Introduction to the rstpm2 package

Mark Clements
Karolinska Institutet

Abstract

This vignette outlines the methods and provides some examples for generalised survival models as implemented in the R **rstpm2** package.

Keywords: survival, splines.

1. Background and theory

Generalised 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 inverse 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, with the main constraint that the time effects be smooth and twice differentiable. 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 articles by Liu, Pawitan and Clements (available on request).

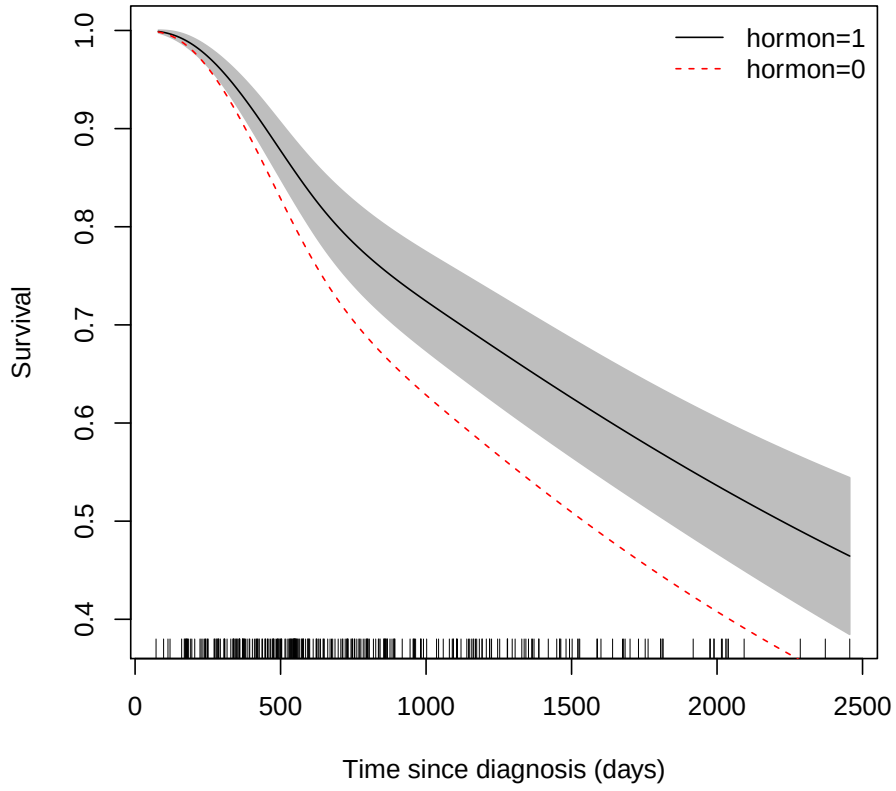
2. Independent survival analysis

We begin with some simple proportional hazard models using the brcancer dataset. We can fit the models using very similar syntax to coxph, except that we need to specify the degrees of freedom for the baseline smoother. Typical values for df are 3-6. For this model the model parameters include an intercept term, time-invariant log-hazard ratios, and parameters for the baseline smoother. The default for the baseline smoother is to use the nsx function, which is a limited extension to the splines::ns function, with log of the time effect.

```
> fit <- stpm2(Surv(rectime,censrec==1)~hormon,
+              data=brcancer, df=4)
> summary(fit)

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), maxit = 300),
    lower = -Inf, upper = Inf)

Coefficients:
                           Estimate Std. Error z value     Pr(z)
(Intercept)                -6.79773    0.72643 -9.3578 < 2.2e-16 ***
hormon                     -0.36406    0.12491 -2.9144  0.003563 **
nsx(log(rectime), df = 4)1  5.69995    0.71677  7.9523 1.831e-15 ***
nsx(log(rectime), df = 4)2  4.85614    0.48002 10.1166 < 2.2e-16 ***
nsx(log(rectime), df = 4)3 10.13328    1.41268  7.1731 7.331e-13 ***
nsx(log(rectime), df = 4)4  4.70626    0.33016 14.2545 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

-2 log L: 5212.943

> ## utility to exponentiate the ith components
> expi <- function(x,i=1:length(x)) { x[i] <- exp(x[i]); x }
> fit.cox <- coxph(Surv(rectime,censrec==1)~hormon, data=brcancer)
> rbind(coxph=coef(summary(fit.cox)),
+       stpm2=expi(coef(summary(fit))["hormon",c(1,1,2:4)],2))

             coef exp(coef)  se(coef)         z    Pr(>|z|)
hormon -0.3640099 0.6948843 0.1250446 -2.911041 0.003602266
stpm2  -0.3640574 0.6948513 0.1249147 -2.914449 0.003563175

>
```

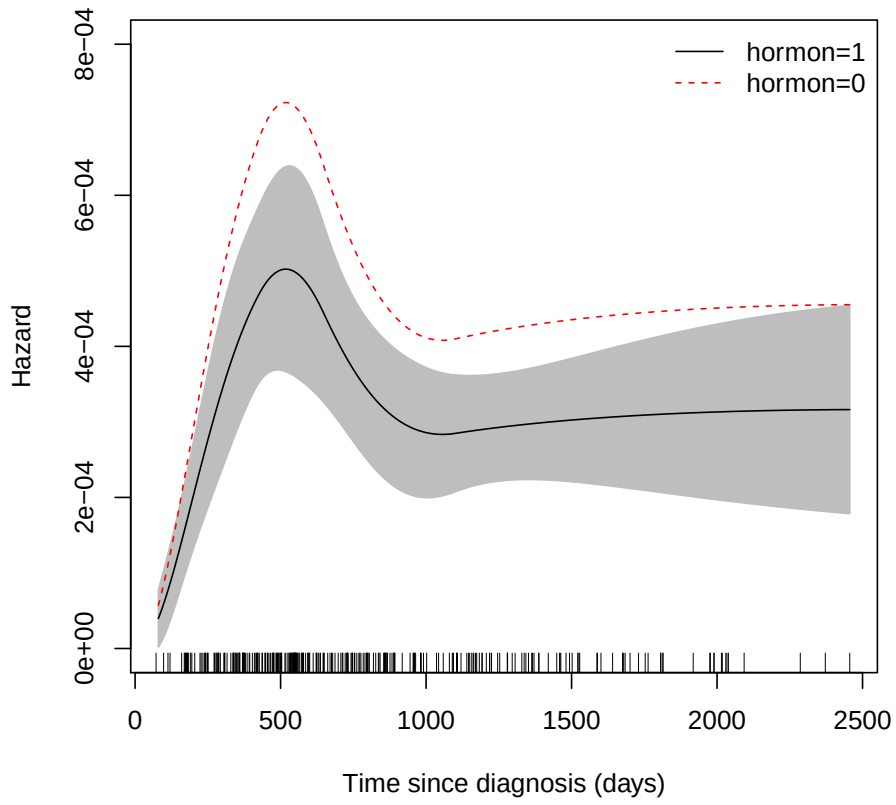
We see that the hazard ratios are very similar to the coxph model. The model fit can also be used to estimate a variety of parameters. For example, we can easily estimate survival for a given parameter.

```
> # tms=seq(0,10,length=301)[-1]
> plot(fit,newdata=data.frame(hormon=1),
+      xlab="Time since diagnosis (days)")
> lines(fit, newdata=data.frame(hormon=0), col=2, lty=2)
> legend("topright", c("hormon=1","hormon=0"),lty=1:2,col=1:2,bty="n")
>
```



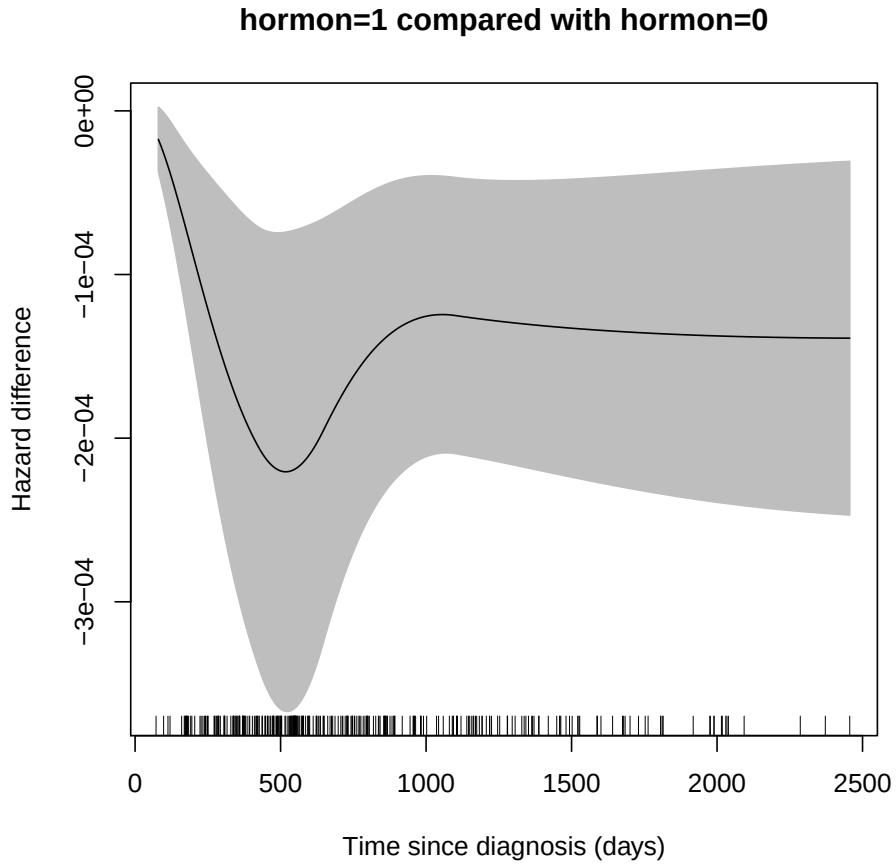
We can also calculate the hazards.

```
> # tms=seq(0,10,length=301)[-1]
> plot(fit,newdata=data.frame(hormon=1), type="hazard",
+      xlab="Time since diagnosis (days)", ylim=c(0,8e-4))
> lines(fit, newdata=data.frame(hormon=0), col=2, lty=2, type="hazard")
> legend("topright", c("hormon=1","hormon=0"),lty=1:2,col=1:2,bty="n")
>
```

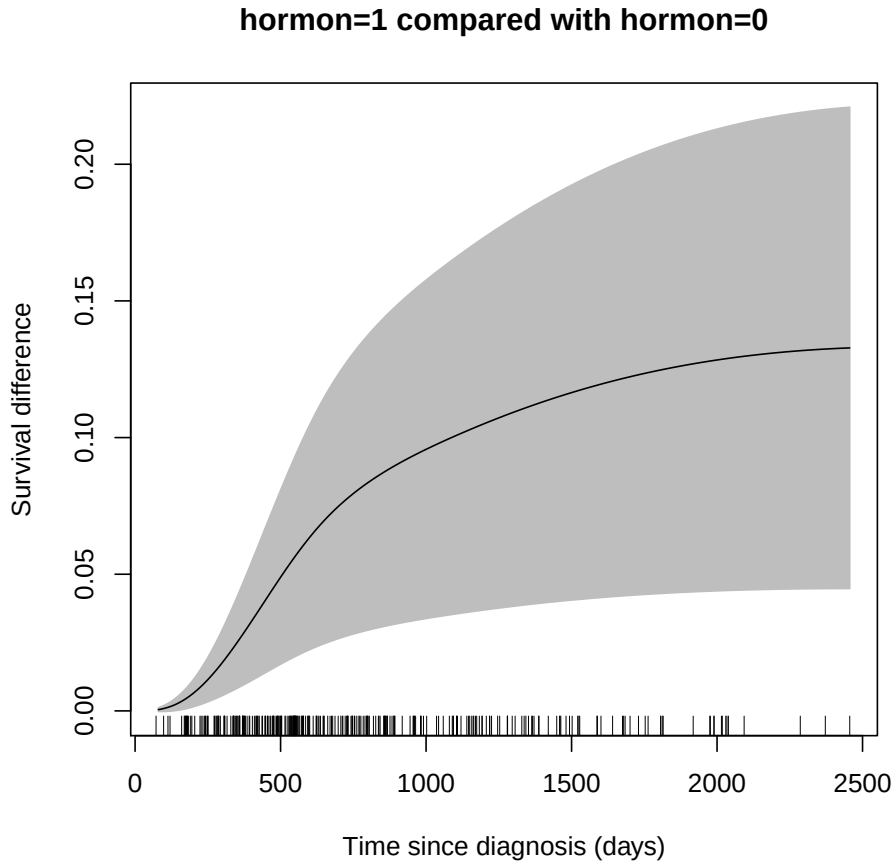


Usefully, we can also estimate survival differences and hazard differences. We define the survival differences using a reference covariate pattern using the newdata argument, and then define an exposed function which takes the newdata and transforms for the 'exposed' covariate pattern. As an example

```
> plot(fit,newdata=data.frame(hormon=0), type="hdiff",
+      exposed=function(data) transform(data, hormon=1),
+      main="hormon=1 compared with hormon=0",
+      xlab="Time since diagnosis (days)")
>
```




```
> plot(fit,newdata=data.frame(hormon=0), type="sdiff",
+      exposed=function(data) transform(data, hormon=1),
+      main="hormon=1 compared with hormon=0",
+      xlab="Time since diagnosis (days)")
>
```



3. 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 <- stpm2(...)
predict(fit,type="meansurv",newdata=data)
```

4. 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)                          -3.977663   0.054783 -72.6075 < 2.2e-16
I(year8594 == "Diagnosed 85-94")TRUE -0.155511   0.025089  -6.1984 5.704e-10
nsx(log(tm), df = 5, cure = TRUE)1    3.323382   0.053170  62.5043 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)2    3.628899   0.053164  68.2580 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)3    1.634974   0.022466  72.7744 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)4    6.592489   0.111515  59.1177 < 2.2e-16
nsx(log(tm), df = 5, cure = TRUE)5    3.371954   0.042790  78.8021 < 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 0.861104.... 0.8544595 0.8677491
2 0.793496.... 0.7851875 0.8018048
3 0.696783.... 0.6864973 0.7070695
4 0.861104.... 0.8544595 0.8677491
5 0.822150.... 0.8144974 0.8298042
6 0.861104.... 0.8544595 0.8677491

>
```

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 0.606094.... 0.6208646 0.5913254
2 0.551251.... 0.5657907 0.5367131
```

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

```
      Estimate     lower     upper
1 0.591297.... 0.6054432 0.5771519
2 0.535085.... 0.5484842 0.5216862
```

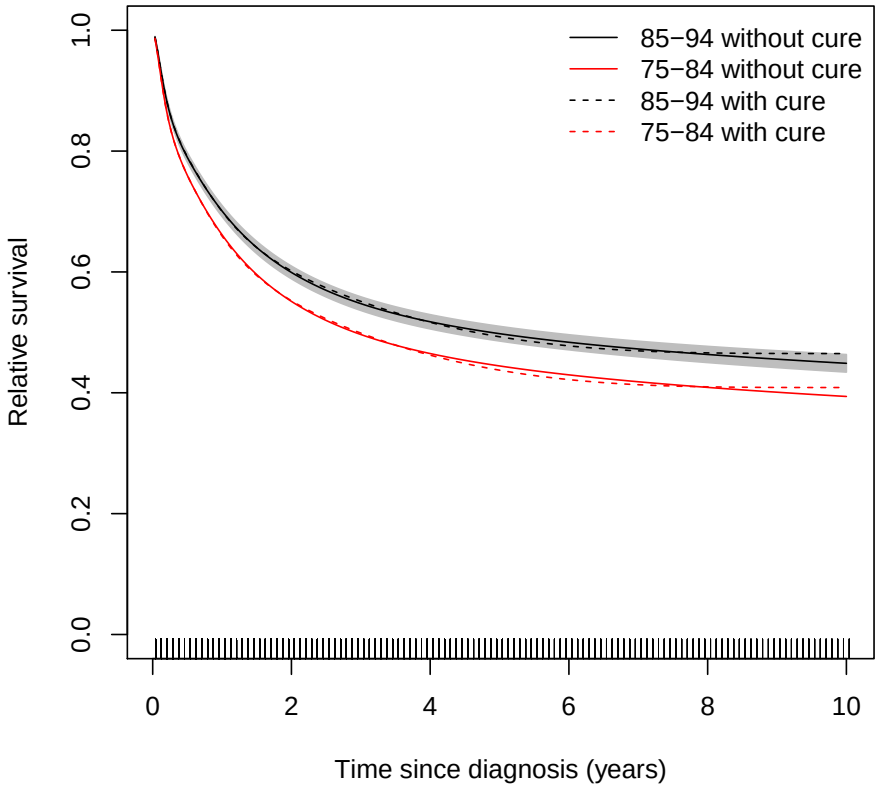
```
> predict(fit, newdata.eof, type="haz", se.fit=TRUE)
```

```
      Estimate        lower        upper
1 1.253896.... 1.081493e-06 1.426299e-06
2 1.073306.... 9.234322e-07 1.223181e-06
```

```
>
```

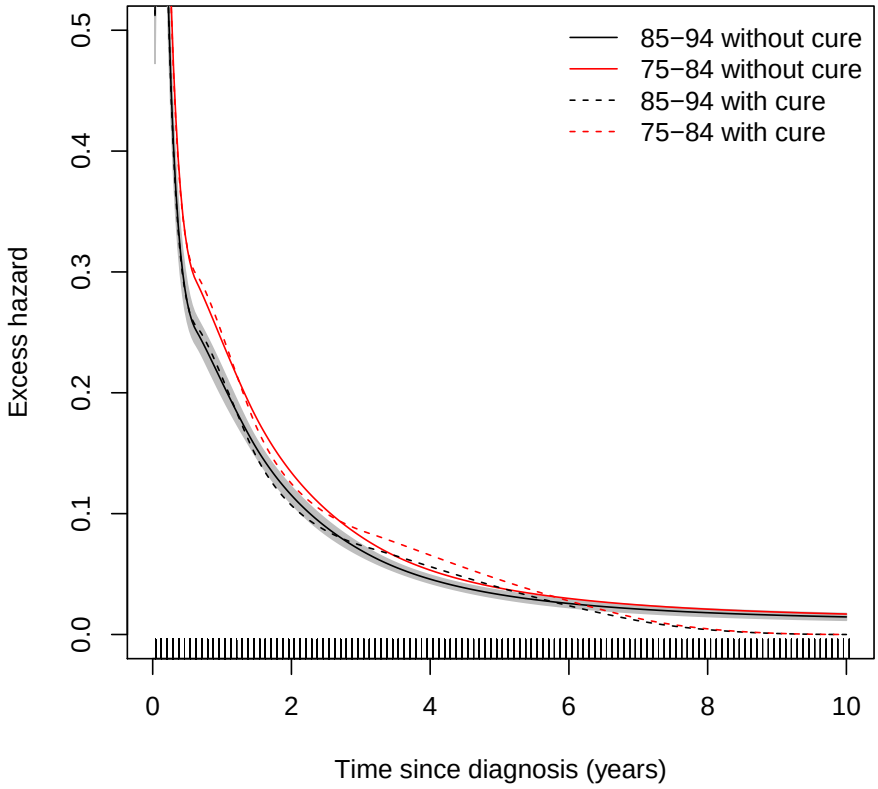
We can plot the predicted survival estimates:

```
> 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