Simulation-Efficient Shortest Probability Intervals." Arvix. http://arxiv.org/pdf/1302.2142v1.pdf.

See Also

simGG, survival, strata, and coxph

Examples

```
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
M1 <- coxph(Surv(acttime, censor) ~ prevgenx + lethal + deathrt1 +
            acutediz + hosp01  + hhosleng + mandiz01 + femdiz01 +
            peddiz01 + orphdum + natreg + I(natreg^2) +
            I(natreg^3) + vandavg3 + wpnoavg3 +
            condavg3 + orderent + stafcder, data = CarpenterFdaData)

# Simulate simpoly First Difference
Sim1 <- coxsimPoly(M1, b = "natreg", qi = "First Difference",
                pow = 3, Xj = seq(1, 150, by = 5), nsim = 100)

## Not run:
# Simulate simpoly Hazard Ratio with spin probibility interval
Sim2 <- coxsimPoly(M1, b = "natreg", qi = "Hazard Ratio",
              pow = 3, Xj = seq(1, 150, by = 5), spin = TRUE)

## End(Not run)
```

`coxsimSpline` *Simulate quantities of interest for penalized splines from Cox Proportional Hazards models*

Description

`coxsimSpline` simulates quantities of interest from penalized splines using multivariate normal distributions.

Usage

```
coxsimSpline(obj, bspline, bdata, qi = "Relative Hazard", Xj = 1, Xl = 0,
  nsim = 1000, ci = 0.95, spin = FALSE, extremesDrop = TRUE)
```

Arguments

`obj`	a `coxph` class fitted model object with a penalized spline. These can be plotted with `simGG`.
`bspline`	a character string of the full `pspline` call used in `obj`. It should be exactly the same as how you entered it in `coxph`. You also need to enter a white spece before and after all equal (=) signs.
`bdata`	a numeric vector of splined variable's values.

qi	quantity of interest to simulate. Values can be "Relative Hazard", "First Difference", "Hazard Ratio", and "Hazard Rate". The default is qi = "Relative Hazard". Think carefully before using qi = "Hazard Rate". You may be creating very many simulated values which can be very computationally intensive to do. Adjust the number of simulations per fitted value with nsim.
Xj	numeric vector of fitted values for b to simulate for.
Xl	numeric vector of values to compare Xj to. Note if qi = "Relative Hazard" or "Hazard Rate" only Xj is relevant.
nsim	the number of simulations to run per value of Xj. Default is nsim = 1000.
ci	the proportion of simulations to keep. The default is ci = 0.95, i.e. keep the middle 95 percent. If spin = TRUE then ci is the confidence level of the shortest probability interval. Any value from 0 through 1 may be used.
spin	logical, whether or not to keep only the shortest probability interval rather than the middle simulations. Currently not supported for hazard rates.
extremesDrop	logical whether or not to drop simulated quantity of interest values that are Inf, NA, NaN and $>$ 1000000 for spin = FALSE or $>$ 800 for spin = TRUE. These values are difficult to plot simGG and may prevent spin from finding the central interval.

Details

Simulates relative hazards, first differences, hazard ratios, and hazard rates for penalized splines from Cox Proportional Hazards models. These can be plotted with simGG. A Cox PH model with one penalized spline is given by:

$$h(t|\mathbf{X}_i) = h_0(t)\mathrm{e}^{g(x)}$$

where $g(x)$ is the penalized spline function. For our post-estimation purposes $g(x)$ is basically a series of linearly combined coefficients such that:

$$g(x) = \beta_{k_1}(x)_{1+} + \beta_{k_2}(x)_{2+} + \beta_{k_3}(x)_{3+} + \ldots + \beta_{k_n}(x)_{n+}$$

where k are the equally spaced spline knots with values inside of the range of observed x and n is the number of knots.

We can again draw values of each $\beta_{k_1}, \ldots \beta_{k_n}$ from the multivariate normal distribution described above. We then use these simulated coefficients to estimates quantities of interest for a range covariate values. For example, the first difference between two values x_j and x_l is:

$$\%\triangle h_i(t) = (\mathrm{e}^{g(x_j)-g(x_l)} - 1) * 100$$

FD(h[i](t)) = (exp(g(x[j]) - g(x[l])) - 1) * 100

Relative hazards and hazard ratios can be calculated by extension.

Currently coxsimSpline does not support simulating hazard rates form multiple stratified models.

Value

a simspline object

References

Luke Keele, "Replication data for: Proportionally Difficult: Testing for Nonproportional Hazards In Cox Models", 2010, http://hdl.handle.net/1902.1/17068 V1 [Version].

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." American Journal of Political Science 44(2): 347-61.

Liu, Ying, Andrew Gelman, and Tian Zheng. 2013. "Simulation-Efficient Shortest Probability Intervals." Arvix. http://arxiv.org/pdf/1302.2142v1.pdf.

See Also

simGG, survival, strata, and coxph

Examples

```
## Not run:
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
# From Keele (2010) replication data
M1 <- coxph(Surv(acttime, censor) ~  prevgenx + lethal + deathrt1 +
           acutediz + hosp01  + pspline(hospdisc, df = 4) +
           pspline(hhosleng, df = 4) + mandiz01 + femdiz01 + peddiz01 +
           orphdum + natreg + vandavg3 + wpnoavg3 +
           pspline(condavg3, df = 4) + pspline(orderent, df = 4) +
           pspline(stafcder, df = 4), data = CarpenterFdaData)

# Simulate Relative Hazards for orderent
Sim1 <- coxsimSpline(M1, bspline = "pspline(stafcder, df = 4)",
                    bdata = CarpenterFdaData$stafcder,
                    qi = "Hazard Ratio",
                    Xj = seq(1100, 1700, by = 10),
                    Xl = seq(1099, 1699, by = 10), spin = TRUE)

# Simulate Hazard Rates for orderent
Sim2 <- coxsimSpline(M1, bspline = "pspline(orderent, df = 4)",
                       bdata = CarpenterFdaData$orderent,
                       qi = "Hazard Rate",
                       Xj = seq(2, 53, by = 3), nsim = 100)

## End(Not run)
```

coxsimtvc *Simulate time-interactive quantities of interest from Cox Proportional Hazards models*

Description

coxsimtvc simulates time-interactive relative hazards, first differences, and hazard ratios from models estimated with coxph using the multivariate normal distribution. These can be plotted with simGG.

Usage

```
coxsimtvc(obj, b, btvc, qi = "Relative Hazard", Xj = NULL, Xl = NULL,
  tfun = "linear", pow = NULL, nsim = 1000, from, to, by = 1,
  ci = 0.95, spin = FALSE, extremesDrop = TRUE)
```

Arguments

obj	a coxph fitted model object with a time interaction.
b	the non-time interacted variable's name.
btvc	the time interacted variable's name.
qi	character string indicating what quantity of interest you would like to calculate. Can be 'Relative Hazard', 'First Difference', 'Hazard Ratio', 'Hazard Rate'. Default is qi = 'Relative Hazard'. If qi = 'First Difference' or qi = 'Hazard Ratio' then you can set Xj and Xl.
Xj	numeric vector of fitted values for b. Must be the same length as Xl or Xl must be NULL.
Xl	numeric vector of fitted values for Xl. Must be the same length as Xj. Only applies if qi = 'First Difference' or qi = 'Hazard Ratio'.
tfun	function of time that btvc was multiplied by. Default is "linear". It can also be "log" (natural log) and "power". If tfun = "power" then the pow argument needs to be specified also.
pow	if tfun = "power", then use pow to specify what power the time interaction was raised to.
nsim	the number of simulations to run per point in time. Default is nsim = 1000.
from	point in time from when to begin simulating coefficient values
to	point in time to stop simulating coefficient values.
by	time intervals by which to simulate coefficient values.
ci	the proportion of simulations to keep. The default is ci = 0.95, i.e. keep the middle 95 percent. If spin = TRUE then ci is the confidence level of the shortest probability interval. Any value from 0 through 1 may be used.
spin	logical, whether or not to keep only the shortest probability interval rather than the middle simulations. Currently not supported for hazard rates.

`extremesDrop` logical whether or not to drop simulated quantity of interest values that are `Inf`, `NA`, `NaN` and > 1000000 for `spin = FALSE` or > 800 for `spin = TRUE`. These values are difficult to plot `simGG` and may prevent `spin` from finding the central interval.

Details

Simulates time-varying relative hazards, first differences, and hazard ratios using parameter estimates from `coxph` models. Can also simulate hazard rates for multiple strata.

Relative hazards are found using:

$$RH = e^{\beta_1 + \beta_2 f(t)}$$

where $f(t)$ is the function of time.

First differences are found using:

$$FD = (e^{(X_j - X_l)(\beta_1 + \beta_2 f(t))} - 1) * 100$$

where X_j and X_l are some values of X to contrast.

Hazard ratios are calculated using:

$$FD = e^{(X_j - X_l)(\beta_1 + \beta_2 f(t))}$$

When simulating non-stratifed time-varying harzards `coxsimtvc` uses the point estimates for a given coefficient $\hat{\beta}_x$ and its time interaction $\hat{\beta}_{xt}$ along with the variance matrix ($\hat{V}(\hat{\beta})$) estimated from a `coxph` model. These are used to draw values of β_1 and β_2 from the multivariate normal distribution $N(\hat{\beta},\ \hat{V}(\hat{\beta}))$.

When simulating stratified time-varying hazard rates H for a given strata k, `coxsimtvc` uses:

$$H_{kxt} = \hat{\beta}_{k0t} e^{\hat{\beta}_1 + \beta_2 f(t)}$$

The resulting simulation values can be plotted using `simGG`.

Value

a `simtvc` object

References

Golub, Jonathan, and Bernard Steunenberg. 2007. "How Time Affects EU Decision-Making." European Union Politics 8(4): 555-66.

Licht, Amanda A. 2011. "Change Comes with Time: Substantive Interpretation of Nonproportional Hazards in Event History Analysis." Political Analysis 19: 227-43.

King, Gary, Michael Tomz, and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." American Journal of Political Science 44(2): 347-61.

Liu, Ying, Andrew Gelman, and Tian Zheng. 2013. "Simulation-Efficient Shortest Probability Intervals." Arvix. http://arxiv.org/pdf/1302.2142v1.pdf.

See Also

`simGG`, `survival`, `strata`, and `coxph`

Examples

```
## Not run:
# Load Golub & Steunenberg (2007) Data
data("GolubEUPData")

# Load survival package
library(survival)

# Expand data (not run to speed processing time, but should be run)
GolubEUPData <- SurvExpand(GolubEUPData, GroupVar = 'caseno',
                     Time = 'begin', Time2 = 'end', event = 'event')

# Create time interactions
BaseVars <- c('qmv', 'backlog', 'coop', 'codec', 'qmvpostsea', 'thatcher')
GolubEUPData <- tvc(GolubEUPData, b = BaseVars, tvar = 'end', tfun = 'log')

# Run Cox PH Model
M2 <- coxph(Surv(begin, end, event) ~ qmv + qmvpostsea + qmvpostteu +
                coop + codec + eu9 + eu10 + eu12 + eu15 + thatcher +
                agenda + backlog + qmv_log + qmvpostsea_log + coop_log +
                codec_log + thatcher_log + backlog_log,
            data = GolubEUPData, ties = "efron")

# Create simtvc object for Relative Hazard
Sim1 <- coxsimtvc(obj = M1, b = "qmv", btvc = "qmv_log",
                  tfun = "log", from = 80, to = 2000,
                  Xj = 1, by = 15, ci = 0.99, nsim = 100)

# Create simtvc object for First Difference
Sim2 <- coxsimtvc(obj = M1, b = "qmv", btvc = "qmv_log",
                 qi = "First Difference", Xj = 1,
                 tfun = "log", from = 80, to = 2000,
                 by = 15, ci = 0.95)

# Create simtvc object for Hazard Ratio
Sim3 <- coxsimtvc(obj = M1, b = "backlog", btvc = "backlog_log",
                  qi = "Hazard Ratio", Xj = c(191, 229),
                  Xl = c(0, 0),
                  tfun = "log", from = 80, to = 2000,
                  by = 15, ci = 0.5)

## End(Not run)
```

`ggfitStrata` *Graph fitted stratified survival curves from Cox Proportional Hazards models*

Description

This function largely improves `plot.survfit`. It plots the curves using ggplot2 rather than base R graphics. One major advantage is the ability to split the survival curves into multiple plots and

arrange them in a grid. This makes it easier to examine many strata at once. Otherwise they can be very bunched up.

Note: the strata legend labels need to be changed manually (see `revalue`) in the `survfit` object with the `strata` component.

Usage

```
ggfitStrata(obj, byStrata = FALSE, xlab = "", ylab = "", title = "",
  lcolour = "#2C7FB8", rcolour = "#2C7FB8")
```

Arguments

`obj`	a `survfit` object.
`byStrata`	logical, whether or not you want to include all of the stratified survival curves on one plot or separate them into a grid arranged plot.
`xlab`	a label for the plot's x-axis
`ylab`	a label of the plot's y-axis
`title`	plot title.
`lcolour`	line color. Currently only works if `byStrata = TRUE`. The default it `lcolour = "#2C7FB8"` (a bluish hexadecimal colour)
`rcolour`	confidence bounds ribbon color. Either a single color or a vector of colours. The default it `lcolour = "#2C7FB8"` (a bluish hexadecimal colour)

Details

`ggfitStrata` graphs fitted survival curves created with `survfit` using ggplot2.

See Also

`survfit`, `ggplot2` and `strata`

Examples

```
## Not run:
# Load survival
library(survival)
# Subset data
bladder1 <- bladder[bladder$enum < 5, ]
# Estimate coxph model
M1 <- coxph(Surv(stop, event) ~ (rx + size + number) * strata(enum) +
cluster(id), bladder1)
# Survfit
M1Fit <- survfit(M1)
# Plot strata in a grid
ggfitStrata(M1Fit, byStrata = TRUE)

## End(Not run)
```

GolubEUPData *A data set from Golub & Steunenberg (2007)*

Description

A data set from Golub & Steunenberg (2007)

Format

A data set with 3001 observations and 17 variables.

Source

Golub, Jonathan, and Bernard Steunenberg. 2007. "How Time Affects EU Decision-Making." European Union Politics 8(4): 555-66.

Amanda A. Licht, 2011, "Replication data for: Change Comes with Time". http://hdl.handle.net/1902.1/15633. IQSS Dataverse Network [Distributor] V3 [Version].

MinMaxLines *Transform the simulation object to include only the min and max of the constricted intervals, as well as the lower and upper bounds of the middle 50 percent of the constricted intervals*

Description

`MinMaxLines` is an internal function to transform the simulation object to include only the min and max of the intervals set by `ci` in the `coxsim` command, as well as the lower and upper bounds of the middle 50 percent of these intervals. It also returns the medians.

Usage

```
MinMaxLines(df, byVars = "Xj", hr = FALSE, strata = FALSE,
  clean = FALSE)
```

Arguments

`df`	a data frame or a simulation class object.
`byVars`	character vector of the variables to subset the data frame by. The default is `'Xj'`.
`hr`	logical indicating whether or not `df` contains a hazard rate.
`strata`	logical indicating whether or not `df` contains a stratified hazard rate.
`clean`	logical, whether or not to clean up the output data frame to only include `byVars`, `Min_CI`, `Lower50_CI`, `median`, `Upper50_CI`, `Max_CI`.

Examples

```
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
M1 <- coxph(Surv(acttime, censor) ~ prevgenx + lethal +
           deathrt1 + acutediz + hosp01  + hhosleng +
           mandiz01 + femdiz01 + peddiz01 + orphdum +
           vandavg3 + wpnoavg3 + condavg3 + orderent +
           stafcder, data = CarpenterFdaData)

 # Simulate Hazard Ratios
 Sim1 <- coxsimLinear(M1, b = "stafcder",
                      Xj = c(1237, 1600),
                      Xl = c(1000, 1000),
                      qi = "Hazard Ratio",
                      spin = TRUE, ci = 0.99)

# Find summary statistics of the constricted interval
Sum <- MinMaxLines(Sim1, clean = TRUE)
```

---

setXl *Create a sequence of Xl values*

Description

`setXl` creates a sequence of `Xl` values given a sequence of `Xj` values and a fixed difference.

Usage

```
setXl(Xj, diff = 1)
```

Arguments

`Xj`	numeric vector of fitted values for the covariate of interest to simulate for.
`diff`	numeric vector of length 1. It specifies the difference between `Xj` and `Xl`. `Xl` is always smaller than `Xj`.

Value

a vector

Examples

```
# Set Xj
setXj = seq(1100, 1700, by = 10)

# Find Xl that are 1 less than Xj
setXl(Xj = setXj, diff = 1)
```

---

simGG *A method for plotting simulation objects created by simPH*

Description

simGG a method for ploting simulation objects created by simPH.

Usage

```
simGG(obj, ...)
```

Arguments

obj	an object created by one of simPH's simulation commands.
...	arguments to be passed to methods.

See Also

simGG.siminteract, simGG.simtvc, simGG.simlinear, simGG.simpoly, simGG.simspline

Examples

```
## Not run:
 # Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
M1 <- coxph(Surv(acttime, censor) ~ lethal*prevgenx,
            data = CarpenterFdaData)

# Simulate Marginal Effect of lethal for multiple
# values of prevgenx
Sim1 <- coxsimInteract(M1, b1 = "lethal", b2 = "prevgenx",
                       X2 = seq(2, 115, by = 5), spin = TRUE)

# Plot simulations
simGG(Sim1)

## End(Not run)
```

simGG.siminteract *Plot simulated linear multiplicative interactions from Cox Proportional Hazards Models*

Description

simGG.siminteract uses ggplot2 to plot the quantities of interest from siminteract objects, including marginal effects, first differences, hazard ratios, and hazard rates.

Usage

```
## S3 method for class 'siminteract'
simGG(obj, from = NULL, to = NULL, xlab = NULL,
  ylab = NULL, title = NULL, method = "auto", spalette = "Set1",
  legend = "legend", leg.name = "", lcolour = "#2B8CBE", lsize = 1,
  pcolour = "#A6CEE3", psize = 1, alpha = 0.2, type = "points", ...)
```

Arguments

obj	a siminteract class object
from	numeric time to start the plot from. Only relevant if qi = "Hazard Rate".
to	numeric time to plot to. Only relevant if qi = "Hazard Rate".
xlab	a label for the plot's x-axis.
ylab	a label of the plot's y-axis. The default uses the value of qi.
title	the plot's main title.
method	what type of smoothing method to use to summarize the center of the simulation distribution.
spalette	colour palette for when there are multiple sets of comparisons to plot. Not relevant for qi = "Marginal Effect". Default palette is "Set1". See scale_colour_brewer.
legend	specifies what type of legend to include (if applicable). The default is legend = "legend". To hide the legend use legend = FALSE. See the discrete_scale for more details.
leg.name	name of the legend (if applicable).
lcolour	character string colour of the smoothing line. The default is hexadecimal colour lcolour = '#2B8CBE'. Only relevant if qi = "Marginal Effect".
lsize	size of the smoothing line. Default is 1. See ggplot2.
pcolour	character string colour of the simulated points or ribbons (when there are not multiple sets of simulations). Default is hexadecimal colour pcolour = '#A6CEE3'.
psize	size of the plotted simulation points. Default is psize = 1. See ggplot2.
alpha	numeric. Alpha (e.g. transparency) for the points, lines, or ribbons. Default is alpha = 0.2. See ggplot2. Note, if type = "lines" or type = "points" then alpah sets the maximum value per line or point at the center of the distribution. Lines or points further from the center are more transparent the further they get from the middle.

type	character string. Specifies how to plot the simulations. Can be points, lines, or ribbons. If points then each simulation value will be plotted. If lines is chosen then each simulation is plotted using a different line. Note: any simulation with a value along its length that is outside of the specified central interval will be dropped. This is to create a smooth plot. If type = "ribbons" a plot will be created with shaded areas ('ribbons') for the minimum and maximum simulation values (i.e. the middle interval set with qi in coxsimSpline) as well as the central 50 percent of this area. It also plots a line for the median value of the full area, so values in method are ignored. One of the key advantages of using ribbons rather than points is that it creates plots with smaller file sizes.
...	Additional arguments. (Currently ignored.)

Details

Uses ggplot2 to plot the quantities of interest from siminteract objects, including marginal effects, first differences, hazard ratios, and hazard rates. If there are multiple strata, the quantities of interest will be plotted in a grid by strata.

Note: A dotted line is created at y = 1 (0 for first difference), i.e. no effect, for time-varying hazard ratio graphs. No line is created for hazard rates.

Note: if qi = "Hazard Ratio" or qi = "First Difference" then you need to have choosen more than one fitted value for X1 in coxsimInteract.

Value

a gg ggplot class object

References

Brambor, Thomas, William Roberts Clark, and Matt Golder. 2006. "Understanding Interaction Models: Improving Empirical Analyses." Political Analysis 14(1): 63-82.

Keele, Luke. 2010. "Proportionally Difficult: Testing for Nonproportional Hazards in Cox Models." Political Analysis 18(2): 189-205.

Carpenter, Daniel P. 2002. "Groups, the Media, Agency Waiting Costs, and FDA Drug Approval." American Journal of Political Science 46(3): 490-505.

See Also

coxsimInteract, simGG.simlinear, and ggplot2

Examples

```
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
M1 <- coxph(Surv(acttime, censor) ~ lethal*prevgenx,
```

```
            data = CarpenterFdaData)

# Simulate Marginal Effect of lethal for multiple values of prevgenx
Sim1 <- coxsimInteract(M1, b1 = "lethal", b2 = "prevgenx",
                    X2 = seq(2, 115, by = 2), nsim = 100)

# Plot quantities of interest
simGG(Sim1)

## Not run:
# Change the order of the covariates to make a more easily
# interpretable hazard ratio graph.
M2 <- coxph(Surv(acttime, censor) ~ prevgenx*lethal,
             data = CarpenterFdaData)

# Simulate Hazard Ratio of lethal for multiple values of prevgenx
Sim2 <- coxsimInteract(M2, b1 = "prevgenx", b2 = "lethal",
                   X1 = seq(2, 115, by = 2),
                   X2 = c(0, 1),
                   qi = "Hazard Ratio", ci = 0.9)

# Simulate First Difference
Sim3 <- coxsimInteract(M2, b1 = "prevgenx", b2 = "lethal",
                       X1 = seq(2, 115, by = 2),
                       X2 = c(0, 1),
                       qi = "First Difference", spin = TRUE)

# Simulate Hazard Rate
Sim4 <- coxsimInteract(M2, b1 = "prevgenx", b2 = "lethal",
                       X1 = 100, X2 = 1, qi = "Hazard Rate")

# Plot quantities of interest
simGG(Sim1, xlab = "\nprevgenx",
       ylab = "Marginal Effect of lethal\n")
simGG(Sim2, ribbons = TRUE)
simGG(Sim3)
simGG(Sim4, to = 150, ribbons = TRUE, legend = FALSE)

## End(Not run)
```

`simGG.simlinear` *Plot simulated linear, non-time interacted quantities of interest from Cox Proportional Hazards Models*

Description

`simGG.simlinear` uses ggplot2 to plot the quantities of interest from `simlinear` objects, including relative hazards, first differences, hazard ratios, and hazard rates.

Usage

```
## S3 method for class 'simlinear'
simGG(obj, from = NULL, to = NULL, xlab = NULL,
  ylab = NULL, title = NULL, method = "auto", spalette = "Set1",
  legend = "legend", leg.name = "", lcolour = "#2B8CBE", lsize = 1,
  pcolour = "#A6CEE3", psize = 1, alpha = 0.2, type = "points", ...)
```

Arguments

obj	a simlinear class object.
from	numeric time to start the plot from. Only relevant if qi = "Hazard Rate".
to	numeric time to plot to. Only relevant if qi = "Hazard Rate".
xlab	a label for the plot's x-axis.
ylab	a label of the plot's y-axis. The default uses the value of qi.
title	the plot's main title.
method	what type of smoothing method to use to summarize the center of the simulation distribution.
spalette	colour palette for when there are multiple sets of comparisons to plot. Default palette is "Set1". See scale_colour_brewer.
legend	specifies what type of legend to include (if applicable). The default is legend = "legend". To hide the legend use legend = FALSE. See the discrete_scale for more details.
leg.name	name of the legend (if applicable).
lcolour	character string colour of the smoothing line. The default is hexadecimal colour lcolour = '#2B8CBE'. Only relevant if qi = "First Difference".
lsize	size of the smoothing line. Default is 1. See ggplot2.
pcolour	character string colour of the simulated points or ribbons (when there are not multiple sets of simulations). Default is hexadecimal colour pcolour = '#A6CEE3'.
psize	size of the plotted simulation points. Default is psize = 1. See ggplot2.
alpha	numeric. Alpha (e.g. transparency) for the points, lines, or ribbons. Default is alpha = 0.2. See ggplot2. Note, if type = "lines" or type = "points" then alpah sets the maximum value per line or point at the center of the distribution. Lines or points further from the center are more transparent the further they get from the middle.
type	character string. Specifies how to plot the simulations. Can be points, lines, or ribbons. If points then each simulation value will be plotted. If lines is chosen then each simulation is plotted using a different line. Note: any simulation with a value along its length that is outside of the specified central interval will be dropped. This is to create a smooth plot. If type = "ribbons" a plot will be created with shaded areas ('ribbons') for the minimum and maximum simulation values (i.e. the middle interval set with qi in coxsimSpline) as well as the central 50 percent of this area. It also plots a line for the median value of the full area, so values in method are ignored. One of the key advantages of using ribbons rather than points is that it creates plots with smaller file sizes.
...	Additional arguments. (Currently ignored.)

Details

Uses ggplot2 to plot the quantities of interest from `simlinear` objects, including relative hazards, first differences, hazard ratios, and hazard rates. If there are multiple strata, the quantities of interest will be plotted in a grid by strata. Note: A dotted line is created at y = 1 (0 for first difference), i.e. no effect, for time-varying hazard ratio graphs. No line is created for hazard rates.

Value

a gg `ggplot` class object

References

Licht, Amanda A. 2011. "Change Comes with Time: Substantive Interpretation of Nonproportional Hazards in Event History Analysis." Political Analysis 19: 227-43.

Keele, Luke. 2010. "Proportionally Difficult: Testing for Nonproportional Hazards in Cox Models." Political Analysis 18(2): 189-205.

Carpenter, Daniel P. 2002. "Groups, the Media, Agency Waiting Costs, and FDA Drug Approval." American Journal of Political Science 46(3): 490-505.

See Also

`coxsimLinear`, `simGG.simtvc`, and `ggplot2`

Examples

```
# Load survival package
library(survival)
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Estimate basic model
M1 <- coxph(Surv(acttime, censor) ~ prevgenx + lethal +
            deathrt1 + acutediz + hosp01  + hhosleng +
            mandiz01 + femdiz01 + peddiz01 + orphdum +
            vandavg3 + wpnoavg3 + condavg3 + orderent +
            stafcder, data = CarpenterFdaData)

# Simulate and plot Hazard Ratios for stafcder variable
Sim1 <- coxsimLinear(M1, b = "stafcder",
                     Xj = c(1237, 1600),
                     Xl = c(1000, 1000),
                     qi = "Hazard Ratio",
                     spin = TRUE, ci = 0.99)
simGG(Sim1, method = 'lm')

## Not run:
# Simulate and plot Hazard Rate for stafcder variable
Sim2 <- coxsimLinear(M1, b = "stafcder", nsim = 100,
                      qi = "Hazard Rate",
                      Xj = c(1237, 1600))
```

```
simGG(Sim2, type = 'lines')

## End(Not run)
```

`simGG.simpoly` *Plot simulated polynomial quantities of interest from Cox Proportional Hazards Models*

Description

`simGG.simpoly` uses ggplot2 to plot simulated relative quantities of interest from a `simpoly` class object.

Usage

```
## S3 method for class 'simpoly'
simGG(obj, from = NULL, to = NULL, xlab = NULL,
  ylab = NULL, title = NULL, method = "auto", spalette = "Set1",
  legend = "legend", leg.name = "", lcolour = "#2B8CBE", lsize = 1,
  pcolour = "#A6CEE3", psize = 1, alpha = 0.2, type = "points", ...)
```

Arguments

`obj`	a `simpoly` class object.
`from`	numeric time to start the plot from. Only relevant if `qi = "Hazard Rate"`.
`to`	numeric time to plot to. Only relevant if `qi = "Hazard Rate"`.
`xlab`	a label for the plot's x-axis.
`ylab`	a label of the plot's y-axis. The default uses the value of `qi`.
`title`	the plot's main title.
`method`	what type of smoothing method to use to summarize the center of the simulation distribution.
`spalette`	colour palette for when there are multiple sets of comparisons to plot. Default palette is `"Set1"`. See `scale_colour_brewer`.
`legend`	specifies what type of legend to include (if applicable). The default is `legend = "legend"`. To hide the legend use `legend = FALSE`. See the `discrete_scale` for more details.
`leg.name`	name of the legend (if applicable).
`lcolour`	character string colour of the smoothing line. The default is hexadecimal colour `lcolour = '#2B8CBE'`. Only relevant if `qi = "First Difference"`.
`lsize`	size of the smoothing line. Default is 1. See `ggplot2`.
`pcolour`	character string colour of the simulated points or ribbons (when there are not multiple sets of simulations). Default is hexadecimal colour `pcolour = '#A6CEE3'`.
`psize`	size of the plotted simulation points. Default is `psize = 1`. See `ggplot2`.

alpha	numeric. Alpha (e.g. transparency) for the points, lines, or ribbons. Default is alpha = 0.2. See ggplot2. Note, if type = "lines" or type = "points" then alpah sets the maximum value per line or point at the center of the distribution. Lines or points further from the center are more transparent the further they get from the middle.
type	character string. Specifies how to plot the simulations. Can be points, lines, or ribbons. If points then each simulation value will be plotted. If lines is chosen then each simulation is plotted using a different line. Note: any simulation with a value along its length that is outside of the specified central interval will be dropped. This is to create a smooth plot. If type = "ribbons" a plot will be created with shaded areas ('ribbons') for the minimum and maximum simulation values (i.e. the middle interval set with qi in coxsimSpline) as well as the central 50 percent of this area. It also plots a line for the median value of the full area, so values in method are ignored. One of the key advantages of using ribbons rather than points is that it creates plots with smaller file sizes.
...	Additional arguments. (Currently ignored.)

Details

Uses ggplot2 to plot the quantities of interest from simpoly objects.

Value

a gg ggplot class object

See Also

coxsimPoly and ggplot2

Examples

```
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
M1 <- coxph(Surv(acttime, censor) ~ prevgenx + lethal +
       deathrt1 + acutediz + hosp01  + hhosleng + mandiz01 +
       femdiz01 + peddiz01 + orphdum + natreg +
       I(natreg^2) + I(natreg^3) + vandavg3 + wpnoavg3 +
       condavg3 + orderent + stafcder, data = CarpenterFdaData)

# Simulate simpoly First Difference
Sim1 <- coxsimPoly(M1, b = "natreg", qi = "First Difference",
           pow = 3, Xj = seq(1, 150, by = 5), nsim = 100)

# Plot simulations
simGG(Sim1, type = 'lines')
```

```
## Not run:
# Simulate simpoly Hazard Ratio with spin probibility interval
Sim2 <- coxsimPoly(M1, b = "natreg", qi = "Hazard Ratio",
          pow = 3, Xj = seq(1, 150, by = 5), spin = TRUE,
          nsim = 100)

# Plot simulations
simGG(Sim2, type = 'ribbons')

Sim3 <- coxsimPoly(M1, b = "natreg", qi = "Hazard Rate",
           pow = 3, Xj = c(1, 150), nsim = 100)

# Plot simulations
simGG(Sim3, type = 'lines')

## End(Not run)
```

`simGG.simspline` *Plot simulated penalised spline hazards from Cox Proportional Hazards Models*

Description

`simGG.simspline` uses ggplot2 and scatter3d to plot quantities of interest from `simspline` objects, including relative hazards, first differences, hazard ratios, and hazard rates.

Usage

```
## S3 method for class 'simspline'
simGG(obj, SmoothSpline = TRUE, FacetTime = NULL,
  from = NULL, to = NULL, xlab = NULL, ylab = NULL, zlab = NULL,
  title = NULL, method = "auto", lcolour = "#2B8CBE", lsize = 1,
  pcolour = "#A6CEE3", psize = 1, alpha = 0.2, surface = TRUE,
  fit = "linear", type = "points", ...)
```

Arguments

`obj`	a `simspline` class object
`SmoothSpline`	logical whether or not to fit the simulations with smoothing splines. Creates a smoother plot. See `smooth.spline` for more information. Note: currently the degrees of freedom are set at 10.
`FacetTime`	a numeric vector of points in time where you would like to plot Hazard Rates in a facet grid. Only relevant if `qi == 'Hazard Rate'`. Note: the values of Facet Time must exactly match values of the `time` element of `obj`.
`from`	numeric time to start the plot from. Only relevant if `qi = "Hazard Rate"`.
`to`	numeric time to plot to. Only relevant if `qi = "Hazard Rate"`.

xlab	a label for the plot's x-axis.
ylab	a label of the plot's y-axis. The default uses the value of qi.
zlab	a label for the plot's z-axis. Only relevant if qi = "Hazard Rate" and FacetTime == NULL.
title	the plot's main title.
method	what type of smoothing method to use to summarize the center of the simulation distribution.
lcolour	character string colour of the smoothing line. The default is hexadecimal colour lcolour = '#2B8CBE'. Only relevant if qi = "Relative Hazard" or qi = "First Difference".
lsize	size of the smoothing line. Default is 1. See ggplot2.
pcolour	character string colour of the simulated points or ribbons (when there are not multiple sets of simulations). Default is hexadecimal colour pcolour = '#A6CEE3'. Only relevant if qi = "Relative Hazard" or qi = "First Difference" or qi = "Hazard Rate" with facets.
psize	size of the plotted simulation points. Default is psize = 1. See ggplot2.
alpha	numeric. Alpha (e.g. transparency) for the points, lines, or ribbons. Default is alpha = 0.2. See ggplot2. Note, if type = "lines" or type = "points" then alpah sets the maximum value per line or point at the center of the distribution. Lines or points further from the center are more transparent the further they get from the middle.
surface	plot surface. Default is surface = TRUE. Only relevant if qi == 'Hazard Rate' and FacetTime = NULL.
fit	one or more of "linear", "quadratic", "smooth", "additive"; to display fitted surface(s); partial matching is supported e.g., c("lin", "quad"). Only relevant if qi == 'Relative Hazard' and FacetTime = NULL.
type	character string. Specifies how to plot the simulations. Can be points, lines, or ribbons. If points then each simulation value will be plotted. If lines is chosen then each simulation is plotted using a different line. Note: any simulation with a value along its length that is outside of the specified central interval will be dropped. This is to create a smooth plot. If type = "ribbons" a plot will be created with shaded areas ('ribbons') for the minimum and maximum simulation values (i.e. the middle interval set with qi in coxsimSpline) as well as the central 50 percent of this area. It also plots a line for the median value of the full area, so values in method are ignored. One of the key advantages of using ribbons rather than points is that it creates plots with smaller file sizes.
...	Additional arguments. (Currently ignored.)

Details

Uses ggplot2 and scatter3d to plot the quantities of interest from simspline objects, including relative hazards, first differences, hazard ratios, and hazard rates. If currently does not support hazard rates for multiple strata.

It can graph hazard rates as a 3D plot using scatter3d with the dimensions: Time, Hazard Rate, and the value of Xj. Ribbon plots are not available with 3D plots. You can also choose to plot

hazard rates for a range of values of Xj in two dimensional plots at specific points in time. Each plot is arranged in a facet grid.

Note: A dotted line is created at y = 1 (0 for first difference), i.e. no effect, for time-varying hazard ratio graphs. No line is created for hazard rates.

Value

a gg ggplot class object. See scatter3d for values from scatter3d calls.

See Also

coxsimLinear, simGG.simtvc, ggplot2, and scatter3d

Examples

```
## Not run:
# Load Carpenter (2002) data
data("CarpenterFdaData")

# Load survival package
library(survival)

# Run basic model
# From Keele (2010) replication data
M1 <- coxph(Surv(acttime, censor) ~  prevgenx + lethal + deathrt1 +
                acutediz + hosp01  + pspline(hospdisc, df = 4) +
                pspline(hhosleng, df = 4) + mandiz01 + femdiz01 +
                peddiz01 + orphdum + natreg + vandavg3 + wpnoavg3 +
                pspline(condavg3, df = 4) + pspline(orderent, df = 4) +
                pspline(stafcder, df = 4), data = CarpenterFdaData)

# Simulate Relative Hazards for orderent
Sim1 <- coxsimSpline(M1, bspline = "pspline(stafcder, df = 4)",
                    bdata = CarpenterFdaData$stafcder,
                    qi = "Hazard Ratio",
                    Xj = seq(1100, 1700, by = 10),
                    Xl = seq(1099, 1699, by = 10), spin = TRUE)

# Simulate Hazard Rate for orderent
Sim2 <- coxsimSpline(M1, bspline = "pspline(orderent, df = 4)",
                    bdata = CarpenterFdaData$orderent,
                    qi = "Hazard Rate",
                    Xj = seq(1, 30, by = 2), ci = 0.9, nsim = 10)

# Plot relative hazard
simGG(Sim1, alpha = 0.5)

# 3D plot hazard rate
simGG(Sim2, zlab = "orderent", fit = "quadratic")

# Create a time grid plot
# Find all points in time where baseline hazard was found
```

```
unique(Sim2$Time)

# Round time values so they can be exactly matched with FacetTime
Sim2$Time <- round(Sim2$Time, digits = 2)

# Create plot
simGG(Sim2, FacetTime = c(6.21, 25.68, 100.64, 202.36),
       ribbons = TRUE, alpha = 0.5)

# Simulated Fitted Values of stafcder
Sim3 <- coxsimSpline(M1, bspline = "pspline(stafcder, df = 4)",
                     bdata = CarpenterFdaData$stafcder,
                     qi = "Hazard Ratio",
                     Xj = seq(1100, 1700, by = 10),
                     Xl = seq(1099, 1699, by = 10), ci = 0.90)

## End(Not run)

# Plot simulated Hazard Ratios
# simGG(Sim3, xlab = "\nFDA Drug Review Staff", alpha = 0.2)
# simGG(Sim3, xlab = "\nFDA Drug Review Staff", alpha = 0.2,
#       SmoothSpline = TRUE, type = 'lines')
```

`simGG.simtvc` *Plot simulated time-interactive hazard ratios or stratified time-interactive hazard rates from Cox Proportional Hazards Models*

Description

`simGG.simtvc` uses ggplot2 to plot the simulated hazards from a `simtvc` class object created by `coxsimtvc` using ggplot2.

Usage

```
## S3 method for class 'simtvc'
simGG(obj, from = NULL, to = NULL, xlab = NULL,
  ylab = NULL, title = NULL, method = "auto", spalette = "Set1",
  legend = "legend", leg.name = "", lsize = 1, psize = 1, alpha = 0.2,
  type = "points", ...)
```

Arguments

`obj`	a `simtvc` class object
`from`	numeric time to start the plot from.
`to`	numeric time to plot to.
`xlab`	a label for the plot's x-axis.
`ylab`	a label of the plot's y-axis. The default uses the value of `qi`.

title	the plot's main title.
method	what type of smoothing method to use to summarize the center of the simulation distribution.
spalette	colour palette for when there are multiple sets of comparisons to plot. Default palette is "Set1". See scale_colour_brewer.
legend	specifies what type of legend to include (if applicable). The default is legend = "legend". To hide the legend use legend = FALSE. See the discrete_scale for more details.
leg.name	name of the legend (if applicable).
lsize	size of the smoothing line. Default is 1. See ggplot2.
psize	size of the plotted simulation points. Default is psize = 1. See ggplot2.
alpha	numeric. Alpha (e.g. transparency) for the points, lines, or ribbons. Default is alpha = 0.2. See ggplot2. Note, if type = "lines" or type = "points" then alpah sets the maximum value per line or point at the center of the distribution. Lines or points further from the center are more transparent the further they get from the middle.
type	character string. Specifies how to plot the simulations. Can be points, lines, or ribbons. If points then each simulation value will be plotted. If lines is chosen then each simulation is plotted using a different line. Note: any simulation with a value along its length that is outside of the specified central interval will be dropped. This is to create a smooth plot. If type = "ribbons" a plot will be created with shaded areas ('ribbons') for the minimum and maximum simulation values (i.e. the middle interval set with qi in coxsimSpline) as well as the central 50 percent of this area. It also plots a line for the median value of the full area, so values in method are ignored. One of the key advantages of using ribbons rather than points is that it creates plots with smaller file sizes.
...	Additional arguments. (Currently ignored.)

Details

Plots either a time-interactive hazard ratios, first differences, and relative hazards, or the hazard rates for multiple strata. Currently the strata legend labels need to be changed manually (see revalue in the plyr package) in the simtvc object with the strata component. Also, currently the x-axis tick marks and break labels must be adjusted manually for non-linear functions of time. Note: A dotted line is created at y = 1 (0 for first difference), i.e. no effect, for time-varying hazard ratio graphs. No line is created for hazard rates.

Value

a gg ggplot class object

References

Licht, Amanda A. 2011. "Change Comes with Time: Substantive Interpretation of Nonproportional Hazards in Event History Analysis." Political Analysis 19: 227-43.

Examples

```
## Not run:
# Load Golub & Steunenberg (2007) Data
data("GolubEUPData")

# Load survival package
library(survival)

# Expand data
GolubEUPData <- SurvExpand(GolubEUPData, GroupVar = 'caseno',
                     Time = 'begin', Time2 = 'end', event = 'event')

# Create time interactions
BaseVars <- c('qmv', 'backlog', 'coop', 'codec', 'qmvpostsea', 'thatcher')
GolubEUPData <- tvc(GolubEUPData, b = BaseVars, tvar = 'end', tfun = 'log')

# Run Cox PH Model
M2 <- coxph(Surv(begin, end, event) ~ qmv + qmvpostsea + qmvpostteu +
                coop + codec + eu9 + eu10 + eu12 + eu15 + thatcher +
                agenda + backlog + qmv_log + qmvpostsea_log + coop_log +
                codec_log + thatcher_log + backlog_log,
            data = GolubEUPData, ties = "efron")

# Create simtvc object for Relative Hazard
Sim1 <- coxsimtvc(obj = M1, b = "qmv", btvc = "qmv_log",
                  tfun = "log", from = 80, to = 2000,
                  Xj = 1, by = 15, ci = 0.99, nsim = 100)

# Create simtvc object for Relative Hazard
Sim1 <- coxsimtvc(obj = M1, b = "qmv", btvc = "qmv_log",
                  tfun = "log", from = 80, to = 2000,
                  Xj = 1, by = 15, ci = 0.99)

# Create simtvc object for First Difference
Sim2 <- coxsimtvc(obj = M1, b = "qmv", btvc = "qmv_log",
                 qi = "First Difference", Xj = 1,
                 tfun = "log", from = 80, to = 2000,
                 by = 15, ci = 0.95)

# Create simtvc object for Hazard Ratio
Sim3 <- coxsimtvc(obj = M1, b = "backlog", btvc = "backlog_log",
                  qi = "Hazard Ratio", Xj = c(191, 229),
                  Xl = c(0, 0),
                  tfun = "log", from = 100, to = 2000,
                  by = 15, ci = 0.99)

# Create plots
simGG(Sim1, legend = FALSE)
simGG(Sim2)
simGG(Sim3, leg.name = "Comparision", from = 1200, type = 'lines')

## End(Not run)
```

simPH *An R package for simulating and plotting quantities of interest from Cox Proportional Hazard models.*

Description

An R package for simulating and plotting quantities of interest (relative hazards, first differences, and hazard ratios) for linear coefficients, multiplicative interactions, polynomials, penalised splines, and non-proportional hazards, as well as stratified survival curves from Cox Proportional Hazard models.

The package includes the following simulation functions:

- coxsimLinear: a function for simulating relative hazards, first differences, hazard ratios, and hazard rates for linear, non-time interacted covariates from Cox Proportional Hazard (coxph) models.
- coxsimtvc: a function for simulating time-interactive hazards (relative hazards, first differences, and hazard ratios) for covariates from Cox Proportional Hazard models. The function will calculate time-interactive hazard ratios for multiple strata estimated from a stratified Cox Proportional Hazard model.
- coxsimSpline: a function for simulating quantities of interest from penalised splines using multivariate normal distributions. Currently does not support simulating hazard rates from stratified models. Note: be extremely careful about the number of simulations you ask the function to find. It is very easy to ask for more than your computer can handle.
- coxsimPoly: a function for simulating quantities of interest for a range of values for a polynomial nonlinear effect from Cox Proportional Hazard models.
- coxsimInteract: a function for simulating quantities of interest for linear multiplicative interactions, including marginal effects and hazard rates.

Results from these functions can be plotted using the simGG method.

The package also includes two functions that make it easier to create time interactions:

- SurvExpand: a function to convert a data frame of non-equal interval continuous observations into equal interval continuous observations.
- tvc: a function to create time interaction variables that can be used in a coxph model (or any other model with time interactions).
- setXl: a function for setting valid Xl values given a sequence of fitted Xj values. This makes it more intituitive to find hazard ratios and first differences for comparisons between some Xj fitted values and Xl values other than 0.

SurvExpand *Convert a data frame of non-equal interval continuous observations into equal interval continuous observations*

Description

SurvExpand convert a data frame of non-equal interval continuous observations into equal interval continuous observations. This is useful for creating time-interactions with tvc.

Usage

```
SurvExpand(data, GroupVar, Time, Time2, event, PartialData = TRUE,
  messages = TRUE)
```

Arguments

data	a data frame.
GroupVar	a character string naming the unit grouping variable.
Time	a character string naming the variable with the interval start time.
Time2	a character string naming the variable with the interval end time.
event	a character string naming the event variable. Note: must be numeric with 0 indicating no event.
PartialData	logical indicating whether or not to keep only the expanded data required to find the Cox partial likelihood.
messages	logical indicating if you want messages returned while the function is working.

Details

The function primarily prepares data from the creation of accurate time-interactions with the tvc command. Note: the function will work best if your original time intervals are recorded in whole numbers. It also currently does not support repeated events data.

Value

Returns a data frame where observations have been expanded into equally spaced time intervals.

See Also

tvc

Examples

```
## Not run:
# Load Golub & Steunenberg (2007) Data
data("GolubEUPData")

# Subset PURELY TO SPEED UP THE EXAMPLE
GolubEUPData <- GolubEUPData[1:500, ]

# Expand data
GolubEUPDataExpand <- SurvExpand(GolubEUPData, GroupVar = 'caseno',
                        Time = 'begin', Time2 = 'end', event = 'event')

## End(Not run)
```

---

tvc *Create a time interaction variable*

Description

tvc creates a time interaction variable that can be used in a coxph model (or any other model with time interactions)

Usage

```
tvc(data, b, tvar, tfun = "linear", pow = NULL, vector = FALSE)
```

Arguments

data	a data frame
b	the non-time interacted variable's name. Either a single value or a vector of variable names can be entered.
tvar	the time variable's name
tfun	function of time that btvc was multiplied by. Default is `tfun = "linear"`. Can also be `tfun = 'log'` (natural log) and `tfun = 'power'`. If `tfun = 'power'` then the pow argument needs to be specified also.
pow	if `tfun = 'power'`, then use pow to specify what power to raise the time interaction to.
vector	logical. Whether or not to return one vector a or a data frame. Can only be used if only one b is included.

Details

Interacts a variable with a specified function of time. Possible functions of time include `'linear'`, natural `'log'`, and exponentiated (`'power'`).

Value

a data frame or vector. If a data frame is returned it will include all of the original variables as well as the interactions denoted by a variable name 'bn_tfun', where bn is one variable name from b and tfun as entered into the function.

See Also

SurvExpand, simGG.simtvc, coxsimtvc, survival, and coxph

Examples

```
# Load Golub & Steunenberg (2007) Data
data('GolubEUPData')

# Subset PURELY TO SPEED UP THE EXAMPLE
GolubEUPData <- GolubEUPData[1:500, ]

# Expand data into equally spaced time intervals
GolubEUPData <- SurvExpand(GolubEUPData, GroupVar = 'caseno',
                  Time = 'begin', Time2 = 'end', event = 'event')

# Create natural log time interaction with the qmv variable
GolubEUPData$Lqmv <- tvc(GolubEUPData, b = 'qmv', tvar = 'end', tfun = 'log',
                         vector = TRUE)

# Create interactions for a vector of variables
BaseVars <- c('qmv', 'backlog', 'coop', 'codec', 'qmvpostsea', 'thatcher')

Test <- tvc(GolubEUPData, b = BaseVars, tvar = 'end', tfun = 'log')
```

Index

∗Topic **datasets**
  CarpenterFdaData, 2
  GolubEUPData, 16
∗Topic **internals**
  MinMaxLines, 16
∗Topic **utilities**
  setXl, 17
  SurvExpand, 33
  tvc, 34

CarpenterFdaData, 2
coxph, *3–9*, *11–13*, *32*, *35*
coxsimInteract, 3, *20*, *32*
coxsimLinear, 5, *23*, *28*, *32*
coxsimPoly, 7, *25*, *32*
coxsimSpline, 9, *20*, *22*, *25*, *27*, *30*, *32*
coxsimtvc, 12, *29*, *32*, *35*

discrete_scale, *19*, *22*, *24*, *30*

ggfitStrata, 14
ggplot2, *14, 15*, *19–30*
GolubEUPData, 16

I, *3*, *6–8*

MinMaxLines, 16

plot.survfit, *14*
pspline, *9*

scale_colour_brewer, *19*, *22*, *24*, *30*
scatter3d, *26–28*
setXl, 17
simGG, *3–13*, 18, *32*
simGG.siminteract, *18*, 19
simGG.simlinear, *18*, *20*, 21
simGG.simpoly, *18*, 24
simGG.simspline, *18*, 26
simGG.simtvc, *18*, *23*, *28*, 29, *35*
simPH, 32
smooth.spline, *26*
strata, *4*, *6*, *8*, *11*, *13*, *15*
SurvExpand, *32*, 33, *35*
survfit, *15*
survival, *4*, *6*, *8*, *11*, *13*, *35*

tvc, *32, 33*, 34