flexsurv: a platform for parametric survival modelling in R

Christopher H. Jackson
MRC Biostatistics Unit, Cambridge, UK
chris.jackson@mrc-bsu.cam.ac.uk

Abstract

flexsurv is an R package for fully-parametric modelling of survival data. Any parametric time-to-event distribution may be fitted if the user supplies at minimum a probability density or hazard function. Many standard survival distributions are built in, and also the three and four-parameter generalized gamma and F models. Any parameter of the distribution can be modelled as a linear or log-linear function of covariates. Another built-in model is the spline model of Royston and Parmar, in which both baseline survival and covariate effects can be arbitrarily flexible parametric functions of time.

The main model-fitting function, `flexsurvreg`, uses the familiar syntax of `survreg` from the standard **survival** package — censoring or left-truncation are specified in `Surv` objects. Estimates and confidence intervals for any function of the model parameters can be printed or plotted. **flexsurv** also enhances the **mstate** package (Putter et al) by providing cumulative incidences for fully-parametric multi-state models.

This article explains the methods and design principles of the package, giving several worked examples of its use.

Keywords: survival.

1. Motivation and design

The Cox model for survival data is ubiquitous in medical research, since the effects of predictors can be estimated without needing to supply a baseline survival distribution that might be inaccurate. However, fully-parametric models have many advantages, and even the originator of the Cox model has expressed a preference for parametric modelling (see Reid 1994). Fully-specified models help to understand the change in hazard through time, and help with prediction and extrapolation. For example, the mean survival $E(T) = \int_0^\infty S(t)$, used in health economic evaluations (Latimer 2013), needs the survivor function $S(t)$ to be fully-specified for all times t.

flexsurv allows parametric distributions of arbitrary complexity to be fitted to survival data, gaining the convenience of parametric modelling, while avoiding the risk of model misspecification. Built-in choices include splines with any number of knots (Royston and Parmar 2002) and 3–4 parameter generalized gamma and F distribution families. Any user-defined model may be employed by supplying at minimum an R function to compute the probability density or hazard, and ideally also its cumulative form. Any parameters may be modelled in terms of covariates, and any function of the parameters may be printed or plotted in model

summaries.

flexsurv is intended as a general platform for survival modelling in R. The `survreg` function in the R package **survival** (Therneau 2014) only supports two-parameter (location/scale) distributions, though users can supply their own distributions if they can be parameterised in this form. Many other contributed R packages can fit survival models, e.g. **eha** (Broström 2014), **VGAM** (Yee and Wild 1996), though these are either limited to specific distribution families, not specifically designed for survival analysis, or (**ActuDistns** Nadarajah and Bakar 2013) contain only the definitions of distribution functions. **flexsurv** enables distribution functions provided by such packages to be used as survival models.

It is similar in spirit to the Stata packages **stpm2** (Lambert and Royston 2009) for spline-based survival modelling, and **stgenreg** (Crowther and Lambert 2013) for fitting survival models with user-defined hazard functions using numerical integration. Though in **flexsurv**, numerical integration can be avoided if the analytic cumulative distribution or hazard can be supplied, and optimisation can also be speeded by supplying analytic derivatives. **flexsurv** also has features for multi-state modelling and interval censoring, and general output reporting. It employs functional programming to work with user-defined or existing R functions.

2. General parametric survival model

2.1. Definitions

The general model that **flexsurv** fits has probability density function

$$f(t|\mu(\mathbf{z}), \boldsymbol{\alpha}(\mathbf{z})), \quad t \geq 0 \tag{1}$$

The cumulative distribution function $F(t)$, survivor function $S(t) = 1 - F(t)$, cumulative hazard $H(t) = -\log S(t)$ and hazard $h(t) = f(t)/S(t)$ are also defined (suppressing the conditioning for clarity). $\mu = \alpha_0$ is the parameter of primary interest, which usually governs the mean or location of the distribution. Other parameters $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_R$ are called "ancillary" and determine the shape, variance or higher moments.

Covariates All parameters may depend on a vector of covariates $\mathbf{z}$ through link-transformed linear models $g_0(\mu) = \boldsymbol{\beta}_0'\mathbf{z}$ and $g_r(\alpha_r) = \boldsymbol{\beta}_r'\mathbf{z}$. $g()$ will typically be log() if the parameter is defined to be positive, or the identity function if the parameter is unrestricted. In all models, $\boldsymbol{\beta}$ includes at least an intercept, so that the full set of parameters is given by $\{\boldsymbol{\beta}_r : r = 0, \ldots, R\}$.

Suppose that the location, but not the ancillary parameters, depends on covariates. If the hazard function factorises as $h(t|\alpha, \mu(\mathbf{z})) = \mu(\mathbf{z})h_0(t|\alpha)$, then this is a *proportional hazards* (PH) model, so that the hazard ratio between two groups (defined by different values of $\mathbf{z}$) is constant over time.

Alternatively, if $S(t|\mu(\mathbf{z}), \alpha) = S(\mu(\mathbf{z})t|\alpha)$ then we have an *accelerated failure time* (AFT) model, so that the effect of covariates is to speed or slow the passage of time. For example, doubling the value of a covariate with coefficient $\beta = \log(2)$ would give half the expected survival time.

Data and likelihood Let $t_i : i = 1, \ldots, n$ be a sample of times from individuals i. Let $c_i = 1$ if t_i is an observed death time, or $c_i = 0$ if t_i is a right-censoring time, thus the true death time is known only to be greater than t_i. Also let s_i be corresponding left-truncation (or delayed-entry) times, meaning that individual i is only observed conditionally on having survived up to s_i, thus $s_i = 0$ if there is no left-truncation. Additionally let t_i^{max} be left-censoring times. If there is no left-censoring then these are infinite, so that $S(t_i^{max}) = 0$; or if the ith death time is interval-censored then $c_i = 0$ and t_i^{max} is finite.

The likelihood for the parameters $\boldsymbol{\beta}$ in model (1), given the corresponding data vectors, is

$$l(\{\boldsymbol{\beta}_r\}|\mathbf{t}, \mathbf{c}, \mathbf{s}, \mathbf{t}^{max}) = \left\{ \prod_{i:\ c_i=1} f_i(t_i) \prod_{i:\ c_i=0} (S_i(t_i) - S_i(t_i^{max})) \right\} / \prod_i S_i(s_i) \qquad (2)$$

The individuals are independent, so that **flexsurv** does not currently support frailty, clustered or random effects models.

An example dataset used throughout this paper is from 686 patients with primary node positive breast cancer, available in the package as `bc`. This was originally provided with `stpm` (Royston 2001), and analysed in much more detail by Royston and Parmar (2002) and Sauerbrei and Royston (1999).

3. Model fitting syntax

The main model-fitting function is called `flexsurvreg`. Its first argument is an R *formula* object. The left hand side of the formula gives the response as a survival object, using the `Surv` function from the **survival** package. Here, this indicates that the response variable is `recyrs`, which represents observed death or censoring times when the variable `censrec` is 1 or 0 respectively. The covariate `group` is a factor representing a prognostic score, with three levels `"Good"` (the baseline), `"Medium"` and `"Poor"`. All of these variables are in the data frame `bc`.

```
> library(flexsurv)
> fs1 <- flexsurvreg(Surv(recyrs, censrec) ~ group, data=bc, dist="weibull")
```

If we also had left-truncation times in a variable called `start`, the response would be `Surv(start,recyrs,censrec)`. Or if all responses were interval-censored between lower and upper bounds `tmin` and `tmax`, then we would write `Surv(tmin,tmax,type="interval2")`.

If the argument `dist` is a string, this denotes a built-in survival distribution. In this case we fit a Weibull survival model. Printing the fitted model object gives estimates and confidence intervals for the model parameters and other useful information. Note that these are the *same parameters* as represented by the R distribution function `dweibull`: the `shape` α and the `scale` μ of the survivor function $S(t) = \exp(-(t/\mu)^\alpha)$, and `group` has a linear effect on $\log(\mu)$.

```
> fs1

Call:
flexsurvreg(formula = Surv(recyrs, censrec) ~ group, data = bc,     dist = "weibull")
```

```
Estimates:
             data mean  est      L95%     U95%     se       exp(est)  L95%
shape             NA     1.3797   1.2548   1.5170   0.0668       NA        NA
scale             NA    11.4229   9.1818  14.2110   1.2728       NA        NA
groupMedium   0.3338    -0.6136  -0.8623  -0.3649   0.1269   0.5414    0.4222
groupPoor     0.3324    -1.2122  -1.4583  -0.9661   0.1256   0.2975    0.2326
             U95%
shape             NA
scale             NA
groupMedium   0.6943
groupPoor     0.3806

N = 686,  Events: 299,  Censored: 387
Total time at risk: 2113.425
Log-likelihood = -811.9419, df = 4
AIC = 1631.884
```

The same model can be fitted using `survreg` in **survival**:

```
> survreg(Surv(recyrs, censrec) ~ group, data=bc, dist="weibull")

Call:
survreg(formula = Surv(recyrs, censrec) ~ group, data = bc, dist = "weibull")

Coefficients:
(Intercept) groupMedium   groupPoor
  2.4356168  -0.6135892  -1.2122137

Scale= 0.7248206

Loglik(model)= -811.9   Loglik(intercept only)= -873.2
        Chisq= 122.53 on 2 degrees of freedom, p= 0
n= 686
```

The maximised log-likelihoods are the same, however the parameterisation is different: the first coefficient `(Intercept)` reported by `survreg` is $\log(\mu)$, and `survreg`'s `"scale"` is `dweibull`'s (thus `flexsurvreg`)'s 1 / `shape`. The covariate effects $\boldsymbol{\beta}$, however, have the same "accelerated failure time" interpretation, as linear effects on $\log(\mu)$. The multiplicative effects $\exp(\boldsymbol{\beta})$ are printed in the output as `exp(est)`.

3.1. Built-in survival models

`flexsurvreg`'s currently built-in distributions are listed in Table 1. In each case, the probability density $f()$ and parameters of the fitted model are taken from an existing R function of the same name but beginning with the letter `d`. For the Weibull, exponential (`dexp`), gamma (`dgamma`) and log-normal (`dlnorm`), the density functions are provided with standard installations of R. These density functions, and the corresponding cumulative distribution function

(with the first letter `d` replaced by `p`) are used internally in `flexsurvreg` to compute the likelihood.

flexsurv provides some additional survival distributions, including a Gompertz distribution with unrestricted shape parameter (`dist="gompertz"`), and the three- and four-parameter families described below. For all built-in distributions, **flexsurv** also defines functions beginning `h` giving the hazard, and `H` for cumulative hazard.

Generalized gamma This three-parameter distribution includes the Weibull, gamma and log-normal as special cases. The original parameterisation from Stacy (1962) is available as `dist="gengamma.orig"`, however the newer parameterisation (Prentice 1974) is preferred: `dist="gengamma"`. This has parameters (μ,σ,q), and survivor function

$$\begin{array}{ll} 1 - I(\gamma, u) & (q > 0) \\ 1 - \Phi(z) & (q = 0) \end{array}$$

where $I(a, x) = \int_0^x x^{a-1} \exp(-x)/\Gamma(a)$ is the incomplete gamma function (the cumulative gamma distribution with shape a and scale 1), Φ is the standard normal cumulative distribution, $u = \gamma \exp(|q|z)$, $z = (\log(t) - \mu)/\sigma$, and $\gamma = q^{-2}$. The Prentice (1974) parameterisation extends the original one to include a further class of models with negative q, and survivor function $I(\gamma, u)$, where z is replaced by $-z$. This stabilises estimation when the distribution is close to log-normal, since $q = 0$ is no longer near the boundary of the parameter space. In R notation, [1] the parameter values corresponding to the three special cases are

```
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 generalized gamma model is fitted to the breast cancer survival data. `fs2` is an AFT model, where only the parameter μ depends on the prognostic covariate `group`. In a second model `fs3`, the first ancillary parameter `sigma` (α_1) also depends on this covariate, giving a model with a time-dependent effect that is neither PH nor AFT. The second ancillary parameter `Q` is still common between prognostic groups.

```
> fs2 <- flexsurvreg(Surv(recyrs, censrec) ~ group, data=bc, dist="gengamma")
> fs3 <- flexsurvreg(Surv(recyrs, censrec) ~ group + sigma(group),
+                    data=bc, dist="gengamma")
```

Table 3 compares all the models fitted to the breast cancer data, showing absolute fit to the data as measured by the maximised -2× log likelihood $-2LL$, number of parameters p, and Akaike's information criterion $-2LL + 2p$ which estimates predictive ability.

Generalized F This four-parameter distribution includes the generalized gamma, and also the log-logistic, as special cases. The variety of hazard shapes that can be represented is discussed by Cox (2008). It is provided here in alternative "original" (`dist="genf.orig"`) and "stable" parameterisations (`dist="genf"`) as presented by Prentice (1975). See `help(GenF)` and `help(GenF.orig)` in the package documentation for the exact definitions.

[1] The parameter called q here and in previous literature is called Q in `dgengamma` and related functions, since the first argument of a cumulative distribution function is conventionally named `q`, for quantile, in R.

	Parameters	Density R function	dist
Exponential	rate	dexp	"exp"
Weibull	shape, scale	dweibull	"weibull"
Gamma	shape, rate	dgamma	"gamma"
Log-normal	meanlog, sdlog	dlnorm	"lnorm"
Gompertz	shape, rate	dgompertz	"gompertz"
Generalized gamma (Prentice 1975)	mu, sigma, Q	dgengamma	"gengamma"
Generalized gamma (Stacy 1962)	shape, scale, k	dgengamma.orig	"gengamma.orig"
Generalized F (stable)	mu, sigma, Q, P	dgenf	"genf"
Generalized F (original)	mu, sigma, s1, s2	dgenf.orig	"genf.orig"

Table 1: Built-in parametric survival distributions in **flexsurv**

3.2. Plotting outputs

The `plot()` method for `flexsurvreg` objects is used as a quick check of model fit. By default, this draws a Kaplan-Meier estimate of the survivor function $S(t)$, one for each combination of categorical covariates, or just a single "population average" curve if there are no categorical covariates. The corresponding estimates from the fitted model are overlaid. Fitted values from further models can be added with the `lines()` method.

`scale="hazard"` can be used to plot hazards from parametric models against kernel density estimates (obtained from **muhaz**, original by Kenneth Hess and port by R. Gentleman 2010; Mueller and Wang 1994). This shows more clearly why the Weibull model is inadequate: the hazard must be increasing or decreasing — while the generalized gamma can represent the increase and subsequent decline in hazard seen in the data.

Similarly, `scale="cumhaz"` plots cumulative hazards. Confidence intervals are produced by simulating a large sample from the asymptotic normal distribution of the maximum likelihood estimates of $\{\boldsymbol{\beta}_r : r = 0, \ldots, R$, via the function `normboot.flexsurvreg`.

In this example, there is only a single categorical covariate, and the `plot` and `summary` methods return one observed and fitted trajectory for each level of that covariate. For more complicated models, users should specify exactly what covariate values they want summaries for, rather than relying on the default [2]. This is done by supplying the `newdata` argument, a data frame or list containing covariate values, just as in standard R functions like `predict.lm`.

For more than casual plots, it is advised to set up the axes beforehand, and use the `lines()` method. Or for even more flexibility, the data underlying the plots is available from the `summary.flexsurvreg()` method.

3.3. Custom model summaries

Any function of the parameters of a fitted model can be summarised or plotted by supplying the argument `fn` to `summary.flexsurvreg` or `plot.flexsurvreg`. This should be an R function, with mandatory first two arguments `t` representing time, and `start` representing a left-truncation point (so that the result is conditional on survival up to that time). The

[2] If there are only factor covariates, all combinations are plotted. If there are any continuous covariates, these methods by default return a "population average" curve, with the linear model design matrix set to its average values, including the 0/1 contrasts defining factors, which doesn't represent a meaningful covariate combination.

```
> plot(fs1, col="gray", lwd.obs=2)
> lines(fs2, col="red", lty=2)
> lines(fs3, col="red")
```

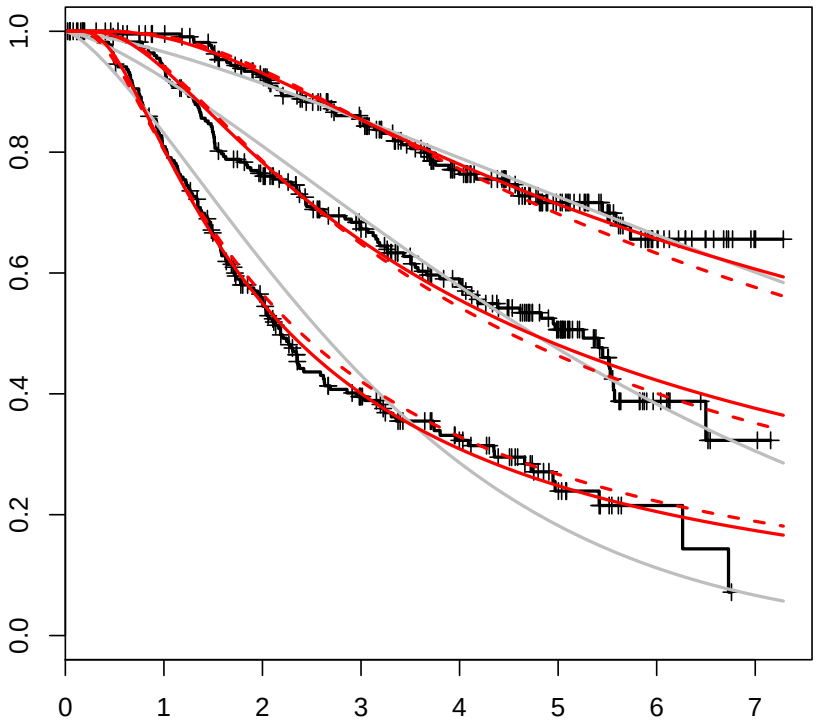

Figure 1: Estimated survival from parametric models and Kaplan-Meier estimates.

remaining arguments must be the parameters of the survival distribution. For example, median survival under the Weibull model `fs1` can be summarised as follows

```
> median.weibull <- function(t, start, shape, scale) {
+     qweibull(0.5, shape=shape, scale=scale)
+ }
> summary(fs1, fn=median.weibull, t=1, B=10000)

group=Good
  time     est      lcl      ucl
1    1 8.75794 7.106009 10.78504

group=Medium
  time      est      lcl      ucl
```

```
> plot(fs1, type="hazard", col="gray", lwd.obs=2)
> lines(fs2, type="hazard", col="red", lty=2)
> lines(fs3, type="hazard", col="red")
```

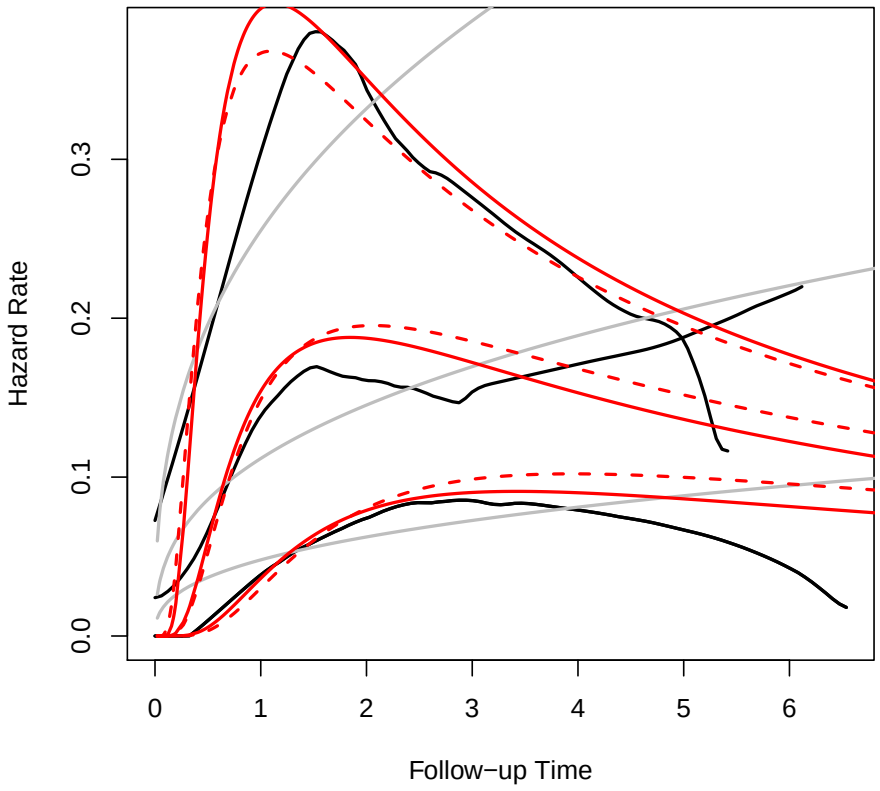

Figure 2: Estimated hazards from parametric models and kernel density estimates.

```
1     1 4.741585 4.114637 5.456919

group=Poor
  time      est      lcl      ucl
1    1 2.605819 2.308561 2.931716
```

Although the median of the Weibull has an analytic form as $\mu \log(2)^{1/\alpha}$, the form of the code given here generalises to other distributions. The argument `t` is not used in `median.weibull`, because the median is a time-constant function of the parameters, unlike the survival or hazard. `10000` random samples are drawn to produce a slightly more precise confidence interval than the default — users should adjust this until the desired level of precision is obtained. A useful future extension of the package would be to allow users to supply derivatives of their custom summary function, so that the delta method can be used to obtain approximate confidence intervals without simulation.

3.4. Computation

The likelihood is maximised in `flexsurvreg` using the optimisation methods available through the standard R `optim` function. By default, this is the `"BFGS"` method ((Nash 1990)) using the analytic derivatives of the likelihood with respect to the model parameters, if these are available, to improve the speed of convergence to the maximum. These are built-in for the exponential, Weibull and Gompertz. For custom distributions, the user can optionally supply functions with names beginning `"DLd"` and `"DLS"` respectively (e.g. `DLdweibull,DLSweibull`) to calculate the derivatives of the log density and log survivor functions with respect to the transformed parameters γ.

Initial values are difficult: ideally two would come from moments of the distribution, then defaults that reduce to simpler distributions. example

3.5. Custom survival distributions

flexsurv is not limited to its built-in distributions. Any survival model of the form (1–2) can be fitted if we can provide either the density function $f()$ or the hazard $h()$. Many contributed R packages provide probability density and cumulative distribution functions for positive distributions. Though survival models may be more naturally characterised by their hazard function, representing the changing risk of death through time. For example, for survival following major surgery we may want a "U-shaped" hazard curve, representing a high risk soon after the operation, which then decreases, but increases naturally as survivors grow older.

To supply a custom distribution, the `dist` argument to `flexsurvreg` is defined to be an R list object, rather than a character string. The list has the following elements.

`name` Name of the distribution. For example, if this is `"llogis"` then there is assumed to be at least either

- a function called `dllogis` to compute the probability density, or
- `hllogis` to compute the hazard.

Ideally there will also be a function called `pllogis` for the cumulative distribution (if `d` is given), or `H` for the cumulative hazard (to complement `h`).

These functions must be *vectorised*, and the density function must also accept an argument `log`, which when `TRUE`, returns the log density. See the examples below.

`pars` Character vector naming the parameters of the distribution $\mu, \alpha_1, ..., \alpha_R$. These must match the arguments of the R distribution function or functions.

`location` Character: quoted name of the location parameter μ. The location parameter will not necessarily be the first one, e.g. in `dweibull` the `scale` comes after the `shape`.

`transforms` A list of functions $g()$ which transform the parameter from its natural range to the real line, for example, `c(log,identity)` [3]

[3] This is a *list*, not an *atomic vector* of functions, so if the distribution only has one parameter, we should write `transforms=c(log)` or `transforms=list(log)`, not `transforms=log`

`inv.transforms` List of corresponding inverse functions.

`inits` A function which provides plausible initial values of the parameters for maximum likelihood estimation. This is optional, but if not provided, then each call to `flexsurvreg` must have an `inits` argument containing a vector of initial values, which is inconvenient. Implausible initial values may produce a likelihood of zero, and a fatal error message (`initial value in 'vmmin' is not finite`) from the optimiser.

Each distribution will ideally have a heuristic for initialising parameters from summaries of the data. For example, since the median of the Weibull is $\mu \log(2)^{1/\alpha}$, a sensible estimate of μ will commonly be the median log uncensored survival time divided by $\log(2)$, with $\alpha = 1$, assuming that in practice the true value of α is not often far from 1. Then we define the function, of one argument `t` assumed to be the uncensored survival times, returning the initial values for the Weibull `shape` and `scale` respectively.

```
inits = function(t) c(1, median(t[t>0]) / log(2))
```

More complicated initial value functions may use other data such as the covariate values and censored observations: for an example, see the function `flexsurv.splineinits` in the package source that computes initial values for spline models (§4.1).

Example: Using functions from a contributed package The following custom model uses the log-logistic distribution functions (`dllogis` and `pllogis`) available in the package **eha**. The survivor function is $S(t|\mu, \alpha) = 1/(1 + (t/\mu)^\alpha)$, so that the odds $(1 - S(t))/S(t)$ of having died are a linear function of log time.

```
> library(eha)
> 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)) })
> fs4 <- flexsurvreg(Surv(recyrs, censrec) ~ group, data=bc, dist=custom.llogis)
```

This fits the breast cancer data better than the Weibull, since it can represent a peaked hazard, but less well than the generalized gamma (Table 3).

Example: Wrapping functions from a contributed package Sometimes there may be probability density and similar functions in a contributed package, but in a different format. For example, **eha** also provides a three-parameter Gompertz-Makeham distribution with hazard $h(t|\mu, \alpha_1, \alpha_2) = \alpha_2 + \alpha_1 \exp(t/\mu)$. The shape parameters α_1, α_2 are provided to `dmakeham` as a vector argument of length two. However, `flexsurvreg` expects distribution functions to have one argument for each parameter. Therefore we write our own functions that wrap around the third-party functions.

```
> dmakeham3 <- function(x, shape1, shape2, scale, ...)  {
+     dmakeham(x, shape=c(shape1, shape2), scale=scale, ...)
+ }
> pmakeham3 <- function(q, shape1, shape2, scale, ...)  {
+     pmakeham(q, shape=c(shape1, shape2), scale=scale, ...)
+ }
```

`flexsurvreg` also requires these functions to be *vectorized*, as the standard distribution functions in R are. That is, we can supply a vector of alternative values for one or more arguments, and expect a vector of the same length to be returned. The R base function `Vectorize` can be used to do this here.

```
> dmakeham3 <- Vectorize(dmakeham3)
> pmakeham3 <- Vectorize(pmakeham3)
```

and this allows us to write, for example,

```
> pmakeham3(c(0, 1, 1, Inf), 1, c(1, 1, 2, 1), 1)

[1] 0.0000000 0.9340120 0.9757244 1.0000000
```

We could then use `dist=list(name="makeham3", pars=c("shape1","shape2","scale"),...)` in a `flexsurvreg` model, though in the breast cancer example, the second shape parameter is poorly identifiable.

Example: Changing the parameterisation of a distribution We may want to fit a Weibull model like `fs1`, but parameterised as $S(t) = \exp(-\mu t^{\alpha})$, so that the covariate effects reported in the printed `flexsurvreg` object can be interpreted as hazard ratios or log hazard ratios without any further transformation. Here instead of the density and cumulative distribution functions, we provide the hazard and cumulative hazard.[4]

```
> detach("package:eha")
> hweibullPH <- function(x, shape, scale = 1, log=FALSE){
+     hweibull(x, shape=shape, scale=scale^{-1/shape}, log=log)
+ }
> HweibullPH <- function(x, shape, scale=1, log=FALSE){
+     Hweibull(x, shape=shape, scale=scale^{-1/shape}, log=log)
+ }
> custom.weibullPH <- list(name="weibullPH",
+                          pars=c("shape","scale"), location="scale",
+                          transforms=c(log, log), inv.transforms=c(exp, exp),
+                          inits = function(t){
+                              c(1, median(t[t>0]) / log(2))
+                          })
> fs6 <- flexsurvreg(Surv(recyrs, censrec) ~ group, data=bc, dist=custom.weibullPH)
> 1 / fs1$res["scale","est"]^fs1$res["shape","est"]

[1] 0.03472474

> 1 / exp(fs1$res["groupMedium","est"]) ^ fs1$res["shape","est"]

[1] 2.331564
```

[4]The **eha** package needs to be detached first so that **flexsurv**'s built-in `hweibull` is used, which returns `NaN` if the parameter values are zero, rather than failing as **eha**'s does.

The fitted model is the same as `fs1`, therefore the maximised likelihood is the same, and the parameter estimates of `fs1` can be transformed to those of `fs6` as shown.

A slightly more complicated example is given in the examples vignette of constructing a proportional hazards generalized gamma model.

Example: Omitting the cumulative distribution or hazard If there is no analytic form for $F(t)$ or $H(t)$ as the integral of the density or hazard respectively, then **flexsurv** can compute these internally by numerical integration, as in **stgenreg** (Crowther and Lambert 2013). The default options of the built-in R routine `integrate` for adaptive quadrature are used, though these may be changed using the `integ.opts` argument to `flexsurvreg`. Models specified this way will take much longer to fit, by an order of magnitude.

EXAMPLE IN SECTION 4.2

4. Any-dimension models

flexsurv also supports models where the number of parameters is arbitrary. In the models discussed previously, the number of parameters in the model family is fixed (e.g. three for the generalized gamma). In this section, the model complexity can be chosen by the user, given the model family. We may want to represent more irregular hazard curves by more flexible functions, or use bigger models if a bigger sample size makes it feasible to estimate more parameters.

4.1. Royston and Parmar spline model

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. This is available here as the function `flexsurvspline`, and is also available in the Stata package `stpm2` (Lambert and Royston 2009) (and historically `stpm`, Royston (2001, 2004)).

$$g(S(t, z)) = s(x, \boldsymbol{\gamma})$$

Typically we use $g(S(t, \mathbf{z}) = \log(-\log(S(t, \mathbf{z}))) = \log(H(t, \mathbf{z}))$, the log cumulative hazard, giving a proportional hazards model.

Spline parameterisation The complexity of the model, thus the dimension of $\boldsymbol{\gamma}$, is governed by the number of *knots* m in the spline function $s()$. Natural cubic splines are piecewise cubic polynomials defined to be continuous, with continuous first and second derivatives at the knots, and also constrained to be linear beyond boundary knots k_{min}, k_{max}. As well as the boundary knots there may be up to $m \geq 0$ *internal* knots $k_1, \ldots k_m$. Various spline parameterisations exist — the one used here is from Royston and Parmar (2002).

$$s(x, \boldsymbol{\gamma}) = \gamma_0 + \gamma_1 x + \gamma_2 v_1(x) + \ldots + \gamma_{m+1} v_m(x) \tag{3}$$

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

Model	$g(S(t,\mathbf{z}))$	In `flexsurvspline`	With $m=0$
Proportional hazards	$\log(-\log(S(t,\mathbf{z})))$ (log cumulative hazard)	`scale="hazard"`	Weibull
`pweibull(t, shape=a, scale=b) == psurvspline(t, gamma=c(log(1 / b`^{a}`), a))`			
Proportional odds	$\log(S(t,\mathbf{z})^{-1}-1)$ (log cumulative odds)	`scale="odds"`	Log-logistic
`eha::pllogis(t, shape=a, scale=b) == psurvspline(t, gamma=c(-a*log(b), a), scale="odds")`			
Normal / probit	$\Phi^{-1}(S(t,\mathbf{z}))$ (inverse normal CDF, `qnorm`)	`scale="normal"`	Log-normal
`plnorm(t, meanlog=a, sdlog=b) == psurvspline(t, gamma=c(-a/b, 1/b), scale="normal")`			

Table 2: Alternative modelling scales for `flexsurvspline`, and equivalent distribution families and parameter values for $m=0$ explained in R notation.

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

and $(x-a)_+ = max(0, x-a)$. If $m=0$ then there are only two parameters γ_0, γ_1. In fact if $g()$ is the log cumulative hazard, this is equivalent to a Weibull model. Table 2 explains two further choices of $g()$, and the parameter values and distributions they simplify to for $m=0$. The probability density and cumulative distribution functions for this model are available as `dsurvspline` and `psurvspline`

Covariates on spline parameters Covariates can be placed on any parameter γ through a linear model (with identity link function). Most straightforwardly, we can let the intercept γ_0 vary with covariates $\mathbf{z}$, giving a proportional hazards or odds model (depending on $g()$).

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

The spline coefficients $\gamma_j : j = 1, 2\ldots$, the "ancillary parameters", may also be modelled as linear functions of covariates $\mathbf{z}$, as

$$\gamma_j(\mathbf{z}) = \gamma_{j0} + \gamma_{j1}z_1 + \gamma_{j2}z_2 + \ldots$$

giving a model where the effects of covariates are arbitrarily flexible functions of time: a non-proportional hazards or odds model.

Spline models in flexsurv The package provides the function `flexsurvspline` to fit this general model. Internal knots are chosen by default from quantiles of the log uncensored death times, however users can supply their own knot locations in the `knots` argument to `flexsurvspline`. Initial values for numerical likelihood maximisation are chosen using the

method described by Royston and Parmar (2002) of Cox regression combined with transforming an empirical survival estimate.

For example, the best-fitting model for the breast cancer dataset identified in Royston and Parmar (2002), a proportional odds model with one internal spline knot, is

```
> sp1 <- flexsurvspline(Surv(recyrs, censrec) ~ group, data=bc, k=1,
+                       scale="odds")
```

A further model where the first ancillary parameter also depends on the prognostic group, giving a time-varying odds ratio, is fitted as

```
> sp2 <- flexsurvspline(Surv(recyrs, censrec) ~ group + gamma1(group),
+                       data=bc, k=1, scale="odds")
```

These models give qualitatively similar results to the generalized gamma in this dataset (Figure **??**), and have similar predictive ability as measured by AIC (Table 3). Though in general, an advantage of spline models is that extra flexibility is available where necessary.

4.2. Implementing general-dimension models

The spline model above is an example of the general parametric form (1), but the number of parameters ($R + 1$ in (1), $m + 2$ in (3)) is arbitrary. **flexsurv** has the tools to deal with any model of this form. `flexsurvspline` works internally by building a custom distribution and then calling `flexsurvreg`. Similar models may in principle be built by users using the same method. This relies on a functional programming trick.

Creating distribution functions dynamically The R distribution functions supplied to custom models are expected to have a fixed number of arguments, including one for each scalar parameter. However, the distribution functions for the spline model (e.g. `dsurvspline`) have an argument `gamma` representing the vector of parameters γ, whose length is determined by the user through the choice of the number of knots. Just as the *scalar parameters* of conventional distribution functions can be supplied as *vector arguments* (as explained in §3.5), similarly, the vector parameters of spline-like distribution functions can be supplied as *matrix arguments*, representing alternative parameter values.

To convert a spline-like distribution function into the correct form, **flexsurv** provides the utility `unroll.function`. This converts a function with one (or more) vector parameters (matrix arguments) to a function with an arbitrary number of scalar parameters (vector arguments). For example, the 5-year survival probability for the baseline group under the model `sp1` is

```
> gamma <- sp1$res[c("gamma0","gamma1","gamma2"),"est"]
> 1 - psurvspline(5, gamma=gamma, knots=sp1$knots)

[1] 0.6896969
```

An alternative function to compute this can be built by `unroll.function`. We tell it that the vector parameter `gamma` should be provided instead as three scalar parameters named `gamma0`,`gamma1`,`gamma2`. The resulting function `pfn` is in the correct form for a custom `flexsurvreg` distribution.

```
> pfn <- unroll.function(psurvspline, gamma=0:2)
> 1 - pfn(5, gamma0=gamma[1], gamma1=gamma[2], gamma2=gamma[3], knots=sp1$knots)

[1] 0.6896969
```

Users wishing to fit a new spline-like model with a known number of parameters could just as easily write distribution functions specific to that number of parameters, and use the methods in §3.5. However the `unroll.function` method is intended to simplify the process of extending the **flexsurv** package to new classes of models. The intention is that wrappers similar to `flexsurvspline` could be included in the package in the future for useful classes of models.

Example: splines on alternative scales stgenreg has demo of spline modelling on the log hazard scale. Can we do this using a generic distribution? (advantage: when there are multiple time dependent effects, the interpretation of the time-dependent hazard ratios is simplified as they do not depend on values of other covariates, which is the case when modelling on the cumulative hazard scale (Royston and Lambert 2011).

Other arbitrary-dimension models A potential application is to fractional polynomials (**?**). These are of the form $\sum_{m=1}^{M} \alpha_m x^{p_m} log(x)^n$ where the power p_m is in the standard set $\{2, -1, -0.5, 0, 0.5, 1, 2, 3\}$ (except that $log(x)$ is used instead of x^0), and n is a non-negative integer. They are similar to splines in that they can give arbitrarily close approximations to a nonlinear function, such as a hazard curve, and are particularly useful for modelling the effects of continuous predictors in regression models. See e.g. **?**, and several other publications by the same authors, for applications and discussion of their advantages over splines. The R package **gamlss** CITE has a function to construct a fractional polynomial basis that may be employed in **flexsurv** models.

Polyhazard models (**?**) are another potential use of this technique. These express an overall hazard as a sum of latent cause-specific hazards, each one typically from the same class of distribution, e.g. a *poly-Weibull* model if they are all Weibull. For example, a U-shaped hazard curve following surgery may be the sum of early hazards from surgical mortality and later deaths from natural causes. However, such models may not always be identifiable without external information to fix or constrain the parameters of particular hazards (**?**).

```
> res <- t(sapply(list(fs1, fs2, fs3, fs4, sp1, sp2),
+                 function(x)rbind(-2*x$loglik, x$npars, x$AIC)))
> rownames(res) <- c("Weibull","Generalized gamma","Generalized gamma (time-varying effect
+                    "Spline", "Spline (time-varying effects)")
> colnames(res) <- c("-2 log likelihood","Parameters","AIC")
```

5. Multi-state models

A *multi-state model* represents how an individual moves between multiple states through time. Survival analysis is a special case with two states, "alive" and "dead". *Competing risks*

```
> res
```

	-2 log likelihood	Parameters	AIC
Weibull	1623.884	4	1631.884
Generalized gamma	1575.137	5	1585.137
Generalized gamma (time-varying effects)	1572.434	7	1586.434
Log-logistic	1598.105	4	1606.105
Spline	1577.964	5	1587.964
Spline (time-varying effects)	1574.848	7	1588.848

Table 3: Summary of all models fitted to the breast cancer data

are a further special case where there are multiple causes of death, that is, multiple possible destination states for the same starting state.

Instead of a single event time for each individual, we may now have a series of event times $t_1, \ldots, t_n$. The last of these may be an observed event or censoring. Given that an individual is in state $S(t)$ at time t, the next state to which they move, and the time of the change, are governed by a set of *transition intensities* $q_{rs}(t, \mathbf{z}(t), \mathcal{H}_t) = \lim_{\delta t \to 0} P(S(t + \delta t) = s | S(t) = r)/\delta t$ for states $r, s = 1, \ldots, R$, which for a survival model are equivalent to the hazard $h(t)$. The intensity represents the instantaneous risk of moving from state r to state s. It may depend on the time t since the start of the process, patient characteristics $\mathbf{z}(t)$, and possibly also the "history" of the process up to that time, $\mathcal{H}_t$, the states previously visited or the length of time spent in them.

Alternative time scales In semi-Markov (clock-reset) models, $q_{rs}(t)$ depends on the time t since entry into the current state, but otherwise, time since the beginning of the process is forgotten. Any software to fit survival models can also fit this kind of model.

In an inhomogeneous Markov (clock-forward) model, $q_{rs}(t)$ is depends on the time t since the beginning of the process, but not otherwise on $\mathcal{H}_t$. The transition intensity out of state r. Again any survival modelling software can be used, with the additional requirement that it can deal with left-truncation or *counting process* data.

Implementing multi-state models as multiple survival models **?** discuss how to implement multi-state models using standard survival modelling software. For each permitted $r \to s$ transition in the multi-state model (ILLUSTRATION) there is a corresponding survival model, with hazard rates defined by $q_{rs}(t)$.

For a patient who moves into state s at time t_j, their next event at t_{j+1} is defined by the model structure (FIGURE) to be one of a set of competing events $s^*_1, \ldots, s^*_{n_s}$. Therefore there are N survival models, where N is the number of distinct permitted transitions from states r to a different state s in the multi-state model structure.

At time t_j, a series of n_s observations are constructed. Each has an indicator for whether the transition to the corresponding state $s^*_1, \ldots, s^*_{n_s}$ is observed or censored at t_{j+1}, coupled with:

- (for a semi-Markov model) the time elapsed $dt_j = t_{j+1} - t_j$ from state r entry to state s entry. This data informs the "survival" model for the $r \to s$ transition.

- (for an inhomogeneous Markov model) the start and stop time (t_j, t_{j+1}). The $r \to s$ model is fitted to the right-censored time t_{j+1} from the *start of the process*, but is conditional on not experiencing the $r \to$ transition until after the state r entry time. In other words, the $r \to s$ transition model is *left-truncated* at the state r entry time.

FLEXSURVREG EXAMPLE

mstate The **mstate** R package (**??**) has a utility `msprep` to produce data of this form from "wide-format" datasets where rows represent individuals, and times of different events appear in different columns, and **mstate** has a utility `msm2Surv` for

illustrates Cox models flexible parametric multi-state models

Prediction from multi-state models Define cumulative incidence functions

The `mstate` package is designed to work with piecewise-constant cumulative incidence functions baseline hazards are estimated non-parametrically (**??**)

function `msfit` that produces the cumulative incidences for each transition and a given covariate category, and their covariances, given a Cox model fitted using `coxph` from the **survival** package.

Aalen-Johansen estimator, simulation

contrast Markov and semi-Markov models

Multi-state models for panel data Note the contrast with multi-state models for *panel data*, that is, observations of the state of the process at a series of times (**?**). In panel data, we do not necessarily know know the time of each transition, or even whether transitions of a certain type have occurred at all between a pair of observations. Multi-state models for this type of data (and also for the exact event time data discussed above) can be fitted with the **msm** package for R, but are restricted to (piecewise) exponentially-distributed event times.

`survSplit` function in **survival**

6. Potential extensions

relative survival frailty many extensions may come from user-contributed models

A. Acknowledgements

Thanks to Milan Bouchet-Valat.

References

Broström G (2014). *eha: Event History Analysis*. R package version 2.4-1, URL http://CRAN.R-project.org/package=eha.

Cox C (2008). "The generalized F distribution: An umbrella for parametric survival analysis." **27**, 4301–4312.

Crowther MJ, Lambert PC (2013). "stgenreg: A Stata package for general parametric survival analysis." *Journal of Statistical Software*, **53**, 1–17.

de Wreede L, Fiocco M, Putter H (2010). "The `mstate` package for estimation and prediction in non-and semi-parametric multi-state and competing risks models." *Computer Methods and Programs in Biomedicine*, **99**(3), 261–274.

de Wreede LC, Fiocco M, Putter H (2011). "`mstate`: an R package for the analysis of competing risks and multi-state models." *J Stat Softw*, **38**, 1–30.

Demiris N, Lunn D, Sharples L (2011). "Survival extrapolation using the poly-Weibull model." *Statistical Methods in Medical Research*.

Kalbfleisch J, Lawless J (1985). "The Analysis of Panel Data under a Markov Assumption." *Journal of the American Statistical Association*, **80**(392), 863–871.

Lambert PC, Royston P (2009). "Further development of flexible parametric models for survival analysis." *Stata Journal*, **9**(2), 265.

Latimer NR (2013). "Survival analysis for economic evaluations alongside clinical trialsâĂŤextrapolation with patient-level data inconsistencies, limitations, and a practical guide." *Medical Decision Making*, **33**(6), 743–754.

Louzada-Neto F (1999). "Polyhazard models for lifetime data." *Biometrics*, **55**, 1281–1285.

Mueller HG, Wang JL (1994). "Hazard rates estimation under random censoring with varying kernels and bandwidths." *Biometrics*, **50**, 61–76.

Nadarajah S, Bakar S (2013). "A new R package for actuarial survival models." *Computational Statistics*, **28**(5), 2139–2160.

Nash JC (1990). *Compact numerical methods for computers: linear algebra and function minimisation*. CRC Press.

original by Kenneth Hess S, port by R Gentleman R (2010). *muhaz: Hazard Function Estimation in Survival Analysis*. R package version 1.2.5, URL http://CRAN.R-project.org/package=muhaz.

Prentice RL (1974). "A log gamma model and its maximum likelihood estimation." *Biometrika*, **61**(3), 539–544.

Prentice RL (1975). "Discrimination among some parametric models." *Biometrika*, **62**(3), 607–614.

Putter H, Fiocco M, Geskus RB (2007). "Tutorial in biostatistics: competing risks and multi-state models." **26**, 2389–2430.

Reid N (1994). "A conversation with Sir David Cox." *Statistical Science*, pp. 439–455.

Royston P (2001). "Flexible parametric alternatives to the Cox model, and more." *Stata Journal*, **1**(1), 1–28.

Royston P (2004). "Flexible parametric alternatives to the Cox model: update." *The Stata Journal*, **4**(1), 98–101.

Royston P, Altman DG (1994). "Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling." *Applied Statistics*, pp. 429–467.

Royston P, 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." **21**(1), 2175–2197.

Sauerbrei W, Royston P (1999). "Building multivariable prognostic and diagnostic models: transformation of the predictors by using fractional polynomials." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **162**(1), 71–94.

Sauerbrei W, Royston P, Binder H (2007). "Selection of important variables and determination of functional form for continuous predictors in multivariable model building." *Statistics in medicine*, **26**(30), 5512–5528.

Stacy EW (1962). "A generalization of the gamma distribution." *Annals of Mathematical Statistics*, (33), 1187–92.

Therneau T (2014). "A Package for Survival Analysis in S." R package version 2.37-7. http://CRAN.R-project.org/package=survival.

Yee TW, Wild C (1996). "Vector generalized additive models." *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 481–493.