Modelling survival data using flexible parametric survival models in Stata using `stpm3`: concepts and modelling choices.

Paul C Lambert[1,2]
[1]Cancer Registry of Norway, Norwegian Institute of Public Health, Oslo, Norway
[2]Department of Medical Epidemiology and Biostatistics, Karolinska Institutet, Stockholm, Sweden

Pre-conference course
2024 Northern European Stata Conference
9th September 2024

Cancer Registry of Norway

Karolinska Institutet

Outline

Outline 2

Introduction to the course

Introduction to the Course

- This course will give an overview of flexible parametric survival models.
- It will focus on the implementation in Stata using `stpm3`.
- There are different reasons why we model
  - Description
  - Prediction
  - Causality
- I will touch on all three of these today, but this course is not intended to cover theory/issues for all three of these areas.
- I want to demonstrate the advantages that `stpm3` can give you when modelling survival data.

Some questions for you

- Who would classify themselves as a (bio)statistician?
- Who would classify themselves as an epidemiologist?
- Who has used flexible parametric models previously?
  - Using `stpm2`
  - Using `stpm3`
  - In R (`flexsurv` or `rstpm2`)

Timetable

Time		Topic
11:00–12:00	-	Introduction to the course
	-	Introduction to flexible parametric survival models
	-	Various type of predictions
12:00–12:15	-	Break
12:15–13:00	-	Contrasts
	-	Incorporating non-linear functions and interactions
13:00–13:45	-	Lunch
13:45–15:00	-	Time-dependent effects
	-	Marginal contrast
	-	User-defined functions in standsurv
15:00–15:15	-	Break
15:15–16:15	-	Models on the log-hazard (and other) scales
	-	What to present?
	-	Convergence issues
16:15–16:30	-	Break
16:30–17:30	-	Some extensions
	-	Final questions and wrap-up

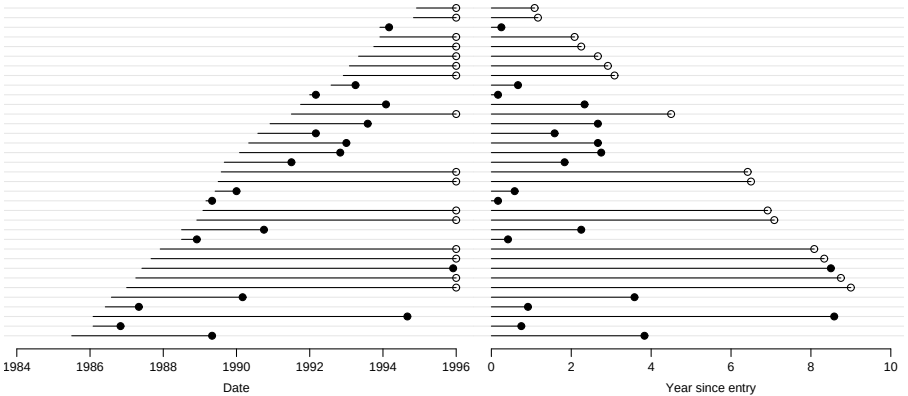
Computing

- There is not time for computer lab sessions today.
- I have included in the course meterial some exercises that I will run through today.
- You can choose to run code simultaneously with me or just watch as I explain what the code does.
- All my code is in Stata. There are some packages in R, where you can fit similar models (`Rstpm2`,`flexsurv`).
- The slides and code are avaiable on my website.
  `www.pclambert.net/courses/stpm3course`

Code to produce the slides

- I have made code and data available on the course website.
- you can use this if you want to replicate graphs or output.
- Note that most graphs have a footnote which gives the name of the do file which contains the code used to plot the graph.
- Note that I use a graph scheme Mark Rutherford developed for the 2nd edition of our book. This is available in the course files.

Censored survival data



Calendar time (left) and time from entry (diagnosis) in years (right)

Survival analysis

- We have censored data and all survival analysis methods need to account for this
- We need to define when time 0 is
  - e.g. date of diagnosis, date of randomization.
- We need to define what our event is and when it occured.
  - e.g. date of death from any cause.
- For those who did not have the event, we need to know the latest time we knew they were event free.
  - e.g. date of emmigration / end of study follow-up

Measures of interest

- We are interested in the proportion with/without an event. The survival or failure function.

$$S(t) = P(T > t) \qquad F(t) = 1 - S(t)$$

- We are interested in the rate of the event for those still at risk at time t.

$$h(t) = \lim_{dt \to 0} \frac{P(t \leq T < t + dt | T \geq t)}{dt}$$

- Hazard is a general term. If our outcome is death then the hazard rate is a mortality rate.
- We use hazards because
  - They give information about those still alive/at risk at different points in time.
  - Rates are a good way to deal with censored data
  - It is often convenient to make assumptions about rates, e.g. proportional hazards.

Measures of interest 2

- If we have $h(t)$ we can obtain $S(t)$.

$$S(t) = \exp\left(-\int_0^t h(u)du\right)$$

- If we have $S(t)$ we can obtain $h(t)$.

$$h(t) = -\frac{d}{dt}\ln\left[S(t)\right]$$

- I will mention cumulative hazards. This is just the 'amount' of hazard experienced by time t.

$$H(t) = \int_0^t h(u)du$$

Example datasets

- I will use some example datasets to illustrate the methods.
- These are all publically available datasets, so you can run all examples.
- The example datasets are
  - North West England breast cancer data
  - England and Wales under 50 breast cancer data
  - Rotterdam breast cancer data

North West England breast cancer data (breast_NW.dta)

- 14,823 women diagnosed with breast cancer in the North West region of England between 1996 and 1990 with follow-up to the end of 1995

```
. describe
Contains data from breast_NW.dta
 Observations:        14,823                  Ch28 Adult Breast 174, 175
    Variables:             9                  9 May 2024 13:24
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
ident           float   %9.0g                 Identifier
sex             byte    %8.0g      sexlb      Sex
dep             byte    %8.0g      caquinlb   GB quintile Carstairs score
datediag        int     %9.0g                 Date of diagnosis
agediag         float   %9.0g                 Age at diagnosis in years
dead            byte    %8.0g      deadlb     Vital status
survtime        float   %9.0g                 Follow-up time in years
dateexit        int     %9.0g                 Date of exit
agegrp          float   %9.0g      agelab     ICSS age groups
-------------------------------------------------------------------------------
Sorted by: sex
```

England and Wales breast cancer data aged < 50 (breast_EW50.dta)

- 24,883 women diagnosed with breast cancer in England and Wales between 1996 and 1990 with follow-up is to the end of 1995

```
. describe
Contains data from breast_EW50.dta
 Observations:        24,883                  Ch28 Adult Breast 174, 175
    Variables:            10                  9 May 2024 13:24
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
ident           float   %9.0g                 Identifier
sex             byte    %8.0g      sexlb      Sex
dep             byte    %8.0g      caquinlb   GB quintile Carstairs score
region          byte    %9.0g      regionlb   NHS Region 1998
datediag        int     %9.0g                 Date of diagnosis
agediag         float   %9.0g                 Age at diagnosis in years
dead            byte    %8.0g      deadlb     Vital status
survtime        float   %9.0g                 Follow-up time in years
dateexit        int     %9.0g                 Date of exit
agegrp          float   %9.0g      agelab     ICSS age groups
-------------------------------------------------------------------------------
Sorted by: sex
```

Rotterdam breast cancer data

- 2,982 patients with primary breast cancer whose records were included in the tumor bank at Rotterdam, The Netherlands.
- Follow-up time ranged from 1 to 231 months (median, 107 months).
- Various covariates are included including use of hormonal therapy (`hormon`), age at surgery (`age`), tumor size in 3 classes, (`size`), tumor grade 2 or 3 (`grade`), number of positive lymph nodes (`nodes`), progesterone receptors, fmol/l (`pr`), estrogen receptors, fmol/l (`er`).
- We can use overall (all cause) survival, cause-specific survival and relapse free survival.

Rotterdam breast cancer data

```
. describe
Contains data from rott3.dta
 Observations:         2,982                  Rotterdam breast cancer data (augmented with cause of death)
    Variables:            23                  21 Sep 2023 15:30
                                              (_dta has notes)
```

Variable name	Storage type	Display format	Value label	Variable label
pid	int	%4.0f		Patient ID number
year	int	%4.0f		Year of surgery
rf	float	%5.1f		Relapse free interval [mo]
rfi	byte	%3.0f		Relapse indicator
mf	float	%5.1f		Metastasis free [m]
mfi	byte	%3.0f	noyes	Metastasis status
os	float	%5.1f		Overall survival (m)
osi	byte	%8.0f	osi	Overall survival
age	byte	%3.0f		age at surgery
meno	byte	%4.0f	post	pre/post meno
size	byte	%9.0g	size	Tumour size, 3 classes (t)
grade	byte	%8.0g		Differentiation grade (diff)
nodes	byte	%8.0g		Number of positive nodes (nrpos)
pr	int	%8.0g		PgR (fmol/l)
er	int	%8.0g		ER (fmol/l)
hormon	byte	%5.0f	adjhormo	Hormonal therapy
chemo	byte	%8.0f	adjchemo	Chemo therapy
enodes	float	%9.0g		exp(-0.12 * nodes)
pr_1	double	%10.0g		log(pr + 1)
enodes_1	double	%10.0g		enodes^2
recent	byte	%9.0g	recent	year of surgery, dichotomized
dcause	byte	%13.0f	dcause	cause of death
cause	float	%12.0g	causelab	

Spline functions

Introduction to spline functions

- This is a very brief introduction to spline functions.
- You have probably come accross using polynomials to model a non-linear function, e.g. a quadratic function.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$$

- Splines are an alternative way to model a non-linear function.
- Like polynomials, we derive additional variables and include these in the linear predictor.

$$y_i = \beta_0 + \beta_1 z_{1i} + \beta_2 z_{2i} + \ldots + \beta_k z_{ki}.$$

- There different types of spline function and many different ways to calculate the z variables.

What are splines?

- Flexible mathematical functions defined by piecewise polynomials.
- The points at which the polynomials join are called knots.
- Constraints ensure the function is smooth.
- The most common splines used in practice are cubic splines.
- However, splines can be of any degree, n.
- Function is forced to have continuous 0^{th}, 1^{st} and 2^{nd} derivatives.
- Regression splines can be incorporated into any regression model with a linear predictor.
- Try these interative graphs
  http://pclambert.net/interactivegraphs/

Cubic splines

- Cubic spline functions can be used in any regression model by calculation of some extra variables.
- After defining K knots, $t_1, \ldots, t_K$ the spline function is

$$S(x) = \sum_{j=0}^{3} \beta_{0j} x^j + \sum_{i=4}^{K+4} \beta_{i3} (x^j - t_i)_+^3$$

- Note the "+" notation means that $u_+ = u$ if $u > 0$ and $u_+ = 0$ if $u \leq 0$.
- There will be $K + 4$ parameters (including the intercept) needed in the linear predictor.

Restricted Cubic Splines

- Restricted cubic splines can be fitted by creating $K-1$ derived variables, where K is the number of knots [1].
- For knots, $k_1, \ldots, k_K$, a restricted cubic spline function can be written

$$s(x) = \gamma_0 + \gamma_1 z_1 + \gamma_2 z_2 + \ldots + \gamma_{K-1} z_{K-1}$$

- The derived variables z_j (also known as the basis functions) are calculated as follows:

$$\begin{aligned} z_1 &= x \\ z_j &= (x-k_j)_+^3 - \lambda_j (x-k_{\min})_+^3 - (1-\lambda_j)(x-k_{\max})_+^3 \end{aligned}$$

where

Restricted Cubic Splines 2

$$\lambda_j = \frac{k_{\max} - k_j}{k_{\max} - k_{\min}}$$

- There are a variety of commands for calculating spline variables (sometimes called basis function). I suggest you use `gensplines`. You can install `gensplines` in Stata using,

Installing `gensplines`

```
. ssc install gensplines
```

- I now tend to use natural splines more than restricted cubic splines.
- Natural splines are calculated differently, but give identical fitted values when used in a model.
- `gensplines` calculates spline functions in a same way as the R `splines2` package[2].

Restricted Cubic Splines 3

- Try these interactive graphs to understand more details about continuity corrections and the number and location of knots
  http://pclambert.net/interactivegraphs/

Using splines with survival data

- Can be used to model non-linear effects of continuous covariates.
- We use them a lot for modelling of baseline (excess) hazard.
- Often better to use $\ln(t)$ rather than t.
- Boundary knots are usually placed at the minimum and maximum of the (log) event times.
- Interior knots placed at equally spaced centiles of the distribution of event times.
- When using Poisson regression we are modelling the log hazard, so our model is

$$\ln[h(t|\mathbf{x}_i)] = s\left(\ln(t)|\gamma, \mathbf{k}_0\right) + \mathbf{x}_i\boldsymbol{\beta}$$

where $s\left(\ln(t)|\gamma, \mathbf{k}_0\right)$ is a restricted cubic spline function with knots, $\mathbf{k}_0$

Introduction to flexible parametric models

Flexible Parametric Survival Models

In this lecture we will,

- Look at standard parametric survival models.
- Extend these to be more 'flexible' using spline functions.
- See a number of advantages in terms of model fit.
- See a number of advantages in terms of predictions.
- Introduce you to the `stpm3` and `standsurv` commands.

What is a parametric survival model?

- In a parametric survival model the survival function can be expressed as a mathematical function of follow-up time and a set of parameters.
- Due to the mathematical relationship between the hazard, survival and density functions, there is also a mathematical function for all these functions.
- There are also parameters for the effect of covariates.
- These parameters are estimated when you fit a model.

Note that a Cox model is a *semi-parametric* model as a parametric function is not estimated for the hazard/survival/density functions. It only directly estimates the (relative) effect of covariates.

Examples of Parametric Models

- The most simple model is the exponential model.

$$h(t) = \lambda, \quad S(t) = \exp(-\lambda t), \quad f(t) = \lambda \exp(-\lambda t)$$

- The hazard rate is constant over time.
- The Weibull model is a commonly used survival model.

$$h(t) = \lambda\gamma t^{\gamma-1}, \quad S(t) = \exp(-\lambda t^{\gamma}), \quad f(t) = \lambda\gamma t^{\gamma-1}\exp(-\lambda t^{\gamma})$$

- The hazard rate is a monotonic function, i.e. it cannot have a turning point.

Other common parametric survival models include the lognormal, log-logistic, Gompertz, gamma, generlized gamma

The Weibull model

$$h(t) = \lambda\gamma t^{\gamma-1}, \quad S(t) = \exp(-\lambda t^{\gamma}), \quad f(t) = \lambda\gamma t^{\gamma-1}\exp(-\lambda t^{\gamma})$$

- We are interested in how the survival/hazard functions vary between individuals (covariate patterns).
- We have choices in how we model covariates,

Proportional hazards

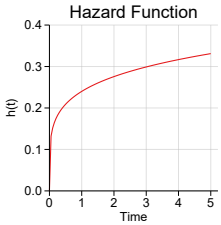
$$h(t|\mathbf{x}_i) = \lambda\gamma t^{\gamma-1}\exp(\mathbf{x}_i\boldsymbol{\beta})$$

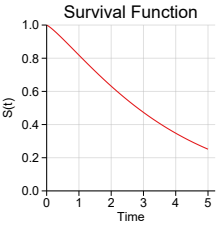
Accelerated Failure

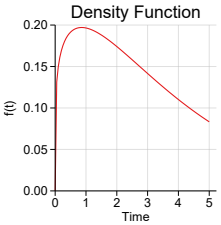
$$S(t|\mathbf{x}_i) = \exp(-\lambda\exp(-\gamma\mathbf{x}_i\boldsymbol{\beta})t^{\gamma})$$

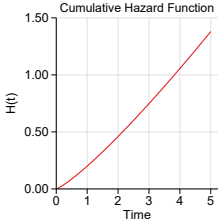
- We concentrate on hazard / cumulative hazard models.

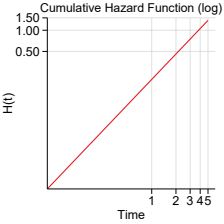
Various Weibull model functions, $\lambda = 0.2$, $\gamma = 1.2$











Graph code in Weibull_example_functions.do

Why use Parametric Survival Models?

- Parametric Models have advantages for
  - Understanding.
  - Prediction (including complex predictions, e.g. marginal effects)
  - Extrapolation.
  - Quantification (e.g., absolute and relative measures of risk).
  - Modelling time-dependent effects.
  - All cause, cause-specific or relative survival.
  - etc etc
- However, standard parametric models are limited in that they impose a certain shape on hazard/survival functions.
- None of the standard models may fit well to your data.

The Cox model I

- Web of Science: over 38,659 citations (April 2023).

$$h_i(t|\mathbf{x}_i) = h_0(t)\exp(\mathbf{x}_i\beta)$$

- Estimates (log) hazard ratios.
- Advantage: The baseline hazard, $h_0(t)$ is not estimated from a Cox model.
- Disadvantage: The baseline hazard, $h_0(t)$ is not estimated from a Cox model.

The Cox model II

- The crucial assumption of the Cox model is that the estimated parameters are not associated with time, i.e., we assume proportional hazards.
- If you are only interested in the relative effect of a covariate on the hazard rate and the assumption of proportional hazards is reasonable, then the Cox model is probably the most appropriate model. In other situations alternative models may be more appropriate.
- However, whenever we estimate a relative effect we should ask "relative to what?"
- Discussion about whether hazard ratios are good causal measures[3, 4, 5, 6].

Quote from Sir David Cox (Reid 1994 [7])

Reid "What do you think of the cottage industry that's grown up around [the Cox model]?"

Cox "In the light of further results one knows since, I think I would normally want to tackle the problem parametrically. ... I'm not keen on non-parametric formulations normally."

Reid "So if you had a set of censored survival data today, you might rather fit a parametric model, even though there was a feeling among the medical statisticians that that wasn't quite right."

Cox "That's right, but since then various people have shown that the answers are very insensitive to the parametric formulation of the underlying distribution. And if you want to do things like predict the outcome for a particular patient, it's much more convenient to do that parametrically."

Flexible parametric models: basic idea

- Consider a Weibull survival curve.

$$S(t) = \exp\left(-\lambda t^{\gamma}\right)$$

- If we transform to the log cumulative hazard scale.

$$\ln\left[H(t)\right] = \ln[-\ln(S(t))]$$

$$\ln\left[H(t)\right] = \ln(\lambda) + \gamma \ln(t)$$

- This is a linear function of $\ln(t)$
- Introducing covariates gives

$$\ln\left[H(t|\mathbf{x}_i)\right] = \ln(\lambda) + \gamma \ln(t) + \mathbf{x}_i\boldsymbol{\beta}$$

- Rather than assuming linearity with $\ln(t)$ flexible parametric models use natural splines for $\ln(t)$.

Flexible parametric models: incorporating splines

- We thus model on the log cumulative hazard scale.

$$\ln[H(t|\mathbf{x}_i)] = \ln\left[H_0(t)\right] + \mathbf{x}_i\boldsymbol{\beta}$$

- This is a proportional hazards model.
- Natural cubic splines with knots, $\mathbf{k_0}$, are used to model the log baseline cumulative hazard.

$$\ln[H(t|\mathbf{x}_i)] = \eta_i(t) = s\left(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0\right) + \mathbf{x}_i\boldsymbol{\beta}$$

- For example, with 4 knots we can write

$$\ln\left[H(t|\mathbf{x}_i)\right] = \eta_i(t) = \underbrace{\gamma_0 + \gamma_1 z_{1i} + \gamma_2 z_{2i} + \gamma_3 z_{3i}}_{\text{log baseline cumulative hazard}} + \underbrace{\mathbf{x}_i\boldsymbol{\beta}}_{\text{log hazard ratios}}$$

Survival and hazard functions

- Log cumulative hazard functions are not that interesting, but We can transform to the survival scale

$$S(t|\mathbf{x}_i) = \exp(-\exp[\eta_i(t)]))$$

- The hazard function is a bit more complex.

$$h(t|\mathbf{x}_i) = \frac{ds\left(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0\right)}{dt} \exp[\eta_i(t)]$$

- This involves the derivatives of the spline functions.
- However, these are easy to calculate.
- Most software will also supply derivatives.

Incorporating spline functions

- The linear predictor is

Linear Predictor

$$\eta_i(t) = s\left(\ln(t)|\boldsymbol{\gamma}, \mathbf{k_0}\right) + \mathbf{x}_i\boldsymbol{\beta}$$

- For models on the log cumulative hazard scale.

Survival and hazard functions

$$S(t) = \exp\left(-\exp\left[\eta_i(t)\right]\right) \qquad h(t) = \frac{ds(\ln(t)|\boldsymbol{\gamma}, \mathbf{k_0})}{dt}\exp\left[\eta_i(t)\right]$$

- Feed these into the likelihood, evaluated at time t_i.

$$\ln L_i = d_i \ln\left[h(t_i)\right] + \ln\left[S(t_i)\right]$$

Incorporating spline functions 2

- This can be extended to delayed entry (left tunctation).

$$\ln L_i = d_i \ln [h(t_i)] + \ln [S(t_i)] - \ln [S(t_{0i})]$$

- As we can write the (log) hazard and survival functions analytically, these models are fast to fit.
- This is not the case when using splines for the linear predictor on the log hazard scale (but less of an issue when modern, fast computers).

A history of software for flexible parametric models

- Patrick Royston wrote `stpm` around 2000 [8].
- I wrote `stpm2` around 2007 [9].
  - Relative survival models
  - Better predictions
- `stpm3` released in 2023.

Why a new command, stpm3*

- stpm2 written before Stata included factor variables
- Use better basis functions for spline functions (natural splines).
- Make predictions and contrasts easier.
- Use frames for predictions.
- Include splines on log hazard scale.
- Include functional forms of covariates in linear predictor (extended functions).
- Make marginal/standardized predictions much, much easier.
  - This was the main reason
- More in Mata (sometimes Python) for speed improvements.

stpm3 - most important syntax*

stpm3 [*(extended) varlist*], [*options*]

- scale() - compulsory option - We will mainly be using lncumhazard and lnhazard).
- df(#) - the number of spline variables for the baseline. Knots are placed at evenly distrributed centiles of the distribution of log event times.
- eform exponentiate coefficients in first equation - gives hazard ratios for a proportional hazards model.
- knots(# # ...) - user defined knots positions
- tvc(varlist) - variables with time-dependent effect - supports factor variables and extended functions.
- dftvc(varlist) - number of spline variables for time-dependent effects.
- knotstvc() - knot positions for time-dependent effects.

Fitting a proportional hazards model

- Example: 24,883 women aged ≤ 50 diagnosed with breast cancer in England and Wales 1986-1990.
- Compare five deprivation groups from most affluent to most deprived.
- No information on cause of death, but given their age, most women who die will die of their breast cancer.

Proportional hazards models

```
. stcox i.dep,
. stpm3 i.dep, df(5) scale(lncumhazard) eform
```

- The `df(5)` option implies using 4 internal knots and 2 boundary knots at their default locations.
- The `scale(lncumhazard)` requests the model to be fitted on the log cumulative hazard scale.

Cox Model

```
. stcox i.dep, nolog noshow
Cox regression with Breslow method for ties
No. of subjects =     24,883                          Number of obs = 24,883
No. of failures =      7,365
Time at risk    = 104,613.63
                                                      LR chi2(4)    =  62.25
Log likelihood = -73291.085                           Prob > chi2   = 0.0000
```

_t	Haz. ratio	Std. err.	z	P>|z|	[95% conf.	interval]
dep						
2	1.048952	.0354078	1.42	0.157	.9818003	1.120698
3	1.105275	.0383099	2.89	0.004	1.032682	1.18297
4	1.213043	.0437555	5.35	0.000	1.130245	1.301906
mostdep	1.309803	.051344	6.88	0.000	1.212939	1.414402

- Only gives relative effects, i.e. hazard ratios.

Flexible parametric proportional hazards model

```
. stpm3 i.dep, scale(lncumhazard) df(5) eform nolog
                                                Number of obs = 24,883
                                                Wald chi2(4)  =  63.40
Log likelihood = -22496.684                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
2	1.048989	.0354091	1.42	0.157	.9818344	1.120737
3	1.105245	.0383089	2.89	0.004	1.032655	1.182939
4	1.213022	.0437548	5.35	0.000	1.130226	1.301884
mostdep	1.309804	.0513441	6.88	0.000	1.21294	1.414403
time						
_ns1	-20.5192	.7302075	-28.10	0.000	-21.95038	-19.08802
_ns2	3.829793	.3917803	9.78	0.000	3.061918	4.597668
_ns3	-1.074997	.0182917	-58.77	0.000	-1.110849	-1.039146
_ns4	-.601024	.0128829	-46.65	0.000	-.6262739	-.575774
_ns5	-.3340791	.0109536	-30.50	0.000	-.3555478	-.3126103
_cons	-1.14467	.023338	-49.05	0.000	-1.190412	-1.098928

```
Note: Estimates are transformed only in the first equation.
```

New variables created

The baseline linear predictor (xb0), and survival functions can be calculated using the _ns variables and the parameters.

```
. list id _t _d _ns* in 1/5, noobs abb(7)
```

ident	_t	_d	_ns1	_ns2	_ns3	_ns4	_ns5
351119	5	0	0	0	0	0	-.00047506
351638	5	0	0	0	0	0	-.00047506
351665	1.191	1	.05313304	.10663582	.83830265	.00229822	0
351723	1.673	1	.01334769	.02707449	.80412155	.15583537	0
351876	5	0	0	0	0	0	-.00047506

```
. gen xb0 = _b[time:_ns1]*_ns1 + _b[time:_ns2]*_ns2 + _b[time:_ns3]*_ns3 + ///
>           _b[time:_ns4]*_ns4 + _b[time:_ns5]*_ns5 + _b[time:_cons]
. gen H0 = exp(xb0)
. gen S0 = exp(-H0)
. list id _t _d xb0 H0 S0 in 1/5, noobs
```

ident	_t	_d	xb0	H0	S0
351119	5	0	-1.144511	.3183795	.7273268
351638	5	0	-1.144511	.3183795	.7273268
351665	1.191	1	-2.729079	.0652794	.9368057
351723	1.673	1	-2.272954	.1030075	.9021202
351876	5	0	-1.144511	.3183795	.7273268

Proportional hazards models

- Do not try and interpret the coefficients of the spline (`_ns`) parameters.
- Together they give us the hazard/survival functions etc.
- FPM and Cox hazard ratios and 95% CIs are very similar.
- I have yet to find an example of a proportional hazards model, where there is a large difference in the estimated hazard ratios.
- Actually, hazard ratios are usually fairly robust to incorrect specification of baseline hazard, though there are some exceptions, e.g. when there is differential follow-up between covariate patterns.
- It is, of course, preferable to model the baseline hazard well! Particularly if you are intersted in absolute risks.
- In `stpm2` we use `scale(hazard)`. In `stpm3` we use `scale(lncumhazard)` as we need to distinguish between `scale(lncumhazard)` and `scale(lnhazard)` models.

Modelling the baseline

- We have used `df(5)` to model the baseline.
- This means there will be 6 knots (4 internal).
- Boundary knots are at the minimum and maximum event times (by default).
- The remaining knots placed at equally spaced percentiles of the event times.
- For, `df(5)` these are 20^{th}, 40^{th}, 60^{th} and 80^{th} percentiles
- Note we use percentiles of event times (i.e. excluding censored observations), so with lots of early events, there will be more knots early on.
- We will discuss issues around selecting the number and location of the knots later.

Flexible parametric models: predictions

Predictions from Parametric Models

- We want to predict various measures after fitting a model. E.g. survival functions, hazard functions.
- We may want contrasts, e.g. differences in survival, hazard ratios.
- An advantage of parametric models is that it is easy to predict survival, hazard functions etc, for any covariate pattern at any point in time.
- Many predictions will be functions of time.
- It is often useful to specify the times at which you want predictions rather than predict at the observed times.
- We are interested in conditional predictions and marginal predictions.
- We will discuss the differences between these and then show you how to do them in Stata.

Different types of predictions

- We want to predict different types of function.
  - hazard functions, survival functions etc
- There three main types of predictions we may be interested in.
  - Predict at observed values of covariates.
  - Predict at user specified values of covariates
  - Take average of predictions (marginal/standardized effects)
- We may also be interested in contrasts in the above, e.g. when comparing unexposed vs exposed.
- Difference choices for time.
  - Predict at observed event/censoring times (`_t`).
  - Predict at single time point for all subjects (e.g. 5 years).
  - Predict at user specified time values (e.g. 100 values between 0 and 10).

Commands for conditional and marginal predictions

- After fitting an `stpm3` model,
  - For conditional predictions, use `predict`.
  - For marginal predictions, use `standsurv`.
- I will initially show some simple examples, but the `predict` and `standsurv` commands are very powerful with many options.
- Note that `stpm2` the `predcict` command had a `meansurv` option to estimate marginal survival. In `stpm3` you have to use `standsurv` for marginal predictions.

Conditional predictions

- Our model will usually contain several covariates, e.g. (age, sex and treatment).
- We may want to predict what the survival function is for a 65 year old male taking a particular treatament.
- We refer to this as a conditional prediction in that it is the predicted survival function conditional on specific values of *all* covariates included in the model.
- We may want to compare betweeen different covariate patterns, e.g., males aged 65 on Treatment A and males aged 65 on Treatment B.
- With many covariates there are many different combinations of predictions.

Conditional predictions 2

- We would write a prediction of the survival function as,

$$\widehat{S}(t|\mathbf{x}^*, \widehat{\beta})$$

- This is for covariate pattern, $\mathbf{x}^*$ and estimated model parameters, $\widehat{\beta}$.
- We need to specify the values of the covariates we want predictions for.
- If a covariate is in the model be have to specify a value for it.
- We can choose the value(s) of time to predict at. If plotting then usually 100 points is sufficient for the function to appear smooth.

Model is a function of deprivation and age

```
. stpm3 i.dep agediag, scale(lncumhazard) df(5) eform nolog
                                                Number of obs =  24,883
                                                Wald chi2(5)  =  148.15
Log likelihood = -22455.915                     Prob > chi2   =  0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
2	1.046464	.0353249	1.35	0.178	.9794686	1.118041
3	1.098292	.0380768	2.70	0.007	1.026142	1.175515
4	1.203149	.0434151	5.13	0.000	1.120996	1.291322
mostdep	1.290682	.0506453	6.50	0.000	1.19514	1.393862
agediag	.9813256	.0020177	-9.17	0.000	.9773788	.9852882
time						
_ns1	-20.52409	.7301018	-28.11	0.000	-21.95506	-19.09312
_ns2	3.828002	.3917218	9.77	0.000	3.060241	4.595762
_ns3	-1.076697	.0183083	-58.81	0.000	-1.112581	-1.040813
_ns4	-.6020499	.0129004	-46.67	0.000	-.6273342	-.5767656
_ns5	-.3347711	.0109738	-30.51	0.000	-.3562794	-.3132628
_cons	-.3407907	.0902046	-3.78	0.000	-.5175885	-.1639929

```
Note: Estimates are transformed only in the first equation.
```

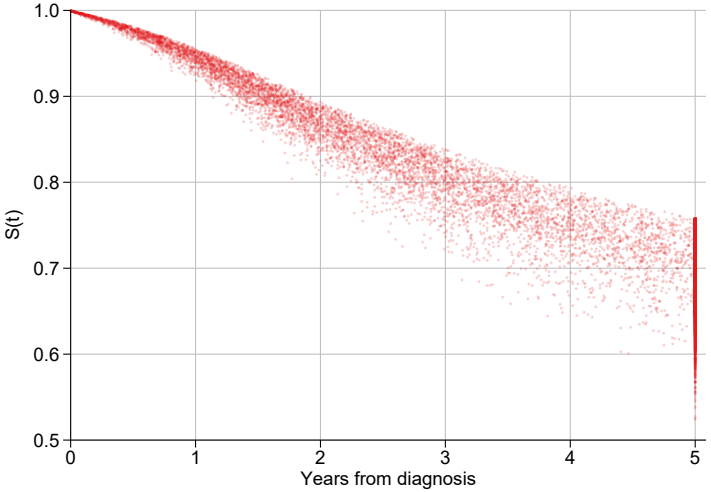
Predict at observed values of covariates.

- The default is to predict at values of _t.

```
. predict s, survival
. twoway (scatter s _t, mcolor(%20) msize(vsmall) mlwidth(none)),  ///
>         ylabel(,format(%3.1f))                                   ///
>         xtitle("Years from diagnosis")                           ///
>         ytitle("S(t)")                                           ///
>         note("Graph code in ${dofile}",                          ///
>               size(vsmall) span color(gs3%50))
```

- I very rarely predict at values of _t and when I do it is usually by mistake.

Predict at observed values of covariates. 2



Graph code in stpm3_predictions

The `timevar()` and `at()` options.

- Around 99% of the time that I use predict I use the `timevar()` option and multiple `at()` options.
- The `timevar()` option specified the time(s) we want predictions at. For example
  - `timevar(0 10, step(0.1))` or `timevar(0 10, n(101))`
- The `at()` options enable multiple predictions from one call to the predict command. This is useful as we can then peform contrasts between different covariate patterns.
  - E.g. Predictions for selected ages
  - `at1(age 40 dep 1) at2(age 60 dep 1) at3(age 80 dep 1)`

Using frames

- Stata introduced frames in Stata 16.
- Frames enable us to have multiple datasets in memory.
- Our predictions for plotting/tabulation are often a different size to our analysis data.
- In `stpm3` I strongly recommend you (nearly always) predict to a frame.
- You can merge predictions to a new frame.
- In `stpm2` predictions were usually 'attached' on the side of our analysis data. This caused confusion and mistakes.

Predict and plot hazard function

```
. predict h40_dep1 h40_dep5, hazard ci frame(hazpred, replace) per(1000) ///
>                            timevar(0.1 5, step(0.1))                    ///
>                            at1(dep 1 agediag 40)                        ///
>                            at2(dep 5 agediag 40)
Predictions are stored in frame - hazpred
.
. frame hazpred {
.   twoway (line h40_dep1 h40_dep5 tt),                                     ///
>          xtitle("Years from diagnosis")                                   ///
>          ytitle("Mortality rate (per 1000 py)")                           ///
>          ylabel(0(20)100)                                                 ///
>          title("Predicted hazard rate for a 40 year old")                 ///
>          legend(order(1 "Least deprived" 2 "Most deprived"))              ///
>          note("Graph code in ${dofile}", size(vsmall) span color(gs3%50))
. }
```

- Predictions saved to a new frame, `hazpred`.
- The `timevar()` option specifies what times to predict at. Between 0.1 amd 5 in steps of 0.1.
- The `per(#)` option multiplies the predicted values by #.
- Note you can have multiple `at` options.

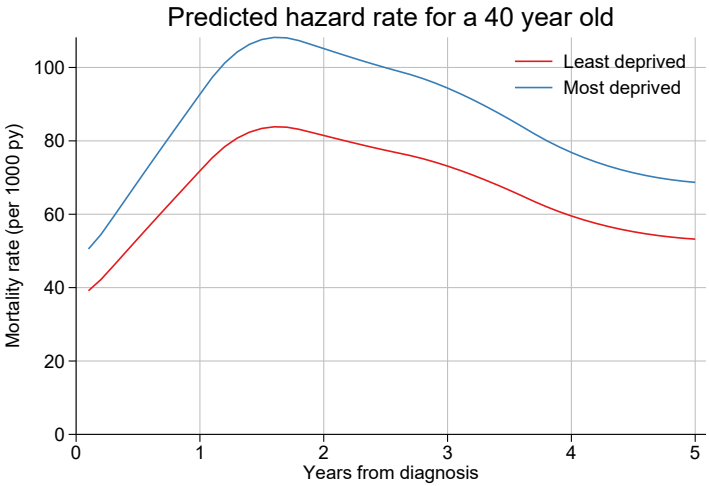
List hazard predictions

```
. frame hazpred: list  in 1/10, noobs
```

tt	h40_dep1	h40~1_lci	h40~1_uci	h40_dep5	h40~5_lci	h40~5_uci
.1	39.152223	35.274082	43.456739	50.533082	45.188214	56.510141
.2	42.181874	38.592435	46.105162	54.443398	49.379116	60.027069
.3	45.8755	42.411563	49.622351	59.210695	54.20825	64.674777
.4	49.647675	46.26021	53.283191	64.079376	59.070154	69.513385
.5	53.40641	50.051484	56.986215	68.930708	63.857744	74.406678
.6	57.135303	53.758485	60.724235	73.743525	68.542869	79.338777
.7	60.8351	57.370058	64.509424	78.518787	73.117777	84.318754
.8	64.510701	60.882671	68.354928	83.262821	77.582753	89.358743
.9	68.16762	64.298758	72.269271	87.98274	81.942702	94.467993
1	71.810984	67.625452	76.255571	92.685167	86.205559	99.651812

- The default name of time variable is `tt`. You can change this.
- With the `ci` option suffixes `_lci` and `_uci` are added to the variable name.

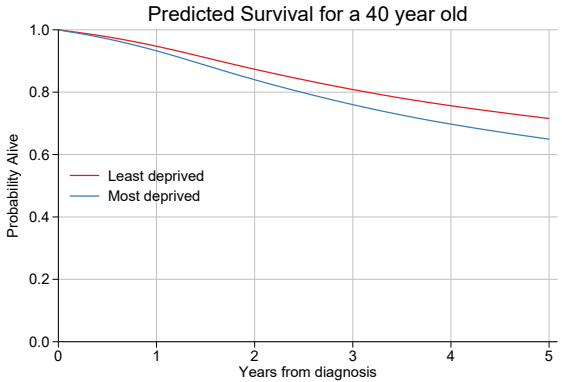
Predicted hazard function



Graph code in stpm3_predictions

Predicted survival function

```
    . predict S40_dep1 S40_dep5, survival ci frame(survpred, replace)        ///
>                           timevar(0 5, step(0.1))                        ///
>                           at1(dep 1 agediag 40) at2(dep 5 agediag 40)
Predictions are stored in frame - survpred
```



Graph code in stpm3_predictions

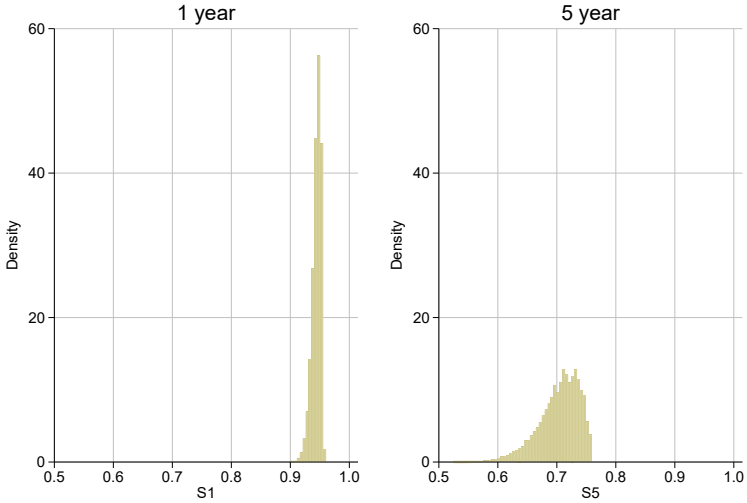
Specifying single time points

- Predict survival at 1 and 5 years.

```
. gen t1 = 1
. gen t5 = 5
. predict S1 S5, surv frame(Spred1_5, replace) ///
>                at1(., attimevar(t1))         ///
>                at2(., attimevar(t5))
Predictions are stored in frame - Spred1_5
.
. frame Spred1_5 {
.   hist S1, name(S1, replace) title("1 year") xlab(,format(%3.1f)) wid(0.005)
(bin=12, start=.90081675, width=.005)
.   hist S5, name(S5, replace) title("5 year") xlab(,format(%3.1f)) wid(0.005)
(bin=47, start=.52427243, width=.005)
.   graph combine S1 S5, xcommon ycommon ///
>   note("Graph code in ${dofile}", size(vsmall) span color(gs3%50))
. }
```

- Predictions saved to a new frame.
- A . (dot) in `at` options predicts at observed values of covariates.
- A separate time variable for each `at` option, using `attimevar()`.

Specifying single time points 2



Graph code in stpm3_predictions

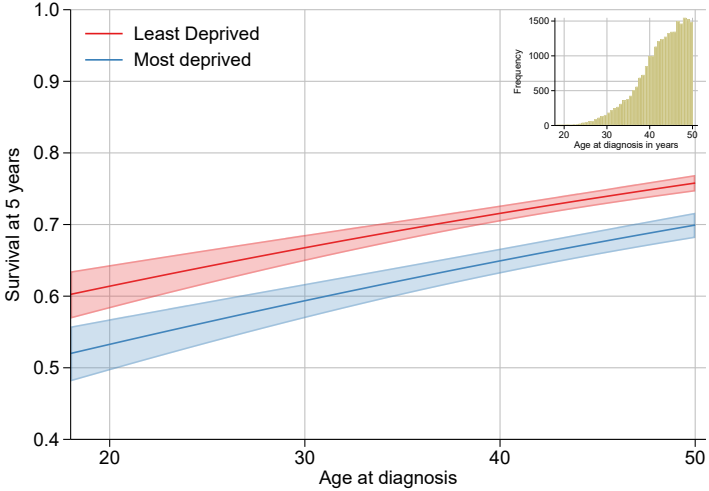
Predict over range of a variable

```
. frame create agepred
. frame agepred {
.   range agediag 18 50 33
Number of observations (_N) was 0, now 33.
.   gen dep = .
(33 missing values generated)
.   gen t5 = 5
.   predict S_dep1 S_dep5, surv ci timevar(t5) merge   ///
>         at1(dep 1, obsvalues) at2(dep 5, obsvalues)
. }
.
. frame agepred {
.   twoway (line S_dep1 S_dep5 agediag)                                    ///
>         (rarea S_dep1_lci S_dep1_uci agediag, pstyle(p1line) color(%30))  ///
>         (rarea S_dep5_lci S_dep5_uci agediag, pstyle(p2line) color(%30))  ///
>         , legend(order(1 "Least Deprived" 2 "Most deprived")              ///
>         cols(1) pos(11))                                                  ///
>         ylabel(0.4(0.1)1, format(%3.1f))                                  ///
>         ytitle("Survival at 5 years")                                     ///
>         xtitle("Age at diagnosis")                                   ///
>         note("Graph code in ${dofile}", size(vsmall) span color(gs3%50))
. }
. grinset t=5 r=5: hist agediag, freq
(bin=43, start=18.207001, width=.73934883)
```

Predict over range of a variable II

- Create a new frame containing values you want to predict at.
- Create values you want to predict at (using `range` here).
- Use `merge` as we are predicting within a frame. These are 'out-of-sample' predictions.
- `obsvalues` option predicts at new 'observed' values of `agediag` (in frame `agepred`). A bit confusing, but these are the 'observed' values in our new frame.
- `grinset` is a user written command that places a mini-graph on an existing graph.
- We are assuming linearity for age here, we will relax this later.

Predict over range of a variable III



Graph code in stpm3_predictions

Marginal predictions

- For marginal predictions we are interested in the average (survival) in a (study) population.
- For example, we could estimate the average (marginal) survival.

$$\widehat{S}_m(t) = \frac{1}{N}\sum_{i=1}^{N}\widehat{S}(t|\mathbf{x_i}, \widehat{\beta})$$

- This is averaged over all study subjects.
- If calculated for all individuals in the study, this should be similar to the corresponding Kaplan-Meier estimate.
- Later in the course we will 'manipulate' exposures. For example, we predict as if everyone was exposed or everyone was unexposed.

The mean covariate method

- Note that a marginal estimate is different from using the mean value of all covariates [10],
- Some software (e.g., `stcurve` in Stata) uses the mean covariate method.
- Using the mean coviatate method, we obtain

$$\widehat{S}(t|\mathbf{x}^*, \widehat{\beta})$$

where $\mathbf{x}^* = (\bar{x}_1, \bar{x}_2, \bar{x}_3, \ldots)$

- This is the survival of an 'average' individual, who happens to have the average values of all covariates.
- Problem with categorical covariates. May be someone with a proportion of each stage and who is 50% male.

Overall Marginal Survival

- Use `standsurv` for marginal predictions.

```
. range tt 0 5 101
(24,782 missing values generated)
. standsurv , survival at1(.) timevar(tt) ci atvar(Sm) frame(msurv, replace)
```

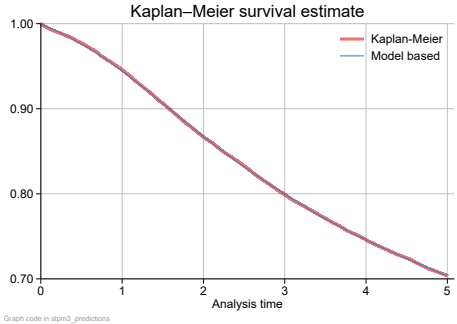
- The `standsurv` command obtains averages of various measures.
- We are obtaining the average survival (`survival` option).
- A variable, `tt`, has been created for the times we want to predict at. This is passed to `standsurv` using `timevar(tt)`.
- Results are saved to a new frame, `msurv`.
- The `at1(.)` option means that we will average over observed values of covariates for each individual.
- The variable containing the prediction will be named, `Sm`.

Overall Marginal Survival 2

- Here, `standsurv` will take the average of 24,883 predicted survival functions.
- Later in the course we will use the `at()` options to force covariates to take specific values.

```
    . sts graph, noshow plotopts(lwidth(*3) lcolor(%60)) legend(on) ///
>     note("Graph code in ${dofile}", size(vsmall) span color(gs3%50))
. frame msurv: addplot: line Sm tt, lcolor(%100) ///
>     legend(order(1 "Kaplan-Meier" 2 "Model based"))
```

Overall Marginal Survival 3



- Agreement, so good it is hard to see differences in lines.

Marginal predictions using `stpm3km`

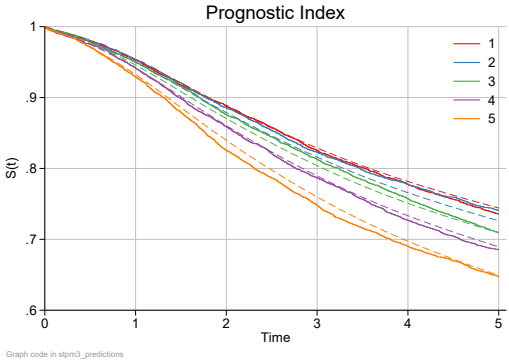
- Often the marginal estimate will be similar to the Kaplan-Meier estimate, even when the model is poor.
- It can be more usful to compare marginal model estimates and Kaplan-Meier estimates in subgroups.
- These subgroups could be based on a prognostic index, or a covariate in (or not in) the model.
- Thus with 5 groups, 5 separate Kaplan-Meier estimates are calculated and the marginal survival is calculated separately in each group.
- The `stpm3km` command makes this easy. `stpm3km` runs the `standsurv` command.

Marginal predictions using `stpm3km` 2

- The `stpm3km` command essentially calls `standsurv` with the following,
  `standsurv if` *varname*`==1, ...`
  `standsurv if` *varname*`==2, ...`
  ⋮
- An alternative way to do this is to use the `over` option. `standsurv , surv over(`*varname*`) ...`
- The key point is the averages are taken *within* groups.
- Note that that we will use `standsurv` to average over the same covariate distribution later. Using the `over()` option or using `if` statements in this way should not be used in that situation as we are averaging over `different` covariate distributions.

Simple use of stpm3km: prognostic index

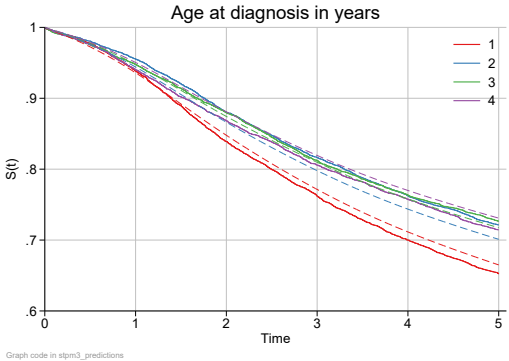
```
. stpm3km
```



- Not bad, but some disagreement.

Simple use of stpm3km: age groups

```
. stpm3km agediag, groups(4)
```



Simple use of `stpm3km`: age groups 2

- Here, we are modelling age continuously, but takeing average of predictions `within` age groups.
- The model makes assumptions (e.g. proportional hazards) and if those assumptions are unrealistic we may get a poor fit.
- Not that bad here. We will do better later when we relax the linearity and proportional hazards assumptions.
- Note we are modelling age continously, but assessing within age groups, we could define different age groups if we wanted.

Some comments on marginal predictions

- This is a very simple example of marginal predictions.
- The key point is we often want to report average effects to summarise survival or differences in survival - we need to think about the population we are averaging over.
- We use the `standsurv` command for many reasons during the course.
- This will include,
  - Population and subgroup summaries
  - Age (and other covariate) standardization
  - Assessment of model fit
  - Obtaining descriptive contrasts
  - Obtaining causal contrasts

Some comments on predictions

- We usually prefer saving results to a `frame`. This separates the analysis data from predictions.
- We nearly always specify the times we want to predict using `timevar`, rather than `_t`
- The rows in the frame are usually less than the analysis data.
- We can predict for various combinations of covariates using multiple `at()` options.

We will run through Example 1 now.

Flexible parametric models: contrasts

Contrasts

- Rather than just predicting one overall measure, we are usually interested in contrasts between different subgroups.
- For example, comparisons betweeen
  - Different countries.
  - Males and females.
  - New vs standard treatment.
- We can predict for various combinations of covariates using `at()` options.
- We can obtain contrasts between conditional predictions using `predict` and marginal predictions using `standsurv`.

Model used to illustrate contrasts

- We fit the following simple model.

```
. stpm3 i.dep agediag, scale(lncumhazard) df(5) eform nolog neq(1)
                                                Number of obs = 24,883
                                                Wald chi2(5)  = 148.15
Log likelihood = -22455.915                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
2	1.046464	.0353249	1.35	0.178	.9794686	1.118041
3	1.098292	.0380768	2.70	0.007	1.026142	1.175515
4	1.203149	.0434151	5.13	0.000	1.120996	1.291322
mostdep	1.290682	.0506453	6.50	0.000	1.19514	1.393862
agediag	.9813256	.0020177	-9.17	0.000	.9773788	.9852882

```
Note: Estimates are transformed only in the first equation.
```

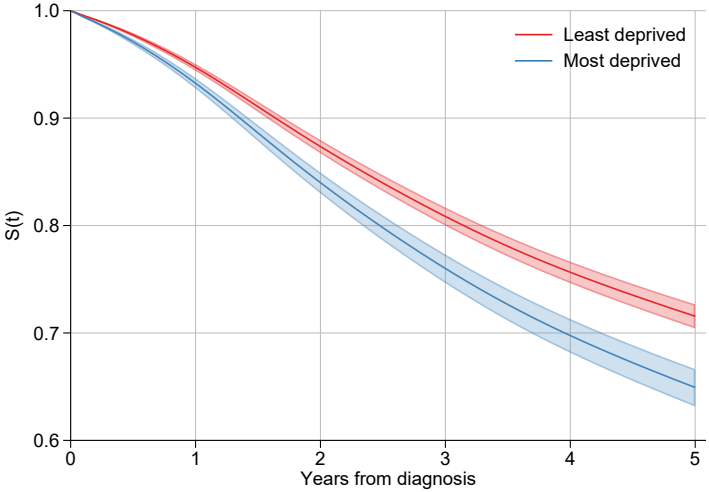
- Note `neq(1)` only shows the first equation (i.e. the spline parameters are not displayed)

Differences in survival functions

```
. predict S40_dep1 S40_dep5, survival ci frame(survpred, replace) ///
>         timevar(0 5, step(0.1))                                 ///
>         at1(dep 1 agediag 40) at2(dep 5 agediag 40)             ///
>         contrast(difference) contrastvar(Sdiff)
Predictions are stored in frame - survpred
```

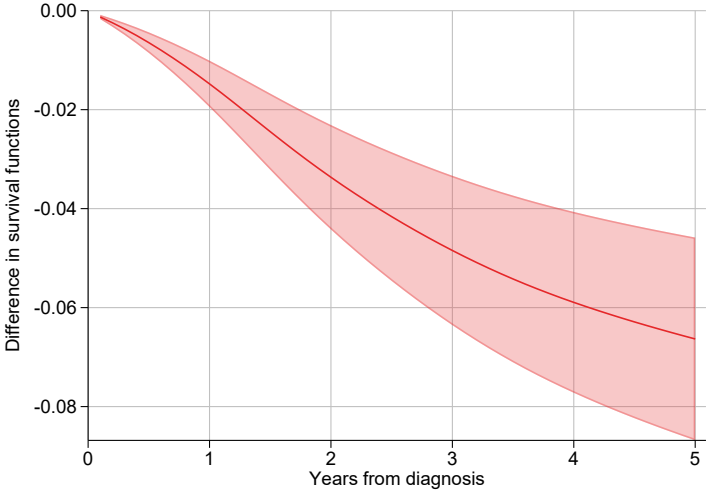
- We have specified that we want to calculate a difference using `contrast(difference)`.
- As we are predicting survival, this will be a difference in survival functions.
- `at1` is the reference, i.e. we calculate `at2-at1`. This can be changed with the `atref(#)` option.
- With K `at` options there will be $K - 1$ contrasts.
- The difference (with CIs) will be saved as `Sdiff` (`Sdiff_lci`,`Sdiff_uci`).

Survival curves to compare



Graph code in stpm3_predictions_contrasts

Differences in survival functions: graph



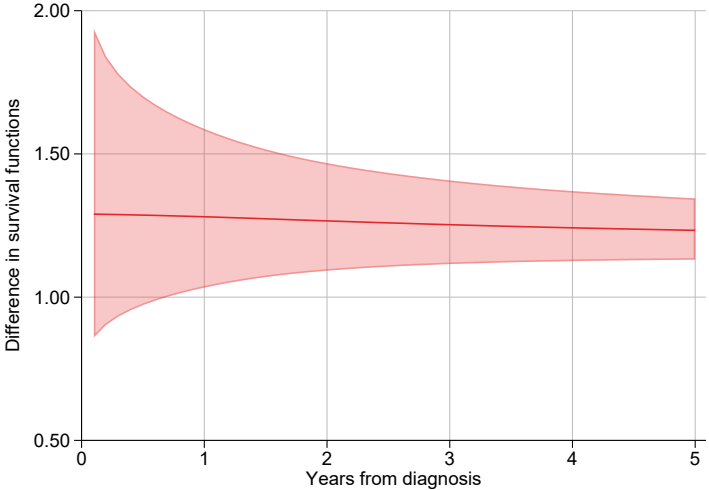
Graph code in stpm3_predictions_contrasts

Ratio of failure functions

- We can also take ratios when performing contrasts.
- This is most common for hazard ratios that change over time, but here we will take ratios of probabilities, i.e relative risk rather than a relative rate.
- Let $F(t) = 1 - S(t)$ be the probability of death by time t. We can calculate a relative risk $\frac{F(t|x=1)}{F(t|x=0)}$
- Note this is a function of time.

```
. predict F40_dep1 F40_dep5, failure ci frame(Fpred, replace)     ///
>         timevar(0 5, step(0.1))                                 ///
>         at1(dep 1 agediag 40) at2(dep 5 agediag 40)             ///
>         contrast(ratio) contrastvar(Fdiff)
Predictions are stored in frame - Fpred
```

Ratio of failure functions: graph



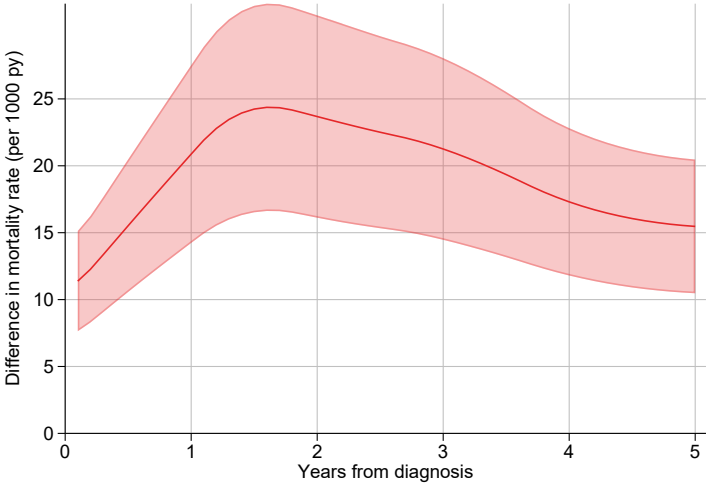
Graph code in stpm3_predictions_contrasts

Difference in hazard functions

```
. predict h40_dep1 h40_dep5, hazard ci frame(hazpred, replace) per(1000) ///
>         timevar(0.1 5, step(0.1))                                       ///
>         at1(dep 1 agediag 40)                                           ///
>         at2(dep 5 agediag 40)                                           ///
>         contrast(difference) contrastvar(hd40_dep)
Predictions are stored in frame - hazpred
```

- The prediction will give how many more deaths (per 1000py) we expect in the most deprived women aged 40 compared to least deprived women aged 40.
- We could use `contrast(ratio)` to get a hazard ratio, but this is a proportional hazards model and so this would just be a horizontal line at 1.29.

Difference in hazard functions: Graph



Graph code in stpm3_predictions_contrasts

How well do splines approximate the hazard?[11]

Journal of Statistical Computation and Simulation, 2013
http://dx.doi.org/10.1080/00949655.2013.845890

Taylor & Francis
Taylor & Francis Group

The use of restricted cubic splines to approximate complex hazard functions in the analysis of time-to-event data: a simulation study

Mark J. Rutherford[a*], Michael J. Crowther[a] and Paul C. Lambert[a,b]

- We do not believe the spline function is the true model, but provides a very good approximation.
- We assessed this in a simulation study.

Simulation Study (Rutherford et al.)

- Want to assess how well splines approximate the true function.
- Generate data assuming a mixture Weibull distribution,

$$S(t) = \pi \exp(-\lambda_1 t^{\gamma_1}) + (1 - \pi) \exp(-\lambda_2 t^{\gamma_2})$$

We will run a simplified version of this simulation study in Example 2

Flexible Parametric Survival Models: non-linear functions

Non-linear functions

- Generally better to model continuous covariates rather than categorise[12].
- However, this raises problems of choosing an appropriate functional form.
- Effects are rarely perfectly linear.
- For example, often a 'U' or 'J' shaped curve for the effect of age.
- Non-linear effects can be modelled in various ways, for example using polynomials, splines or fractional polynomials[13].
- Models appear complex, but we can still report results in a simple way.
- I will use the England breast North West data as an example.
- I restrict analysis to the least and most deprived groups.

Assuming linearity

```
. use breast_nw if inlist(dep,1,5)
(Ch28 Adult Breast 174, 175)
. stset survtime, failure(dead==1) exit(time 5)
  (output omitted)
. stpm3 i.dep agediag, scale(lncumhazard) df(5) eform nolog neq(1)
                                                Number of obs =  6,242
                                                Wald chi2(2)  = 604.18
Log likelihood = -8074.6607                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
mostdep	1.266583	.0486285	6.16	0.000	1.174771	1.365571
agediag	1.034256	.0015005	23.22	0.000	1.031319	1.037201

```
Note: Estimates are transformed only in the first equation.
. estimate store linear
```

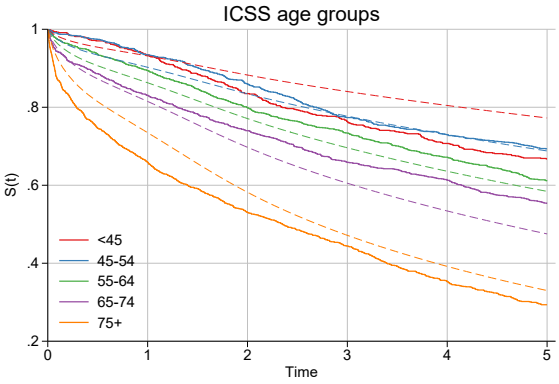
- This is a proportional hazards model.
- There is a 3.4% increase in the mortality rate for each yearly increase in age.
- This 3.4% increase is assumed to be the same for any age.

Assuming linearity 2

- This 3.4% is assumed to be the same at all follow-up times (because we are assuming proportional hazards).
- But, is linearity a reasonable assumption? For many cancers there is a 'U' shaped relationship with age.

Marginal predictions within age groups

```
. stpm3km i.agegrp, legoptions(pos(7))
```



- Very poor fit.

Quadratic function for age

- We relax the assumption of linearity by including a quadratic term for age.

```
. gen agediag2 = agediag^2
. stpm3 i.dep agediag agediag2, scale(lncumhazard) df(5) eform nolog neq(1)
                                                  Number of obs =  6,242
                                                  Wald chi2(3)  = 910.20
Log likelihood = -7998.9853                       Prob > chi2   = 0.0000
------------------------------------------------------------------------------
             |     exp(b)   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
xb           |
         dep |
    mostdep  |   1.283202   .0492616     6.50   0.000     1.190194    1.383478
     agediag |   .9128108   .0087122    -9.56   0.000     .8958939    .9300471
    agediag2 |   1.000972   .0000741    13.13   0.000     1.000827    1.001118
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first equation.
. estimate store quad1
. predict S60dep5, surv ci frame(f1, replace)          ///
>                  at1(dep 5 agediag 60 agediag2 3600)
Predictions are stored in frame - f1
```

- The individual age coefficients are difficult to interpret.

Quadratic function for age 2

- Predictions can get awkward. To predict for a 60 year old, we have to calculate the value of the `agediag2` term for a 60 year old (i.e 3600).
- With many covariates, more complex functions, interactions this is prone to error.
- `stpm3` has extended functions that allow the user to specify the function in the command itself. This makes predictions much easier.

Extended functions

- Extended functions allow you to include non-linear and more general functions when you specify the model.
- This makes predictions for complex non-linear effect with potential interactions much easier.
- The details of the non-linear function (e.g. knots for splines) are saved with the model.
- The current extended functions in `stpm3` are

`@bs()`	-	B-splines
`@fn()`	-	general functions
`@fp()`	-	fractional polynomials
`@ns()`	-	natural cubic splines
`@poly()`	-	polynomials
`@rcs()`	-	restricted cubic splines

Extended functions 2

- These can be incorporated as both main and time-dependent effects.
- See the help file for details of syntax.

Quadratic function for age: extended function

```
. stpm3 i.dep @poly(agediag,degree(2)), scale(lncumhazard) df(5) eform nolog neq(1)
                                                Number of obs =  6,242
                                                Wald chi2(3)  = 910.20
Log likelihood = -7998.9853                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
mostdep	1.283202	.0492616	6.50	0.000	1.190194	1.383478
_poly_f1_agediag1	.9128108	.0087122	-9.56	0.000	.8958939	.9300471
_poly_f1_agediag2	1.000972	.0000741	13.13	0.000	1.000827	1.001118

```
Note: Estimates are transformed only in the first equation.
Extended functions
 (1) @poly(agediag, degree(2))
. estimate store quad2
. predict S60dep5, surv ci frame(f1, replace)         ///
>                  at1(dep 5 agediag 60)
Predictions are stored in frame - f1
```

- The extended function is used directly in the *varlist*.
- `stpm3` has created new variables.

Quadratic function for age: extended function 2

- For the `predict` command we only need to know the age we want to predict at.
- This is a big advantage in more complex models.

Quadratic function for age: compare coeficients

```
. estimate tab quad1 quad2
```

Variable	quad1	quad2
xb		
dep		
mostdep	.24935823	.24935823
agediag	-.09122665	
agediag2	.00097182	
_poly_f1_a~1		-.09122665
_poly_f1_a~2		.00097182
time		
_ns1	-15.900347	-15.900347
_ns2	4.121899	4.121899
_ns3	-1.3906653	-1.3906653
_ns4	-.70938006	-.70938006
_ns5	-.4341717	-.4341717
_cons	1.0303667	1.0303667

- Coefficients are identical.

Natural spline function for age

- We generally prefer splines functions. Here the `@ns()` extended function is used.

```
. stpm3 i.dep @ns(agediag,df(3)), scale(lncumhazard) df(5) eform nolog neq(1)
                                                  Number of obs =  6,242
                                                  Wald chi2(4)  = 919.95
Log likelihood = -7998.7223                       Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
mostdep	1.287756	.0495411	6.57	0.000	1.194227	1.388609
_ns_f1_agediag1	.001194	.0008423	-9.54	0.000	.0002996	.0047584
_ns_f1_agediag2	.5379266	.1906011	-1.75	0.080	.268609	1.077272
_ns_f1_agediag3	.0548944	.012089	-13.18	0.000	.0356511	.0845244

```
Note: Estimates are transformed only in the first equation.
Extended functions
 (1) @ns(agediag, df(3))
. predict S60dep5, surv ci frame(f1, replace)         ///
>                   at1(dep 5 agediag 60)
Predictions are stored in frame - f1
```

Natural spline function for age 2

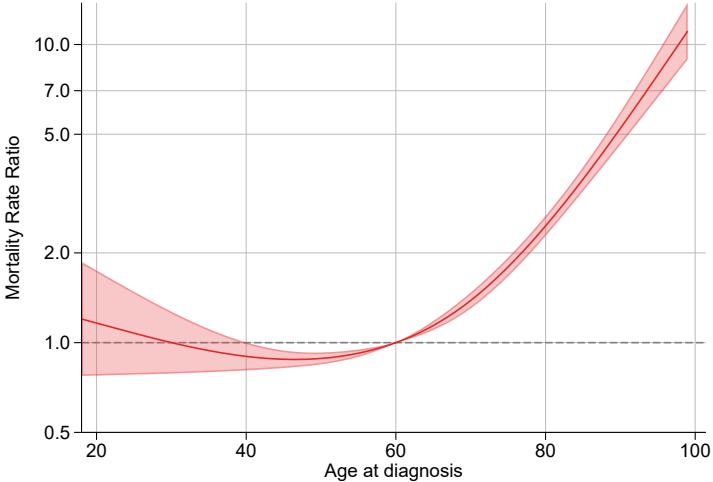
- `stpm3` has created new variables.
- Importantly, the `predict` command does not change from the polynomial model. We just need to think about the values of age and deprivation group we want to predict for.
- In the above code the `predict` command will calculate the relevant natural spline variables for a 60 year old.
- Irrespective of the extended function we use, the `predict` command will stay the same. This is true even when you have interactions and/or time-dependent effects.

Hazard ratio for non-linear function

- When quantify the non-linear function in terms of a hazard ratio, we need to specify a reference age.
- Similar to us setting a reference group with categorical variables.

```
. frame create agepred
. frame agepred {
.   range agediag 18 99 82
Number of observations (_N) was 0, now 82.
.   gen dep = .
(82 missing values generated)
.   gen t1 = 1
.   predict h_dep1 h_dep5, hazard ci timevar(t1) merge ///
>         at1(dep 1 agediag 60) at2(dep 1, obsvalues) ///
>         contrast(ratio) contrastvar(hr)
. }
```

Hazard ratio as a function of age



Graph code in stpm3_breast_NW_non_linear_functions

Non-linear functions with interactions

```
. stpm3 i.dep##@ns(agediag,df(3)), scale(lncumhazard) df(5) eform nolog neq(1) vsquish
                                                Number of obs =  6,242
                                                Wald chi2(7)  = 922.33
Log likelihood = -7987.6138                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
mostdep	.749069	.1537846	-1.41	0.159	.500921	1.120145
_ns_f1_agediag1	.0020386	.0020793	-6.07	0.000	.0002761	.0150503
_ns_f1_agediag2	.1990167	.1051681	-3.05	0.002	.070645	.5606573
_ns_f1_agediag3	.0368143	.0115166	-10.55	0.000	.0199405	.0679669
dep#c._ns_f1_agediag1						
mostdep	.4840744	.683222	-0.51	0.607	.0304458	7.696564
dep#c._ns_f1_agediag2						
mostdep	5.343581	3.816513	2.35	0.019	1.317904	21.66611
dep#c._ns_f1_agediag3						
mostdep	2.137918	.9350963	1.74	0.082	.9071666	5.038428

```
Note: Estimates are transformed only in the first equation.
Extended functions
 (1) @ns(agediag, df(3))
. predict S60dep5, surv ci frame(f1, replace)         ///
>                 at1(dep 5 agediag 60)
Predictions are stored in frame - f1
```

Non-linear functions with interactions 2

- By fitting an interaction between deprivation and age there is now a differernt non-linear function for the least and most deprived groups.
- Using the ## notation will fit the main effects and the interaction.
- Using the # notation will just fit the interaction.
- As before the `predict` command does not change. This greatly simplifies how we can predict from complex models.

Models with interactions

Interactions

- It is easy to add interactions to any model. The challenge is in quantifying what that interactions means.
- Sometimes our research question means that it is necessary to incorporate an interaction, e.g. Improvements in survival over calendar time are greater for for females than males.
- Sometimes we want to fit an interaction to model the complexity in our data, but are not interested in quantifying that interaction, e.g. we want our model to allow a differential effect of age for males and females, because we beleive it exists, but are only intersted in marginal differences in survival.
- I will fit a simple model using the Breast North West England data restricted to the least and most deprived to show some examples.

Main effects model

- We start with a main effects model

```
. stpm3 @ns(agediag,df(3)) i.dep, scale(lncumhazard) df(4) nolog neq(1)
                                                  Number of obs =  6,242
                                                  Wald chi2(4)  = 919.44
Log likelihood = -8008.3369                       Prob > chi2   = 0.0000
------------------------------------------------------------------------------
             |      Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
xb           |
_ns_f1_agedi~1 |  -6.731391   .7054777    -9.54   0.000    -8.114101    -5.34868
_ns_f1_agedi~2 |  -.6183224   .3543468    -1.74   0.081    -1.312829    .0761844
_ns_f1_agedi~3 |  -2.902752   .2202527   -13.18   0.000     -3.33444   -2.471065
               |
           dep |
       mostdep |   .2530977    .038471     6.58   0.000      .177696    .3284994
------------------------------------------------------------------------------
Extended functions
 (1) @ns(agediag, df(3))
. estimates store main
```

- We can interpret the effect of dep (it is a log hazard ratio).

Now with an interaction

```
. stpm3 @ns(agediag,df(3))##i.dep, scale(lncumhazard) df(4) nolog neq(1) vsquish
                                                Number of obs =  6,242
                                                Wald chi2(7)  = 921.73
Log likelihood = -7997.2579                     Prob > chi2   = 0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
_ns_f1_agediag1	-6.189548	1.020047	-6.07	0.000	-8.188803	-4.190294
_ns_f1_agediag2	-1.61466	.5284833	-3.06	0.002	-2.650468	-.5788519
_ns_f1_agediag3	-3.299452	.3128195	-10.55	0.000	-3.912567	-2.686337
dep |						
mostdep	-.2864709	.2053254	-1.40	0.163	-.6889013	.1159594
dep#c._ns_f1_agediag1 |						
mostdep	-.7368452	1.411549	-0.52	0.602	-3.50343	2.02974
dep#c._ns_f1_agediag2 |						
mostdep	1.679111	.7142818	2.35	0.019	.2791448	3.079078
dep#c._ns_f1_agediag3 |						
mostdep	.7551497	.43744	1.73	0.084	-.102217	1.612516

```
Extended functions
 (1) @ns(agediag, df(3))
. estimates store inter
```

- None of the parameters have a useful interpretion alone.

Likelihood ratio test

- Likelihood ratio test indicates differential effect of age for the least and most deprived, i.e. the effect of deprivation varies by age.

```
. lrtest main inter
Likelihood-ratio test
Assumption: main nested within inter
 LR chi2(3) =  22.16
Prob > chi2 = 0.0001
```

- We have used natural splines for age, so now we need to understand/quantify the interaction.

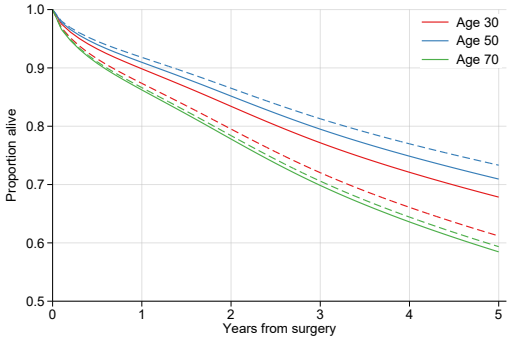
Compare survival for selected ages

- Predict for 30, 50 and 70 year old women in the least deprived group.

```
. estimates restore main
(results main are active now)
. predict h30m h50m  h70m, surv ci timevar(0 5,step(0.1))              ///
>                          frame(haz, replace)                         ///
>                          at1(agediag 30 dep 1) at2(agediag 50 dep 1) ///
>                          at3(agediag 70 dep 1)
Predictions are stored in frame - haz
. estimate restore inter
(results inter are active now)
. predict h30i  h50i h70i, surv ci                                     ///
>                          frame(haz, merge)                           ///
>                          at1(agediag 30 dep 1) at2(agediag 50 dep 1) ///
>                          at3(agediag 70 dep 1)
Predictions are stored in frame - haz
```

- Note that the prediction command is identical for the main effects and interaction model (except we are now merging to a frame).
- As a user you need to state the covariates you want a prediction for, `stpm3` will take care of dealing with interactions, non-linear effects, time-dependent effects etc.

Compare survival for selected ages (least deprived group)



- Larger difference (and opposite direction) for younger women.
- Useful to look at these plots: in large datasets significant interaction may lead to small differences in survival.

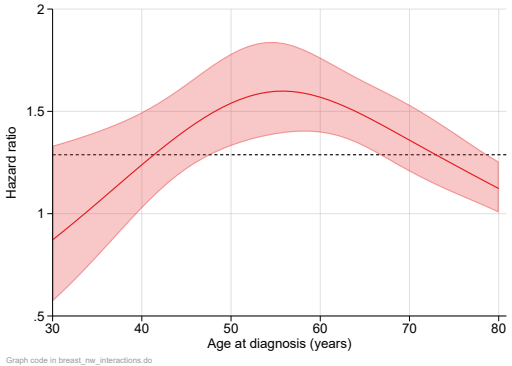
Hazard ratio for deprivation as function of age

- With an interaction the hazard ratio for deprivation will be a function of age.
- First create a frame with the ages you want to predict at.

```
. frame create ageHR
. frame ageHR {
.   range agediag 30 80 51
Number of observations (_N) was 0, now 51.
.   gen dep = .
(51 missing values generated)
.   predict h1 h5, hazard timevar(1) ci merge      ///
>                  at1(dep 1, obsvalues)           ///
>                  at2(dep 5, obsvalues)           ///
>                  contrast(ratio) contrastvar(hr)
. }
```

- We specify the `timevar()` option, but the hazard ratio will not vary by time, so we can use any time. `timevar(1)`, `timevar(2)`, `timevar(7)` would give identical results as we are assuming proportional hazards.
- We use the `obsvalues` suboption as we want to predict at the created values of `agediag`.

Hazard ratio for deprivation as function of age



- Dashed line shows hazard ratio from main effects model.

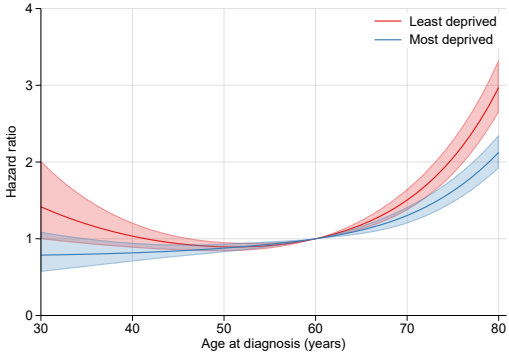
Hazard ratio for age - by deprivation group

- With an interaction the hazard ratio function for age will be different for each deprivation group.
- We can use the same frame as before.
- I will use age 60 as the reference age.

```
. frame ageHR {
.   predict, hazard timevar(tt) ci merge nogen           ///
>            at1(agediag 60 dep 1) at2(dep 1, obsvalues) ///
>            contrast(ratio) contrastvar(hr_dep1)
.
.   predict, hazard timevar(tt) ci merge nogen           ///
>            at1(agediag 60 dep 5) at2(dep 5, obsvalues) ///
>            contrast(ratio) contrastvar(hr_dep5)
. }
```

- Remember `at1()` is the reference (denominator) by default.

Hazard ratio for age - by deprivation group



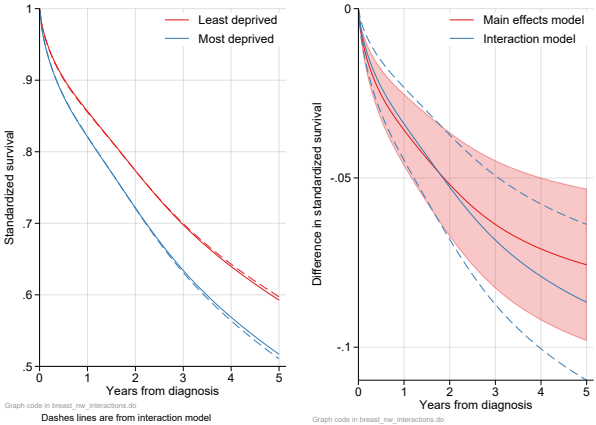
- Slighly more dramatic effect of age for least deprived group.

Impact on marginal survival and differences

- We can explore impact of the interaction on marginal survival and the survival difference.

```
. range tt 0 5 101
(6,141 missing values generated)
. estimates restore main
(results main are active now)
. standsurv S1m S5m, surv timevar(tt) ci frame(surv, replace) ///
>                  at1(dep 1) at2(dep 5)                       ///
>                  contrast(difference) contrastvar(Sdiff_m)
.
. estimates restore inter
(results inter are active now)
. standsurv S1i S5i, surv ci frame(surv, merge)               ///
>                  at1(dep 1) at2(dep 5)                       ///
>                  contrast(difference) contrastvar(Sdiff_i)
```

Impact on marginal survival and differences



- Not huge, but about 1 percentage point difference at 5 years.

Sensitivity to knots

I will run Exercise 3, rather than the slides for this section

- When using splines it is important to ask if the fitted values are sensitive to the number and the location of the knots.
- Too many knots will overfit with local 'humps and bumps'.
- Too few knots will underfit.
- In most situations the exact choice of knots is not crucial.
- We can use the AIC and BIC to help us select how many knots to use, but a simple sensitivity analysis is recommended.

How many knots?

- An obvious question is how many knots to use?
- In proportional hazard models, the number of knots is generally not that important when interest only lies in estimation of hazard ratios.
- The models are (usually) not nested and models can be compared using the AIC or the BIC where.

$$AIC = -2 \ln L + 2p \qquad BIC = -2 \ln L + \ln(N)p$$

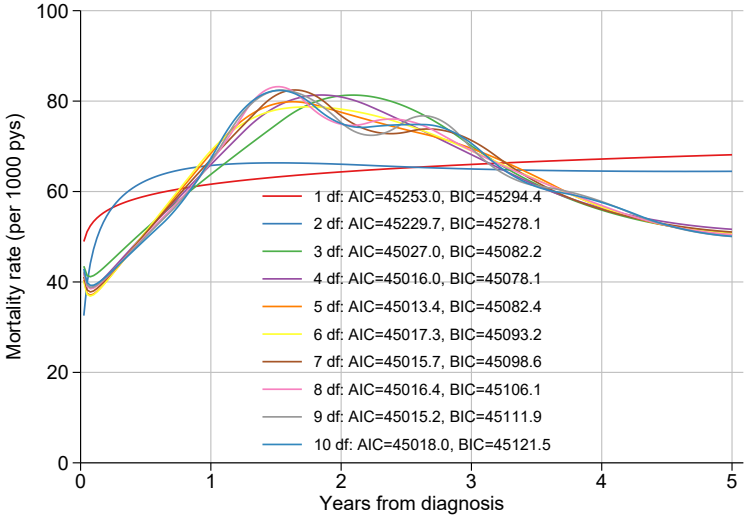
- The AIC and BIC differ in terms of the penalty function applied to the number of parameters (p).
- Note that in survival data, N is usually taken to be the number of events.
- Selecting the number of knots in these models is an area where more research is needed. However, it is very unlikely that you will change your conclusions by including some extra knots.

Use loops when doing sensitivity analysis

```
forvalues i = 1/10 {
  stpm3 i.dep, scale(lncumhazard) df(`i´)
  estimates store df`i´
  predict h`i´, hazard zeros timevar(0 5, step(0.1)) per(1000) frame(hpred, mergecreate)
  predict S`i´, surv zeros timevar(0 5, step(0.1)) frame(spred, mergecreate)
}
```

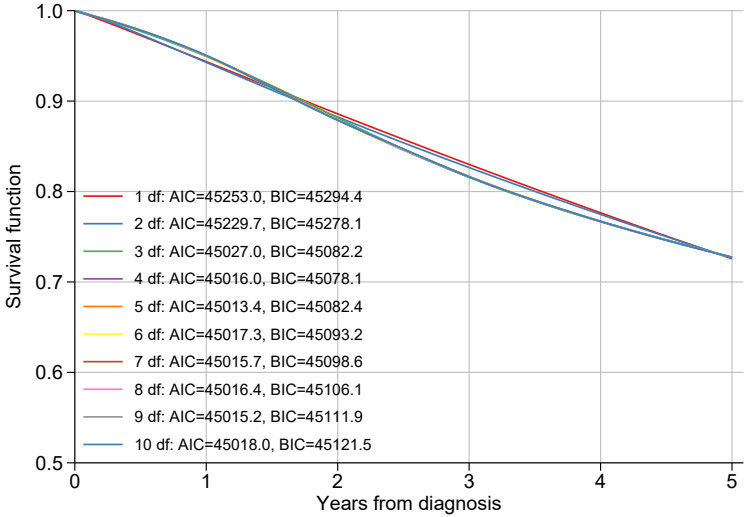
- The `frame(hpred, mergecreate)` option creates frame `hpred` if it does not exist, otherwise it will merge predictions into frame `hpred` and ignore the `timevar()` option.

Example of different knots for baseline hazard



Graph code in stpm3_sensitivity_to_knots.do

Example of different knots for baseline survival



Graph code in stpm3_sensitivity_to_knots.do

Effect of number of knots on hazard ratios

```
. estimates table df1 df3 df5 df7 df9, keep(2.dep 3.dep 4.dep 5.dep) eform se
```

Variable	df1	df3	df5	df7	df9
dep					
2	1.0492382	1.0490249	1.0489889	1.0489795	1.0489547
	.03541747	.03541028	.03540907	.03540875	.03540792
3	1.1054619	1.1052441	1.1052453	1.1052494	1.1052519
	.03831642	.03830887	.03830891	.03830906	.03830914
4	1.2144316	1.2129807	1.2130224	1.2130376	1.2130374
	.04380552	.04375326	.04375476	.04375532	.04375531
mostdep	1.310615	1.3100448	1.3098038	1.3098032	1.3097919
	.05137581	.05135347	.05134406	.05134403	.05134359

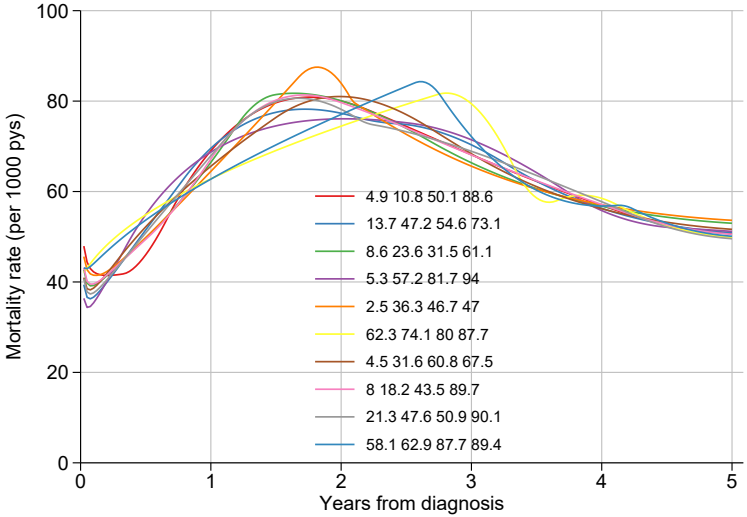
Legend: b/se

- Very similar estimates of hazard ratios and standard errors.
- Even for $df = 1$ or $df = 2$.

Where to place the knots?

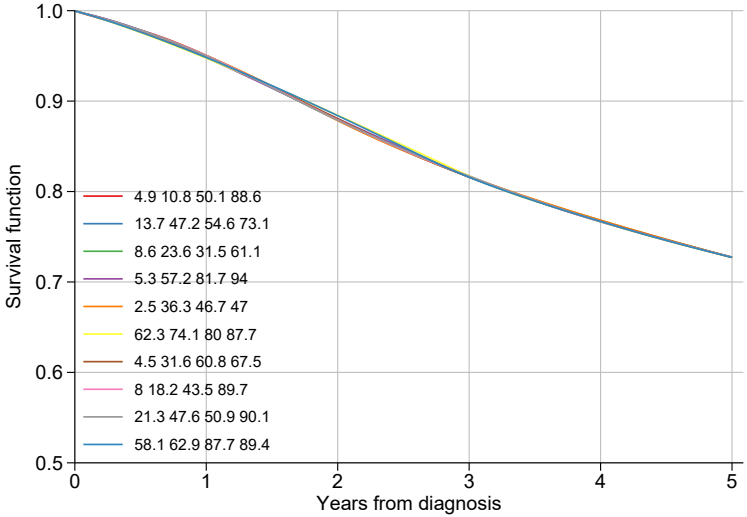
- The default knots positions tend to work fairly well.
- Unless the knots are in silly places then there is usually very little difference in the fitted values.
- The graphs on the following page shows for 5 df (4 interior knots) the fitted hazard and survival functions with the interior knot locations randomly selected.

Random knot positions for baseline hazard



Graph code in stpm3_sensitivity_to_knots.do

Effect of location of knots on baseline survival



Graph code in stpm3_sensitivity_to_knots.do

Flexible parametric models: time-dependent effects

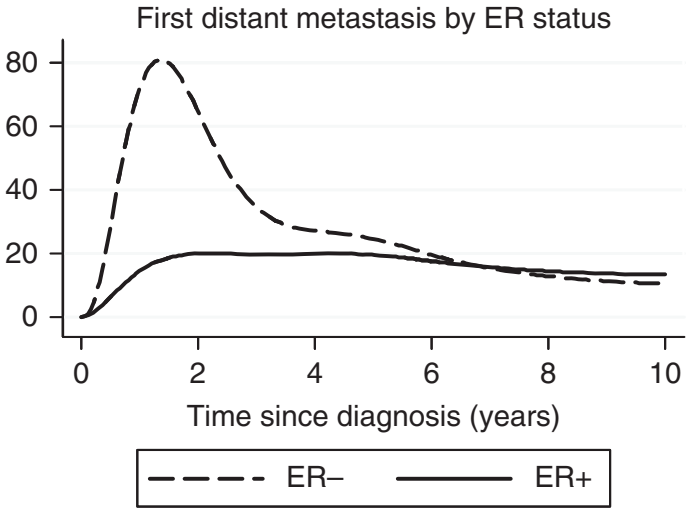
Time-dependent effects

- When the relative effect of a covariate varies over follow-up time, then we no longer have proportional hazards. In other words the effect of the covariate is time-dependent.
- Note that this is different to a time-dependent covariate, where the value of a covariate can change over follow-up time
- One way of fitting a model with non-proportional hazards is to fit the model on an alternative scale. E.g. in a proportional odds models the hazard rates are forced to converge as follow-up time increases.
- FPMs include the effect of time as covariates in the linear predictor, so time-dependent effects can be included by fitting interactions between the covariate of interest and the covariates defining the effect of time.

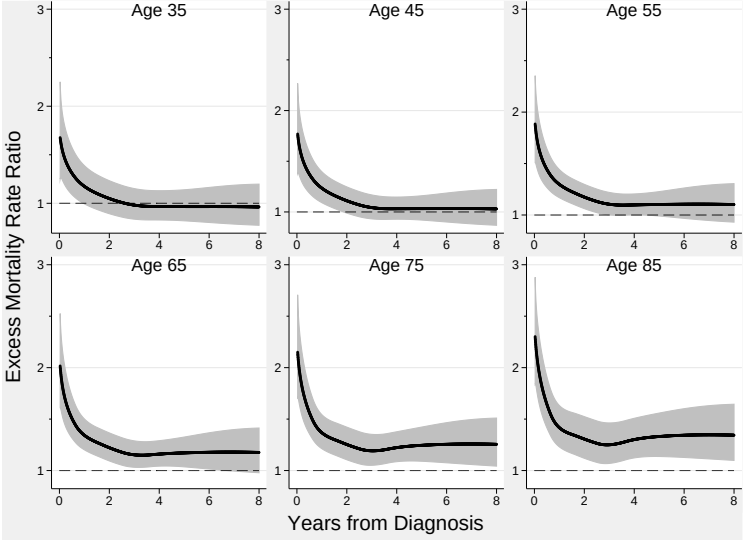
Time-dependent effects 2

- Time-dependent effects can be estimated in a piecewise fashion by categorization of time scale or in continuous time using splines.
- The fitting is the easy part - need to think about why the effect is time-dependent. Is it a true causal effect? Is it due to unobserved frailty?

Rate of Metastasis by Estrogen receptor status[14]



Excess Mortality Ratios for breast cancer (England vs Norway)



Non-proportional hazards

Proportional hazards

$$h_i(t|\mathbf{x_i}) = h_0(t)\exp\left(\mathbf{x}_i\boldsymbol{\beta}\right)$$

- Alternatively on the log scale.

$$\ln\left[h_i(t|\mathbf{x_i})\right] = \ln\left[h_0(t)\right] + \mathbf{x}_i\boldsymbol{\beta}$$

- The log hazard ratio, β is a single value, assumed to be the same throughout follow-up.
- For non-proportional hazards, the log hazard ratio is a function of time.
- This could be a step function, a linear function of time (or log time), or a spline function.

Time-dependent effects

- A proportional cumulative hazards model can be written

$$\ln\left[H_i(t|\mathbf{x_i})\right] = \eta_i(t) = s\left(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0\right) + \mathbf{x}_i\boldsymbol{\beta}$$

- We introduce is a new set of spline variables for each time-dependent effect.
- If there are D time-dependent effects then

$$\ln\left[H_i(t|\mathbf{x}_i)\right] = s\left(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0\right) + \sum_{j=1}^{D} s\left(\ln(t)|\boldsymbol{\delta}_j, \mathbf{k}_j\right)x_{ij} + \mathbf{x}_i\boldsymbol{\beta}$$

- The number of spline variables for a particular time-dependent effect will depend on the number of knots, $\mathbf{k}_j$

Time-dependent effects 2

- For any time-dependent effect there is an interaction between the covariate and the spline variables.
- The model is allowing for non-proportional cumulative hazards, but as we can estimate a hazard function for any covariate combination, we can take the ratio, between two (or more) different covariate combinations.

stpm3 and Time-Dependent Effects

- Non-proportional effects can be fitted by use of the `tvc()` and `dftvc()` options.

```
. stpm3 i.dep, scale(lncumhazard) df(5) tvc(i.dep) dftvc(3)
```

- There is no need to split the time-scale when fitting time-dependent effects.
- When time-dependence is a linear function of $\ln(t)$ and $N = 100,000$, 50% censored and no ties.
  - `stcox` using `tvc() texp(ln(_t))` - 43 minutes, 6 seconds.
  - `stpm3` using `dftvc(1)` - 0 minutes, 1.1 seconds.

Using the tvc() and dftvc() options

```
 . use breast_EW50 if agediag<=50 & inlist(dep,1,5)
(Ch28 Adult Breast 174, 175)
. stset survtime, failure(dead==1) exit(time 5)
  (output omitted)
. stpm3 i.dep, scale(lncumhazard) df(5) tvc(i.dep) dftvc(3) ///
>               eform nolog vsquish
                                                Number of obs =  9,719
                                                Wald chi2(1)  =  40.80
Log likelihood = -8750.6755                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
mostdep	1.286488	.0507348	6.39	0.000	1.190795	1.38987
time						
_ns1	-20.82268	1.513862	-13.75	0.000	-23.78979	-17.85556
_ns2	3.824477	.786076	4.87	0.000	2.283796	5.365157
_ns3	-1.156724	.0349299	-33.12	0.000	-1.225186	-1.088263
_ns4	-.6240747	.0245301	-25.44	0.000	-.6721529	-.5759965
_ns5	-.3733175	.0209403	-17.83	0.000	-.4143598	-.3322752
dep#c._ns_tvc1						
mostdep	1.895422	2.089921	0.91	0.364	-2.200747	5.991592
dep#c._ns_tvc2						
mostdep	-.1955392	1.109001	-0.18	0.860	-2.36914	1.978062
dep#c._ns_tvc3						
mostdep	.1387845	.0481613	2.88	0.004	.04439	.233179
_cons	-1.14041	.0234124	-48.71	0.000	-1.186297	-1.094522

Note: Estimates are transformed only in the first equation.

Comments on model/output

- Data reduced to least and most deprived groups (`inlist(dep,1,5)`).
- `stpm3` has formed an interaction between deprivation and some newly created spline variables.

```
. list dep _t 5.dep#c.(_ns_tvc*) in 1/7,noobs
```

dep	_t	5.dep# c._ns_t~1	5.dep# c._ns_t~2	5.dep# c._ns_tvc3
leastdep	5	0	0	0
leastdep	1.191	0	0	0
mostdep	1.673	.06921045	.13859791	.60150914
leastdep	5	0	0	0
leastdep	5	0	0	0
leastdep	4.0110002	0	0	0
mostdep	5	0	0	-.00080274

- I have shown the parameters estimates, but now none of them have a useful interpretation individually. However, by combining them we get predictions of hazard/survival functions etc.

Predictions

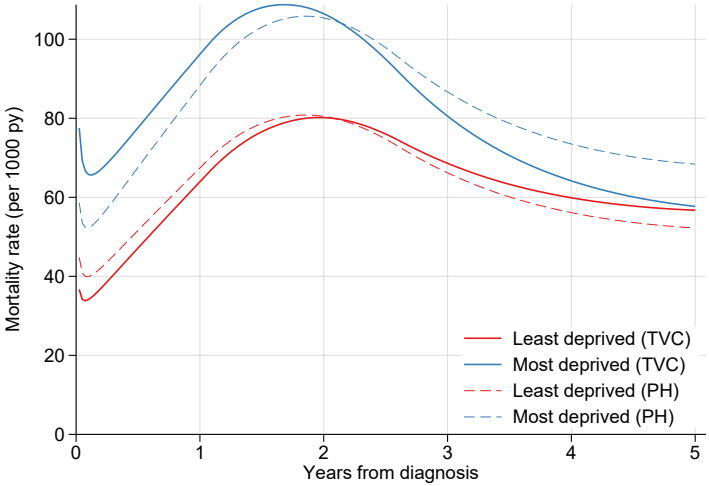
- The `predict` command will automatically incorprate any time dependent effects.
- This means that the `predict` syntax is identical whether there is a time dependent effect of not.

Predict hazard functions

```
. predict h0tvc h5tvc, hazard frame(hpred, replace) per(1000) ci ///
>                       timevar(0 5, step(0.025))                  ///
>                       at1(dep 1) at2(dep 5)                  ///
>                       contrast(ratio) contrastvar(hr)
Predictions are stored in frame - hpred
```

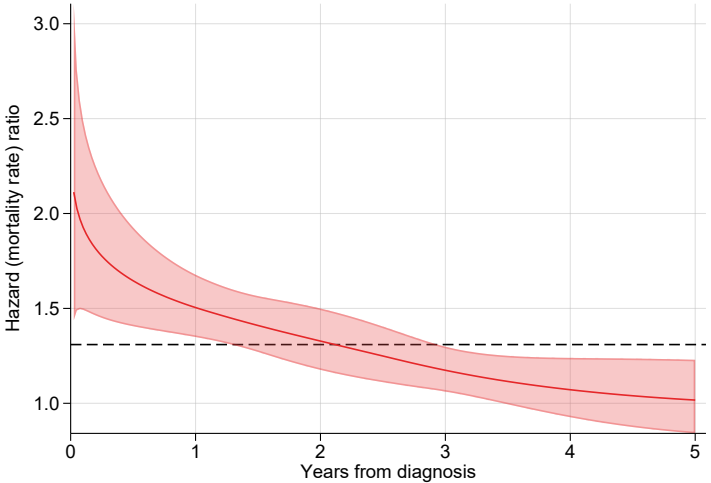
- The hazard ratio will now be a function of time. We can predict this using the `contrast(ratio)` option.

Non-proportional hazards



Graph code in stpm3_breast_EW50_non_linear_functions

Hazard ratio is a function of time

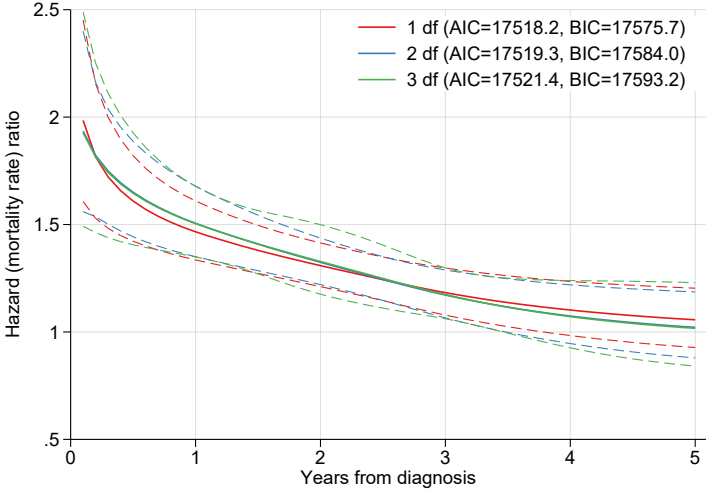


Graph code in stpm3_breast_EW50_non_linear_functions

Comments on fitting non-proportional hazards

- I have used 3 df to model the time-dependent effect of deprivation group.
- We are modelling the difference between the baseline and deprivation group 5. We often need fewer df for departures from the baseline than for the baseline itself. In a PH model the difference is a single value - the (log) hazard ratio.
- As we are modelling on the log cumulative hazard scale, the time-dependent hazard ratio may not be constant over other covariates that are also time-dependent. This is not the case for models on the log hazard scale.
- We report hazard ratios less often than we used to. However, most of the models we fit allow for non-proportional hazards.
- As always, it may be useful to perform a sensitivity analysis to the choice of df for time-dependent effects.

Hazard ratio: Sensitivity to knots



Graph code in stpm3_breast_EW50_non_linear_functions

Hazard ratio: Comments

- We (like others) present hazard ratios (with or without time-dependent effects) far less than we used too.
- Most of the models we fit incorporate time-dependent effects as the type of data we see rarely has proportional hazards.
- However, we choose to present more interpretable/understandable metrics.
  - Differences in marginal survival
  - Differences in restricted mean survival time
  - 'Avoidable' deaths
  - Differences in life expectancy
  - Reference adjusted measures
- We obtain better estimates of the above if we allow for non-proportional hazards.

Flexible Parametric Models: Marginal contrasts

Marginal measures and contrasts

- When analysing time-to-event data, it is often of interest to compare the prognosis of one population group to another e.g. 5-year survival of least deprived vs most deprived groups.
- It is common to fit a regression model, usually a Cox model, to adjust for several confounders.
- The most common reported parameter is an adjusted hazard ratio. But how informative is this?

Some thoughts on hazard ratios (HRs)

- Despite the popularity and broad use of HRs, these are often misinterpreted as relative *risks* [3, 4, 5, 6].
  - The relative risk is the ratio of the probability of experiencing the event by a specific time for the exposed to the probability for the unexposed.
  - The HR for an exposure is defined as the ratio of the hazard *rates* for the exposed and unexposed.
- Time-dependent HRs can be obtained but they are overlooked and a single HR is estimated for the whole study follow-up - can be an unrealistic assumption e.g. the effect of a treatment may lose effectiveness over time.

More thoughts on hazard ratios

- HRs are conditional on those who have survived up to a particular time. Even after adjusting for confounders at baseline, there may be emerging differences between survivors with time, resulting in an imbalanced comparison between exposure groups (built-in selection bias).
- Moreover, the HR is a relative measure, making it difficult to understand whether this effect is clinically meaningful. Absolute measures can be more informative than relative measures.
- It has an informative interpretation in terms of risk (that is more often the quantity of interest).

Marginal Contrasts

- `standsurv` has similar `contrast()` and `contrastvar()` options to `stpm3`'s `predict` command.
- We are now comparing marginal survival (and other) functions.
- When making contrasts we usually want to average over the same covariate distribution.
- These covariates may be confounders and thus we are averaging over the same confounder distribution,
- For confounders, Z, we can write this as,

$$E[S(t|X=1,Z)] - E[S(t|X=0,Z)]$$

- The key point is that this is the expectation (average) over the same covariate distribution, Z.

Estimation

- Fit a survival model for exposure X and confounders Z.
- Estimation of a marginal survival function is based on predicting a survival function for each individual and taking an average.

$$\frac{1}{N}\sum_{i=1}^{N}\widehat{S}\left(t|X=1,Z=z_i\right)-\frac{1}{N}\sum_{i=1}^{N}\widehat{S}\left(t|X=0,Z=z_i\right)$$

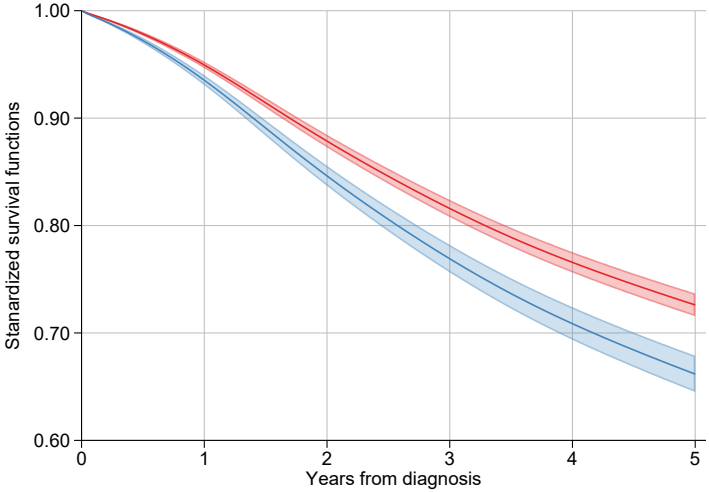
- Force everyone to be exposed and then unexposed.
- We use their observed covariate pattern, z_i.
- Epidemiologists call this model based or regression standardization[15].
- Also know as marginal effect or G-computation / G-formula.
- Can restrict to a subset of the population, e.g. the average causal effect in the exposed.
- We will go over more details of this later.

Differences in survival functions

```
. standsurv, survival ci timevar(tt) frame(msdiff, replace)  ///
>            at1(dep 1) at2(dep 5)                           ///
>            atvar(S1 S5)                                    ///
>            contrast(difference)                            ///
>            contrastvar(Sdiff)
```

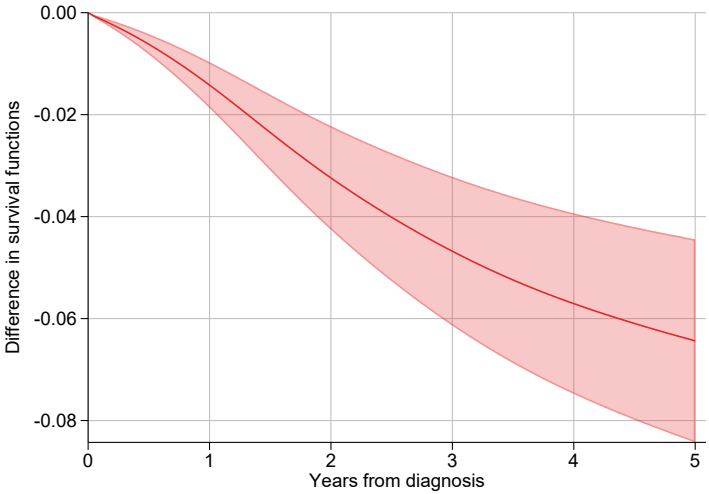
- `at1(dep 1)` forces all individuals to have `dep=1`.
- Need to be careful with (separate) `if` statements for the `at()` options as it is important to average over the same population.
- A single `if` statements will average over the same sub-population.
- Not sensible to use the `over()` option.
- Do not have expect to agree with Kaplan-Meier estimates as we are assigning individuals to covariate patterns they do not have.

Standardized survival functions



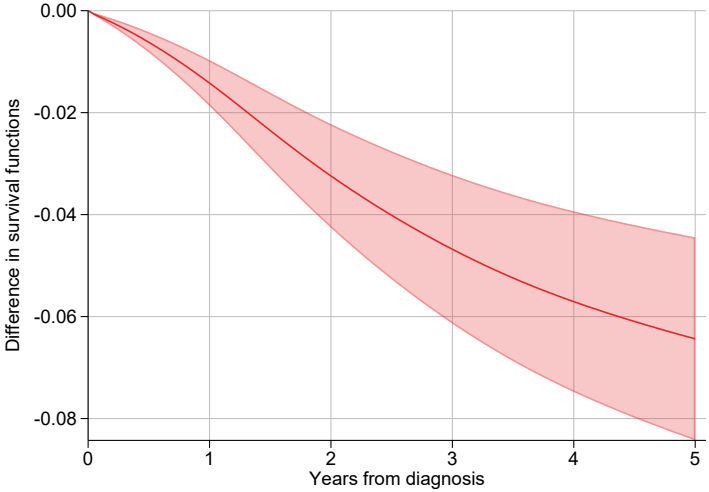
Graph code in stpm3_predictions_contrasts

Difference in standardized survival functions



Graph code in stpm3_predictions_contrasts

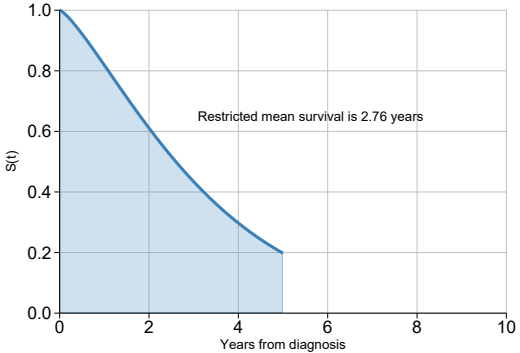
Difference in standardized survival functions



Graph code in stpm3_predictions_contrasts

Restricted mean survival time (RMST)

- An alternative measure to quantify survival is restricted mean survival where the mean is calculated up to time t^*.
- This is the area under the survival curve up to t, so we can obtain using (numeric) integration.



Restricted mean survival time (RMST)

restricted mean survival time

$$RMST(t^*) = E\left[min(T, t^*)\right]$$

$$RMST_s(t^*|X = x, Z) = \int_0^{t^*} E\left[S(t|X = x, Z)\right]$$

and is estimated by

$$\widehat{RMST}_s(t^*|X = x, Z) = \int_0^{t^*} \frac{1}{N} \sum_{i=1}^{N} S(t|X = x, Z = z_i)$$

- we can then take differences or ratios.
- Various authors suggest a better causal effect than HR[6]

RMST example

```
. gen t5 = 5 in 1
(24,882 missing values generated)
. standsurv, rmst ci timevar(t5)      ///
>            frame(rmst, replace)     ///
>            at1(dep 1) at2(dep 5)    ///
>            atvar(rmst1 rmst5)       ///
>            contrast(difference)     ///
>            contrastvar(rmstdiff)
```

- Just use `rmst` option - the rest of the code is the same.
- I choose to estimate at a single time point.

RMST output

```
. frame rmst: list rmst1* rmst5*, noobs
```

rmst1	rmst1_lci	rmst1_uci	rmst5	rmst5_lci	rmst5_uci
4.2725531	4.2421609	4.3031631	4.0895111	4.0404619	4.1391558

```
. frame rmst: list rmstdiff*, noobs
```

rmstdiff	rmstdi~lci	rmstdi~uci
-.18304201	-.24009462	-.12598939

- A difference of 0.18 years over the first 5 years after diagnosis.

Causal effects?

- Are we estimating causal effects?
- If we believe the cofounders are sufficient for confounding control (and we have modelled them appropriately), then yes.
- We are using regression standardization, or applying the G-formula.
- We meed to make additional assumptions (consistency, positivity, well-defined interventions) [15].
- However, we use standardization all the time is descriptive epidemiology, we want to standardize over age/sex and other covariate distributions.
- The code/maths is the same, but to infer causality we have to make lots of untestable assumptions.

I will run Example 4, so clarify what `standsurv` is doing.

Different populations to standardize over

- When we use `standsurv` the default is to standardize (average) over the covariate distribution in memory.
- Covariates that are no included in the model do not impact the estimate.
- We can restrict the population we standardize over using an `if` statement or by using the `atif()` suboption within any `at` option.
- We do not need to use the same data we fitted the model to. This can be useful for external validation of a prognostic model or when models have to be fitted separately as data sources can't be combined.
- Sometimes we standardize as we want to make fair comparisons between groups and sometimes we standardize as we want the average survival in a specific group (perhaps for assessment of model fit).
- The next few slides go through different syntax of `standsurv` to illusrate some of these issues.

Model to illustrate standardization issues

```
. stpm3 i.hormon @ns(age,df(3)) @fn(exp(-0.12 * nodes),stub(enodes)), ///
>       scale(lncumhazard) df(4) nolog neq(1)
                                                Number of obs =  2,982
                                                Wald chi2(5)  = 624.29
Log likelihood = -2670.2995                     Prob > chi2   = 0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
hormon						
yes	-.1270688	.0910185	-1.40	0.163	-.3054619	.0513242
_ns_f1_age1	-3.226227	.9001483	-3.58	0.000	-4.990485	-1.461968
_ns_f1_age2	-.8273933	.3834533	-2.16	0.031	-1.578948	-.0758386
_ns_f1_age3	-2.285906	.3964405	-5.77	0.000	-3.062915	-1.508897
_fn_enodes	-2.189403	.0979397	-22.35	0.000	-2.381362	-1.997445

```
Extended functions
 (1) @ns(age, df(3))
 (2) @fn(exp(-0.12 * nodes), stub(enodes))
```

- The research question of interest is to compare the effect of receiving hormonal therapy vs no receiving hormonal therapy.
- We (naively) assume that including age and nodes is sufficent for confounding control and we are thus able to estimate causal effects.

Averaging over the full study population

```
. standsurv, surv ci timevar(tt) frame(f1, replace)   ///
>            at1(hormon 0) at2(hormon 1) atvar(S0 S1) ///
>            contrast(difference) contrastvar(Sdiff)
. frame f1: list tt S0 S1 Sdiff if tt == 10, noobs
```

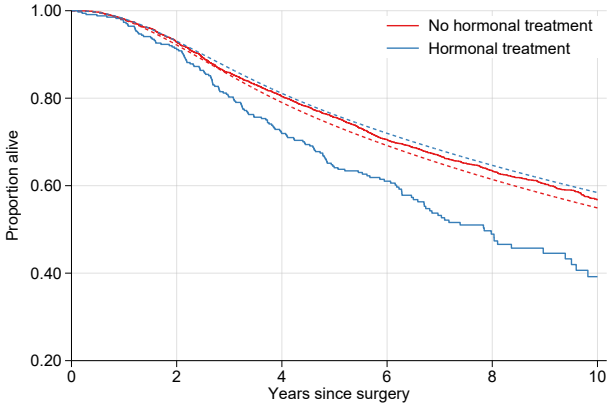
tt	S0	S1	Sdiff
10	.54911641	.58422915	.03511274

- We are averaging over the combined covariate distribution, so do not expect these marginal curves to agree with the Kaplan-Meier estimates.
- There are more in the untreated group, so expect the marginal curves to be closer to this group.

```
. dtable  age nodes, by(hormon, nototal) cont(, stat(mean)) sample(,stat(freq))
```

	Hormonal therapy	
	no	yes
N	2,643	339
age at surgery	54.098	62.549
Number of positive nodes (nrpos)	2.327	5.720

Averaging over the full study population



Graph code in standsurv_rott3_warnings

Dashed lines are standardized model based estimates

Averaging over the untreated

```
. standsurv if hormon==0, surv ci timevar(tt) frame(f2, replace)   ///
>                         at1(hormon 0) at2(hormon 1) atvar(S0 S1) ///
>                         contrast(difference) contrastvar(Sdiff)
. frame f2: list tt S0 S1 Sdiff if tt == 10, noobs
```

tt	S0	S1	Sdiff
10	.57092317	.60552487	.03460171

- We are averaging over the covariate distribution of the untreated. We would expect that the model based and Kaplan-Meier curves for the untreated should be similar if the model is reasonable.
- We are forcing the covariate distribution of the treated to be less severe than observed.
- We are estimating how much the survival would improve if the untreated were treated (under assumptions!).

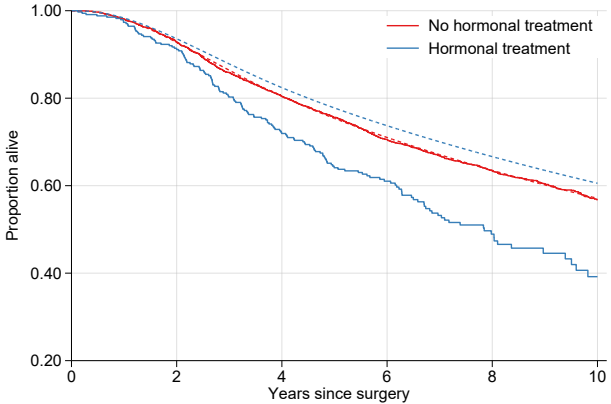
Averaging over the untreated 2

- We can estimate the above using different syntax.

```
. standsurv, surv ci timevar(tt) frame(f3, replace)                          ///
>            at1(hormon 0, atif(hormon==0)) at2(hormon 1, atif(hormon==0)) ///
>            atvar(S0 S1)                                                    ///
>            contrast(difference) contrastvar(Sdiff)
. frame f2: list tt S0 S1 Sdiff if tt == 10, noobs
```

tt	S0	S1	Sdiff
10	.57092317	.60552487	.03460171

Averaging over the untreated



Graph code in standsurv_rott3_warnings

Dashed lines are standardized model based estimates

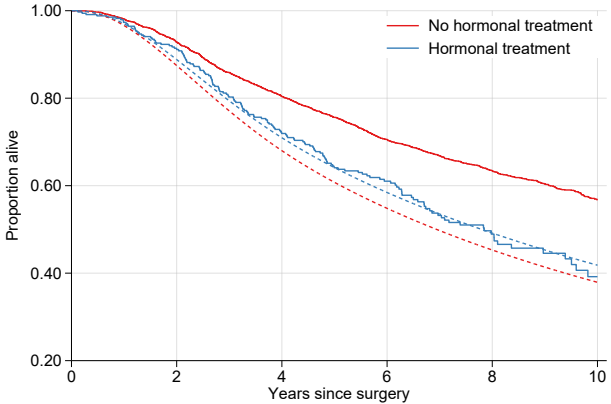
Averaging over the treated

```
. standsurv if hormon==1, surv ci timevar(tt) frame(f4, replace)   ///
>                         at1(hormon 0) at2(hormon 1) atvar(S0 S1) ///
>                         contrast(difference) contrastvar(Sdiff)
. frame f4: list tt S0 S1 Sdiff if tt == 10, noobs
```

tt	S0	S1	Sdiff
10	.3791009	.4181979	.039097

- We are averaging over the covariate distribution of the treated. We would expect that the model based and Kaplan-Meier curves for the treated should be similar if the model is reasonable.
- We are forcing the covariate distribution of the untreated to be more severe than observed.
- We are estimating how much the survival would change if the treated were untreated (under assumptions!).

Averaging over the untreated



Graph code in standsurv_rott3_warnings

Dashed lines are standardized model based estimates

Averaging within each group

```
. standsurv, surv ci timevar(tt) frame(f5, replace)  ///
>            over(hormon) atvar(S0 S1)               ///
>            contrast(difference) contrastvar(Sdiff)
. frame f5: list tt S0 S1 Sdiff if tt == 10, noobs
```

tt	S0	S1	Sdiff
10	.57092317	.4181979	-.15272527

- We are averaging over the covariate distribution of each group separately. We would expect that the model based and Kaplan-Meier curves for both groups to be similar if the model is reasonable.
- We are not adjusting for confounding, each group has its own covariate distribution and we know that the treated group were older and had more severe disease.

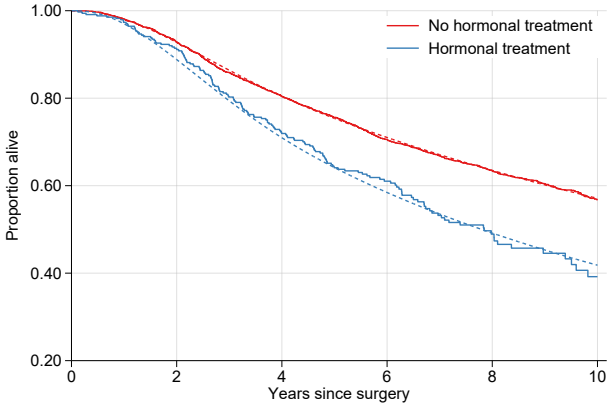
Averaging within each group 2

- Good for assessing model fit and/or understanding average survival in subgroups, but this does not answer a causal question.
- Alternative syntax

```
. standsurv, surv ci timevar(tt) frame(f6, replace)                          ///
>            at1(hormon 0, atif(hormon==0)) at2(hormon 1, atif(hormon==1)) ///
>            atvar(S0 S1)                                                    ///
>            contrast(difference) contrastvar(Sdiff)
. frame f6: list tt S0 S1 Sdiff if tt == 10, noobs
```

tt	S0	S1	Sdiff
10	.57092317	.4181979	-.15272527

Averaging within each group



Graph code in standsurv_rott3_warnings

Dashed lines are standardized model based estimates

User defined functions in `standsurv`

Used defined functions in `standsurv`

- When using `standsurv` we can give a number of `at` options.
- We use the `contrast()` option to perform contrasts of what is defined in each `at` option.
  - `contrast(difference)` will take absolute differences
  - `contrast(ratio)` will take ratios.
- Sometimes we need to use a more complicated function than a difference or a ratio.
- We can use the `userfunction()` option to do this.

Example: Population attributable fraction

- The (population) attributable fraction is the proportion of preventable outcomes if all subjects had not been exposed to a particular exposure. i.e.

$$AF = \frac{P(D=1) - P(D=1|X=0)}{P(D=1)}$$

  where where $P(D=1)$ is proportion diseased in the whole population, and $P(D=1|X=0)$ is the probability of being diseased in the unexposed.

- In observation studies there will be confounding and thus we need to consider potential confounders, Z.

$$AF = \frac{E(D=1|Z) - E(D=1|X=0,Z)}{E(D=1|Z)}$$

Attributable fraction in survival studies

- In survival studies the probability of being diseased is a function of time, so we define the AF using the failure function, $F(t) = 1 - S(t)$, so $AF(t)$ is defined as

$$AF(t) = \frac{E[F(t|Z)] - E[F(t|X=0,Z)]}{E[F(t|Z)]} = 1 - \frac{E[F(t|X=0,Z)]}{E[F(t|Z)]}$$

  - $E[F(t|Z)]$ is the standardized failure function over covariate distribution, Z
  - $E[F(t|X=0,Z)]$ is the standardized failure function over covariate distribution, Z where all subjects are forced to be unexposed.

- See Samualson (2008)[16] for some background.

Rotterdam data

- We fit a model to the Rotterdam data.
- The exposure is *not* being treated with hormonal therapy.

```
. stpm3 i.hormon age enodes pr_1, scale(lncumhazard) df(4) eform nolog
                                                Number of obs =  2,982
                                                Wald chi2(4)  = 619.62
Log likelihood = -2668.4925                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
hormon						
yes	.7906432	.0715077	-2.60	0.009	.66221	.9439854
age	1.013244	.0024119	5.53	0.000	1.008528	1.017983
enodes	.1132534	.0110135	-22.40	0.000	.0935998	.1370337
pr_1	.9064855	.0119282	-7.46	0.000	.8834055	.9301685
time						
_ns1	-25.90082	1.871965	-13.84	0.000	-29.5698	-22.23184
_ns2	7.980587	1.003724	7.95	0.000	6.013324	9.947851
_ns3	-1.091126	.0461407	-23.65	0.000	-1.18156	-1.000691
_ns4	-.70103	.0504635	-13.89	0.000	-.7999366	-.6021234
_cons	.801967	.161537	4.96	0.000	.4853603	1.118574

```
Note: Estimates are transformed only in the first equation.
```

Using standsurv

- `standsurv` will calculate the ingredients for the AF.

```
. range tt 0 10 101
(2,881 missing values generated)
. standsurv,  failure timevar(tt) ci frame(AF1, replace)  ///
>             at1(hormon 0) at2(hormon 1) at3(.)          ///
>             atvar(F_hormon0 F_hormon1 F_all)
```

- We can plug in the relevant standardized estimates to calculate the AF.

```
. frame AF1: gen AF_tmp = 1 - F_hormon1/F_all
(1 missing value generated)
. frame AF1: list tt F_hormon1 F_all AF_tmp if inlist(tt,1,5,10), noobs
```

tt	F_hormon1	F_all	AF_tmp
1	.01685169	.02035349	.17204904
5	.22362896	.26167585	.14539701
10	.39250923	.44808119	.12402208

- Unfortunately, this will not give us a confidence interval.

Write a Mata function

- We write a short Mata function to calculate the AF.

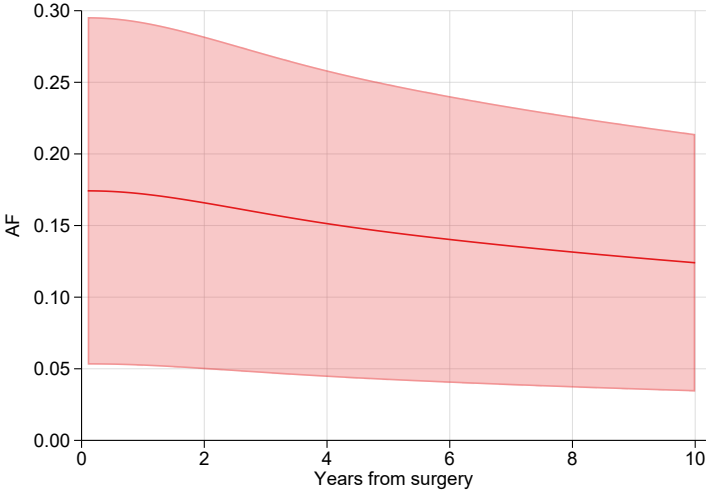
```
. mata:
------------------------------------------------- mata (type end to exit) -----------------
: function calcAF(at) {
>   // at2 is F(t|unexposed,Z)
>   // at1 is F(t,Z)
>   return(1 - at[2]/at[1])
> }
: end
--------------------------------------------------------------------------------------------
```

- The function receives the argument `at`.
- The `at` options in standsurv need to be specified in the same order as used in the Mata function.

```
. standsurv, failure timevar(tt) ci frame(AF2, replace) ///
>            at1(.) at2(hormon 1)                       ///
>            userfunction(calcAF) userfunctionvar(AF)
```

- Note the use of the `userfunction()` option.

Plot the attributable fraction



Graph code in standsurv_PAF.do

Flexible parametric surival models: other scales

*Models on other scales

- Proportionality on one scale will generally leads to non-proportionality on another scale.
- `stpm3` can fit models on other scales. These are
  - The log cumulative odds: use `scale(lnodds)`.
  - The probit scale: use `scale(probit)`.
- May get more parsimonious model on alternative scales, which may have advantages in smaller datasets.
- We hardly ever use these other scales (apart from `scale(lnhazard)`)

*Proportional Odds

- Proportional odds models described by Bennett as extension of logistic regression to censored data [17].
- For two groups OR could be a function of time

$$\text{OR}(t) = \frac{F_1(t)}{1-F_1(t)} / \frac{F_2(t)}{1-F_2(t)}$$

$$\frac{F_1(t)}{1-F_1(t)} = \frac{F_2(t)}{1-F_2(t)}\text{OR}(t)$$

- If $\text{OR}(t)$ is constant then this is a proportional odds model
- The log-logistic model can be expressed as a PO model.

*Other models

proportional hazards

$$\log(-\log(S(t|\mathbf{x}_i))) = s(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0) + \mathbf{x}_i\boldsymbol{\beta}$$

- With 1 df (linear in $\ln t$) equivalent to Weibull model

proportional odds

$$\text{logit}(1 - S(t|\mathbf{x}_i)) = s(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0) + \mathbf{x}_i\boldsymbol{\beta}$$

- With 1 df (linear in $\ln t$) equivalent to log-logistic model

probit model

$$-\Phi^{-1}(S(\ln t|\mathbf{x}_i)) = s(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0) + \mathbf{x}_i\boldsymbol{\beta}$$

- With 1 df (linear in $\ln t$) equivalent to log-normal model

Flexible parametric models: log hazard scale

Modelling on the log hazard scale

- We have also worked on using splines on the log hazard scale (rather than the log cumulative hazard scale) [18, 19, 20].
- `stpm3` has the option `scale(lnhazard)` or `scale(loghazard)` to fit these models.
- Why model on the log hazard scale?
  - More natural scale when modelling multiple time scales.
  - More natural scale when modelling SMRs/SIRs.
  - Some interpretation issues for cumulative hazard models when having multiple time-dependent effects and wanting to quantify using hazard ratios.
  - May be more sensible when extrapolating in relative survival/excess mortality models.

Modelling on the log hazard scale

- The model is

$$\ln\left[h_i(t)\right] = s\left(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0\right) + \mathbf{x}_i\boldsymbol{\beta}$$

- We have changed from $H_i(t)$ to $h_i(t)$.
- The splines now directly model the log baseline hazard function.
- Extension to time-dependent effects is the same, i.e. include interactions between covariates and a spline function.

Estimation

- The contribution of the i^{th} subject to the the Log-likelihood is,

$$\ln L_i = d_i \ln [h(t_i)] - \int_{t_{0i}}^{t_i} h(u)du$$

- Need to integrate the hazard function in order to estimate the parameters.
- However, it is not possible to derive these integrals analytically. We therefore, use numerical integration using Gaussian quadrature.
- Estimation slower, but allows a wider range of models to be fitted.
- See papers [18, 19, 20] for more details.

NW England Breast Cancer Example

```
. stpm3 i.dep, scale(lnhazard) df(5) eform nolog

                                                Number of obs = 24,883
                                                Wald chi2(4)  =  63.42
Log likelihood = -22496.272                     Prob > chi2   = 0.0000
```

	exp(b)	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
2	1.049003	.0354095	1.42	0.156	.9818471	1.120751
3	1.10526	.0383094	2.89	0.004	1.032668	1.182954
4	1.213058	.0437561	5.35	0.000	1.130259	1.301923
mostdep	1.30988	.051347	6.89	0.000	1.21301	1.414485
time						
_ns1	1.291467	.5156812	2.50	0.012	.2807504	2.302184
_ns2	-1.691936	.3192245	-5.30	0.000	-2.317605	-1.066268
_ns3	.6101762	.0637901	9.57	0.000	.4851499	.7352025
_ns4	.3895094	.0599444	6.50	0.000	.2720205	.5069982
_ns5	.343812	.1204279	2.85	0.004	.1077777	.5798463
_cons	-3.02479	.0552183	-54.78	0.000	-3.133016	-2.916564

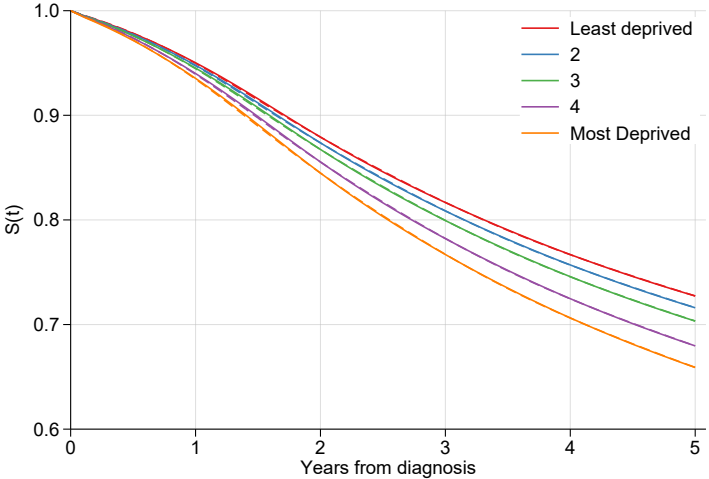
```
Note: Estimates are transformed only in the first equation.
Quadrature method: tanh-sinh with 30 nodes.
Analytical integration before first and after last knot.
```

NW England Breast Cancer Example

- The syntax for `predict` is identical to when using `scale(lncumhazard)` (or any other scale).

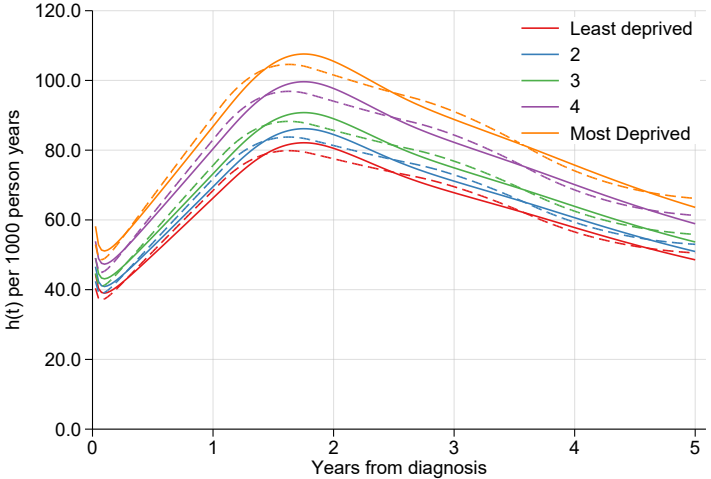
```
. predict Slnh*, surv ci frame(surv, replace)     ///
>                timevar(0 5, step(0.1))          ///
>                at1(dep 1) at2(dep 2) at3(dep 3) ///
>                at4(dep 4) at5(dep 5)
Predictions are stored in frame - surv
```

Comparison of survival functions



Solid lines are from log hazard model, dashed lines are from log cumulative hazard model
Graph code in log_hazard_models

Comparison of hazard functions



Solid lines are from log hazard model, dashed lines are from log cumulative hazard model
Graph code in log_hazard_models

Comments on models on log hazard scale

- The `lnhazard` (or `loghazard`) option did not exist in `stpm2`, so is a new addition to `stpm3`.
- You could fit log hazard spline models using `strcs`.
- Also implemented for relative survival models.
- You should be able to repeat exercises for models on the log hazard scale if you are interested.
- We are starting to use models on the log hazard scale more for work on extrapolation of survival functions.
- If you are interested in trying these models just change the `scale()` option in `stpm3`.

Model convergence issues

Model convergence issues

- You may have seen one of the following error messages.

```
initial values not feasible
r(1400);
```

```
Iteration 4: Log likelihood = -20517.528 (backed up)
Iteration 5: Log likelihood = -20517.407 (backed up)
iteration 6: Log likelihood = -20517.347 (backed up)
...
```

```
Iteration 9: log likelihood = -22921.452
Hessian is not negative semidefinite
r(430);
```

- I will discuss some reasons why you may get convergence problems.

Initial values

- `stpm3` used maximum likelihood to obtain the parameter estimates and their variances.
- This requires reasonable starting values.
- These are obtained by fitting a Cox (or other) model and then using least squares.
- The code on the following page is a simplified version of how starting values are obtained.

Obtaining initial values (simplified)

```
use breast_NW if inlist(dep,1,5)
stset survtime, failure(dead==1)
// we want to fit a model with age and i.dep included in the model
// we will use 3 df
gen lnt = ln(_t)
gensplines lnt, type(ns) df(3) gen(_ns) subcentile(_d==1)
// start by fitting a Cox model
stcox agediag i.dep
// obtain the linear predictor
predict xb, xb
// obtain the baseline cumulative hazard
predict CH, basechazard
// obtain the cumulative hazard at _t for each subject
replace CH = CH*exp(xb)
// calculate log cumulative hazard
gen logCH = ln(CH)
// initial values obtained using least squares for those with an event
regress logCH agediag i.dep _ns1 _ns2 _ns3 if _d
matrix initb = e(b)
matrix list initb
```

Using alternative models for the initial values

- The initial values not feasible error means that the log-likelihood can not be evaluated.
- One reason (more common in relative survival models) is that one or more of the hazard functions evaluted at the event times is negative.
- For `scale(logcumhazard)` models the (log) cumulative hazard should be monotonically increasing, i.e. have no turning points.
- The splines are not constrained to be monotonic, but usually this is not a problem as there is never observed negative hazards.
- We can try using a different model other than a Cox model to obtain starting values - use the `initmodel(`*model*`)` option. You can use `cox`, `exp`, `weibull` or `stpm2`. I find `exp` is usually a good option.
- If you use the option `initvaluesloop` then `stpm3` will loop over these different models to try different initial values.
- However, none of the initial values models may give usable initial values.

Use a simpler model for starting values

- You can try fitting a simpler model and using this for initial values of the parameters.
- Fit a simpler model, store the parameters and then pass these to `stpm3` using the `from()` option.

```
stpm3 @ns(agediag,df(3)), scale(lncumhazard) df(5)
matrix initb = e(b)

stpm3 @ns(agediag,df(3)), scale(lncumhazard) df(5)  from(initb)  ///
                          tvc(@ns(agediag,df(3))) dftvc(3)
```

Backed up error

- Nearly always means there is a problem with your model and changing initial values will not cure the problem.
- Things to think about
  - Do you have risk groups with only a small number (or zero) events. Use `sts graph, by(varname) risktable`.
  - Have you have a few individuals with very long follow-up time? Are you interested in long term follow-up. If not restrict the follow-up time, e.g. `stset ... , .....exit(time 10)`.
  - If you have a small dataset (actually a small number of events) then you should not be fitting a complex model.

Time-dependent effects

- This is where the problems usually occur.
- Consider modelling age with a time-dependent effect, you are allowing a very flexible function that changes over follow-up time.
- However, we need data to estimate things. Think how many 85 year olds will still be alive 10 years after diagnosis.
- Try restricting the follow-up.
- We have used winsorizing a lot that can help when it is the extreme ages causing problems.

Winsorizing

- Winsorizing sets observations lower or higher than specifed percentiles to the values of the percentiles, whilst leaving all other observations unchanged.
- We have used this (mainly in relative survival models) to stabalize model predictions at the extremes of a continous distribution.

```
. centile agediag, c(2 98)
                                                      Binom. interp.
    Variable |       Obs  Percentile    Centile        [95% conf. interval]
-------------+-------------------------------------------------------------
     agediag |    14,823           2      34.287        33.71434    34.86864
                                98     89.28256        88.85043    89.75771
```

- A wisorized variable would replace all values less than 34.287 to 34.287 and all values greater than 89.282 to 89.282
- We can use the `winsor()` suboptions when using an extended varlist in `stpm3`.

Winsorizing: Standard model

```
. // model without winsorizing
. stpm3 @ns(agediag,df(3)), scale(lncumhazard) df(5) neq(1) nolog
                                                      Number of obs =  14,823
                                                      Wald chi2(3)  = 2214.54
Log likelihood = -18712.099                           Prob > chi2   =  0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
_ns_f1_agediag1	-7.150054	.4829609	-14.80	0.000	-8.09664	-6.203468
_ns_f1_agediag2	-.7180313	.242339	-2.96	0.003	-1.193007	-.2430555
_ns_f1_agediag3	-3.075457	.1473045	-20.88	0.000	-3.364169	-2.786746

```
Extended functions
 (1) @ns(agediag, df(3))
. estimates store standard
```

Winsorizing: Winsorized model

```
. // model with winsorizing
. stpm3 @ns(agediag,df(3) winsor(2 98)), scale(lncumhazard) df(5) neq(1) nolog
                                                Number of obs =  14,823
                                                Wald chi2(3)  = 2170.15
Log likelihood = -18716.11                      Prob > chi2   =  0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
_ns_f1_agediag1	-5.576178	.2167664	-25.72	0.000	-6.001032	-5.151324
_ns_f1_agediag2	-.0392185	.1279677	-0.31	0.759	-.2900306	.2115937
_ns_f1_agediag3	-2.146475	.091622	-23.43	0.000	-2.326051	-1.966899

```
Extended functions
 (1) @ns(agediag, df(3) winsor(2 98))
. estimates store winsorized
```

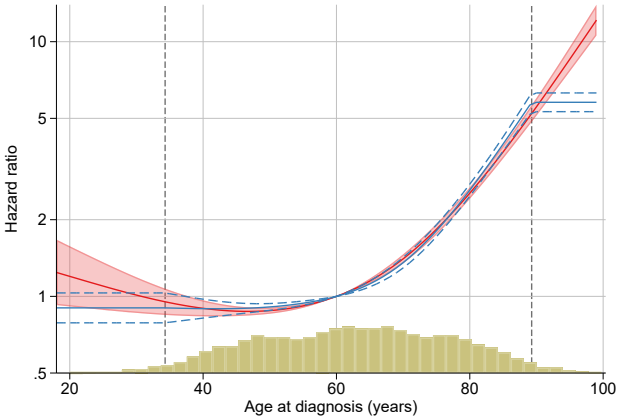
- 96% of the age variable is unchanged.

Winsorizing: predict hazard ratios

```
. frame create agehr
. frame agehr {
.   range agediag 18 99 82
Number of observations (_N) was 0, now 82.
.
.   gen t1 = 1
.   estimates restore standard
(results standard are active now)
.   predict , hazard ci merge timevar(t1) nogen           ///
>             at1(agediag 60) at2(.)                      ///
>             contrast(ratio) contrastvar(hr_standard)
.
.   estimates restore winsorized
(results winsorized are active now)
.   predict , hazard ci merge timevar(t1) nogen           ///
>             at1(agediag 60) at2(.)                      ///
>             contrast(ratio) contrastvar(hr_winsor)
. }
```

- Note that the code for the prediction is identical when we have winsorizing!

Winsorizing: compare hazard ratios



Graph code in stpm3_winsorizing.do

What to present?

What to report?

- We have seen many ways to present survival data
  - (Conditional) hazard ratios (proportional and time-dependent)
  - Survival for selected covariates patterns.
  - Marginal survival and differences.
  - Survival as a function of age (or other covariates)
  - Marginal restricted mean survival time (and differnces)
- There are things I have not presented
  - Marginal hazards ratios
  - Median survival (or other percentiles).
  - Attributable fractions / avoidable deaths.
- The answer, as always, is "it depends", but I would like to hear your views of what you like/don't like of the many different ways to present survival data.

Some extensions

Some Extensions

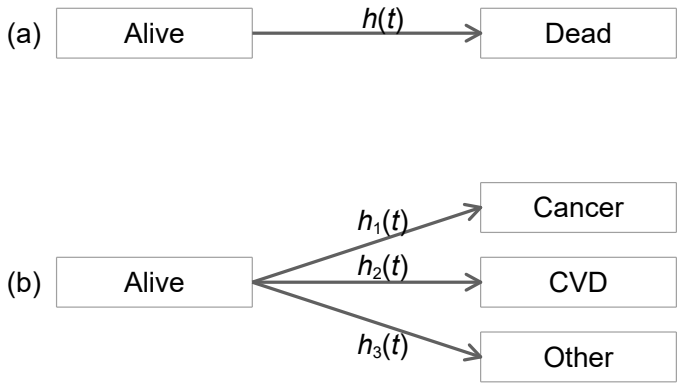
- Today I presented an overview of some standard flexible parametric survival models.
- There are lots of extensions and I will very briefly cover some of these. Namely:
  - Competing risks
  - Relative survival models
  - Using constraints
  - Assessment of calibration

Competing risks

- We are at risk of more than one event.
- For example, people diagnosed with cancer are at risk of dying from their cancer, but also at risk of dying from other causes.
- A competing event is an event that prevent the occurrence of the event of interest may be present.
  - Dying from a cardiovascular event means that the (potential) time-to-death for cancer never observed.
- Flexible parametric survival models also useful for competing risks models (and more general multistate models).

Predictions are based on estimates from > 1 model.

Competing risks



Graph code in competing_risks_schematic.do

Cause-specific Cumulative Incidence Function (CIF)

$$F_k(t) = \int_0^t S(u)h_k(u)du$$

Probability of dying due to cause k

Partitioning all-cause probability of death

$$F(t) = \sum_{k=1}^{K} F_k(t)$$

- CIFs estimated using numerical integration - using ODEs.

Different causes

```
. table cause, statistic(frequency) statistic(percent)
```

	Frequency	Percent
cause		
Censored	1,710	57.34
Cancer	996	33.40
Other causes	276	9.26
Total	2,982	100.00

A model for each cause

Death due to breast cancer

```
. stset os, failure(cause=1) exit(time 120) scale(12)
. stpm3 @ns(age, df(5)) i.size i.grade pr_1,   ///
        scale(lnodds) df(3)
. estimates store cancer
```

Death due to other causes

```
. stset os, failure(cause=2) exit(time 120) scale(12)
. stpm3 @ns(age,df(3)), scale(lncumhazard) df(3)
. estimates store other
```

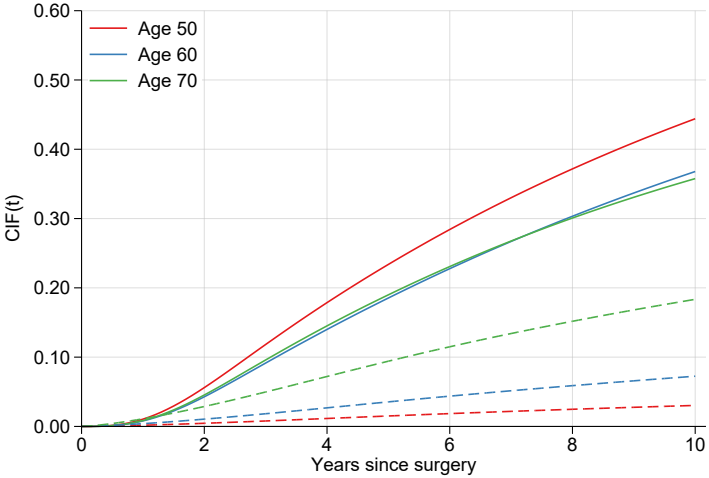
Store model estimates so can pass to `predict` command.

Predictions

```
. // Conditional predictions
. predict cif50 cif60 cif70, cif crmodels(cancer other) ci              ///
>                            timevar(0 10, step(0.1))                   ///
>                            at1(age 50 size 1 grade 2 nodes 3 pr_1 0) ///
>                            at2(age 60 size 1 grade 2 nodes 3 pr_1 0) ///
>                            at3(age 70 size 1 grade 2 nodes 3 pr_1 0) ///
>                            frame(cifs, replace)
Predictions are stored in frame - cifs

.
. // Marginal predictions
. standsurv CIF_size1 CIF_size3, cif crmodels(cancer other) ci          ///
>                            timevar(tt)                               ///
>                            at1(size 1)                                ///
>                            at2(size 3)                                ///
>                            contrast(difference) contrastvar(cifdiff) ///
>                            frame(cifstand, replace)
```

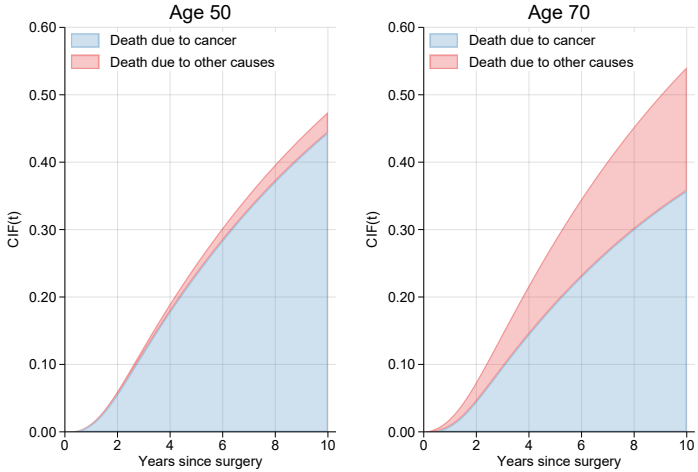
Competing Risks: Predictions



Graph code in competing_risks_example.do

Death due to cancer (solid) and other causes (dashed)

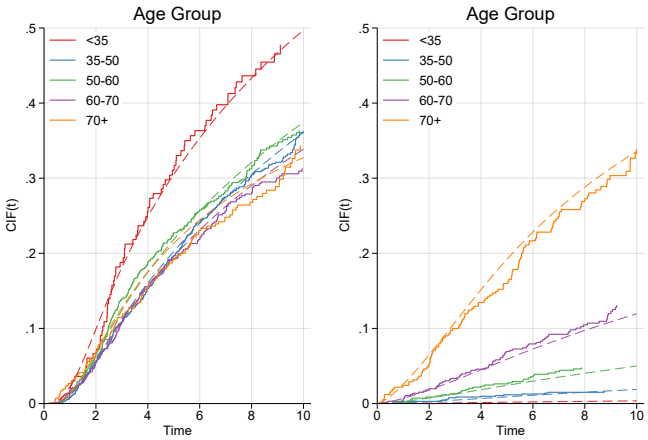
Competing Risks: Prediction - Stacked Graphs



Graph code in competing_risks_example.do

Assess model fit with stpm3aj

```
. stpm3aj i.agegrp , crmodels(cancer other) compet1(2)
```



Graph code in competing_risks_example.do

Competing Risks Extensions

- Causal Inference and competing risks using `standsurv` [21].
- Competing risks and prognostic models [22].
- Parametric version of Fine and Gray model [23, 24].

Competing causes

- Individuals diagnosed with a specific cancer are at risk for
  - dying from their cancer.
  - dying from other causes.

$$h(t|Z_i) = h_o(t|Z_i) + h_c(t|Z_i)$$

- If we had reliable, accurate cause of death information we can estimate $h_c(t|Z_i)$. This is just a cause-specific analysis.
- However, lots of evidence that death certificates not well completed. Accuracy varies ...
  - over time
  - between countries
  - between cancers
  - by age (particularly poor in the elderly)

So we estimate $h_c(t|Z_i)$ without using cause of death information.

Excess mortality/Relative survival

Incorporate expected mortality rates

$$\begin{array}{ccccc} \text{All-cause mortality} & = & \text{expected mortality} & + & \text{excess mortality} \\ h(t|Z_i) & = & h^*(t|Z_i) & + & \lambda(t|Z_i) \end{array}$$

- Need expected rates stratified by levels of Z, e.g. (age, sex, calendar year, region, deprivation group, ...).
- In a perfect world $h_c(t|Z_i) = \lambda(t|Z_i)$.
- The world is not perfect....

Transform to survival

$$S(t|Z_i) = S^*(t|Z_i)R(t|Z_i)$$

$$R(t|Z_i) = \frac{S(t|Z_i)}{S^*(t|Z_i)} \quad \text{hence 'relative survival'}$$

Merging in expected mortality

- The expected mortality at the time of death is required.
- Make use of `stset` information to obtain attained age and calendar year.

```
 . use breast_NW
(Ch28 Adult Breast 174, 175)
.
. stset dateexit, origin(datediag) failure(dead==1) ///
> exit(time datediag+5*365.24) id(ident) scale(365.24)
  (output omitted)
.
. gen age = int(min(agediag + _t,99))
. gen year = year(datediag+ _t*365.24)
.
. //Merge in the expected rate
. merge m:1 sex dep year age using popmort_NW ///
>       , keepusing(rate) keep(match)
    Result                      Number of obs
    -----------------------------------------
    Not matched                             0
    Matched                            14,823  (_merge==3)
    -----------------------------------------
```

Adding the bhazard() option

- If we add the bhazard(rate) option we fit a relative surival model. Then
  - predict, survival ... wil predict relatve survival
  - predict, hazard ... will predict excess hazard rates
  - Exactly the same in standsurv.
- Thus, all we have learnt about 'standard' models apply to relative survival models.

```
 . //Excess mortality model - Proportional excess hazards for agegroup
. stpm3 @ns(agediag, df(3))##i.dep, scale(lncumhazard) df(7) bhazard(rate) neq(0) ///
>                                tvc(@ns(agediag,df(3)) i.dep) dftvc(3) nolog
                                                      Number of obs = 14,823
                                                      Wald chi2(19) = 365.32
Log likelihood = -17280.981                           Prob > chi2   = 0.0000
------------------------------------------------------------------------------
                             |  Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-----------------------------+------------------------------------------------
Extended functions
 (1) @ns(agediag, df(3))
```

Some predictions just for relative survival models

$$S(t|Z_i) = S^*(t|Z_i)R(t|Z_i)$$

- In relative survival models we can either predict relative survival or all cause survival
- We all-cause survival measures (also crude probabilities and life expectancy) we need to incorporate expected rates.
- This is done using the `expsurv()` option that merges in the expected rates.

Predictions

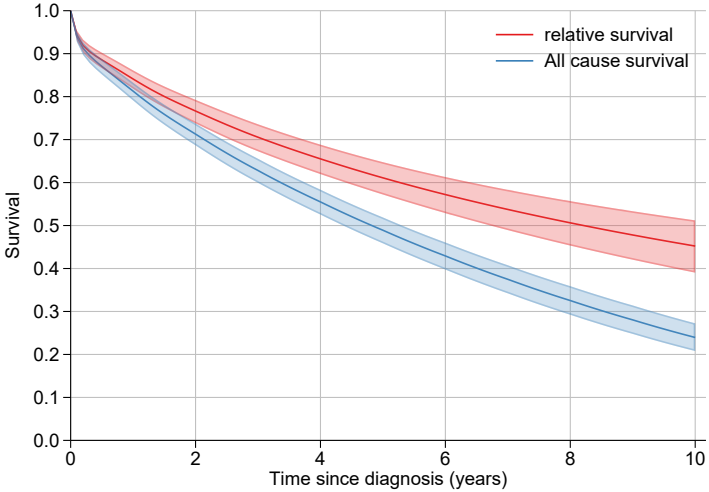
- Predict relative survival.

```
. predict RS_a75_d1, survival at1(agediag 75 dep 1) ci    ///
>           timevar(0 10, step(0.1))                      ///
>           frame(rs_pred,replace)
Predictions are stored in frame - rs_pred
```

- Predict all cause survival

```
. predict AC_a75_d1, survival at1(agediag 75 dep 1) ci  ///
>           frame(rs_pred,merge)                        ///
>           expsurv(using(popmort_NW)                   ///
>           agediag(75) datediag(1990-1-1)              ///
>           pmage(age) pmrate(rate) pmyear(year)        ///
>           pmother(sex dep) at1(sex 2 dep 1)           ///
>                       pmmaxyear(1995) pmmaxage(99))
Predictions are stored in frame - rs_pred
```

We can then plot and compare



Graph code in stpm3_relative_survival.do

Lots more on relative survival

- Loss in life expectancy[25, 26]
- Causal Inference[27] and mediation analysis[28].
- Reference adjustment[29, 30]
- A marginal model for relative survival[31]

Using constraints

- Most Stata estimaton commands allow incorporation of constraints on parameters and `stpm3` is no exception.
- In `stpm2` there was a `cure` option which enabled cure models to be fitted. This applied constraints on some parameters.
- I have not implemented cure models in `stpm3`, partly because I am far less keen on them than I used to be.
- I will use constraints in two examples
  1. Constraining the excess hazard to be zero after the last knot and therefore fitting a cure model.
  2. Constraining a time-dependent hazard ratio to be constant, i.e. proportional, after a specified time.

Cure models

- Recall that in a relative survival model the excess hazard is made up two components, the expected mortality rate, $h^*(t)$ and the excess mortality rate, $\lambda(t)$.

$$h(t|X,Z) = h^*(t|X,Z) + \lambda(t|X,Z)$$

- If at some timepoint the excess mortality rate, $\lambda(t)$ is at, and remains at zero, we have statistical cure.
- This means that those still at risk are dying as the same rate as would be expected in the general population.
- One of the nice things about natural splines is that the final spline variable is the gradient of the linear effect after the last knot.
- In a flexible parametric cure model the gradient of the log cumualtive excess hazard is zero after the last knots, the survival function will stabalize to platea. In other words the excess hazard is zero[32].

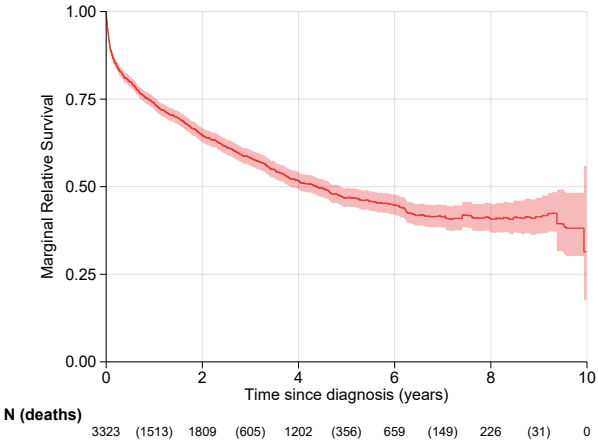
NW England Breast Cancer Example

- Only look at those aged 75+
- Load data and plot non-parametrcic Pohar Perme estimate.

```
 . use breast_nw if agegrp==5
(Ch28 Adult Breast 174, 175)
. stset survtime, f(dead=1) id(ident)
  (output omitted)
.
. stpp R_pp using "https://pclambert.net/data/popmort_NW.dta", ///
>                 agediag(agediag) datediag(datediag)                          ///
>                 pmage(age) pmyear(year) ///
>                 pmother(sex dep)  graphname(R_pp2, replace)
```

- If cure is a reasonable assumption, then we should see the relative survival curve plateau.

Pohar Perme estimate of relative survival



- Cure seems a resonable assumption here.

Fit initial model (not assuming cure)

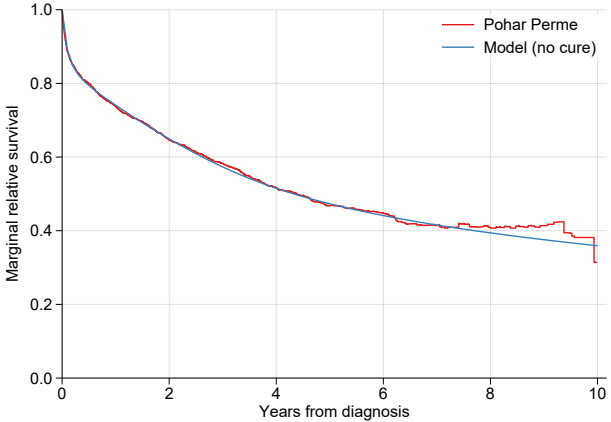
- Only modelling age here.

```
    . stpm3 @ns(agediag, df(3)), scale(lncumhazard) df(5) ///
>                             bhazard(rate) nolog
                                                  Number of obs =  3,323
                                                  Wald chi2(3)  = 106.14
Log likelihood = -4684.9507                       Prob > chi2   = 0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
_ns_f1_agediag1	-3.652952	.8636969	-4.23	0.000	-5.345767	-1.960137
_ns_f1_agediag2	.2453203	.2238792	1.10	0.273	-.193475	.6841155
_ns_f1_agediag3	-.421499	.5111492	-0.82	0.410	-1.423333	.5803351
time						
_ns1	-15.92283	.544663	-29.23	0.000	-16.99035	-14.85531
_ns2	4.738218	.2797482	16.94	0.000	4.189921	5.286514
_ns3	-1.456713	.0726732	-20.04	0.000	-1.59915	-1.314277
_ns4	-.8845034	.0611872	-14.46	0.000	-1.004428	-.7645787
_ns5	-.488664	.0960268	-5.09	0.000	-.6768731	-.3004549
_cons	.8267461	.2568451	3.22	0.001	.323339	1.330153

```
Extended functions
 (1) @ns(agediag, df(3))
```

Compare to Pohar Perme estimate of relative survival



Graph code in constraints_cure_model.do

- Reasonable fit

Adding an extra knot

- In our previous work on cure models we found that models fitted better if we added an extra knot towards the end of follow-up.
- The default in `stpm2` was to add a knot at the 95th percentile.
- We also found it sometime useful to place the upper boundary knot after the end of follow-up.
- I will use the `knots()` option to specify internal knots at specific percentiles.
- You can control the location of boundary knots using the `bknots()` option or specify all knots using the `allknots()` option.

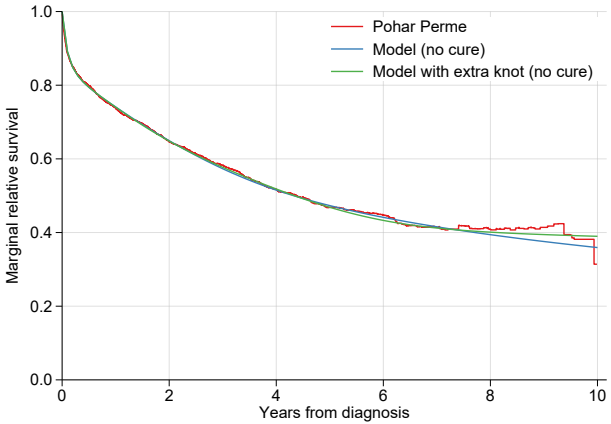
Fit model with additional knot (still not assuming cure)

```
. stpm3 @ns(agediag, df(3)), scale(lncumhazard)              ///
>       knots(20 40 60 80 95, percentile) bhazard(rate) nolog
                                                    Number of obs =  3,323
                                                    Wald chi2(3)  = 105.30
Log likelihood = -4682.6021                         Prob > chi2   = 0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
_ns_f1_agediag1	-3.675761	.8637086	-4.26	0.000	-5.368599	-1.982923
_ns_f1_agediag2	.2576028	.223777	1.15	0.250	-.1809921	.6961977
_ns_f1_agediag3	-.43859	.5111091	-0.86	0.391	-1.440345	.5631654
time						
_ns1	-15.69225	.5561798	-28.21	0.000	-16.78235	-14.60216
_ns2	4.762887	.2806516	16.97	0.000	4.21282	5.312954
_ns3	-1.37851	.081273	-16.96	0.000	-1.537802	-1.219217
_ns4	-.7836407	.076685	-10.22	0.000	-.9339405	-.6333408
_ns5	-.4101389	.0684764	-5.99	0.000	-.5443502	-.2759276
_ns6	-.0743801	.1026938	-0.72	0.469	-.2756563	.126896
_cons	.7455238	.2595853	2.87	0.004	.2367459	1.254302

```
Extended functions
 (1) @ns(agediag, df(3))
```

Compare to Pohar Perme estimate of relative survival



Graph code in constraints_cure_model.do

- Model estimate has reduced gradient towards end of follow-up

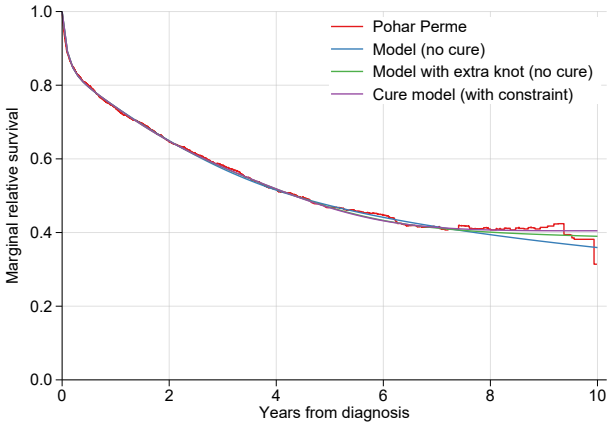
Add a constraint to fit a cure model

```
    . constraint 1 _ns6==0
. stpm3 @ns(agediag, df(3)), scale(lncumhazard) knots(20 40 60 80 95, percentile) ///
>                            bhazard(rate) constraints(1) nolog
                                                    Number of obs =  3,323
                                                    Wald chi2(3)  = 105.16
Log likelihood = -4682.8677                         Prob > chi2   = 0.0000
 ( 1)  [time]_ns6 = 0
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
_ns_f1_agediag1	-3.676204	.8639616	-4.26	0.000	-5.369538	-1.982871
_ns_f1_agediag2	.258445	.2239541	1.15	0.248	-.1804969	.6973869
_ns_f1_agediag3	-.4389886	.5112358	-0.86	0.391	-1.440992	.5630151
time						
_ns1	-15.56957	.5304502	-29.35	0.000	-16.60924	-14.52991
_ns2	4.766312	.2807062	16.98	0.000	4.216138	5.316486
_ns3	-1.337454	.0587382	-22.77	0.000	-1.452579	-1.222329
_ns4	-.7394701	.0469931	-15.74	0.000	-.8315749	-.6473653
_ns5	-.373215	.0459909	-8.11	0.000	-.4633556	-.2830744
_ns6	0	(omitted)				
_cons	.703227	.2530923	2.78	0.005	.2071753	1.199279

```
Extended functions
 (1) @ns(agediag, df(3))
```

Compare to Pohar Perme estimate of relative survival



Graph code in constraints_cure_model.do

- We are now fitting a simple cure model!

Predict cure

- The estimate cure proportion is just the predicted relative survival at or after the last knot.
- We use `predict` or `standsurv` in the same way as any other model.
- The marginal cure proportion, i.e averaged over all individuals is....

```
. gen t10 = 10 in 1
(3,322 missing values generated)
. standsurv cure, surv timevar(t10) frame(cure, replace) ci at1(.)
. frame cure: list cure*, noobs
```

cure	cure_lci	cure_uci
.40480216	.37698911	.43466716

Predict cure differece

- The difference in the cure proportion between 75 and 85 years olds is.

```
. predict cure75 cure85, surv timevar(t10) ci frame(cure2, replace)   ///
>                        at1(agediag 75) at2(agediag 85)              ///
>                        contrast(difference) contrastvar(curediff)
Predictions are stored in frame - cure2
. frame cure2: list cure75* cure85*, noobs
```

cure75	cure7~lci	cure7~uci	cure85	cure8~lci	cure8~uci
.55255618	.47924827	.61976937	.31430022	.27348844	.35585248

```
. frame cure2: list curediff*, noobs abbrev(12)
```

curediff	curediff_lci	curediff_uci
-.23825596	-.31384277	-.16266915

- All this would be easier with specific options or a wrapper command.

Constraining hazard ratios

- An advantage of FPSMs is the ability the model non-proportional hazards.
- Sometimes it may be useful to constrain a hazard ratio to be proportional after a specific time.
- This has been discussed in the context of extrapolation in health economics[33].
- We used constraints to impose 'treatment waning' on a hazard ratio [Jennings under review].
- It may help with some non-convergence problems???
- The following few slides give the general idea how how to do this using constraints.
- It is more natural to constrain hazard ratios on the log hazard scale.

Breast NW data: Time-dependent effect for deprivation group.

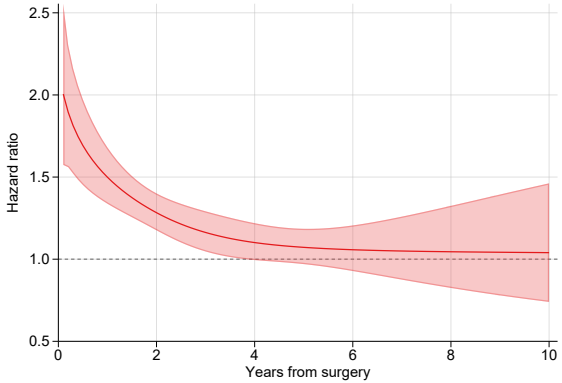
```
. stpm3 i.dep, scale(lnhazard) df(5) tvc(i.dep) dftvc(3) nolog vsquish
                                                Number of obs =  9,719
                                                Wald chi2(1)  =   0.06
Log likelihood = -10314.656                     Prob > chi2   = 0.8072
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
mostdep	.0400289	.1640139	0.24	0.807	-.2814324	.3614901
time						
_ns1	2.996079	1.063506	2.82	0.005	.9116456	5.080513
_ns2	-1.856115	.5998345	-3.09	0.002	-3.031769	-.6804611
_ns3	1.30735	.145643	8.98	0.000	1.021895	1.592805
_ns4	.8237268	.1313701	6.27	0.000	.5662462	1.081207
_ns5	1.126959	.2571798	4.38	0.000	.6228955	1.631022
dep#c._ns_tvc1						
mostdep	2.117289	1.509185	1.40	0.161	-.8406583	5.075236
dep#c._ns_tvc2						
mostdep	.0748991	.8512222	0.09	0.930	-1.593466	1.743264
dep#c._ns_tvc3						
mostdep	.0252986	.2991369	0.08	0.933	-.5609989	.6115961
_cons	-3.643122	.1292035	-28.20	0.000	-3.896356	-3.389888

```
Quadrature method: tanh-sinh with 30 nodes.
Analytical integration before first and after last knot.
```

Calculate hazard ratio

```
. predict , hazard timevar(0 10,n(101)) ci frame(f1, replace) nogen ///
>           at1(dep 1) at2(dep 5)                                   ///
>           contrast(ratio) contrastvar(hr1)
Predictions are stored in frame - f1
```



Graph code in constraints_hazard_ratio.do

Thinking about constraints

- I will constrain the hazard ratio to be proportional from 5 years.
- The time-dependent effect for deprivation is due to includion of spline variables calculated when using the `tvc()` and `dftvc()` options.
- I need to define the knots to be in the range (0,5) with the upper boundary knots at 5.
- I do this before running `stpm3`
- Like the cure models I will add an extra knot at the 95 percentile of the event times.
- Note that there are no constraints on the spline terms for the baseline.

Calculating the knot positions

```
. summ _t, meanonly
. global mintime `r(min)´
. _pctile _t if _d & _t<=5, p(33.33 66.67 95 )
. global allknots  ${mintime} `r(r1)´ `r(r2)´ `r(r3)´ `r(r4)´ 5
. di "${allknots}"
.003000000026077 1.560999989509583 2.913000106811523 4.651999950408936  5
```

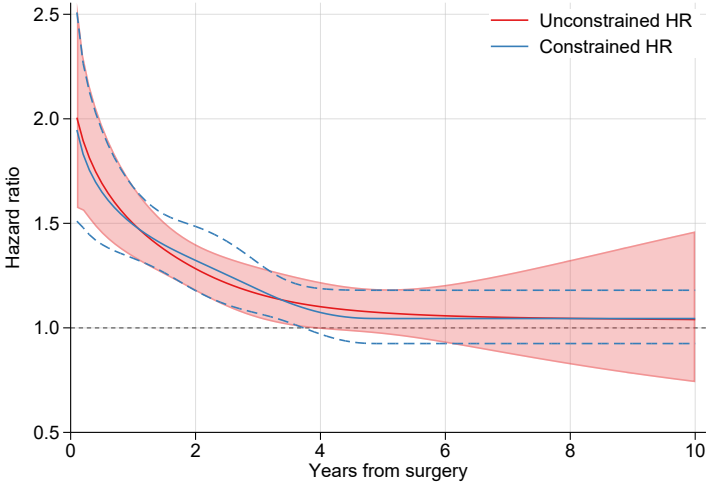
Fit model with user defined kinots and constraint

```
. constraint 1 5.dep#c._ns_tvc4 = 0
. stpm3 i.dep, scale(lnhazard) df(5) tvc(i.dep) allknotstvc(${allknots}) ///
>              constraints(1) nolog vsquish
  (output omitted)
 ( 1)  [time]5.dep#c._ns_tvc4 = 0
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
dep						
mostdep	.0437316	.0621109	0.70	0.481	-.0780036	.1654668
time						
_ns1	2.88449	1.079271	2.67	0.008	.7691566	4.999823
_ns2	-1.720327	.637552	-2.70	0.007	-2.969906	-.4707477
_ns3	1.299459	.1369801	9.49	0.000	1.030983	1.567935
_ns4	.8322954	.1170678	7.11	0.000	.6028467	1.061744
_ns5	1.157298	.2504617	4.62	0.000	.6664016	1.648193
dep#c._ns_tvc1						
mostdep	2.380574	1.426375	1.67	0.095	-.4150694	5.176218
dep#c._ns_tvc2						
mostdep	-.2183354	.901481	-0.24	0.809	-1.985206	1.548535
dep#c._ns_tvc3						
mostdep	.1765361	.1176695	1.50	0.134	-.0540919	.4071641
dep#c._ns_tvc4						
mostdep	0	(omitted)				
_cons	-3.651555	.1204318	-30.32	0.000	-3.887597	-3.415513

```
Quadrature method: tanh-sinh with 30 nodes.
Analytical integration before first and after last knot.
```

Compare constrained and unconstrained hazard ratio



Graph code in constraints_hazard_ratio.do

Some thoughts on constraints

- I did the two examples using constraints last week.
- I like the idea of constraining hazard ratios.
- Time-dependence is most common early after diagnosis, but we put effort into modelling non-proportional hazards throughout the time-scale.
- Could be very useful for extrapolation.
- Need to think about sensitivity to knot positions. Is the 95th percentile sensible for the additional knot?

Calibration

- A prognostic model is a regression model intended to enable prediction of future outcomes given values of several covariates measures at or before the time origin.
- Used to make health care decisions, e.g. treatment, timings of follow-up etc.
- We are interested in both callibration and discrimination of the model.
- A common way to assess calibration is a calibration plot.

Calibration the agreement between observed and predicted probabilities.

Discrimination the ability of the prognostic model to distinguish between patients at different levels of risk

Calibration plots

- A visual tool to assess agreement between predicted and observed probabilities.
- With survival data (due to censoring) often define groups based on predicted probabilities and compare marginal predictions with non-parametric estimates.
- More recently use pseudo observations to enable visualization over the complete range of predictions[34].
- Useful to add other summaries of model performance to plot.
- `stpm3calplot` does some of this work for you.
- It will be added in a future release, soon(ish).

Rotterdam data

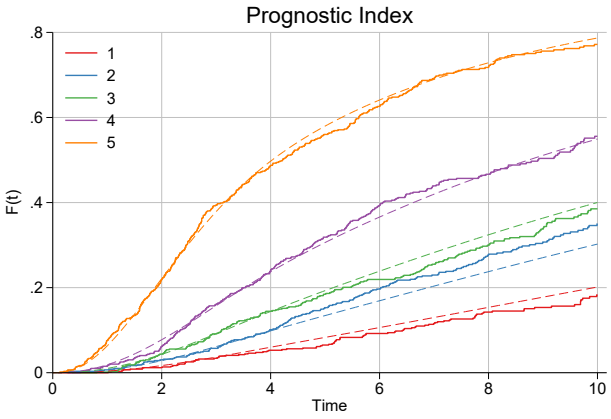
```
. stpm3 age @fn(exp(-0.12 * nodes),stub(enodes)) i.size i.hormon i.grade pr_1, ///
>       scale(lnodds) df(4) neq(1) nolog
                                                Number of obs =  2,982
                                                Wald chi2(7)  = 604.36
Log likelihood = -2607.772                      Prob > chi2   = 0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
xb						
age	.0148001	.0029896	4.95	0.000	.0089405	.0206596
_fn_enodes	-2.664496	.1550357	-17.19	0.000	-2.96836	-2.360631
size						
>20-50mmm	.4698654	.0854911	5.50	0.000	.3023059	.6374249
>50 mm	.8191977	.1311011	6.25	0.000	.5622443	1.076151
hormon						
yes	-.4521206	.1220432	-3.70	0.000	-.6913209	-.2129203
3.grade	.3962003	.0933199	4.25	0.000	.2132966	.579104
pr_1	-.138221	.0176075	-7.85	0.000	-.172731	-.103711

```
(1) @fn(exp(-0.12 * nodes), stub(enodes))
```

Calibration: `stpm3km` with failure option

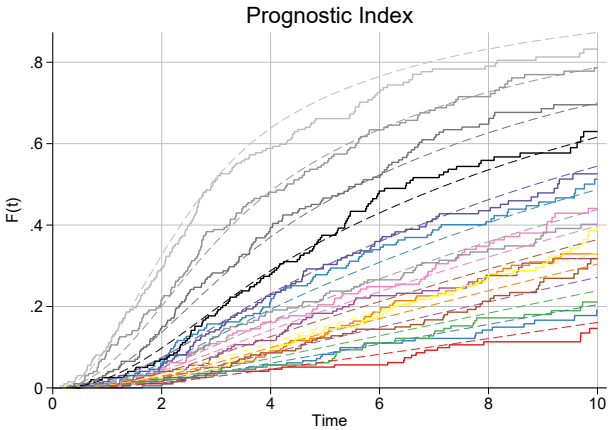
```
. stpm3km, failure
```



Graph code in stpm3calplot.do

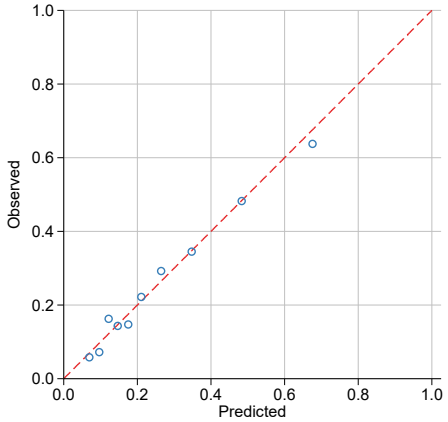
Calibration: stpm3km with lots of groups

```
. stpm3km, groups(15) legend(off)
```



Calibration: stpm3calplot at 5 years

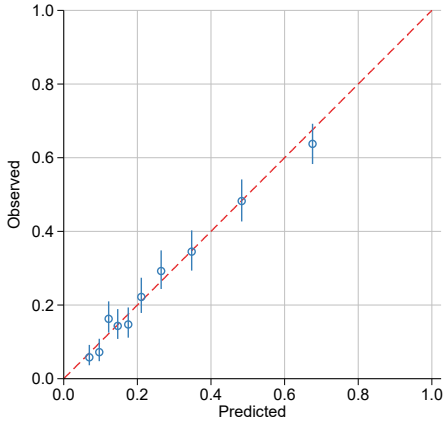
```
. stpm3calplot, time(5)
```



Graph code in stpm3calplot.do

Calibration: stpm3calplot with Observed CIs

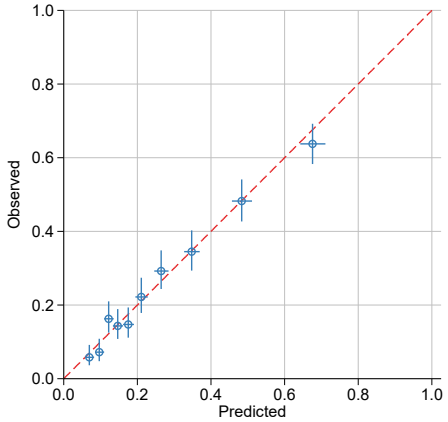
```
. stpm3calplot, time(5) ciobs
```



Graph code in stpm3calplot.do

Calibration: stpm3calplot with Expected CIs

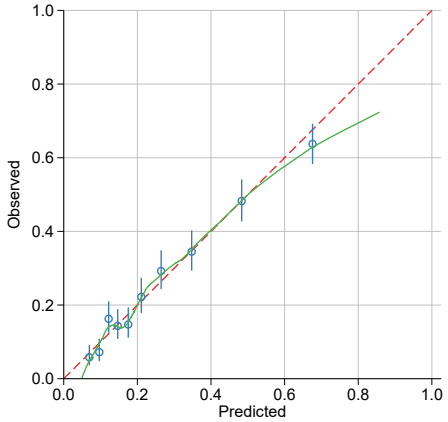
```
. stpm3calplot, time(5) cipred
```



Graph code in stpm3calplot.do

Calibration: stpm3calplot with pseudo observations smoother

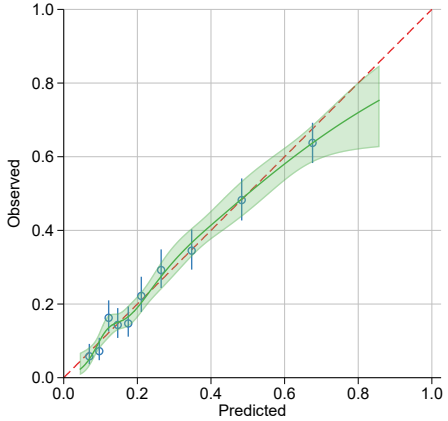
```
. stpm3calplot, time(5) ciobs pseudo
```



Graph code in stpm3calplot.do

Calibration: stpm3calplot with pseudo observations smoother

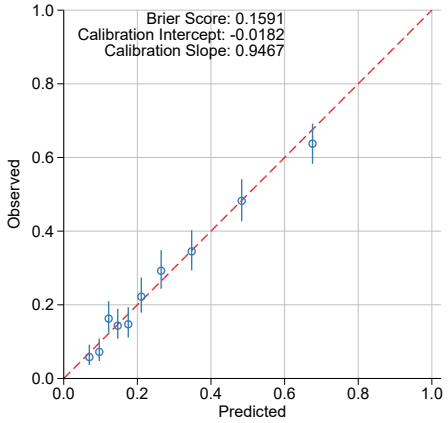
```
. stpm3calplot, time(5) pseudo smoother(ns) smootherci
```



Graph code in stpm3calplot.do

Calibration: stpm3calplot with performance statistics

```
. stpm3calplot, time(5) ciobs pseudo smoother(glm) smootherci ///
          stats(brier calint calslope)
```



Graph code in stpm3calplot.do

Final thoughts

Final thoughts and wrap-up

- We have found these models very useful in a range of areas from simple descriptive models, to prognostic models and causal models.
- The Cox model is a great approach, but I think in most cases the advantages of using of these flexible models outweighs the disadvantages.
- I hope I have convinced you of this (if you were not convinced already).
- I think the most powerful part of `stpm3` is the predictions, both conditional and marginal. I can fit a complex Cox model with non-linear effects , relax the proportional hazards assumption etc, but then it is much, much harder to obtain predictions, obtain marginal estimates etc etc.

stpm3

- Let me know if you find any bugs in `stpm3`.
- If you have a question, it is far easier for me if you create a small working example that demonstrates the problem (you are much more likely to get a response).
- Some things that may or may not come in the future.
  - `stpm3calplot` will be released when it is tidied up.
  - In `standsurv` you should be able to specify `timevar(0 10, step(0.1))` rather than create the variable yourself.
  - `tvcoffset()` option: would allow multiple time-scales for `scale(lnhazard)` models.
  - Anything that improves convergence or gives you explanation of what the problem is.
  - Think of useful syntax for introducing constraints.
- Please add suggestions of your own.

Last slide!

- Any final questions?.
- Thank you attending!

References

[1] Durrleman S, Simon R. Flexible regression models with cubic splines. *Statistics in Medicine* 1989; **8**:551–561.

[2] Wang W, Yan J. Shape-restricted regression splines with R package splines2. *Journal of Data Science* 2021;**19**:498–517.

[3] Aalen OO, Cook RJ, RÅÂ¸ysland K. Does cox analysis of a randomized survival study yield a causal treatment effect? *Lifetime data analysis* 2015;**21**:579–593.

[4] Hernán MA. The hazards of hazard ratios. *Epidemiology* 2010;**21**:13–15.

[5] Sutradhar R, Austin PC. Relative rates not relative risks: addressing a widespread misinterpretation of hazard ratios ;**28**:54–57.

[6] Stensrud MJ, Aalen JM, Aalen OO, Valberg M. Limitations of hazard ratios in clinical trials. *European heart journal* 2018;.

[7] Reid N. A conversation with Sir David Cox. *Statistical Science* 1994;**9**:439–455.

[8] Royston P. Flexible parametric alternatives to the Cox model, and more. *The Stata Journal* 2001;**1**:1–28.

[9] Lambert PC, Royston P. Further development of flexible parametric models for survival analysis. *The Stata Journal* 2009;**9**:265–290.

References 2

[10] Cupples LA, Gagnon DR, Ramaswamy R, D'Agostino RB. Age-adjusted survival curves with application in the Framingham study. *Statistics in Medicine* 1995;**14**:1731–1744.

[11] Rutherford MJ, Crowther MJ, Lambert PC. The use of restricted cubic splines to approximate complex hazard functions in the analysis of time-to-event data: a simulation study. *Journal of Statistical Computation and Simulation* 2015;**85**:777–793.

[12] Royston P, Altman DG, Sauerbrei W. Dichotomizing continuous predictors in multiple regression: a bad idea. *Statistics in Medicine* 2006;**25**:127–141.

[13] Royston P, Sauerbrei W. *Multivariable model-building: A pragmatic approach to regression analysis based on fractional polynomials for modelling continuous variables.*. Wiley, 2008.

[14] Colzani E, Johansson ALV, Liljegren A, Foukakis T, Clements M, Adolfsson J, *et al.*. Time-dependent risk of developing distant metastasis in breast cancer patients according to treatment, age and tumour characteristics. *Br J Cancer* 2014;**110**:1378–1384.

[15] Vansteelandt S, Keiding N. Invited commentary: G-computation–lost in translation? *Am J Epidemiol* 2011;**173**:739–742.

[16] Samuelsen SO, Eide GE. Attributable fractions with survival data. *Stat Med* 2008;**27**:1447–1467.

References 3

[17] Bennett D. Analysis of survival data by the proportional odds model. *Statistics in Medicine* 1983; **2**:273–277.

[18] Crowther M, Lambert P. A general framework for parametric survival analysis. *Statistics in Medicine* 2014; **33**:5280–5297.

[19] Crowther MJ, Lambert PC. stgenreg: A stata package for general parametric survival analysis. *Journal of Statistical Software* 2013;**53**:1–17.

[20] Bower H, Crowther MJ, Lambert PC. strcs: A command for fitting flexible parametric survival models on the log-hazard scale. *The Stata Journal* 2016;**16**:989–1012.

[21] Syriopoulou E, Mozumder SI, Rutherford MJ, Lambert PC. Estimating causal effects in the presence of competing events using regression standardisation with the stata command standsurv. *BMC Medical Research Methodology* 2022;**22**:226.

[22] Booth S, Mozumder SI, Archer L, Ensor J, Riley RD, Lambert PC, Rutherford MJ. Using temporal recalibration to improve the calibration of risk prediction models in competing risk settings when there are trends in survival over time. *Statistics in Medicine* 2023;**30**:5007–5024.

[23] Lambert PC, Wilkes SR, Crowther M. Flexible parametric modelling of the cause-specific cumulative incidence function. *Statistics in Medicine* 2017;**36**:1429–1446.

References 4

[24] Lambert PC. The estimation and modelling of cause-specific cumulative incidence functions using time-dependent weights. *The Stata Journal* 2017;**17**:181–207.

[25] Andersson TML, Dickman PW, Eloranta S, Lambe M, Lambert PC. Estimating the loss in expectation of life due to cancer using flexible parametric survival models. *Statistics in Medicine* 2013;**32**:5286–5300.

[26] Rutherford MJ, Andersson TML, Björkholm M, Lambert PC. Loss in life expectancy and gain in life years as measures of cancer impact. *Cancer Epidemiology* 2019;**60**:168–173.

[27] Syriopoulou E, Rutherford M, Lambert P. Marginal measures and causal effects using the relative survival framework. *International Journal of Epidemiology* 2020;**49**:619–628.

[28] Syriopoulou E, Rutherford MJ, Lambert PC. Understanding disparities in cancer prognosis: An extension of mediation analysis to the relative survival framework. *Biometrical journal* 2021;**63**:341–353.

[29] Lambert PC, Andersson TML, Rutherford MJ, Myklebust TÅ, Møller B. Reference-adjusted and standardized all-cause and crude probabilities as an alternative to net survival in population-based cancer studies. *International Journal of Epidemiology* 2020;**49**:1614–1623.

[30] Rutherford MJ, Andersson TML, Myklebust TÅ, Møller B, Lambert PC. Non-parametric estimation of reference adjusted, standardised probabilities of all-cause death and death due to cancer for population group comparisons. *BMC Medical Research Methodology* 2022;**22**:2.

References 5

[31] Lambert PC, Syriopoulou E, Rutherford MJ. Direct modelling of age standardized marginal relative survival through incorporation of time-dependent weights. *BMC Medical Research Methodology* 2021;**21**:84.

[32] Andersson TML, Dickman PW, Eloranta S, Lambert PC. Estimating and modelling cure in population-based cancer studies within the framework of flexible parametric survival models. *BMC Medical Research Methodology* 2011;**11**:96.

[33] Jackson C, Stevens J, Ren S, Latimer N, Bojke L, Manca A, Sharples L. Extrapolating survival from randomized trials using external data: A review of methods. *Medical Decision Making* 2017;**37**:377–390.

[34] Gerds TA, Andersen PK, Kattan MW. Calibration plots for risk prediction models in the presence of competing risks. *Statistics in Medicine* ;**33**:3191–3203.