Package ‘flexsurv’

July 2, 2014

Type Package

Title Flexible parametric survival models

Version 0.3

Date 2014-01-19

Author Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

Maintainer Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

Description Flexible parametric models for time-to-event data, including the Royston-Parmar spline model, generalized gamma and generalized F distributions. Easily extensible to fit user-defined parametric distributions.

License GPL (>= 2)

Depends survival

Imports muhaz,mvtnorm

Suggests eha

LazyLoad yes

Repository CRAN

Repository/R-Forge/Project flexsurv

Repository/R-Forge/Revision 24

Repository/R-Forge/DateTimeStamp 2014-01-19 20:04:42

Date/Publication 2014-01-20 11:12:59

NeedsCompilation no

R topics documented:

flexsurv-package 2
bc 3
flexsurvreg 5
flexsurvspline 9
GenF 12
GenF.orig 14
GenGamma 16
GenGamma.orig 18
Gompertz 20
lines.flexsurvreg 21
plot.flexsurvreg 22
qgeneric 24
summary.flexsurvreg 26
Survspline 27

Index **29**

`flexsurv-package` *flexsurv: Flexible parametric models for time-to-event data*

Description

flexsurv: Flexible parametric models for time-to-event data, including the generalized gamma, the generalized F and the Royston-Parmar spline model.

Details

`flexsurvreg` fits parametric models for time-to-event (survival) data. Data may be right-censored or left-truncated. Several built-in parametric distributions are available. Any user-defined parametric model can also be employed by supplying a list with basic information about the distribution, including the density and probability functions.

Covariates can be included using a linear model on any parameter of the distribution, log-transformed to the real line if necessary. This typically defines an accelerated failure time or proportional hazards model, depending on the distribution and parameter.

`flexsurvspline` fits the flexible survival model of Royston and Parmar (2002) in which the log cumulative hazard is modelled as a natural cubic spline function of log time, including a linear term for covariates. Alternative proportional odds or probit parameterisations are available.

Output from the models can be presented as survivor, cumulative hazard and hazard functions (`summary.flexsurvreg`). These can be plotted against nonparametric estimates (`plot.flexsurvreg`) to assess goodness-of-fit.

Distribution ("dpqr") functions for the generalized gamma and F distributions are given in `GenGamma`, `GenF` (preferred parameterisations) and `GenGamma.orig`, `GenF.orig` (original parameterisations).

`flexsurv` also includes the standard Gompertz distribution with unrestricted shape parameter, see `Gompertz`.

Related R packages

flexsurv was written to encourage the use of flexible distributions to account for model uncertainty in survival analysis, initially the three-parameter generalized gamma, four-parameter generalized F and the Royston-Parmar spline model. However it was straightforward to modularise the design of the code to accept any generic parametric distribution.

survreg from the **survival** package, the recommended R package for survival analysis, supports two-parameter location-scale parametric models.

The **eha** package includes functions phreg and aftreg for parametric survival modelling under a variety of distributions and proportional hazards or accelerated failure time parameterisations.

Other facilities for generic maximum likelihood model fitting exist, for example fitdistr in the **MASS** package. flexsurvreg is intended to provide typical outputs and summaries of interest to survival analysts, particularly in medical applications. Feature requests along these lines are welcome.

Note that if an R package provides density and probability functions for a parametric distribution, it can then be used easily in flexsurvreg. For instance, several "reliability" distributions used in industrial statistics are available in the **VGAM** package. Please report unexplained inconsistencies in results between **flexsurv** and other software.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Royston, P. and Parmar, M. (2002). Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 21(1):2175-2197.

Cox, C. (2008). The generalized F distribution: An umbrella for parametric survival analysis. Statistics in Medicine 27:4301-4312.

Cox, C., Chu, H., Schneider, M. F. and Muñoz, A. (2007). Parametric survival analysis and taxonomy of hazard functions for the generalized gamma distribution. Statistics in Medicine 26:4252-4374

---

bc — *Breast cancer survival data*

Description

Survival times of 686 patients with primary node positive breast cancer.

Usage

```
data(bc)
```

Format

A data frame with 686 rows.

`censrec`	(numeric)	1=dead, 0=censored
`rectime`	(numeric)	Time of death or censoring in days
`group`	(numeric)	Prognostic group: `"Good"`,`"Medium"` or `"Poor"`, from a regression model developed by Sauerbrei and Royston (1999).

Source

German Breast Cancer Study Group, 1984-1989. Used as a reference dataset for the spline-based survival model of Royston and Parmar (2002), implemented here in `flexsurvspline`. Originally provided with the `stpm` (Royston 2001, 2004) and `stpm2` (Lambert 2009, 2010) Stata modules.

References

Royston, P. and Parmar, M. (2002). Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 21(1):2175-2197.

Sauerbrei, W. and Royston, P. (1999). Building multivariable prognostic and diagnostic models: transformation of the predictors using fractional polynomials. Journal of the Royal Statistical Society, Series A 162:71-94.

See Also

`flexsurvspline`

`flexsurvreg` *Flexible parametric regression for time-to-event data*

Description

Parametric regression for time-to-event data using the generalized F and other flexible distributions. Users may easily extend this function with their own survival distributions.

Usage

```
flexsurvreg(formula, data, weights, subset, na.action, dist, inits,
            fixedpars=NULL, cl=0.95,...)
```

Arguments

`formula` A formula expression in conventional R linear modelling syntax. The response must be a survival object as returned by the `Surv` function, and any covariates are given on the right-hand side. For example,

`Surv(time, dead) ~ age + sex`

Only `Surv` objects of `type="right"` or `type="counting"`, corresponding to right-censored and/or left-truncated observations, are supported.

If there are no covariates, specify `1` on the right hand side, for example `Surv(time, dead) ~ 1`.

By default, covariates are placed on the "location" parameter of the distribution, typically the "scale" or "rate" parameter, through a linear model, or a log-linear model if this parameter must be positive. This gives an accelerated failure time model or a proportional hazards model, depending on the distribution.

Covariates can be placed on other parameters by using the name of the parameter as a function in the formula. For example, in a Weibull model, the following expresses the scale parameter in terms of age and a treatment variable `treat`, and the shape parameter in terms of sex and treatment.

```
Surv(time, dead) ~ age + treat + shape(sex) + shape(treatment)
```

`data` A data frame in which to find variables supplied in `formula`. If not given, the variables should be in the working environment.

`weights` Optional vector of case weights.

`subset` Vector of integer or logicals specifying the subset of the observations to be used in the fit.

`na.action` a missing-data filter function, applied after any 'subset' argument has been used. Default is `options()$na.action`.

`dist` Either one of the following strings identifying a built-in distribution:

`"gengamma"`	Generalized gamma (stable parameterisation)
`"gengamma.orig"`	Generalized gamma (original parameterisation)
`"genf"`	Generalized F (stable parameterisation)
`"genf.orig"`	Generalized F (original parameterisation)
`"weibull"`	Weibull
`"gamma"`	Gamma
`"exp"`	Exponential
`"lnorm"`	Log-normal
`"gompertz"`	Gompertz

or a list specifying a custom distribution. See section "Custom distributions" below for how to construct this list.

The parameterisations of the built-in distributions used here are the same as in their built-in distribution functions: `dgengamma`, `dgengamma.orig`, `dgenf`, `dgenf.orig`, `dweibull`, `dgamma`, `dexp`, `dlnorm`, `dgompertz`, respectively. The functions in base R are used where available, otherwise, they are provided in this package.

Note that the Weibull parameterisation is different from that in `survreg`, instead it is consistent with `dweibull`. The `"scale"` reported by `survreg` is equivalent to `1/shape` as defined by `dweibull` and hence `flexsurvreg`. The first coefficient `(Intercept)` reported by `survreg` is equivalent to `log(scale)` in `dweibull` and `flexsurvreg`.

Similarly in the exponential distribution, the rate, rather than the mean, is modelled on covariates.

`inits` A numeric vector giving initial values for each unknown parameter. If not specified, default initial values are chosen from a simple summary of the uncensored survival time, for example the mean is often used to initialize scale parameters.

See the object `flexsurv.dists` in the source for the exact methods used. If the likelihood surface may be uneven, it is advised to run the optimisation starting from various different initial values to ensure convergence to the true global maximum.

`fixedpars` Vector of indices of parameters whose values will be fixed at their initial values during the optimisation. The indices are ordered with parameters of the baseline distribution coming first, followed by covariate effects. For example, in a stable generalized Gamma model with two covariates, to fix the third of three generalized gamma parameters (the shape `Q`, see the help for `GenGamma`) and the second covariate, specify `fixedpars = c(3, 5)`

`cl` Width of symmetric confidence intervals for maximum likelihood estimates, by default 0.95.

`...` Optional arguments to the general-purpose R optimisation routine `optim`. For example, the BFGS optimisation algorithm is the default in `flexsurvreg`, but this can be changed, for example to `method="Nelder-Mead"` which can be more robust to poor initial values. If the optimisation fails to converge, consider normalising the problem using, for example, `control=list(fnscale = 2500)`, for example, replacing 2500 by a number of the order of magnitude of the likelihood. If 'false' convergence is reported with a non-positive-definite Hessian, then consider tightening the tolerance criteria for convergence. If the optimisation takes a long time, intermediate steps can be printed using the `trace` argument of the control list. See `optim` for details.

Details

Parameters are estimated by maximum likelihood using the algorithms available in the standard R `optim` function. Parameters defined to be positive are estimated on the log scale. Confidence intervals are estimated from the Hessian at the maximum, and transformed back to the original scale of the parameters.

The usage of `flexsurvreg` is intended to be as similar as possible to `survreg` in the **survival** package.

Value

A list of class `"flexsurvreg"` with the following elements.

`call` A copy of the function call, for use in post-processing.

`dlist` List defining the survival distribution used.

`res` Matrix of maximum likelihood estimates and confidence limits, with parameters on their natural scales.

`res.t` Matrix of maximum likelihood estimates and confidence limits, with parameters all transformed to the real line. The `coef`, `vcov` and `confint` methods for `flexsurvreg` objects work on this scale.

`loglik` Log-likelihood

`AIC` Akaike's information criterion (-2*log likelihood + 2*number of estimated parameters)

Custom distributions

flexsurvreg is intended to be easy to extend to handle new distributions. To define a new distribution for use in flexsurvreg, construct a list with the following elements:

`name`: A string naming the distribution. If this is called `"dist"`, for example, then there must be a function called `ddist` in the working environment which defines the probability density, and a function called `pdist` which defines the probability distribution or cumulative density. These functions may be in an add-on package (see below for an example) or may be user-written. Arguments other than parameters must be named in the conventional way – for example `x` for the first argument of the density function, as in `dnorm(x, ...)` and `q` for the first argument of the probability function.

`pars`: Vector of strings naming the parameters of the distribution. These must be the same names as the arguments of the density and probability functions.

`location`: Name of the parameter which can be modelled as a linear function of covariates, possibly after transformation.

`transforms`: Vector of R functions which transform the range of values taken by each parameter onto the real line. For example, `c(log, log)` for a distribution with two positive parameters.

`inv.transforms`: Vector of R functions defining the corresponding inverse transformations.

`inits`: A function of the uncensored survival times `t`, which returns a vector of reasonable initial values for maximum likelihood estimation of each parameter. For example, `function(t){ c(1, mean(t)) }` will always initialize the first of two parameters at 1, and the second (a scale parameter, for instance) at the mean of `t`.

For example, suppose we want to use a log-logistic survival distribution. This is available in the CRAN package **eha**, which provides conventionally-defined density and probability functions called `dllogis` and `pllogis`. See the Examples below for the custom list in this case, and the subsequent command to fit the model.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Jackson, C. H. and Sharples, L. D. and Thompson, S. G. (2010) Survival models in health economic evaluations: balancing fit and parsimony to improve prediction. International Journal of Biostatistics 6(1):Article 34.

Cox, C. (2008) The generalized F distribution: An umbrella for parametric survival analysis. Statistics in Medicine 27:4301-4312.

Cox, C., Chu, H., Schneider, M. F. and Muñoz, A. (2007) Parametric survival analysis and taxonomy of hazard functions for the generalized gamma distribution. Statistics in Medicine 26:4252-4374

See Also

`flexsurvspline` for flexible survival modelling using the spline model of Royston and Parmar.

`plot.flexsurvreg` and `lines.flexsurvreg` to plot fitted survival, hazards and cumulative hazards from models fitted by `flexsurvreg` and `flexsurvspline`.

Examples

```
data(ovarian)
## Compare generalized gamma fit with Weibull
fitg <- flexsurvreg(formula = Surv(futime, fustat) ~ 1, data = ovarian, dist="gengamma")
fitg
fitw <- flexsurvreg(formula = Surv(futime, fustat) ~ 1, data = ovarian, dist="weibull")
fitw
plot(fitg)
lines(fitw, col="blue", lwd.ci=1, lty.ci=1)
## Identical AIC, probably not enough data in this simple example for a
## very flexible model to be worthwhile.

## Custom distribution
library(eha)  ## make "dllogis" and "pllogis" available to the working environment
custom.llogis <- list(name="llogis",
                      pars=c("shape","scale"),
                      location="scale",
                      transforms=c(log, log),
                      inv.transforms=c(exp, exp),
                      inits=function(t){ c(1, median(t)) })
fitl <- flexsurvreg(formula = Surv(futime, fustat) ~ 1, data = ovarian, dist=custom.llogis)
fitl
lines(fitl, col.fit="purple", col.ci="purple")
```

`flexsurvspline` *Flexible survival regression using the Royston/Parmar spline model.*

Description

Flexible parametric modelling of time-to-event data using the spline model of Royston and Parmar (2002).

Usage

```
flexsurvspline(formula, data, k=0, knots=NULL, scale="hazard", weights,
               subset, na.action, inits=NULL, fixedpars=NULL, cl=0.95,...)
```

Arguments

`formula` A formula expression in conventional R linear modelling syntax. The response must be a survival object as returned by the `Surv` function, and any covariates are given on the right-hand side. For example,

`Surv(time, dead) ~ age + sex`

If there are no covariates, specify `1` on the right hand side, for example `Surv(time, dead) ~ 1`. Only `Surv` objects of `type="right"` or `type="counting"`, corresponding to right-censored and/or left-truncated observations, are supported.

`data` A data frame in which to find variables supplied in `formula`. If not given, the variables should be in the working environment.

`k` Number of knots in the spline. The default `k=0` gives a Weibull, log-logistic or lognormal model, if `"scale"` is `"hazard"`, `"odds"` or `"normal"` respectively. `k` is equivalent to `df-1` in the notation of `stpm` for Stata. The knots are then chosen as equally-spaced quantiles of the log uncensored survival times, for example, at the median with one knot, or at the 33% and 67% quantiles of log time with two knots. To override this default knot placement, specify `knots` instead.

`knots` Locations of knots on the axis of log time. If not specified, knot locations are chosen as described in `k` above. Either `k` or `knots` must be specified. If both are specified, `knots` overrides `k`.

`scale` If `"hazard"`, the log cumulative hazard is modelled as a spline function of log time.

If `"odds"`, the log cumulative odds is modelled as a spline function of log time.

If `"normal"`, $-\Phi^{-1}(S(t))$ is modelled as a spline function of log time, where $\Phi^{-1}()$ is the inverse normal distribution function `qnorm`.

`weights` Optional vector of case weights.

`subset` Vector of integer or logicals specifying the subset of the observations to be used in the fit.

`na.action` a missing-data filter function, applied after any 'subset' argument has been used. Default is 'options()$na.action'.

`inits` A numeric vector giving initial values for each unknown parameter. If not specified, default initial values are chosen by estimating the baseline survival at each observed death time from the equivalent Cox model, transforming to the log cumulative hazard $\log(H)$ (or equivalent under the odds or normal models) then performing a linear regression of $\log(H)$ on the spline basis and covariates.

`fixedpars` Vector of indices of parameters whose values will be fixed at their initial values during optimisation. The indices are ordered with the intercept `"gamma0"` first, then the remaining spline coefficients `"gamma1","gamma2"...` followed by covariate effects.

`cl` Width of symmetric confidence intervals for maximum likelihood estimates, by default 0.95.

`...` Optional arguments to the general-purpose R optimisation routine `optim`. See `flexsurvreg` for examples.

Details

In the spline-based survival model of Royston and Parmar (2002), a transformation $g(S(t,z))$ of the survival function is modelled as a natural cubic spline function of log time $x = \log(t)$ plus linear effects of covariates z.

$$g(S(t,z)) = s(x,\boldsymbol{\gamma}) + \boldsymbol{\beta}^T\mathbf{z}$$

The proportional hazards model (`scale="hazard"`) defines $g(S(t,\mathbf{z})) = \log(-\log(S(t,\mathbf{z}))) = \log(H(t,\mathbf{z}))$, the log cumulative hazard.

The proportional odds model (`scale="odds"`) defines $g(S(t,\mathbf{z})) = \log(S(t,\mathbf{z})^{-1} - 1)$, the log cumulative odds.

The probit model (`scale="normal"`) defines $g(S(t,\mathbf{z})) = -\Phi^{-1}(S(t,\mathbf{z}))$, where $\Phi^{-1}()$ is the inverse normal distribution function `qnorm`.

With no knots, the spline reduces to a linear function, and these models are equivalent to Weibull, log-logistic and lognormal models respectively.

Natural cubic splines are cubic splines constrained to be linear beyond boundary knots k_{min}, k_{max}. The spline function is defined as

$$s(x,\boldsymbol{\gamma}) = \gamma_0 + \gamma_1 x + \gamma_2 v_1(x) + \ldots + \gamma_{m+1} v_m(x)$$

where $v_j(x)$ is the jth basis function

$$v_j(x) = (x-k_j)_+^3 - \lambda_j (x-k_{min})_+^3 - (1-\lambda_j)(x-k_{max})_+^3$$

$$\lambda_j = \frac{k_{max}-k_j}{k_{max}-k_{min}}$$

and $(x-a)_+ = max(0, x-a)$.

Parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ are estimated by maximum likelihood using the algorithms available in the standard R `optim` function. Confidence intervals are estimated from the Hessian at the maximum.

Value

A list of class `"flexsurvreg"` with the following elements.

`call` A copy of the function call, for use in post-processing.

`k` Number of knots.

`knots` Location of knots on the log time axis.

`res` Matrix of maximum likelihood estimates and confidence limits. Spline coefficients are labelled `"gamma..."`, and covariate effects are labelled with the names of the covariates.

Coefficients `gamma1,gamma2,...` here are the equivalent of `s0,s1,...` in Stata `streg`, and `gamma0` is the equivalent of the `xb` constant term. To reproduce results, use the `noorthog` option in Stata, since no orthogonalisation is performed on the spline basis here.

In the Weibull model, for example, `gamma0,gamma1` are `-shape log(scale), shape` respectively in `dweibull` or `flexsurvreg` notation, or (`-Intercept/scale`, `1/scale`) in `survreg` notation.

In the log-logistic model with shape `a` and scale `b` (as in `dllogis` from the **eha** package), `1/b^a` is equivalent to `exp(gamma0)`, and `a` is equivalent to `gamma1`.

In the log-normal model with log-scale mean `mu` and standard deviation `sigma`, `-mu/sigma` is equivalent to `gamma0` and `1/sigma` is equivalent to `gamma1`.

`loglik` The maximised log-likelihood. This will differ from Stata, where the sum of the log uncensored survival times is added to the log-likelihood in survival models, to remove dependency on the time scale.

`AIC` Akaike's information criterion (-2*log likelihood + 2*number of estimated parameters)

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Royston, P. and Parmar, M. (2002). Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 21(1):2175-2197.

See Also

`flexsurvreg` for flexible survival modelling using fully parametric distributions including the generalized F and gamma.

`plot.flexsurvreg` and `lines.flexsurvreg` to plot fitted survival, hazards and cumulative hazards from models fitted by `flexsurvspline` and `flexsurvreg`.

Examples

```
data(bc)
bc$recyrs <- bc$rectime/365

## Best-fitting model to breast cancer data from Royston and Parmar (2002)
## One internal knot (2 df) and cumulative odds scale
spl <- flexsurvspline(Surv(recyrs, censrec) ~ group, data=bc, k=1, scale="odds")

## Fitted survival
plot(spl, ci=TRUE, lwd=3, lwd.ci=1, col.ci="gray")

## Simple Weibull model fits much less well
splw <- flexsurvspline(Surv(recyrs, censrec) ~ group, data=bc, k=0, scale="hazard")
lines(splw, col="blue")

## Alternative way of fitting the Weibull
splw2 <- flexsurvreg(Surv(recyrs, censrec) ~ group, data=bc, dist="weibull")
```

`GenF` *Generalized F distribution*

Description

Density, distribution function, hazards, quantile function and random generation for the generalized F distribution, using the reparameterisation by Prentice (1975).

Usage

```
dgenf(x, mu=0, sigma=1, Q, P, log = FALSE)
pgenf(q, mu=0, sigma=1, Q, P, lower.tail = TRUE, log.p = FALSE)
qgenf(p, mu=0, sigma=1, Q, P, lower.tail = TRUE, log.p = FALSE)
rgenf(n, mu=0, sigma=1, Q, P)
Hgenf(x, mu=0, sigma=1, Q, P)
hgenf(x, mu=0, sigma=1, Q, P)
```

Arguments

`x,q`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mu`	Vector of location parameters.
`sigma`	Vector of scale parameters.
`Q`	Vector of first shape parameters.
`P`	Vector of second shape parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \le x)$, otherwise, $P(X > x)$.

Details

If $y \sim F(2s_1, 2s_2)$, and $w = \log(y)$ then $x = \exp(w\sigma + \mu)$ has the original generalized F distribution with location parameter μ, scale parameter $\sigma > 0$ and shape parameters s_1, s_2.

In this more stable version described by Prentice (1975), s_1, s_2 are replaced by shape parameters Q, P, with $P > 0$, and

$$s_1 = 2(Q^2 + 2P + Q\delta)^{-1}, \quad s_2 = 2(Q^2 + 2P - Q\delta)^{-1}$$

equivalently

$$Q = \left(\frac{1}{s_1} - \frac{1}{s_2}\right)\left(\frac{1}{s_1} + \frac{1}{s_2}\right)^{-1/2}, \quad P = \frac{2}{s_1 + s_2}$$

Define $\delta = (Q^2 + 2P)^{1/2}$, and $w = (\log(x) - \mu)\delta/\sigma$, then the probability density function of x is

$$f(x) = \frac{\delta(s_1/s_2)^{s_1} e^{s_1 w}}{\sigma x(1 + s_1 e^w/s_2)^{(s_1+s_2)} B(s_1, s_2)}$$

The original parameterisation is available in this package as `dgenf.orig`, for the sake of completion / compatibility. With the above definitions,

`dgenf(x, mu=mu, sigma=sigma, Q=Q, P=P) = dgenf.orig(x, mu=mu, sigma=sigma/delta, s1=s1, s2=s2)`

The generalized F distribution with `P=0` is equivalent to the generalized gamma distribution `dgengamma`, so that `dgenf(x, mu, sigma, Q, P=0)` equals `dgengamma(x, mu, sigma, Q)`. The generalized gamma reduces further to several common distributions, as described in the `GenGamma` help page.

The generalized F distribution includes the log-logistic distribution (see `Llogis`) as a further special case:

```
dgenf(x, mu=mu, sigma=sigma, Q=0, P=1) = dllogis(x, shape=sqrt(2)/sigma, scale=exp(mu))
```

The range of hazard trajectories available under this distribution are discussed in detail by Cox (2008). Jackson et al. (2010) give an application to modelling oral cancer survival for use in a health economic evaluation of screening.

Value

`dgenf` gives the density, `pgenf` gives the distribution function, `qgenf` gives the quantile function, `rgenf` generates random deviates, `Hgenf` retuns the cumulative hazard and `hgenf` the hazard.

Note

The parameters `Q` and `P` are usually called q and p in the literature - they were made upper-case in these R functions to avoid clashing with the conventional arguments `q` in the probability function and `p` in the quantile function.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

R. L. Prentice (1975). Discrimination among some parametric models. Biometrika 62(3):607-614.

Cox, C. (2008). The generalized F distribution: An umbrella for parametric survival analysis. Statistics in Medicine 27:4301-4312.

Jackson, C. H. and Sharples, L. D. and Thompson, S. G. (2010). Survival models in health economic evaluations: balancing fit and parsimony to improve prediction. International Journal of Biostatistics 6(1):Article 34.

See Also

`GenF.orig`, `GenGamma`

`GenF.orig` *Generalized F distribution (original parameterisation)*

Description

Density, distribution function, quantile function and random generation for the generalized F distribution, using the less flexible original parameterisation described by Prentice (1975).

Usage

```
dgenf.orig(x, mu=0, sigma=1, s1, s2, log = FALSE)
pgenf.orig(q, mu=0, sigma=1, s1, s2, lower.tail = TRUE, log.p = FALSE)
qgenf.orig(p, mu=0, sigma=1, s1, s2, lower.tail = TRUE, log.p = FALSE)
rgenf.orig(n, mu=0, sigma=1, s1, s2)
Hgenf.orig(x, mu=0, sigma=1, s1, s2)
hgenf.orig(x, mu=0, sigma=1, s1, s2)
```

Arguments

`x,q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mu`	Vector of location parameters.
`sigma`	Vector of scale parameters.
`s1`	Vector of first F shape parameters.
`s2`	vector of second F shape parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.

Details

If $y \sim F(2s_1, 2s_2)$, and $w = \log(y)$ then $x = \exp(w\sigma + \mu)$ has the original generalized F distribution with location parameter μ, scale parameter $\sigma > 0$ and shape parameters $s_1 > 0, s_2 > 0$. The probability density function of x is

$$f(x|\mu, \sigma, s_1, s_2) = \frac{(s_1/s_2)^{s_1} e^{s_1 w}}{\sigma x (1 + s_1 e^w / s_2)^{(s_1+s_2)} B(s_1, s_2)}$$

where $w = (\log(x) - \mu)/\sigma$, $B(s_1, s_2) = \Gamma(s_1)\Gamma(s_2)/\Gamma(s_1 + s_2)$ is the beta function.

As $s_2 \to \infty$, the distribution of x tends towards an original generalized gamma distribution with the following parameters:

`dgengamma.orig(x, shape=1/sigma, scale=exp(mu) / s1^sigma, k=s1)`

See `GenGamma.orig` for how this includes several other common distributions as special cases.

The alternative parameterisation of the generalized F distribution, originating from Prentice (1975) and given in this package as `GenF`, is preferred for statistical modelling, since it is more stable as s_1 tends to infinity, and includes a further new class of distributions with negative first shape parameter. The original is provided here for the sake of completion and compatibility.

Value

`dgenf.orig` gives the density, `pgenf.orig` gives the distribution function, `qgenf.orig` gives the quantile function, `rgenf.orig` generates random deviates, `Hgenf.orig` retuns the cumulative hazard and `hgenf.orig` the hazard.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

R. L. Prentice (1975). Discrimination among some parametric models. Biometrika 62(3):607-614.

See Also

GenF, GenGamma.orig, GenGamma

GenGamma *Generalized gamma distribution*

Description

Density, distribution function, hazards, quantile function and random generation for the generalized gamma distribution, using the parameterisation originating from Prentice (1974). Also known as the (generalized) log-gamma distribution.

Usage

```
dgengamma(x, mu=0, sigma=1, Q, log = FALSE)
pgengamma(q, mu=0, sigma=1, Q, lower.tail = TRUE, log.p = FALSE)
qgengamma(p, mu=0, sigma=1, Q, lower.tail = TRUE, log.p = FALSE)
rgengamma(n, mu=0, sigma=1, Q)
Hgengamma(x, mu=0, sigma=1, Q)
hgengamma(x, mu=0, sigma=1, Q)
```

Arguments

x,q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.
mu	Vector of "location" parameters.
sigma	Vector of "scale" parameters. Note the inconsistent meanings of the term "scale" - this parameter is analogous to the (log-scale) standard deviation of the log-normal distribution, "sdlog" in dlnorm, rather than the "scale" parameter of the gamma distribution dgamma. Constrained to be positive.
Q	Vector of shape parameters.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.

Details

If $\gamma \sim Gamma(Q^{-2}, 1)$, and $w = log(Q^2\gamma)/Q$, then $x = \exp(\mu + \sigma w)$ follows the generalized gamma distribution with probability density function

$$f(x|\mu, \sigma, Q) = \frac{|Q|(Q^{-2})^{Q^{-2}}}{\sigma x \Gamma(Q^{-2})} \exp(Q^{-2}(Qw - \exp(Qw)))$$

This parameterisation is preferred to the original parameterisation of the generalized gamma by Stacy (1962) since it is more numerically stable near to $Q = 0$ (the log-normal distribution), and allows $Q <= 0$. The original is available in this package as `dgengamma.orig`, for the sake of completion and compatibility with other software - this is implicitly restricted to `Q`>0 (or `k`>0 in the original notation). The parameters of `dgengamma` and `dgengamma.orig` are related as follows.

```
dgengamma.orig(x, shape=shape, scale=scale, k=k) =

dgengamma(x, mu=log(scale) + log(k)/shape, sigma=1/(shape*sqrt(k)), Q=1/sqrt(k))
```

The generalized gamma distribution simplifies to the gamma, log-normal and Weibull distributions with the following parameterisations:

```
dgengamma(x, mu, sigma, Q=0)       =  dlnorm(x, mu, sigma)
dgengamma(x, mu, sigma, Q=1)       =  dweibull(x, shape=1/sigma, scale=exp(mu))
dgengamma(x, mu, sigma, Q=sigma)   =  dgamma(x, shape=1/sigma^2, rate=exp(-mu) / sigma^2)
```

The properties of the generalized gamma and its applications to survival analysis are discussed in detail by Cox (2007).

The generalized F distribution `GenF` extends the generalized gamma to four parameters.

Value

`dgengamma` gives the density, `pgengamma` gives the distribution function, `qgengamma` gives the quantile function, `rgengamma` generates random deviates, `Hgengamma` retuns the cumulative hazard and `hgengamma` the hazard.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Prentice, R. L. (1974). A log gamma model and its maximum likelihood estimation. Biometrika 61(3):539-544.

Farewell, V. T. and Prentice, R. L. (1977). A study of distributional shape in life testing. Technometrics 19(1):69-75.

Lawless, J. F. (1980). Inference in the generalized gamma and log gamma distributions. Technometrics 22(3):409-419.

Cox, C., Chu, H., Schneider, M. F. and Muñoz, A. (2007). Parametric survival analysis and taxonomy of hazard functions for the generalized gamma distribution. Statistics in Medicine 26:4252-4374

Stacy, E. W. (1962). A generalization of the gamma distribution. Annals of Mathematical Statistics 33:1187-92

See Also

GenGamma.orig, GenF, Lognormal, GammaDist, Weibull.

GenGamma.orig *Generalized gamma distribution (original parameterisation)*

Description

Density, distribution function, hazards, quantile function and random generation for the generalized gamma distribution, using the original parameterisation from Stacy (1962).

Usage

```
dgengamma.orig(x, shape, scale=1, k, log = FALSE)
pgengamma.orig(q, shape, scale=1, k, lower.tail = TRUE, log.p = FALSE)
qgengamma.orig(p, shape, scale=1, k, lower.tail = TRUE, log.p = FALSE)
rgengamma.orig(n, shape, scale=1, k)
Hgengamma.orig(x, shape, scale=1, k)
hgengamma.orig(x, shape, scale=1, k)
```

Arguments

x,q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.
shape	vector of "Weibull" shape parameters.
scale	vector of scale parameters.
k	vector of "Gamma" shape parameters.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.

Details

If $w \sim Gamma(k, 1)$, then $x = \exp(w/shape + \log(scale))$ follows the original generalised gamma distribution with the parameterisation given here (Stacy 1962). Defining shape= $b > 0$, scale= $a > 0$, x has probability density

$$f(x|a, b, k) = \frac{b}{\Gamma(k)} \frac{x^{bk-1}}{a^{bk}} \exp(-(x/a)^b)$$

The original generalized gamma distribution simplifies to the gamma, exponential and Weibull distributions with the following parameterisations:

```
dgengamma.orig(x, shape, scale, k=1)     =  dweibull(x, shape, scale)
 dgengamma.orig(x, shape=1, scale, k)    =  dgamma(x, shape=k, scale)
 dgengamma.orig(x, shape=1, scale, k=1)  =  dexp(x, rate=1/scale)
```

Also as k tends to infinity, it tends to the log normal (as in `dlnorm`) with the following parameters (Lawless, 1980):

```
dlnorm(x, meanlog=log(scale) + log(k)/shape, sdlog=1/(shape*sqrt(k)))
```

For more stable behaviour as the distribution tends to the log-normal, an alternative parameterisation was developed by Prentice (1974). This is given in `dgengamma`, and is now preferred for statistical modelling. It is also more flexible, including a further new class of distributions with negative shape `k`.

The generalized F distribution `GenF.orig`, and its similar alternative parameterisation `GenF`, extend the generalized gamma to four parameters.

Value

`dgengamma.orig` gives the density, `pgengamma.orig` gives the distribution function, `qgengamma.orig` gives the quantile function, `rgengamma.orig` generates random deviates, `Hgengamma.orig` retuns the cumulative hazard and `hgengamma.orig` the hazard.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Stacy, E. W. (1962). A generalization of the gamma distribution. Annals of Mathematical Statistics 33:1187-92.

Prentice, R. L. (1974). A log gamma model and its maximum likelihood estimation. Biometrika 61(3):539-544.

Lawless, J. F. (1980). Inference in the generalized gamma and log gamma distributions. Technometrics 22(3):409-419.

See Also

GenGamma, GenF.orig, GenF, Lognormal, GammaDist, Weibull.

Gompertz *The Gompertz distribution*

Description

Density, distribution function, hazards, quantile function and random generation for the Gompertz distribution with unrestricted shape.

Usage

```
dgompertz(x, shape, rate = 1, log = FALSE)
pgompertz(q, shape, rate = 1, lower.tail = TRUE, log.p = FALSE)
qgompertz(p, shape, rate = 1, lower.tail = TRUE, log.p = FALSE)
rgompertz(n, shape, rate = 1)
hgompertz(x, shape, rate = 1)
Hgompertz(x, shape, rate = 1)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If `length(n) > 1`, the length is taken to be the number required.
shape, rate	vector of shape and rate parameters.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.

Details

The Gompertz distribution with `shape` parameter a and `rate` parameter b has probability density function

$$f(x|a,b) = be^{ax}\exp(-b/a(e^{ax}-1))$$

and hazard

$$h(x|a,b) = be^{ax}$$

The hazard is increasing for shape $a > 0$ and decreasing for $a < 0$. For $a = 0$ the Gompertz is equivalent to the exponential distribution with constant hazard and rate b.

The probability distribution function is

$$F(x|a,b) = 1 - \exp(-b/a(e^{ax}-1))$$

Thus if a is negative, letting x tend to infinity shows that there is a non-zero probability $1-\exp(b/a)$ of living forever. On these occasions `qgompertz` and `rgompertz` will return `Inf`.

Value

dgompertz gives the density, pgompertz gives the distribution function, qgompertz gives the quantile function, hgompertz gives the hazard function, Hgompertz gives the cumulative hazard function, and rgompertz generates random deviates.

Note

Some implementations of the Gompertz restrict a to be strictly positive, which ensures that the probability of survival decreases to zero as x increases to infinity. The more flexible implementation given here is consistent with streg in Stata.

The functions dgompertz and similar available in the package **eha** label the parameters the other way round, so that what is called the shape there is called the rate here, and what is called 1 / scale there is called the shape here. The terminology here is consistent with the exponential dexp and Weibull dweibull distributions in R.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Stata Press (2007) Stata release 10 manual: Survival analysis and epidemiological tables.

See Also

dexp

---

lines.flexsurvreg *Add fitted flexible survival curves to a plot*

Description

Add fitted survival (or hazard or cumulative hazard) curves from a flexsurvreg model fit to an existing plot.

Equivalent to plot.flexsurvreg(...,add=TRUE).

Usage

```
## S3 method for class 'flexsurvreg'
lines(x, X=NULL, type="survival", t=NULL,
        est=TRUE, ci=NULL, B=1000, cl=0.95,
        col="red", lty=1, lwd=2, col.ci=NULL, lty.ci=2, lwd.ci=1, ...)
```

Arguments

x	Output from flexsurvreg, representing a fitted survival model object.
X	Matrix of covariate values to produce fitted curves for, as described in plot.flexsurvreg.
type	"survival" for survival, "cumhaz" for cumulative hazard, or "hazard" for hazard, as in plot.flexsurvreg.
t	Vector of times to plot fitted values for.
est	Plot fitted curves (TRUE or FALSE.)
ci	Plot confidence intervals for fitted curves.
B	Number of simulations controlling accuracy of confidence intervals, as used in summary.
cl	Width of confidence intervals, by default 0.95 for 95% intervals.
col	Colour of the fitted curve(s).
lty	Line type of the fitted curve(s).
lwd	Line width of the fitted curve(s).
col.ci	Colour of the confidence limits, defaulting to the same as for the fitted curve.
lty.ci	Line type of the confidence limits.
lwd.ci	Line width of the confidence limits, defaulting to the same as for the fitted curve.
...	Other arguments to be passed to the generic plot and lines functions.

Author(s)

C. H. Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

See Also

flexsurvreg

---

plot.flexsurvreg *Plots of fitted flexible survival models*

Description

Plot fitted survival, cumulative hazard or hazard from a parametric model against nonparametric estimates to diagnose goodness-of-fit.

Usage

```
## S3 method for class 'flexsurvreg'
plot(x, X=NULL, type="survival", t=NULL, start=NULL,
        est=TRUE, ci=NULL, B=1000, cl=0.95,
        col.obs="black", lty.obs=1, lwd.obs=1,
        col="red", lty=1, lwd=2,
        col.ci=NULL, lty.ci=2, lwd.ci=1,
        add=FALSE,...)
```

Arguments

x	Output from flexsurvreg or flexsurvspline, representing a fitted survival model object.
X	Matrix of covariate values to produce fitted survival curves for. Columns represent different covariates, and rows represent multiple curves. For example matrix(c(1,2),nrow=2) if there is only one covariate in the model, and we want to plot curves for covariate values of 1 and 2. For "factor" (categorical) covariates, the values of the contrasts representing factor levels (as returned by the contrasts function) should be used. For example, for a covariate agegroup specified as an unordered factor with levels 20-29, 30-39, 40-49, 50-59, and baseline level 20-29, there are three contrasts. To plot curves for groups 20-29 and 40-49, supply X = rbind(c(0,0,0), c(0,1,0)), since all contrasts are zero for the baseline level, and the second contrast is "turned on" for the third level 40-49. If there are only factor covariates in the model, then Kaplan-Meier curves are plotted for all distinct groups, and by default, fitted curves are also plotted for these groups. To plot Kaplan-Meier and fitted curves for only a subset of groups, use plot(survfit()) followed by lines.flexsurvreg(). If there are any continuous covariates, then a single population Kaplan-Meier curve is drawn. By default, a single fitted curve is drawn with the covariates set to their mean values in the data - for categorical covariates, the means of the 0/1 indicator variables are taken.
type	"survival" for survival, to be plotted against Kaplan-Meier estimates from plot.survfit. "cumhaz" for cumulative hazard, plotted against transformed Kaplan-Meier estimates from plot.survfit. "hazard" for hazard, to be plotted against smooth nonparametric estimates from muhaz. The nonparametric estimates tend to be unstable, and these plots are intended just to roughly indicate the shape of the hazards through time. The min.time and max.time options to muhaz may sometimes need to be passed as arguments to plot.flexsurvreg to avoid an error here.
t	Vector of times to plot fitted values for, see summary.flexsurvreg.
start	Left-truncation points, see summary.flexsurvreg.
est	Plot fitted curves (TRUE or FALSE.)
ci	Plot confidence intervals for fitted curves. By default, this is TRUE if one observed/fitted curve is plotted, and FALSE if multiple curves are plotted.
B	Number of simulations controlling accuracy of confidence intervals, as used in summary. Decrease for greater speed at the expense of accuracy, or set B=0 to turn off calculation of CIs.
cl	Width of confidence intervals, by default 0.95 for 95% intervals.
col.obs	Colour of the nonparametric curve.
lty.obs	Line type of the nonparametric curve.
lwd.obs	Line width of the nonparametric curve.
col	Colour of the fitted parametric curve(s).

`lty`	Line type of the fitted parametric curve(s).
`lwd`	Line width of the fitted parametric curve(s).
`col.ci`	Colour of the fitted confidence limits, defaulting to the same as for the fitted curve.
`lty.ci`	Line type of the fitted confidence limits.
`lwd.ci`	Line width of the fitted confidence limits.
`add`	If `TRUE`, add lines to an existing plot, otherwise new axes are drawn.
`...`	Other options to be passed to `plot.survfit` or `muhaz`, for example, to control the smoothness of the nonparametric hazard estimates. The `min.time` and `max.time` options to `muhaz` may sometimes need to be changed from the defaults.

Note

Some standard plot arguments such as `"xlim"`,`"xlab"` may not work. This function was designed as a quick check of model fit. Users wanting publication-quality graphs are advised to set up an empty plot with the desired axes first (e.g. with `plot(...,type="n",...)`), then use suitable `lines` functions to add lines.

Author(s)

C. H. Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

See Also

`flexsurvreg`

`qgeneric` *Generic function to find quantiles of a distribution*

Description

Generic function to find the quantiles of a distribution, given the equivalent probability distribution function.

Usage

```
qgeneric(pdist, p, ...)
```

Arguments

`pdist`	Probability distribution function, for example, `pnorm` for the normal distribution, which must be defined in the current workspace. This should accept and return vectorised parameters and values. It should also return the correct values for the entire real line, for example a positive distribution should have `pdist(x)==0` for $x < 0$.
`p`	Vector of probabilities to find the quantiles for.
`...`	The remaining arguments define parameters of the distribution `pdist`. These MUST be named explicitly. This may also contain the standard arguments `log.p` (logical; default `FALSE`, if `TRUE`, probabilities p are given as log(p)), and `lower.tail` (logical; if `TRUE` (default), probabilities are P[X <= x] otherwise, P[X > x].). If the distribution is bounded above or below, then this should contain arguments `lbound` and `ubound` respectively, and these will be returned if `p` is 0 or 1 respectively. Defaults to `-Inf` and `Inf` respectively.

Details

This function is intended to enable users to define `"q"` functions for new distributions, in cases where the distribution function `pdist` is available analytically, but the quantile function is not.

It works by finding the root of the equation $h(q) = pdist(q) - p = 0$. Starting from the interval $(-1, 1)$, the interval width is expanded by 50% until $h()$ is of opposite sign at either end. The root is then found using `uniroot`.

This assumes a suitably smooth, continuous distribution.

An identical function is provided in the **msm** package.

Value

Vector of quantiles of the distribution at `p`.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

Examples

```
qnorm(c(0.025, 0.975), 0, 1)
qgeneric(pnorm, c(0.025, 0.975), mean=0, sd=1) # must name the arguments
```

summary.flexsurvreg *Summaries of fitted flexible survival models*

Description

Return fitted survival, cumulative hazard or hazard at a series of times from a fitted flexsurvreg or flexsurvspline model.

Usage

```
## S3 method for class 'flexsurvreg'
summary(object, X=NULL, type="survival", t=NULL,
         start=NULL, B=1000, cl=0.95,...)
```

Arguments

object	Output from flexsurvreg or flexsurvspline, representing a fitted survival model object.
X	Matrix of covariate values to produce fitted values for. Columns represent different covariates, and rows represent multiple combinations of covariate values. For example matrix(c(1,2),nrow=2) if there is only one covariate in the model, and we want survival for covariate values of 1 and 2. For "factor" (categorical) covariates, the values of the contrasts representing factor levels (as returned by the contrasts function) should be used. For example, for a covariate agegroup specified as an unordered factor with levels 20-29, 30-39, 40-49, 50-59, and baseline level 20-29, there are three contrasts. To return summaries for groups 20-29 and 40-49, supply X = rbind(c(0,0,0), c(0,1,0)), since all contrasts are zero for the baseline level, and the second contrast is "turned on" for the third level 40-49. If there are only factor covariates in the model, then all distinct groups are used by default. If there are any continuous covariates, then a single summary is provided. By default, this is with all covariates set to their mean values in the data - for categorical covariates, the means of the 0/1 indicator variables are taken.
type	"survival" for survival probabilities. "cumhaz" for cumulative hazards. "hazard" for hazards.
t	Times to calculate fitted values for. By default, these are the sorted unique observation (including censoring) times in the data. If the corresponding left-truncation times start are not supplied, then they all default to 0.
start	Left-truncation times, defaults to those corresponding to the default t in the data.
B	Number of simulations from the normal asymptotic distribution of the estimates used to calculate confidence intervals. Decrease for greater speed at the expense of accuracy, or set B=0 to turn off calculation of CIs.
cl	Width of symmetric confidence intervals, relative to 1.
...	Further arguments passed to or from other methods.

Value

A list with one element for each unique covariate value (if there are only categorical covariates) or one element (if there are no covariates or any continuous covariates). Each of these elements is a matrix with one row for each time in `t`, giving the estimated survival (or cumulative hazard, or hazard) and 95% confidence limits. These list elements are named with the covariate names and values which define them.

If there are multiple summaries, an additional list component named `X` contains a matrix with the exact values of contrasts (dummy covariates) defining each summary.

The `plot.flexsurvreg` function can be used to quickly plot these model-based summaries against empirical summaries such as Kaplan-Meier curves, to diagnose model fit.

Confidence intervals for models fitted with `flexsurvreg` are obtained by random sampling from the asymptotic normal distribution of the maximum likelihood estimates (see, e.g. Mandel (2013)). For models fitted with `flexsurvreg`, intervals for the hazard are obtained in this way, whereas intervals for the survival and cumulative hazard are obtained analytically as in Royston and Parmar (2002).

Author(s)

C. H. Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Royston, P. and Parmar, M. (2002). Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 21(1):2175-2197.

Mandel, M. (2013). "Simulation based confidence intervals for functions with complicated derivatives." The American Statistician (in press).

See Also

`flexsurvreg`, `flexsurvspline`.

`Survspline` *Royston/Parmar spline survival distribution*

Description

Probability density and distribution function for the Royston/Parmar spline model.

Usage

```
dsurvspline(x, gamma, beta=0, X=0, knots=c(-10,10), scale="hazard", offset=0)
psurvspline(q, gamma, beta=0, X=0, knots=c(-10,10), scale="hazard", offset=0)
```

Arguments

`x,q`	Vector of times.
`gamma`	Vector of parameters describing the baseline spline function, as described in `flexsurvspline`.
`beta`	Vector of covariate effects.
`X`	Matrix of covariate values.
`knots`	Locations of knots on the axis of log time, supplied in increasing order. Unlike in `flexsurvspline`, these include the two boundary knots. If there are no additional knots, the boundary locations are not used. If there are one or more additional knots, the boundary knots should be at or beyond the minimum and maximum values of the log times. In `flexsurvspline` these are exactly at the minimum and maximum values.
`scale`	`"hazard"`, `"odds"`, or `"normal"`, as described in `flexsurvspline`. With the default of no knots in addition to the boundaries, this model reduces to the Weibull, log-logistic and log-normal respectively.
`offset`	An extra constant to add to the linear predictor η.

Value

`dsurvspline` gives the density and `psurvspline` gives the distribution function, as described in `flexsurvspline`.

Author(s)

Christopher Jackson <chris.jackson@mrc-bsu.cam.ac.uk>

References

Royston, P. and Parmar, M. (2002). Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. Statistics in Medicine 21(1):2175-2197.

See Also

`flexsurvspline`.

Examples

```
## reduces to the weibull
regscale <- 0.786; cf <- 1.82
a <- 1/regscale; b <- exp(cf)
dweibull(1, shape=a, scale=b)
dsurvspline(1, gamma=c(log(1 / b^a), a)) # should be the same

## reduces to the log-normal
meanlog <- 1.52; sdlog <- 1.11
dlnorm(1, meanlog, sdlog)
dsurvspline(1, gamma = c(-meanlog/sdlog, 1/sdlog), scale="normal")
# should be the same
```

Index

∗Topic **aplot**
lines.flexsurvreg, 21
∗Topic **datasets**
bc, 3
∗Topic **distribution**
GenF, 12
GenF.orig, 14
GenGamma, 16
GenGamma.orig, 18
Gompertz, 20
qgeneric, 24
Survspline, 27
∗Topic **hplot**
plot.flexsurvreg, 22
∗Topic **models**
flexsurvreg, 5
flexsurvspline, 9
lines.flexsurvreg, 21
plot.flexsurvreg, 22
summary.flexsurvreg, 26
∗Topic **package**
flexsurv-package, 2
∗Topic **survival**
flexsurvreg, 5
flexsurvspline, 9

aftreg, *3*

bc, 3

coef, *7*
confint, *7*
contrasts, *23*, *26*

dexp, *6*, *19*, *21*
dgamma, *6*, *16*, *19*
dgenf, *6*
dgenf *(GenF)*, 12
dgenf.orig, *6*, *13*
dgenf.orig *(GenF.orig)*, 14
dgengamma, *6*, *13*, *17*, *19*
dgengamma *(GenGamma)*, 16
dgengamma.orig, *6*, *15*, *17*
dgengamma.orig *(GenGamma.orig)*, 18
dgompertz, *6*, *21*
dgompertz *(Gompertz)*, 20
dllogis, *8*, *11*, *14*
dlnorm, *6*, *16*, *19*
dnorm, *8*
dsurvspline *(Survspline)*, 27
dweibull, *6*, *11*, *19*, *21*

fitdistr, *3*
flexsurv, *2*, *3*
flexsurv *(flexsurv-package)*, 2
flexsurv-package, 2
flexsurvreg, *2*, *3*, 5, *6–8*, *10–12*, *21–24*, *26*, *27*
flexsurvspline, *2*, *5*, *8*, 9, *12*, *23*, *26–28*

GammaDist, *18*, *20*
GenF, *2*, 12, *15–20*
GenF.orig, *2*, *14*, 14, *19*, *20*
GenGamma, *2*, *7*, *13*, *14*, *16*, 16, *20*
GenGamma.orig, *2*, *15*, *16*, *18*, 18
Gompertz, *2*, 20

Hgenf *(GenF)*, 12
hgenf *(GenF)*, 12
Hgenf.orig *(GenF.orig)*, 14
hgenf.orig *(GenF.orig)*, 14
Hgengamma *(GenGamma)*, 16
hgengamma *(GenGamma)*, 16
Hgengamma.orig *(GenGamma.orig)*, 18
hgengamma.orig *(GenGamma.orig)*, 18
Hgompertz *(Gompertz)*, 20
hgompertz *(Gompertz)*, 20

lines, *22*, *24*
lines.flexsurvreg, *8*, *12*, 21

Llogis, *14*
Lognormal, *18*, *20*

muhaz, *23, 24*

optim, *7*, *10, 11*

pgenf *(GenF)*, 12
pgenf.orig *(GenF.orig)*, 14
pgengamma *(GenGamma)*, 16
pgengamma.orig *(GenGamma.orig)*, 18
pgompertz *(Gompertz)*, 20
phreg, *3*
pllogis, *8*
plot, *22*
plot.flexsurvreg, *2*, *8*, *12*, *21, 22*, 22, *23*, *27*
plot.survfit, *23, 24*
pnorm, *25*
psurvspline *(Survspline)*, 27

qgeneric, 24
qgenf *(GenF)*, 12
qgenf.orig *(GenF.orig)*, 14
qgengamma *(GenGamma)*, 16
qgengamma.orig *(GenGamma.orig)*, 18
qgompertz *(Gompertz)*, 20
qnorm, *10, 11*

rgenf *(GenF)*, 12
rgenf.orig *(GenF.orig)*, 14
rgengamma *(GenGamma)*, 16
rgengamma.orig *(GenGamma.orig)*, 18
rgompertz *(Gompertz)*, 20

summary, *22, 23*
summary.flexsurvreg, *2*, *23*, 26
Surv, *5*, *9*
survreg, *3*, *6, 7*, *11*
Survspline, 27

uniroot, *25*

vcov, *7*

Weibull, *18*, *20*