flexsurv: flexible parametric survival modelling in R. Supplementary examples

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

Abstract

This vignette of examples supplements the main **flexsurv** user guide.

Keywords: survival.

1. Examples of custom distributions

1.1. Proportional hazards generalized gamma model

? discuss using the **stgenreg** Stata package to construct a proportional hazards parameterisation of the three-parameter generalised gamma distribution. A similar trick can be used in **flexsurv**. A four-parameter custom distribution is created by defining its hazard (and cumulative hazard) functions. These are obtained by multiplying the built-in functions `hgengamma` and `Hgengamma` by an extra dummy parameter, which is used as the location parameter of the new distribution. The intercept of this parameter is fixed at 1 when calling `flexsurvreg`, so that the new model is no more complex than the generalized gamma AFT model `fs3`, but covariate effects on the dummy parameter are now interpreted as hazard ratios.

```
> library(flexsurv)
> hgengammaPH <- function(x, dummy, mu=0, sigma=1, Q){
+     dummy * hgengamma(x=x, mu=mu, sigma=sigma, Q=Q)
+ }
> HgengammaPH <- function(x, dummy, mu=0, sigma=1, Q){
+     dummy * Hgengamma(x=x, mu=mu, sigma=sigma, Q=Q)
+ }
> custom.gengammaPH <- list(name="gengammaPH",
+                            pars=c("dummy","mu","sigma","Q"), location="dummy",
+                            transforms=c(log, identity, log, identity),
+                            inv.transforms=c(exp, identity, exp, identity),
+                            inits=function(t){
+                                lt <- log(t[t>0])
+                                c(1, mean(lt), sd(lt), 0)
+                            })
```

```
> fs7 <- flexsurvreg(Surv(recyrs, censrec) ~ group, data=bc,
+                    dist=custom.gengammaPH, fixedpars=1)
```

2. Examples of custom model summaries

2.1. Plotting a hazard ratio against time

The following code plots the hazard ratio (Medium versus Good prognostic group) against time for both the proportional hazards model `fs7` and the better-fitting accelerated failure time model `fs2`. It illustrates the use of the following functions.

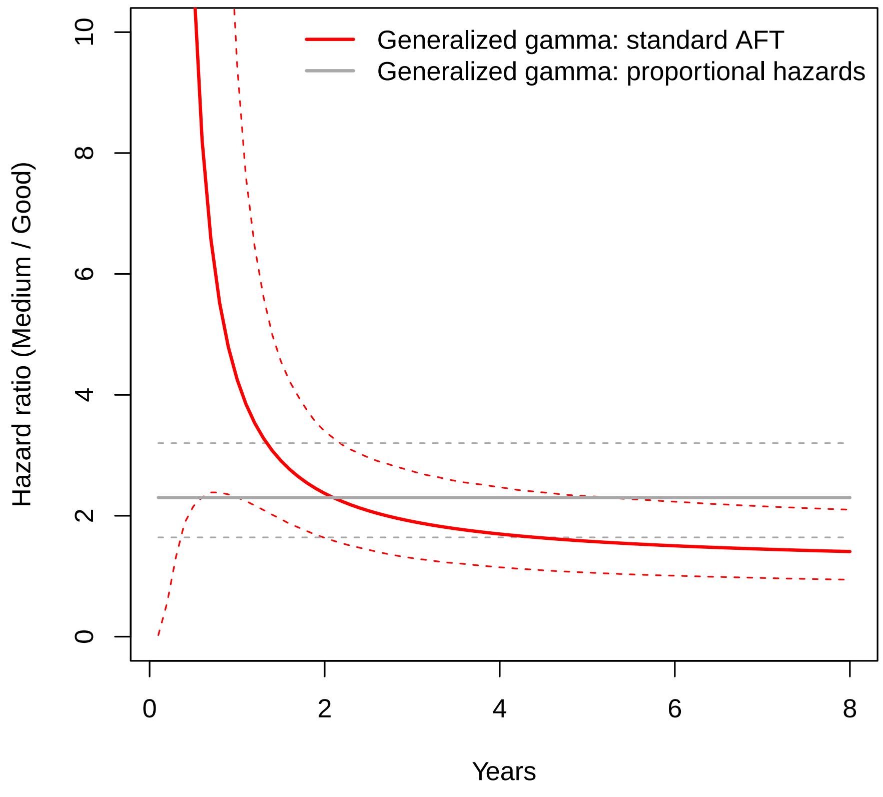
`summary.flexsurvreg` for generating the estimated hazard at a series of times, for particular covariate categories.

`normboot.flexsurvreg` for generating a bootstrap-style sample from the sampling distribution of the parameter estimates, for particular covariate categories.

`do.call` for constructing a function call by supplying a list containing the function's arguments. This is used throughout the source of **flexsurv**.

```
> fs2 <- flexsurvreg(Surv(recyrs, censrec) ~ group + sigma(group),
+                    data=bc, dist="gengamma")
> B <- 5000
> t <- seq(0.1, 8, by=0.1)
> hrAFT.est <-
+     summary(fs2, t=t, type="hazard",
+             newdata=data.frame(group="Medium"),ci=FALSE)[[1]][,"est"] /
+     summary(fs2, t=t, type="hazard",
+             newdata=data.frame(group="Good"),ci=FALSE)[[1]][,"est"]
> pars <- normboot.flexsurvreg(fs2, B=B, newdata=data.frame(group=c("Good","Medium")))
> hrAFT <- matrix(nrow=B, ncol=length(t))
> for (i in seq_along(t)){
+     haz.medium.rep <- do.call(hgengamma, c(list(t[i]), as.data.frame(pars[[2]])))
+     haz.good.rep <- do.call(hgengamma, c(list(t[i]), as.data.frame(pars[[1]])))
+     hrAFT[,i] <- haz.medium.rep / haz.good.rep
+ }
> hrAFT <- apply(hrAFT, 2, quantile, c(0.025, 0.975))
> hrPH.est <-
+     summary(fs7, t=t, type="hazard",
+             newdata=data.frame(group="Medium"),ci=FALSE)[[1]][,"est"] /
+     summary(fs7, t=t, type="hazard",
+             newdata=data.frame(group="Good"),ci=FALSE)[[1]][,"est"]
> pars <- normboot.flexsurvreg(fs7, B=B, newdata=data.frame(group=c("Good","Medium")))
> hrPH <- matrix(nrow=B, ncol=length(t))
> for (i in seq_along(t)){
```

```
+     haz.medium.rep <- do.call(hgengammaPH, c(list(t[i]), as.data.frame(pars[[2]])))
+     haz.good.rep <- do.call(hgengammaPH, c(list(t[i]), as.data.frame(pars[[1]])))
+     hrPH[,i] <- haz.medium.rep / haz.good.rep
+ }
> hrPH <- apply(hrPH, 2, quantile, c(0.025, 0.975))
> plot(t, hrAFT[1,], type="l", ylim=c(0, 10), col="red", xlab="Years",
+      ylab="Hazard ratio (Medium / Good)", lwd=1, lty=2)
> lines(t, hrAFT[2,], col="red", lwd=1, lty=2)
> lines(t, hrPH[1,], col="darkgray", lwd=1, lty=2)
> lines(t, hrPH[2,], col="darkgray", lwd=1, lty=2)
> lines(t, hrAFT.est, col="red", lwd=2)
> lines(t, hrPH.est, col="darkgray", lwd=2)
> legend("topright", lwd=c(2,2), col=c("red","darkgray"), bty="n",
+        c("Generalized gamma: standard AFT", "Generalized gamma: proportional hazards"))
```



Since version 2.2 of **flexsurv**, however, the code above is obsolete, since the function `hr_flexsurvreg` can simply be used to calculate hazard ratios and their confidence intervals against time.

```
> nd <- data.frame(group=c("Good","Medium"))
> hr_aft <- hr_flexsurvreg(fs2, t=t, newdata=nd)
> hr_ph <- hr_flexsurvreg(fs7, t=t, newdata=nd)
> plot(t, hr_aft$lcl, type="l", ylim=c(0, 10), col="red", xlab="Years",
+      ylab="Hazard ratio (Medium / Good)", lwd=1, lty=2)
> lines(t, hr_aft$ucl, col="red", lwd=1, lty=2)
> lines(t, hr_aft$est, col="red", lwd=2)
> lines(t, hr_ph$lcl, col="darkgray", lwd=1, lty=2)
> lines(t, hr_ph$ucl, col="darkgray", lwd=1, lty=2)
> lines(t, hr_ph$est, col="darkgray", lwd=2)
> legend("topright", lwd=c(2,2), col=c("red","darkgray"), bty="n",
+        c("Generalized gamma: standard AFT", "Generalized gamma: proportional hazards"))
```

2.2. Restricted mean survival

The expected survival up to time t, from a model with cumulative distribution $F(t|\alpha)$, is

$$E(T|T < t) = \int_0^t 1 - F(u|\alpha)du$$

An estimate and confidence interval for this, for a specified covariate value, can be computed as follows.

In versions of **flexsurv** before version 2.2, this example used a custom function supplied to `summary.flexsurvreg`. However, specific functions for estimating mean and restricted mean survival for built-in distributions have been available since version 1.1 (2017).

Hence we can now calculate the restricted mean survival in various simpler ways.

```
> summary(fs2, type="rmst", t=100, tidy=TRUE)

  time       est       lcl       ucl  group
1  100 21.501020 13.495570 31.461506   Good
2  100 12.000251  7.978702 17.616684 Medium
3  100  5.539196  3.755992  8.842049   Poor
```

```
> summary(fs2, fn=rmst_gengamma, t=100, tidy=TRUE)

  time       est       lcl       ucl  group
1  100 21.501020 13.078146 32.072089   Good
2  100 12.000251  8.088047 18.125800 Medium
3  100  5.539196  3.720218  8.824741   Poor
```

Or for the baseline category of ”Good”:

```
> est <- fs2$res[,"est"]
> rmst_gengamma(t=100, mu=est[1], sigma=est[2], Q=est[3])

[1] 21.50102
```

Note that the (unrestricted) median is more stable, and less than the restricted mean due to the skewness of this distribution.

```
> summary(fs2, type="median")

group=Good
       est      lcl      ucl
1 9.736555 7.133129 13.94929

group=Medium
       est      lcl      ucl
1 4.717786 3.903108 5.798121

group=Poor
       est      lcl      ucl
1 2.280328 1.965234 2.685015
```

Note also that the custom function `fn` supplied to `summary.flexsurvreg` must be vectorised - previous versions of this example used an unvectorised custom function called `mean.gengamma` which will not work in version 2.2 or later.

3. Spline models

3.1. Prognostic model for the German breast cancer data

The regression model III in **?** used to create the prognostic group from the breast cancer data (supplied as `bc` in **flexsurv** and `GBSG2` in **TH.data**) can be reproduced as follows. Firstly, the required fractional polynomial transformations of the covariates are constructed. `progc` implements the Cox model used by **?**, and `prog3` is a flexible fully-parametric alternative, implemented as a spline with three internal knots. The number of knots was chosen to minimise AIC. The covariate effects are very similar.

After fitting the model, the prognostic index can then be derived from categorising observations in three groups according to the tertiles of the linear predictor in each model. The indices produced by the Cox model (`progc`) and the spline-based model (`progf`) agree exactly.

```
> if (require("TH.data")){
+
+ GBSG2 <- transform(GBSG2,
+                    X1a=(age/50)^-2,
+                    X1b=(age/50)^-0.5,
+                    X4=tgrade %in% c("II","III"),
+                    X5=exp(-0.12*pnodes),
+                    X6=(progrec+1)^0.5
+                    )
+ (progc <- coxph(Surv(time, cens) ~ horTh + X1a + X1b + X4 +
+                    X5 + X6, data=GBSG2))
+ (prog3 <- flexsurvspline(Surv(time, cens) ~ horTh + X1a + X1b + X4 +
+                             X5 + X6, k=3, data=GBSG2))
+ predc <- predict(progc, type="lp")
+ progc <- cut(predc, quantile(predc, 0:3/3))
+ predf <- model.matrix(prog3) %*% prog3$res[-(1:5),"est"]
+ progf <- cut(predf, quantile(predf, 0:3/3))
+ table(progc, progf)
+
+ }
```

```
                  progf
progc              (-9.91,-7.76] (-7.76,-7.11] (-7.11,-3.36]
  (-2.21,-0.0437]            228             0             0
  (-0.0437,0.6]                0           228             0
  (0.6,4.31]                   0             0           229
```

4. Right truncation: retrospective ascertainment

Suppose we want to estimate the distribution of the time from onset of a disease to death, but have only observed cases known to have died by the current date. In this case, times from onset to death for individuals in the data are *right-truncated* by the current date minus the onset date. Predicted survival times for new cases can then be described by an un-truncated version of the fitted distribution. Denote the time from onset to death as the *delay* time.

This is illustrated in the following simulated example. Suppose individual onset times are uniformly distributed between 0 and 30 days. Their delay times are generated from a Gamma distribution.

```
> set.seed(1)
> nsim <- 10000
> onsetday <- runif(nsim, 0, 30)
> deathday <- onsetday + rgamma(nsim, shape=1.5, rate=1/10)
```

The data are examined at 40 days. Therefore we do not observe people who have died after this time. For each individual in the observed data, their delay time is right-truncated by 40 days minus their onset day, since their delay times cannot be greater than this if they are included in the data. The right-truncation point is specified by the variable `rtrunc` in the data.

```
> datt <- data.frame(delay = deathday - onsetday,
+                    event = rep(1, nsim),
+                    rtrunc = 40 - onsetday)
> datt <- datt[datt$delay < datt$rtrunc, ]
```

The truncated Gamma model is fitted with `flexsurvreg` by specifying individual-specific truncation points in the argument `rtrunc`. The fitted model reproduces the gamma parameters that were used to simulate the data. After fitting the model, we can use the fitted model to predict the mean time to death - this is approximately the shape / rate of the untruncated gamma distribution.

```
> fitt <- flexsurvreg(Surv(delay, event) ~ 1, data=datt, rtrunc = rtrunc, dist="gamma")
> fitt

Call:
flexsurvreg(formula = Surv(delay, event) ~ 1, data = datt, rtrunc = rtrunc,
    dist = "gamma")

Estimates:
       est      L95%     U95%     se
shape  1.50886  1.45760  1.56194  0.02661
rate   0.10067  0.09439  0.10737  0.00331

N = 7728,  Events: 7728,  Censored: 0
Total time at risk: 82028.85
Log-likelihood = -24252.16, df = 2
AIC = 48508.32

> summary(fitt, t=1, fn = mean_gamma)

  time      est      lcl      ucl
1    1 14.98784 14.40418 15.62192
```