

Introduction to the `rstpm2` package

Mark Clements
Karolinska Institutet

Abstract

This vignette outlines the methods and provides some examples for link-based survival models as implemented in the R `rstpm2` package.

Keywords: survival, splines.

1. Background and theory

Link-based survival models provide a flexible and general approach to modelling survival or time-to-event data. The survival function $S(t|x)$ to time t for covariates x is defined in terms of a link function G and a linear prediction $\eta(t, x)$, such that

$$S(t|x) = G(\eta(t, x))$$

where η is a function of both time t and covariates x . The linear predictor can be constructed in a flexible manner. Royston and Parmar (2003) focused on time being modelled using natural splines for log-time, including left truncation and relative survival. We have implemented the Royston-Parmar model class and extended it in several ways, allowing for: (i) general parametric models for $\eta(t, x)$, including B-splines and natural splines for different transformations of time; (ii) general semi-parametric models for $\eta(t, x)$ including penalised smoothers together with unpenalised parametric functions; (iii) interval censoring; and (iv) frailties using Gamma and log-Normal distributions. Fully parametric models are estimated using maximum likelihood, while the semi-parametric models are estimated using maximum penalised likelihood with smoothing parameters selected using A more detailed theoretical development is available from the paper by Liu, Pawitan and Clements (available on request). Why would you want to use these models?

2. Mean survival

This has a useful interpretation for causal inference.

$$E_Z(S(t|Z, X = 1)) - E_Z(S(t|Z, X = 0))$$

```
fit <- stpm(...)  
predict(fit, type="meansurv", newdata=data)
```

3. Cure models

For cure, we use the melanoma dataset used by Andersson and colleagues for cure models with Stata's `stpm2` (see <http://www.pauldickman.com/survival/>).

Initially, we merge the patient data with the all cause mortality rates.

```
> popmort2 <- transform(rstpm2::popmort, exitage=age, exityear=year, age=NULL, year=NULL)
> colon2 <- within(rstpm2::colon, {
+   status <- ifelse(surv_mm>120.5, 1, status)
+   tm <- pmin(surv_mm, 120.5)/12
+   exit <- dx+tm*365.25
+   sex <- as.numeric(sex)
+   exitage <- pmin(floor(age+tm), 99)
+   exityear <- floor(yydx+tm)
+   ##year8594 <- (year8594=="Diagnosed 85-94")
+ })
> colon2 <- merge(colon2, popmort2)
```

For comparisons, we fit the relative survival model without and with cure.

```
> fit0 <- stpm2(Surv(tm, status %in% 2:3)~I(year8594=="Diagnosed 85-94"),
+             data=colon2,
+             bhazard=colon2$rate, df=5)

> summary(fit <- stpm2(Surv(tm, status %in% 2:3)~I(year8594=="Diagnosed 85-94"),
+                   data=colon2,
+                   bhazard=colon2$rate,
+                   df=5, cure=TRUE))
```

Maximum likelihood estimation

Call:

```
mle2(minuslogl = negll, start = coef, eval.only = TRUE, vecpar = TRUE,
     gr = function(beta)
     {
       localargs <- args
       localargs$init <- beta
       localargs$return_type <- "gradient"
       return(.Call("model_output", localargs, package = "rstpm2"))
     }, control = list(parscale = c(1, 1, 1, 1, 1, 1, 1), maxit = 300),
     lower = -Inf, upper = Inf)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(z)
(Intercept)	-3.977323	0.054778	-72.6082	< 2.2e-16
I(year8594 == "Diagnosed 85-94")TRUE	-0.155612	0.025088	-6.2027	5.551e-10
nsx(log(tm), df = 5, cure = TRUE)1	3.323191	0.053165	62.5066	< 2.2e-16

```

nsx(log(tm), df = 5, cure = TRUE)2    3.628630    0.053159    68.2594 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)3    1.634847    0.022465    72.7743 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)4    6.592021    0.111504    59.1194 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)5    3.371809    0.042788    78.8027 < 2.2e-16

```

```

(Intercept)                                ***
I(year8594 == "Diagnosed 85-94")TRUE      ***
nsx(log(tm), df = 5, cure = TRUE)1        ***
nsx(log(tm), df = 5, cure = TRUE)2        ***
nsx(log(tm), df = 5, cure = TRUE)3        ***
nsx(log(tm), df = 5, cure = TRUE)4        ***
nsx(log(tm), df = 5, cure = TRUE)5        ***
---

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

-2 log L: 42190.77

```
> predict(fit, head(colon2), se.fit=TRUE)
```

	Estimate	lower	upper
1	0.8610828	0.8542898	0.8675842
2	0.7934651	0.7850103	0.8016309
3	0.6967400	0.6863191	0.7068926
4	0.8610828	0.8542898	0.8675842
5	0.8221243	0.8143226	0.8296334
6	0.8610828	0.8542898	0.8675842

The estimate for the year parameter from the model without cure is within three significant figures with that in Stata. For the predictions, the Stata model gives:

```

+-----+
|      surv   surv_lci   surv_uci |
+-----+
1. | .86108264   .8542898   .8675839 |
2. | .79346526   .7850106   .8016309 |
3. | .69674037   .6863196   .7068927 |
4. | .86108264   .8542898   .8675839 |
5. | .82212425   .8143227   .8296332 |
+-----+
6. | .86108264   .8542898   .8675839 |
+-----+

```

We can estimate the proportion of failures prior to the last event time:

```

> newdata.eof <- data.frame(year8594 = unique(colon2$year8594),
+                             tm=10)
> 1-predict(fit0, newdata.eof, type="surv", se.fit=TRUE)

```

	Estimate	lower	upper
1	0.6060933	0.6208798	0.5913474
2	0.5512425	0.5658371	0.5367647

```
> 1-predict(fit, newdata.eof, type="surv", se.fit=TRUE)
```

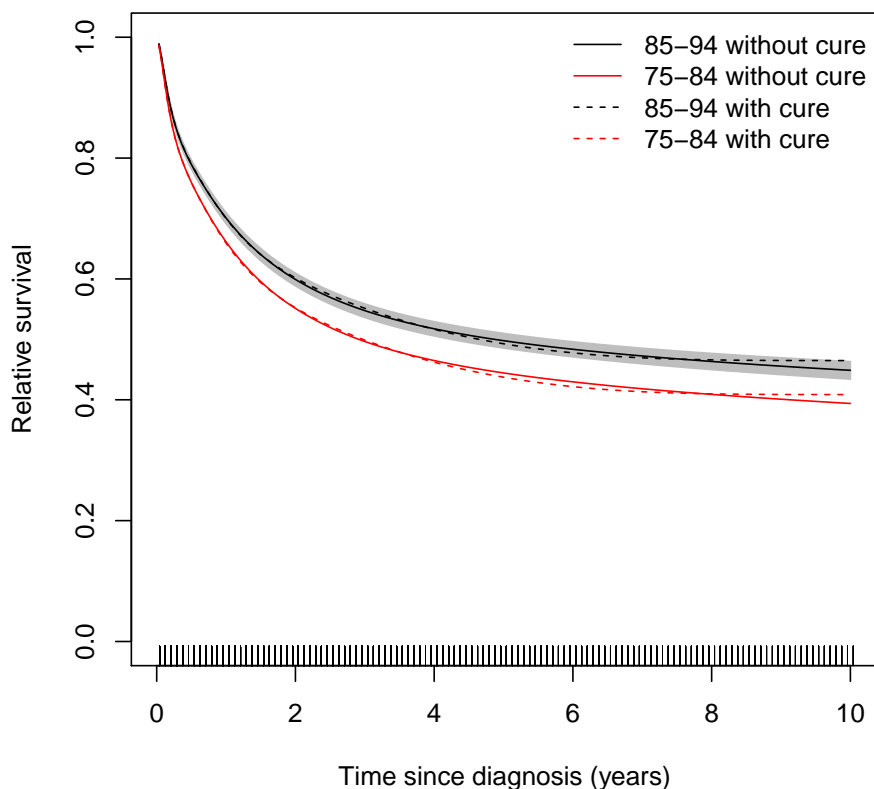
	Estimate	lower	upper
1	0.5913317	0.6055024	0.5772183
2	0.5350824	0.5485383	0.5217445

```
> predict(fit, newdata.eof, type="haz", se.fit=TRUE)
```

	Estimate	lower	upper
1	1.254130e-06	1.093036e-06	1.438966e-06
2	1.073398e-06	9.335145e-07	1.234243e-06

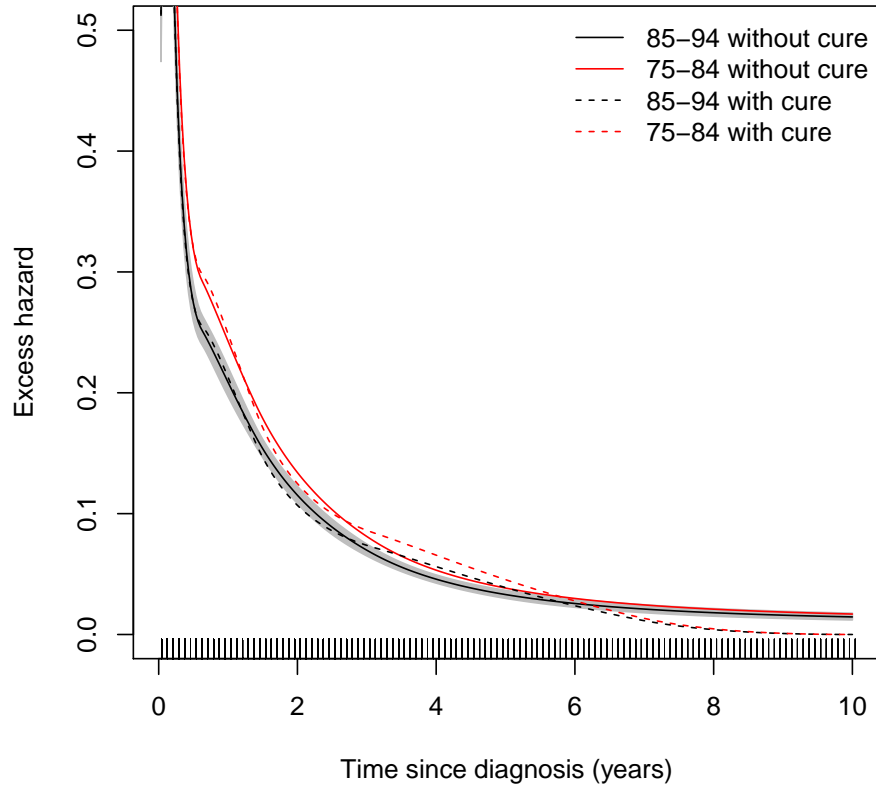
We can plot the predicted survival estimates:

```
> tms=seq(0,10,length=301)[-1]
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 85-94", tm=tms), ylim=0:1,
+       xlab="Time since diagnosis (years)", ylab="Relative survival")
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 75-84",tm=tms),
+       add=TRUE,line.col="red",rug=FALSE)
> ## warnings: Predicted hazards less than zero for cure
> plot(fit,newdata=data.frame(year8594 = "Diagnosed 85-94",tm=tms),
+       add=TRUE,ci=FALSE,lty=2,rug=FALSE)
> plot(fit,newdata=data.frame(year8594="Diagnosed 75-84",tm=tms),
+       add=TRUE,rug=FALSE,line.col="red",ci=FALSE,lty=2)
> legend("topright",c("85-94 without cure","75-84 without cure",
+                     "85-94 with cure","75-84 with cure"),
+       col=c(1,2,1,2), lty=c(1,1,2,2), bty="n")
```



And the hazard curves:

```
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 85-94", tm=tms),
+      ylim=c(0,0.5), type="hazard",
+      xlab="Time since diagnosis (years)",ylab="Excess hazard")
> plot(fit0,newdata=data.frame(year8594 = "Diagnosed 75-84", tm=tms),
+      type="hazard",
+      add=TRUE,line.col="red",rug=FALSE)
> plot(fit,newdata=data.frame(year8594 = "Diagnosed 85-94", tm=tms),
+      type="hazard",
+      add=TRUE,ci=FALSE,lty=2,rug=FALSE)
> plot(fit,newdata=data.frame(year8594="Diagnosed 75-84", tm=tms),
+      type="hazard",
+      add=TRUE,rug=FALSE,line.col="red",ci=FALSE,lty=2)
> legend("topright",c("85-94 without cure","75-84 without cure",
+                    "85-94 with cure","75-84 with cure"),
+      col=c(1,2,1,2), lty=c(1,1,2,2), bty="n")
```

**Affiliation:**

Mark Clements
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet
Email: mark.clements@ki.se