

# 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

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

where  $\eta$  is a function of both time  $t$  and covariates  $x$ , with regression parameters  $\theta$ . We can calculate the hazard from this function, where

$$\begin{aligned} h(t|x; \theta) &= \frac{d}{dt} (-\log(S(t|x; \theta))) \\ &= \frac{-G'(\eta(t, x; \theta))}{G(\eta(t, x; \theta))} \frac{\partial \eta(t, x; \theta)}{\partial t} \end{aligned}$$

We model using a linear predictor  $\eta(t, x; \theta) = X(t, x)\theta$  for a design matrix  $X(t, x)$ . The linear predictor can be constructed in a flexible manner, with the main constraint being that the time effects be smooth and twice differentiable. We calculate the derivative for the linear predictor using finite differences, such that

$$\frac{\partial \eta(t, x; \theta)}{\partial t} = \frac{\partial X(t, x)\theta}{\partial t} = \frac{X(t + \epsilon, x) - X(t - \epsilon, x)}{2\epsilon} \theta = X_D(t, x)\theta$$

for a derivative design matrix  $X_D(t, x)$ . This formulation allows for considerable flexibility in the construction of the linear predictor, with possible interactions between time and covariates. The default smoother for time using natural splines for  $\log(\text{time})$ , which is the flexible parametric survival model developed by Royston and Parmar (2003) and implemented by the Stata command `stpm2`<sup>1</sup>

---

<sup>1</sup>As a technical aside, the Stata implementation uses natural splines using a truncated power basis with

Link description	Inverse link function $G(\eta(t, x; \theta))$	Interpretation	<code>link.type</code>
log-log	$\exp(-\exp(\eta(t, x; \theta)))$	Proportional hazards	"PH"
logit	$\text{expit}(-\eta(t, x; \theta))$	Proportional odds	"PO"
probit	$\Phi(-\eta(t, x; \theta))$	Probit	"probit"
log	$\exp(-\eta(t, x; \theta))$	Additive hazards	"AH"
Aranda-Ordaz	$\exp(-\log(\psi * \exp(\eta(t, x; \theta)) + 1)/\psi)$	Aranda-Ordaz	"AO"

Table 1: Implemented link functions

The models are estimated using maximum likelihood estimation (MLE) for fully parametric models, penalised MLE for penalised smoothers, maximum marginal likelihood estimation (MMLE) for parametric models with clustered data, or penalised MMLE for penalised models with clustered data. The likelihoods include left truncation, right censoring and interval censoring. For clustered data, we include Gamma frailties and normal random effects. Details on these models are available from <https://doi.org/10.1177/0962280216664760> and <https://doi.org/10.1002/sim.7451>.

## 2. Syntax

The main functions for fitting the models are `stpm2` for parametric models, possibly with clustered data, and `pstpm2` for penalised models, possibly with clustered data. A subset of the syntax for `stpm2` is:

```
stpm2(formula, data, smooth.formula = NULL,
      df = 3, tvc = NULL,
      link.type=c("PH","PO","probit","AH","AO"), theta.A0=0,
      bhazard = NULL,
      robust = FALSE, cluster = NULL, frailty = !is.null(cluster) & !robust,
      RandDist=c("Gamma","LogN"),
      ...)
```

The `formula` has a `Surv` object on the left-hand-side and a linear predictor on the right-hand-side that does *not* include time (for `pstpm2`, it also does not include penalised functions). The time effects can be specified in several ways: the most general is using `smooth.formula`, where the right-hand-side of the formula specifies functions for time that are smooth with respect to time. This specification can include interactions between time and covariates. As an example, `smooth.formula=~nsx(log(time),df=3)+x:nsx(log(time),df=2)` specifies a baseline natural spline smoother of the log of the variable `time` used in the `Surv` object with three degrees of freedom, with an interaction between a covariate `x` and a natural spline smoother of `log(time)` with two degrees of freedom. Other specifications of time effects have equivalent formulations: for example, `df = 3` is equivalent to `smooth.formula=~nsx(log(time),df=3)` for the variable `time`. Similarly, `tvc=list(x=2)` is equivalent to `smooth.formula=~x:nsx(log(time),df=2)`. A current limitation of the implementation is that the dataset `data` needs to be specified.

---

orthogonalisation, while the `ns()` function in R uses a matrix projection of B-splines. Note that we have implemented an extended `nsx()` function for natural splines that includes cure splines, centering, and a compatibility argument to use Stata `stpm2`'s unusual specification of quantiles.

Type of link is specified with the `link.type` argument; this defaults to a log–log link for proportional hazards (see Table 1). For the Aranda-Ordaz link, the fixed value of the scale term  $\psi$  is specified using the `theta.A0` argument. For relative survival, a vector for the baseline hazard can be specified using the `bhazard` argument. A vector for the clusters can be specified with the `cluster` argument. The calculation of robust standard errors can be specified with the `robust=TRUE` argument; if `robust` is false, then the model assumes a frailty or random effects model, with either a default Gamma frailty (`RandDist="Gamma"`) or a normal random effect (`RandDist="LogN"`, using notation from the `frailtypack` package). The syntax for the fitting the penalised models with `pstpm2` is very similar. A subset of the arguments are:

```
pstpm2(formula, data, smooth.formula = NULL,
        tvc = NULL,
        bhazard = NULL,
        sp=NULL,
        criterion=c("GCV","BIC"),
        link.type=c("PH","PO","probit","AH","AO"), theta.A0=0,
        robust=FALSE,
        frailty=!is.null(cluster) & !robust, cluster = NULL, RandDist=c("Gamma","LogN"),
        ...)
```

The penalised smoothers are specified using the `s()` function from the `mgcv` package within the `smooth.formula` argument; by default, not specifying `smooth.formula` will lead to `smooth.formula=~s(log(` Interactions with time (both penalised and unpenalised) and penalised covariate effects should be specified using `smooth.formula`.

Note that the `df` argument is not included. By default, the smoothing parameter(s) are using the `criterion` argument; the smoothing parameters can also be fixed using the `sp` argument. The specifications for relative survival, link type, and clustered data follow that for the `stpm2` function.

## 2.1. Post-estimation

One of the strengths of these models is varied post-estimation. Most of the estimators are described in Tables 2 and 3.

## 3. Examples: 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)
```

Description	Formulation <sup>a</sup>	type
Conditional link	$\eta(t, x; \hat{\theta})$	"link"
Conditional survival	$S(t x; \hat{\theta}) = G(\eta(t, x; \hat{\theta}))$	"surv"
Conditional odds	$\text{Odds}(t x; \hat{\theta}) = S(t x; \hat{\theta}) / (1 - S(t x; \hat{\theta}))$	"odds"
Conditional failure	$1 - S(t x; \hat{\theta})$	"fail"
Conditional cumulative hazard	$H(t x; \hat{\theta}) = -\log G(\eta(t, x; \hat{\theta}))$	"cumhaz"
Conditional density	$f(t x; \hat{\theta}) = G'(\eta(t, x; \hat{\theta})) \frac{\partial \eta(t, x; \hat{\theta})}{\partial t}$	"density"
Conditional hazard	$h(t x; \hat{\theta}) = \frac{G'(\eta(t, x; \hat{\theta}))}{G(\eta(t, x; \hat{\theta}))} \frac{\partial \eta(t, x; \hat{\theta})}{\partial t}$	"hazard"
Conditional log hazard	$\log h(t x; \hat{\theta})$	"loghazard"
Conditional survival differences	$S(t x^*; \hat{\theta}) - S(t x; \hat{\theta})$	"survdiff"
Conditional hazard differences	$h(t x^*; \hat{\theta}) - h(t x; \hat{\theta})$	"hazdiff"
Conditional hazard ratios	$h(t x^*; \hat{\theta}) / h(t x; \hat{\theta})$	"hr"
Conditional odds ratios	$\text{Odds}(t x^*; \hat{\theta}) / \text{Odds}(t x; \hat{\theta})$	"or"
Standardised survival	$E_{X^*} S(t X^*; \hat{\theta})$	"meansurv"
Standardised survival differences	$E_{X_1^*} S(t X_1^*; \hat{\theta}) - E_{X_0^*} S(t X_0^*; \hat{\theta})$	"meansurvdiff"
Standardised hazard	$h_{X^*}(t X^*; \hat{\theta}) = \frac{E_{X^*}(S(t X^*; \hat{\theta})h(t X^*; \hat{\theta}))}{E_{X^*}(S(t X^*; \hat{\theta}))}$	"meanhaz"
Standardised hazard ratio	$h_{X_1^*}(t X_1^*; \hat{\theta}) / h_{X_0^*}(t X_0^*; \hat{\theta})$	"meanhazdiff"
Attributable fraction	$\frac{E_{X^*} S(t X^*; \hat{\theta}) - E_X S(t X; \hat{\theta})}{1 - E_X S(t X; \hat{\theta})}$	"af"

<sup>a</sup>Notation:  $x^*$  is a covariate pattern for the “exposed” group;  $X^*$  is a set of possibly counterfactual covariates;  $E_X(g(X))$  is the expectation or average of  $g(X)$  across the set  $X$ ;  $X_0^*$  and  $X_1^*$  are sets of possibly counterfactual covariates for the “unexposed” and “exposed” sets, respectively.

Table 2: Types of conditional post-estimators

Description	Formulation <sup>a</sup>	type
Marginal survival	$S_M(t x; \hat{\theta}) = E_Z G(\eta(t, x, Z; \hat{\theta}))$	"margsurv"
Marginal hazard	$h_M(t x; \hat{\theta}) = E_Z (h(t, x, Z; \hat{\theta}))$	"marghaz"
Marginal survival differences	$S_M(t x^*; \hat{\theta}) - S_M(t x; \hat{\theta})$	"margsurvdiff"
Marginal hazard ratios	$h_M(t x^*; \hat{\theta}) / h_M(t x; \hat{\theta})$	"marghr"
Standardised marginal survival	$E_Z E_{X^*} S(t X^*, Z; \hat{\theta})$	"meanmargsurv"
Standardised marginal survival differences	$E_Z E_{X^*} S(t X^*, Z; \hat{\theta}) - E_Z E_X S(t X, Z; \hat{\theta})$	"meansurvdiff"
Attributable fraction	$\frac{E_Z E_{X^*} S(t X^*, Z; \hat{\theta}) - E_Z E_X S(t X, Z; \hat{\theta})}{1 - E_Z E_X S(t X, Z; \hat{\theta})}$	"af"

<sup>a</sup>Notation:  $Z$  is a random effect or frailty;  $x^*$  is a covariate pattern for the “exposed” group;  $X^*$  is a set of possibly counterfactual covariates;  $E_X(g(X))$  is the expectation or average of  $g(X)$  across the set  $X$ ;  $X_0^*$  and  $X_1^*$  are sets of possibly counterfactual covariates for the “unexposed” and “exposed” sets, respectively.

Table 3: Types of post-estimators for clustered data

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

>

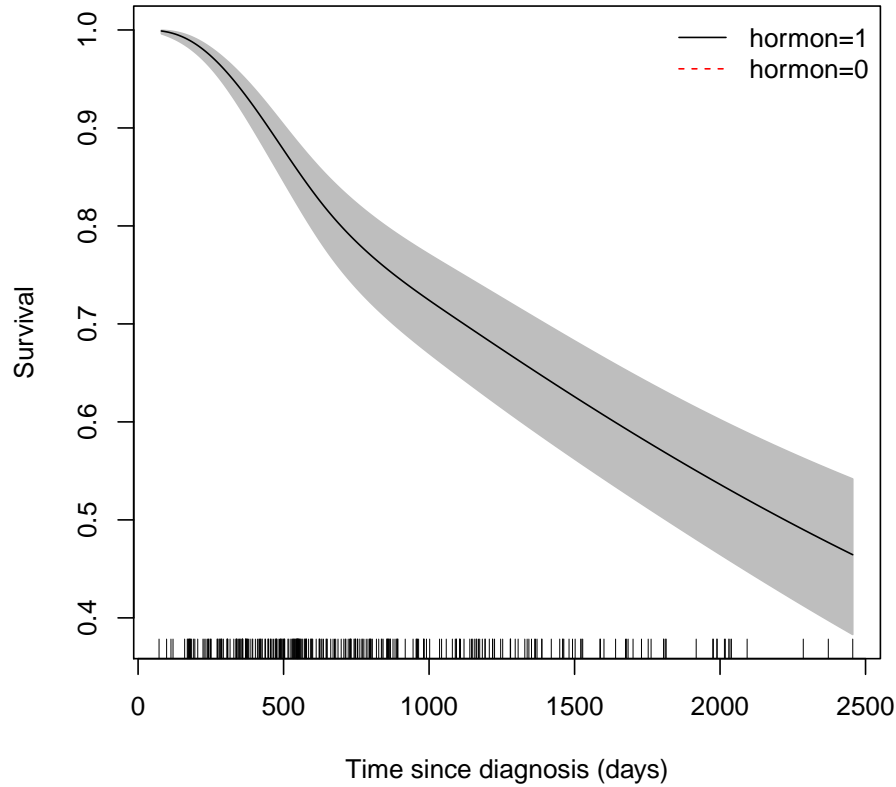
```

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.

```

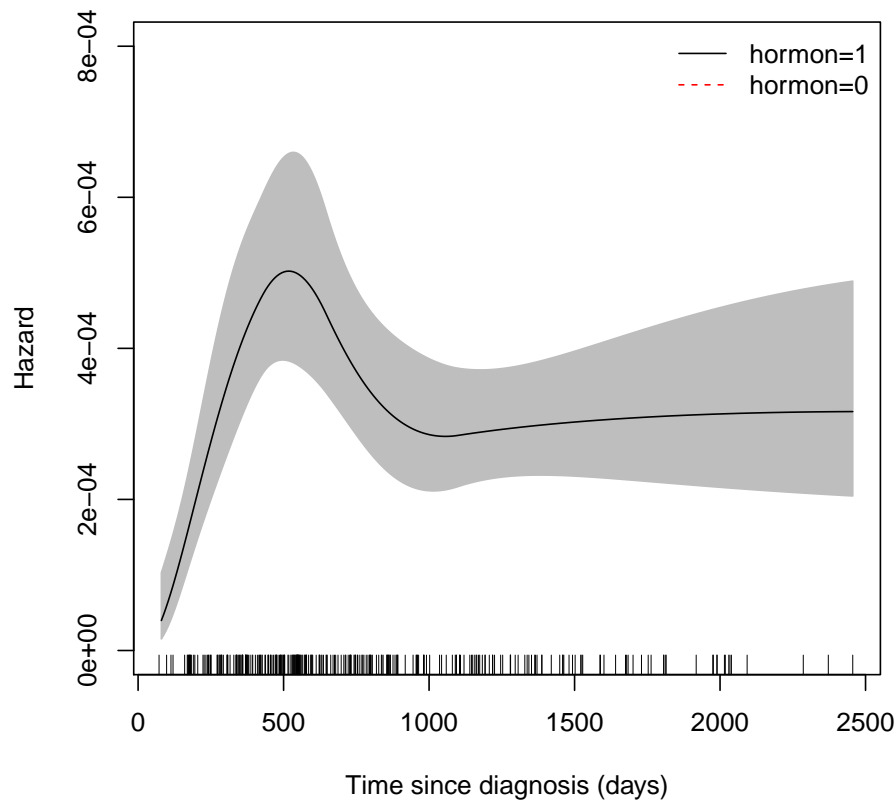
> # tms=seq(0,10,length=301)[-1]
> plot(fit,newdata=data.frame(hormon=1),
+      xlab="Time since diagnosis (days)")
> lines(fit, newdata=data.frame(hormon=0), col=2, lty=2)
> legend("topright", c("hormon=1", "hormon=0"),lty=1:2,col=1:2,bty="n")
>

```



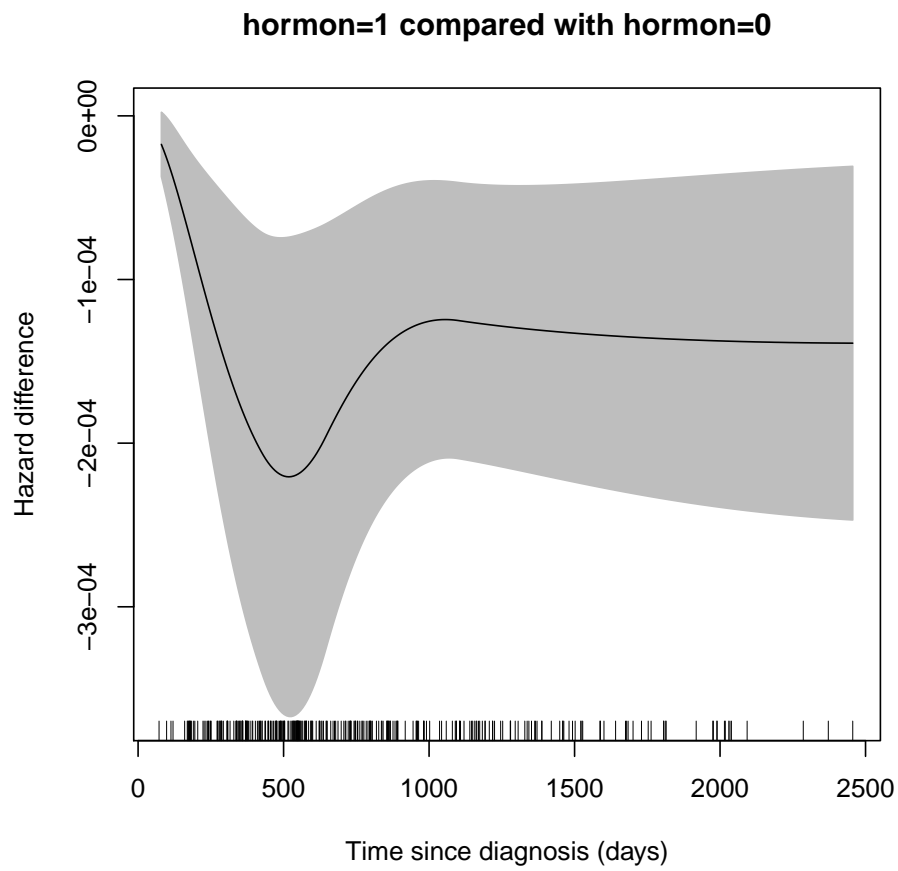
We can also calculate the hazards.

```
> # 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")
>
```



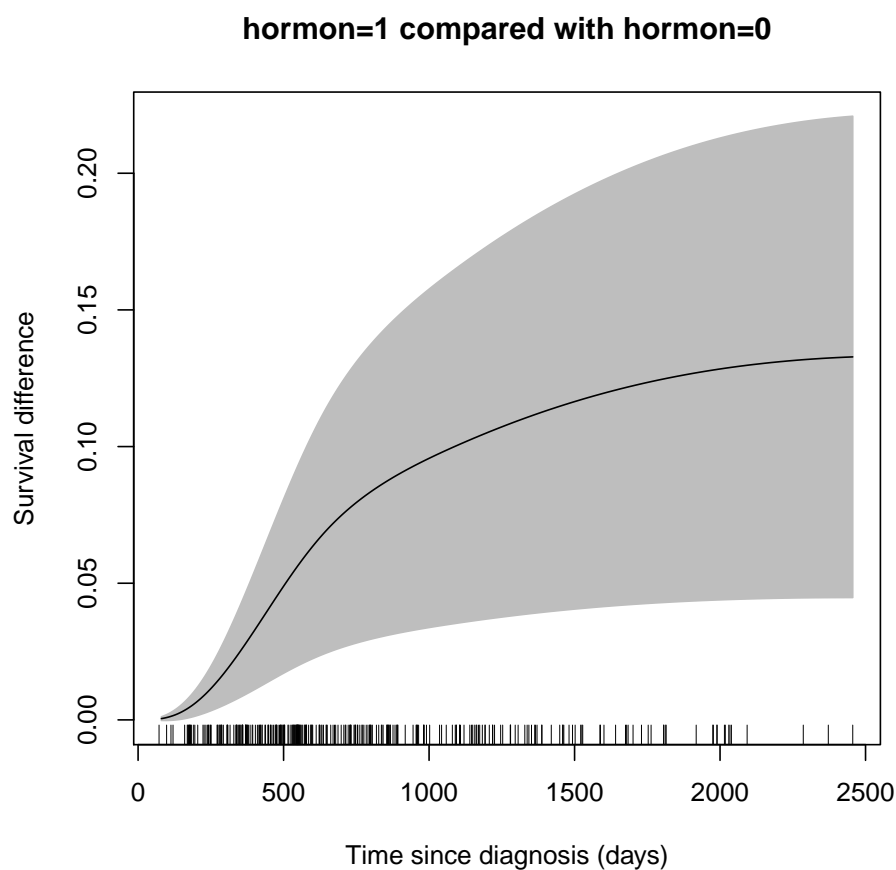
Usefully, we can also estimate survival differences and hazard differences. We define the survival differences using a reference covariate pattern using the `newdata` argument, and then define an exposed function which takes the `newdata` and transforms for the 'exposed' covariate pattern. As an example

```
> 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)")  
>
```





## 4. 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)
```

## 5. 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 0.8611043 0.8543120 0.8676049
2 0.7934962 0.7850420 0.8016613
3 0.6967834 0.6863629 0.7069354
4 0.8611043 0.8543120 0.8676049
5 0.8221508 0.8143498 0.8296592
6 0.8611043 0.8543120 0.8676049

>

```

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 0.6060950 0.6208812 0.5913493
2 0.5512519 0.5658461 0.5367744

```

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

	Estimate	lower	upper
1	0.5912976	0.6054689	0.5771838
2	0.5350852	0.5485409	0.5217474

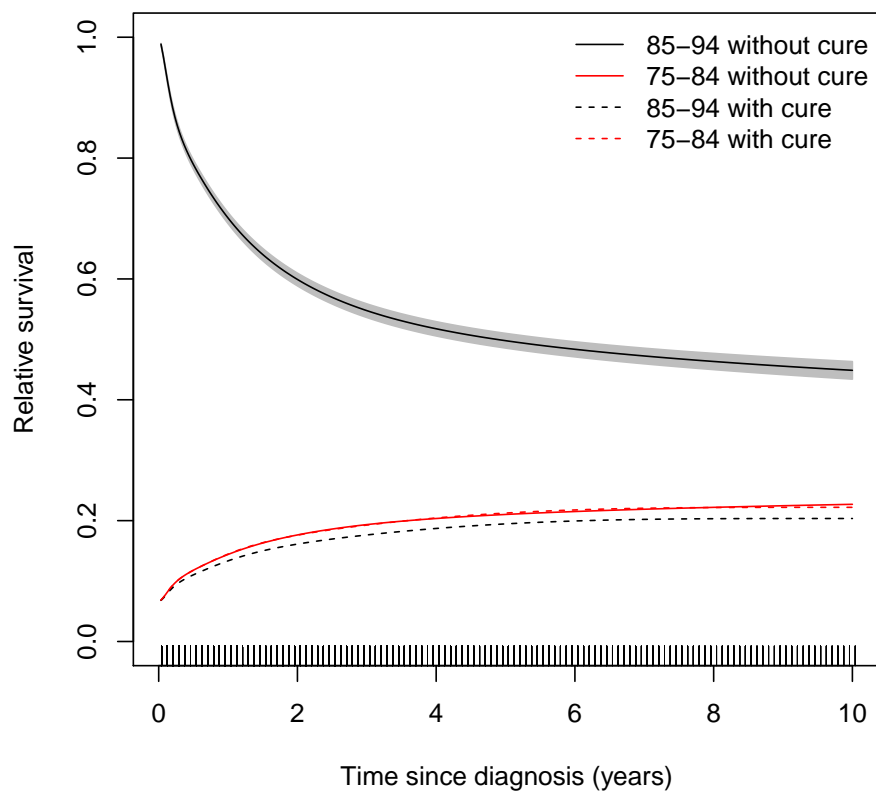
```
> predict(fit, newdata.eof, type="haz", se.fit=TRUE)
```

	Estimate	lower	upper
1	1.253896e-06	1.092820e-06	1.438714e-06
2	1.073307e-06	9.334257e-07	1.234150e-06

```
>
```

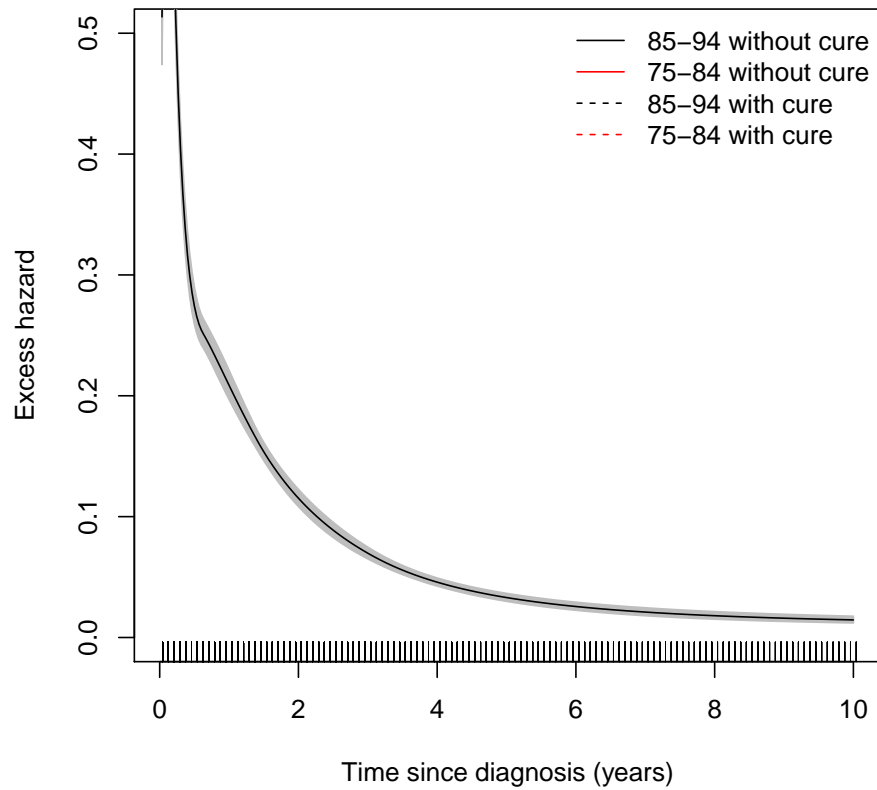
We can plot the predicted survival estimates:

```
> 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](mailto:mark.clements@ki.se)