# Recent developments in the fitting and assessment of flexible parametric survival models

Paul C Lambert[1,2]

[1]Medical Epidemiology and Biostatistics, Karolinska Institutet, Stockholm, Sweden
[2]Biostatistics Research Group, Department of Population Health Sciences, University of Leicester, UK

2024 German Stata Conference
Mannheim, Germany, 7th June 2024



Slides: pclambert.net/pdf/Stata2024_Germany_Paul_Lambert.pdf

# Outline

This will be a broad talk across a range of areas. I will cover

- What are flexible paramatric survival models?
- Extended functions (non-linearity)
- Predictions and Contrasts
- Standardization
- Assessing the fit of models
- Competing risks
- Log hazard models

# Censored survival data

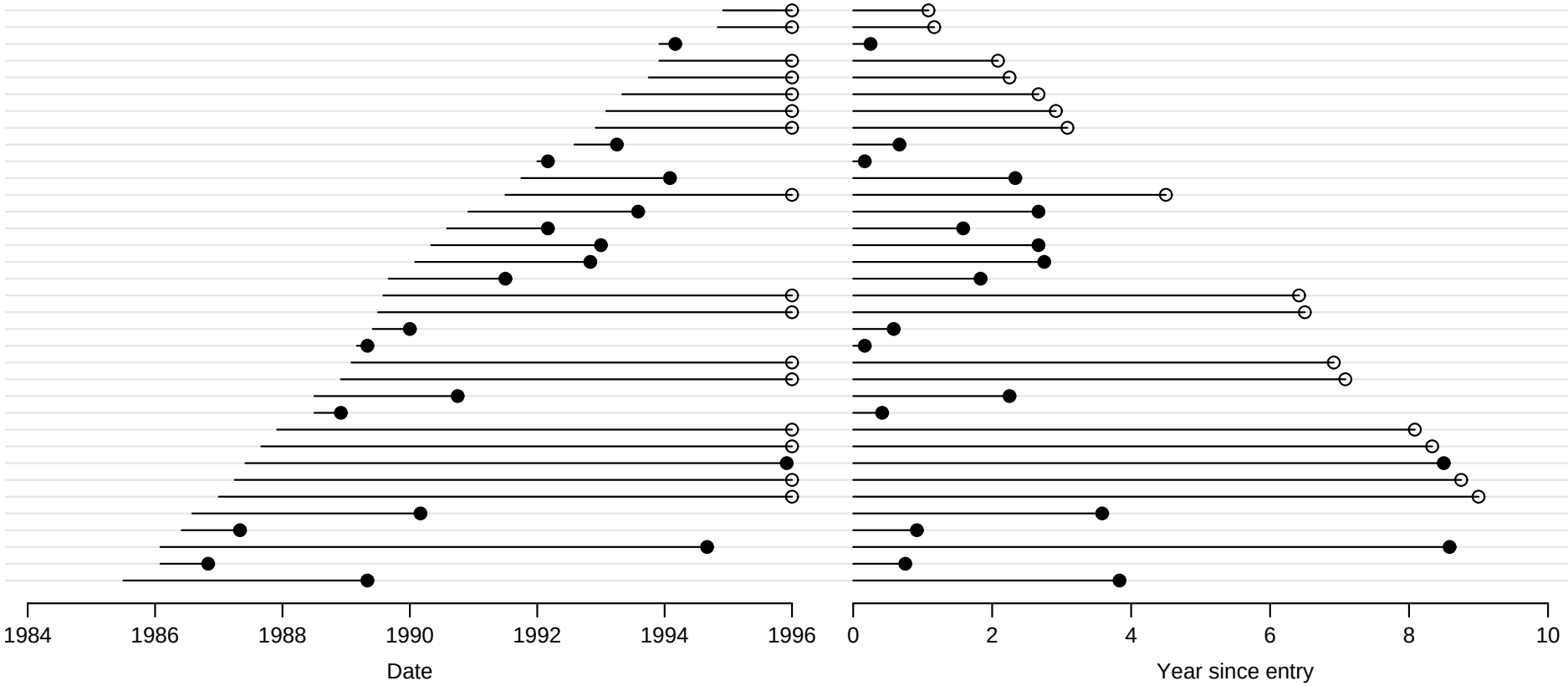



Figure 1: Calendar time (left) and time from entry in years (right)

# What is a parametric survival model

- In a parametric survival model the survival function is expressed as a mathematical function of follow-up time and a set of parameters.
- There is a mathematical relationship between the hazard, survival and density functions, so there are also mathematical function for all these entities.
- There are also parameters for the effects of covariates.
- All 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.

# 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 $\ln(t)$
- 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.

# 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, $s\left(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0\right)$, with knots, $\mathbf{k_0}$, are used to model the log baseline cumulative hazard.

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

# Flexible parametric models: incorporating splines

- We thus model on the log cumulative hazard scale.

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

- This is a proportional hazards model.
- Natural cubic splines, $s(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0)$, with knots, $\mathbf{k_0}$, are used to model the log baseline cumulative hazard.

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

- For example, with 4 knots we can write

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

# Flexible parametric models: incorporating splines

- We thus model on the log cumulative hazard scale.

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

- This is a proportional hazards model.
- Natural cubic splines, $s(\ln(t)|\boldsymbol{\gamma}, \mathbf{k}_0)$, with knots, $\mathbf{k_0}$, are used to model the log baseline cumulative hazard.

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

- For example, with 4 knots we can write

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

# A history of software

Patrick Royston wrote the initial `stpm` command around 2000 [1]

The Stata Journal (2001)
1, Number 1, pp. 1–28

**Flexible parametric alternatives to the Cox model, and more**

Patrick Royston
UK Medical Research Council
patrick.royston@ctu.mrc.ac.uk

# A history of software

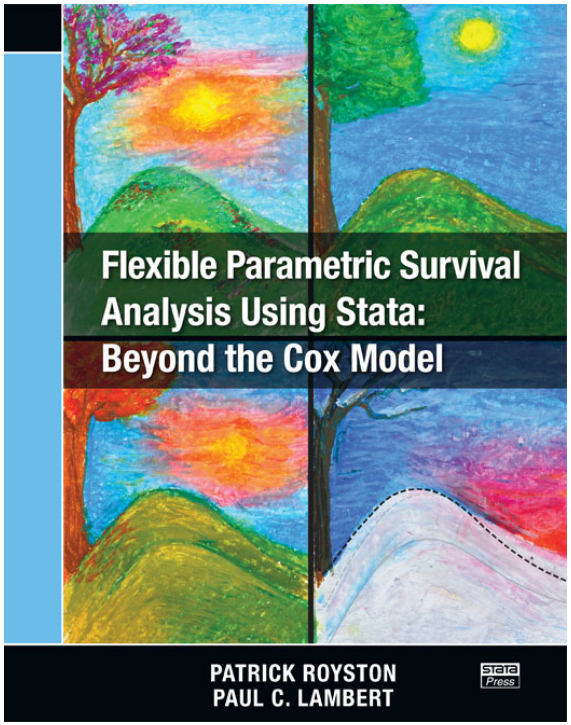
I developed `stpm2` with lots of input from Patrick Royston around 2008. `stpm2` allowed more flexibility for modelling time-dependent effects, relative survival models and more prediction options [2, 3]

The Stata Journal (2009)
**9**, Number 2, pp. 265–290

**Further development of flexible parametric models for survival analysis**

Paul C. Lambert
Centre for Biostatistics and Genetic Epidemiology
Department of Health Sciences
University of Leicester, UK
paul.lambert@le.ac.uk

Patrick Royston
Clinical Trials Unit
Medical Research Council
London, UK
patrick.royston@ctu.mrc.ac.uk



# A history of software

I (with lots and lots of suggestions from Mark Rutherford) wrote `stpm3` in 2022. It incorporated

- Factor variables (properly)
- Extended functions (non-linearity)
- Improved `predict` command
- Prediction to frames
- Better synergy with the `standsurv` postestimation command for obtaining marginal survival curves
- More in Mata (sometimes Python) for speed improvements

# Breast Cancer: Fitting a simple model

```
// Cox Model (deprivation group - 5 levels)
. stcox i.dep

// Flexible parametric survival model
. stpm3 i.dep, scale(lncumhazard) df(5)
```

```
. stpm3 i.dep, scale(lncumhazard) df(5) eform nolog
  (output omitted)
```

| | 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.

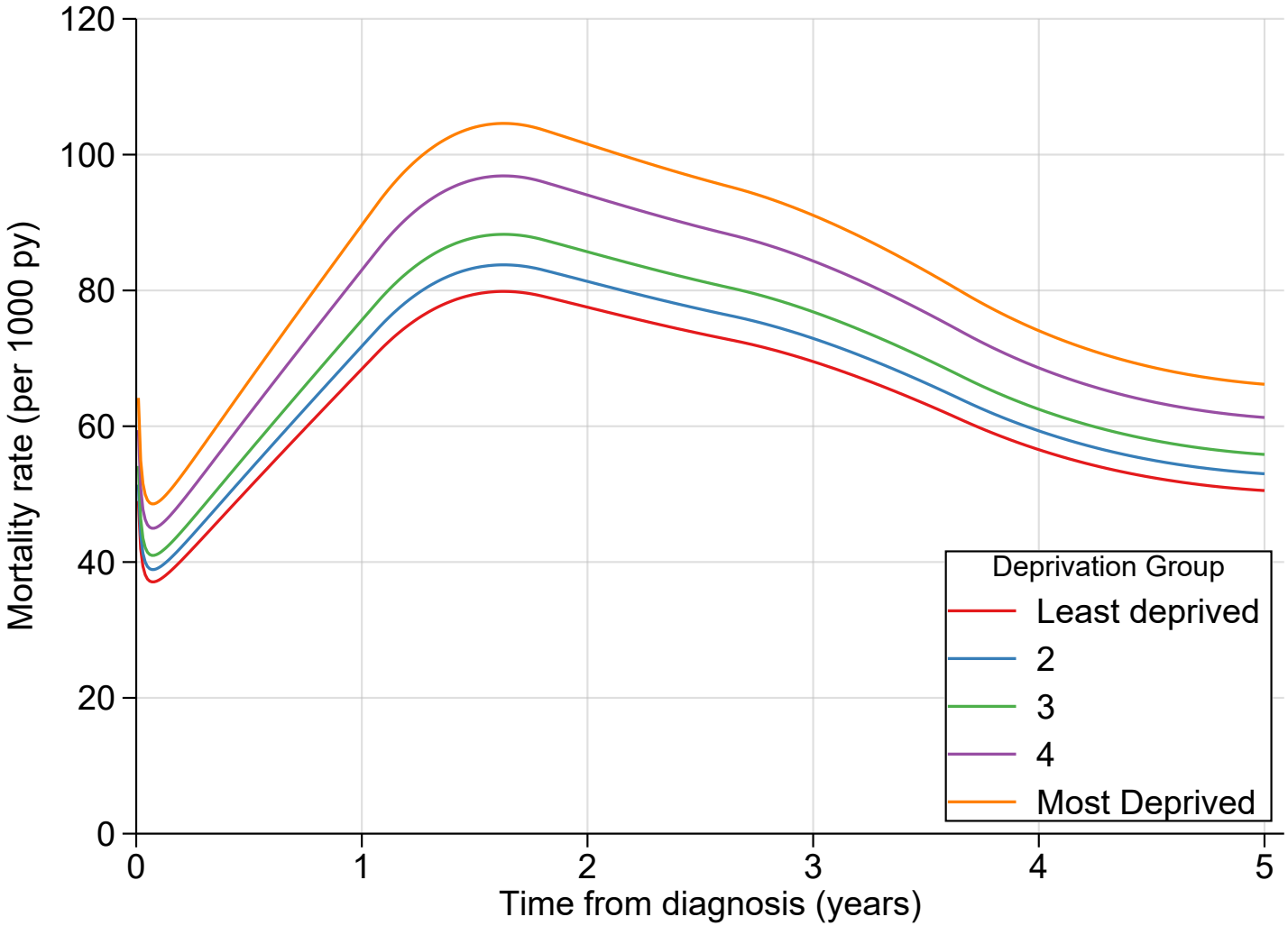
# Comparison of log hazard ratios

```
. estimates table stcox stpm3, eq(1:1)  b(%7.4f)
```

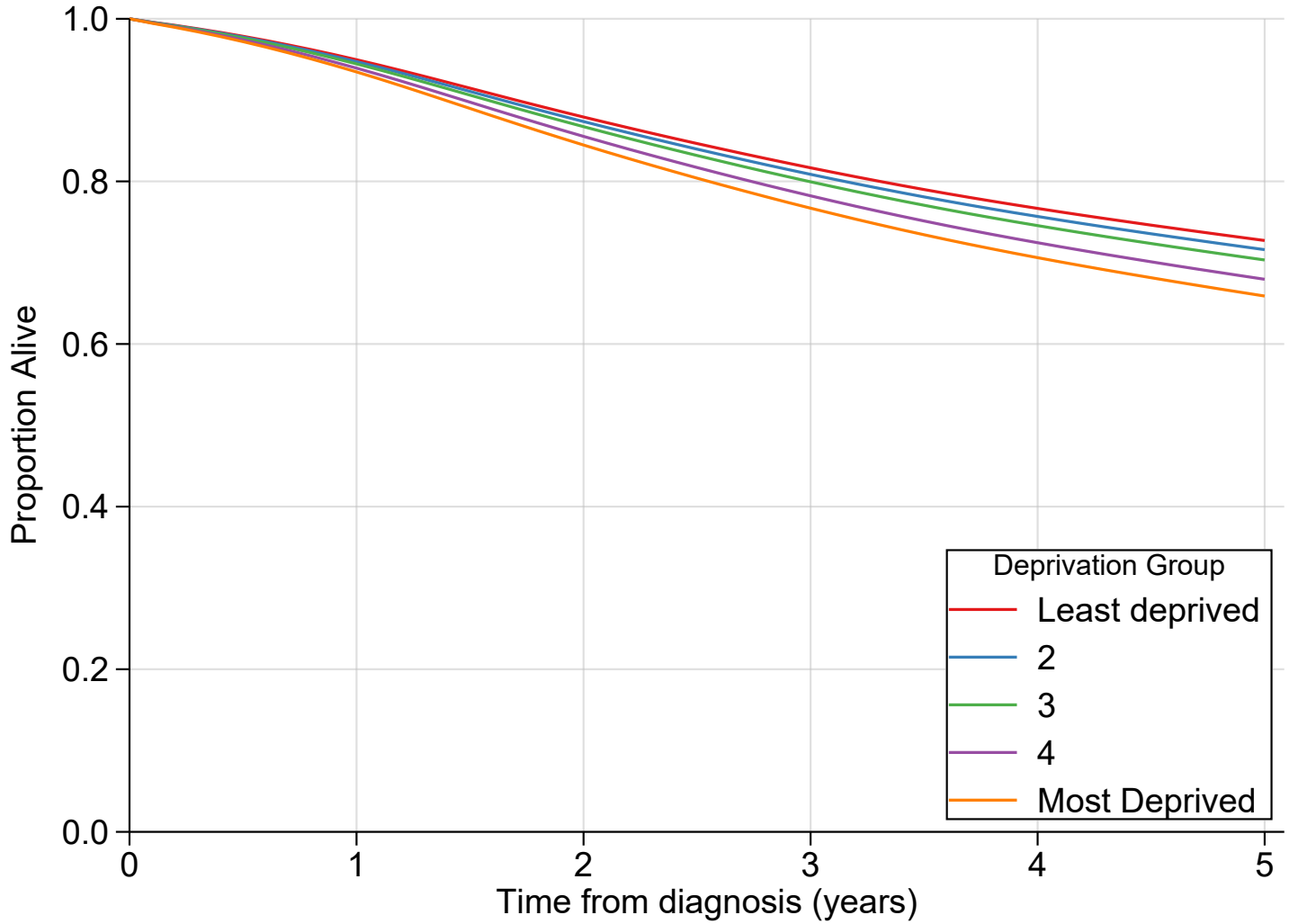
| Variable | stcox | stpm3 |
|---|---|---|
| #1 | | |
| dep | | |
| 2 | 0.0478 | 0.0478 |
| 3 | 0.1001 | 0.1001 |
| 4 | 0.1931 | 0.1931 |
| mostdep | 0.2699 | 0.2699 |
| time | | |
| _ns1 | | -20.5192 |
| _ns2 | | 3.8298 |
| _ns3 | | -1.0750 |
| _ns4 | | -0.6010 |
| _ns5 | | -0.3341 |
| _cons | | -1.1447 |

- Estimation of the baseline (using splines) make postestimation predictions (and uncertainty) much, much easier.

# Hazard and Survival functions



# Hazard and Survival functions



# Extended functions

- If we want to include a spline function for a covariate we usually do something like

```
. gensplines agediag, type(ns) df(3) gen(agens)
. stpm3 i.dep agens1 agens2 agens3, scale(lncumhazard) df(4) nolog
                                                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 | | | | | | |
| dep | | | | | | |
| mostdep | .2530977 | .038471 | 6.58 | 0.000 | .177696 | .3284994 |
| agens1 | -6.731391 | .7054777 | -9.54 | 0.000 | -8.114101 | -5.34868 |
| agens2 | -.6183224 | .3543468 | -1.74 | 0.081 | -1.312829 | .0761844 |
| agens3 | -2.902752 | .2202527 | -13.18 | 0.000 | -3.33444 | -2.471065 |
| time | | | | | | |
| _ns1 | -15.49693 | .4692123 | -33.03 | 0.000 | -16.41657 | -14.57729 |
| _ns2 | 3.966272 | .2513638 | 15.78 | 0.000 | 3.473608 | 4.458936 |
| _ns3 | -1.092355 | .0299249 | -36.50 | 0.000 | -1.151006 | -1.033703 |
| _ns4 | -.5337899 | .0237586 | -22.47 | 0.000 | -.580356 | -.4872238 |
| _cons | 1.439092 | .1061348 | 13.56 | 0.000 | 1.231072 | 1.647112 |

# Extended functions

- Extended functions are included in the model command.

```
. stpm3 i.dep @ns(agediag, df(3)), scale(lncumhazard) df(4) nolog
                                                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 | | | | | | |
| dep | | | | | | |
| mostdep | .2530977 | .038471 | 6.58 | 0.000 | .177696 | .3284994 |
| _ns_f1_agediag1 | -6.731391 | .7054777 | -9.54 | 0.000 | -8.114101 | -5.34868 |
| _ns_f1_agediag2 | -.6183224 | .3543468 | -1.74 | 0.081 | -1.312829 | .0761844 |
| _ns_f1_agediag3 | -2.902752 | .2202527 | -13.18 | 0.000 | -3.33444 | -2.471065 |
| time | | | | | | |
| _ns1 | -15.49693 | .4692123 | -33.03 | 0.000 | -16.41657 | -14.57729 |
| _ns2 | 3.966272 | .2513638 | 15.78 | 0.000 | 3.473608 | 4.458936 |
| _ns3 | -1.092355 | .0299249 | -36.50 | 0.000 | -1.151006 | -1.033703 |
| _ns4 | -.5337899 | .0237586 | -22.47 | 0.000 | -.580356 | -.4872238 |
| _cons | 1.439092 | .1061348 | 13.56 | 0.000 | 1.231072 | 1.647112 |

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

# Types of extended functions

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

# Extended functions

- Multiple extended functions.

```
. stpm3 i.dep @ns(agediag, df(3)) @poly(yeardiag, degree(2)), ///
        scale(lncumhazard) df(4)
```

# Extended functions

- Multiple extended functions.

```
. stpm3 i.dep @ns(agediag, df(3)) @poly(yeardiag, degree(2)), ///
        scale(lncumhazard) df(4)
```

- Interactions with extended functions

```
. stpm3  i.dep##@ns(agediag, df(3)), ///
        scale(lncumhazard) df(4)
```

# Extended functions

- Multiple extended functions.

```
. stpm3 i.dep @ns(agediag, df(3)) @poly(yeardiag, degree(2)), ///
        scale(lncumhazard) df(4)
```

- Interactions with extended functions

```
. stpm3  i.dep##@ns(agediag, df(3)), ///
        scale(lncumhazard) df(4)
```

- Interactions between extended functions

```
. stpm3 i.dep @ns(agediag, df(3))##@ns(yeardiag, degree(2)), ///
        scale(lncumhazard) df(4)
```

# Extended functions

- Multiple extended functions.

```
. stpm3 i.dep @ns(agediag, df(3)) @poly(yeardiag, degree(2)), ///
        scale(lncumhazard) df(4)
```

- Interactions with extended functions

```
. stpm3  i.dep##@ns(agediag, df(3)), ///
        scale(lncumhazard) df(4)
```

- Interactions between extended functions

```
. stpm3 i.dep @ns(agediag, df(3))##@ns(yeardiag, degree(2)), ///
        scale(lncumhazard) df(4)
```

- Using extended functions makes many predictions much simpler.

# Relaxing the proportional hazards assumption

- We often make the proportional hazards assumption in survival models.
- Sometimes reasonable, but needs to be assessed.
- Under proportional hazards the effect of a covariate is assumed the same at all time points.
- Non-proportional hazards means there is an interaction between a covariate and follow-up time.
- `stpm3` forms these interactions for you using the `tvc()` and `dftvc()` options.

```
. stpm3 i.dep @ns(agediag, df(3)), scale(lncumhazard) df(5) ///
        tvc(i.dep) dftvc(3)
```

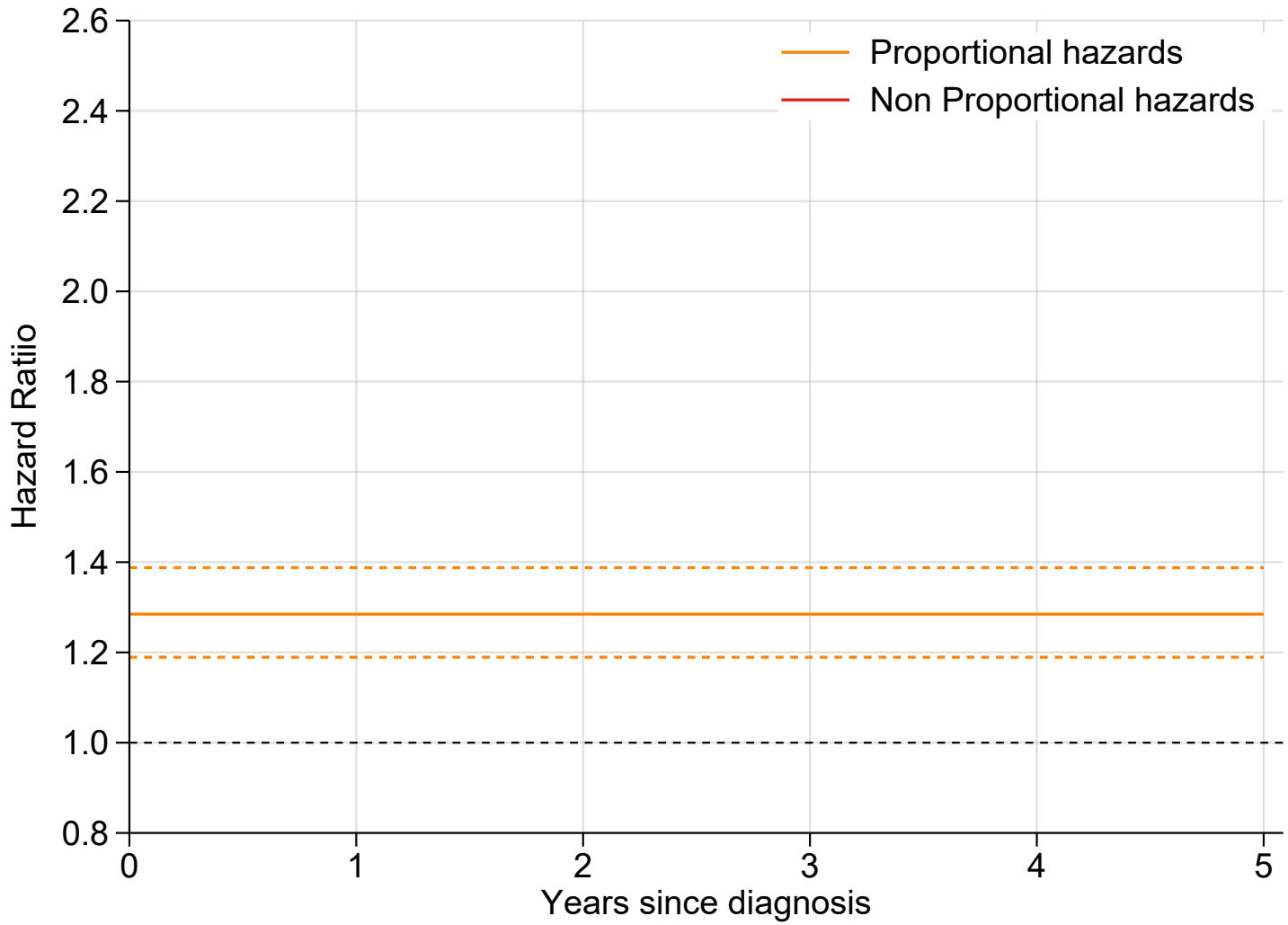
# Model output

```
. stpm3 i.dep @ns(agediag, df(3)), scale(lncumhazard) df(5) ///
>       tvc(i.dep) dftvc(3) vsquish nolog
  (output omitted)
```

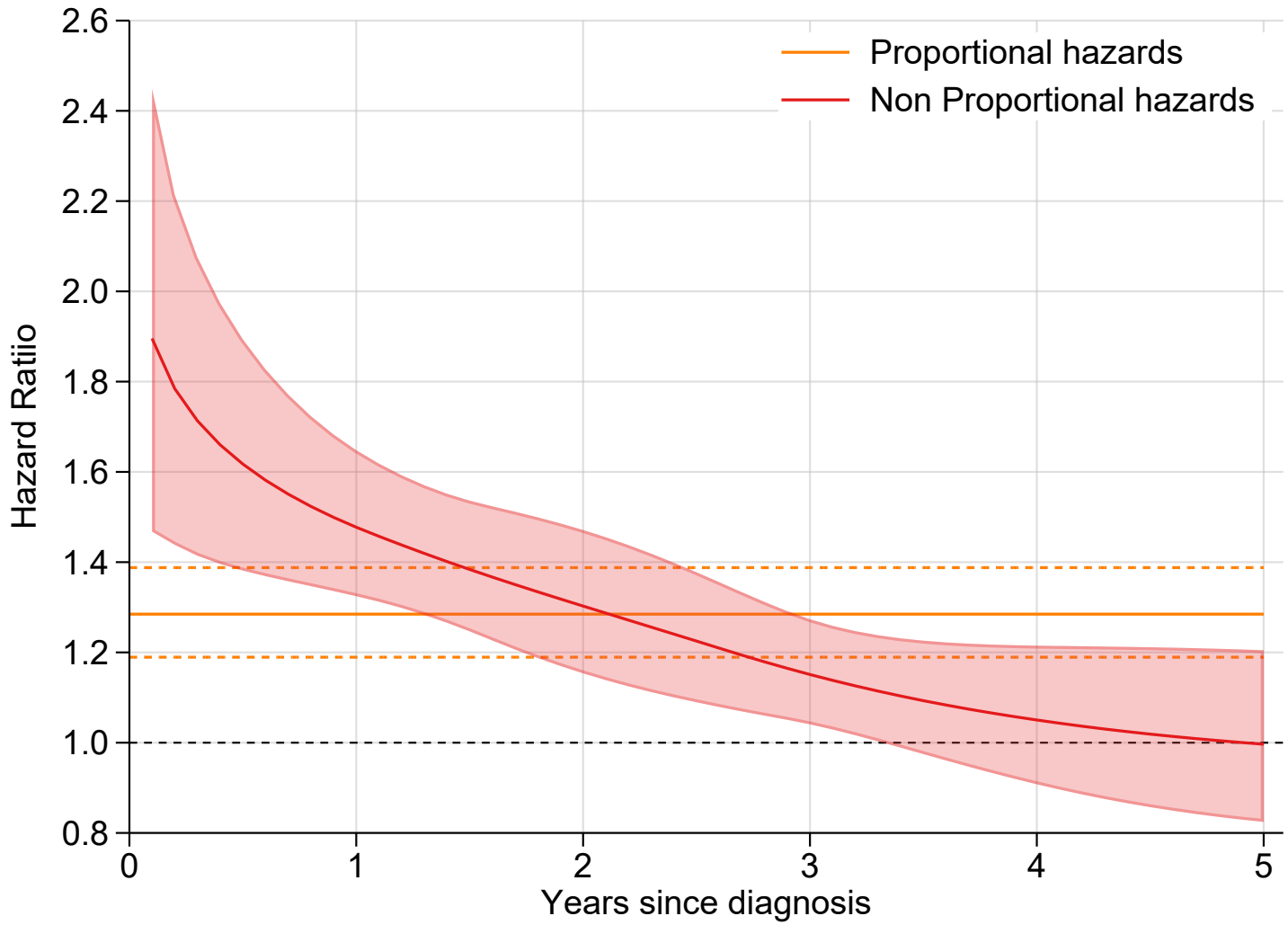
| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| xb | | | | | | |
| dep | | | | | | |
| mostdep | .2326102 | .0396115 | 5.87 | 0.000 | .1549731 | .3102473 |
| _ns_f1_agediag1 | 1.59671 | .6642716 | 2.40 | 0.016 | .2947616 | 2.898659 |
| _ns_f1_agediag2 | -.4834445 | .4039909 | -1.20 | 0.231 | -1.275252 | .3083631 |
| _ns_f1_agediag3 | -.1530129 | .1288731 | -1.19 | 0.235 | -.4055995 | .0995737 |
| time | | | | | | |
| _ns1 | -20.8306 | 1.51351 | -13.76 | 0.000 | -23.79702 | -17.86417 |
| _ns2 | 3.82159 | .7859048 | 4.86 | 0.000 | 2.281245 | 5.361935 |
| _ns3 | -1.159027 | .0349635 | -33.15 | 0.000 | -1.227555 | -1.0905 |
| _ns4 | -.6253963 | .0245694 | -25.45 | 0.000 | -.6735514 | -.5772411 |
| _ns5 | -.3741499 | .020979 | -17.83 | 0.000 | -.415268 | -.3330317 |
| dep#c._ns_tvc1 | | | | | | |
| mostdep | 1.900655 | 2.089832 | 0.91 | 0.363 | -2.195339 | 5.99665 |
| dep#c._ns_tvc2 | | | | | | |
| mostdep | -.1922529 | 1.108999 | -0.17 | 0.862 | -2.365851 | 1.981346 |
| dep#c._ns_tvc3 | | | | | | |
| mostdep | .1395927 | .0482234 | 2.89 | 0.004 | .0450767 | .2341087 |
| _cons | -1.155454 | .0628256 | -18.39 | 0.000 | -1.27859 | -1.032318 |

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

# Hazard Ratio: Most vs Least Deprived



# Hazard Ratio: Most vs Least Deprived



# Different types of predictions

- We want to predict different types of functions.
  - hazard functions, survival functions etc.
- There are 3 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 estimates).

# Different types of predictions

- We want to predict different types of functions.
  - hazard functions, survival functions etc.
- There are 3 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 estimates).
- We may also be interested in contrasts in the above, e.g. when comparing unexposed vs exposed.

# Different types of predictions

- We want to predict different types of functions.
  - hazard functions, survival functions etc.
- There are 3 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 estimates).
- We may also be interested in contrasts in the above, e.g. when comparing unexposed vs exposed.
- Different 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`
- By conditional I mean given values of all covariates specified in the model.

# Standard predictions

- Similar to what would be expected in `streg`
- Predicts at observed covariate values and at `_t`

## Model

```
. stset survtime, failure(dead=1) exit(time 5)
. stpm3 i.dep @ns(agediag, df(3)), scale(lncumhazard) df(4)
```

## Predictions

```
. predict   xb, xb ci          // linear predictor
. predict    S, survival ci    // survival function
. predict    h, hazard ci      // hazard function
```

By default this type of prediction will be saved in the current frame

# predict with multiple at() options

```
. predict S60_d1 S70_d1 S80_d1,     /// new variables
>         survival ci               /// predict survival and CI
>         at1(agediag 60 dep 1)     /// 1st prediction
>         at2(agediag 70 dep 1)     /// 2nd prediction
>         at3(agediag 80 dep 1)     /// 3rd prediction
>         timevar(0 10, step(0.1))  /// times to predict at
>         frame(surv, replace)      //  save in frame
Predictions are stored in frame - surv
```

# predict with multiple at() options

```
. predict S60_d1 S70_d1 S80_d1,     /// new variables
>         survival ci               /// predict survival and CI
>         at1(agediag 60 dep 1)     /// 1st prediction
>         at2(agediag 70 dep 1)     /// 2nd prediction
>         at3(agediag 80 dep 1)     /// 3rd prediction
>         timevar(0 10, step(0.1))  /// times to predict at
>         frame(surv, replace)      //  save in frame
Predictions are stored in frame - surv
```

# predict with multiple at() options

```
. predict S60_d1 S70_d1 S80_d1,     /// new variables
>         survival ci               /// predict survival and CI
>         at1(agediag 60 dep 1)     /// 1st prediction
>         at2(agediag 70 dep 1)     /// 2nd prediction
>         at3(agediag 80 dep 1)     /// 3rd prediction
>         timevar(0 10, step(0.1))  /// times to predict at
>         frame(surv, replace)      //  save in frame
Predictions are stored in frame - surv
```

# predict with multiple at() options

```
. predict S60_d1 S70_d1 S80_d1,     /// new variables
>         survival ci               /// predict survival and CI
>         at1(agediag 60 dep 1)     /// 1st prediction
>         at2(agediag 70 dep 1)     /// 2nd prediction
>         at3(agediag 80 dep 1)     /// 3rd prediction
>         timevar(0 10, step(0.1))  /// times to predict at
>         frame(surv, replace)      //  save in frame
Predictions are stored in frame - surv
```

# predict with multiple at() options

```
. predict S60_d1 S70_d1 S80_d1,     /// new variables
>         survival ci               /// predict survival and CI
>         at1(agediag 60 dep 1)     /// 1st prediction
>         at2(agediag 70 dep 1)     /// 2nd prediction
>         at3(agediag 80 dep 1)     /// 3rd prediction
>         timevar(0 10, step(0.1))  /// times to predict at
>         frame(surv, replace)      //  save in frame
Predictions are stored in frame - surv
```

# predict with multiple at() options

```
. predict S60_d1 S70_d1 S80_d1,     /// new variables
>         survival ci               /// predict survival and CI
>         at1(agediag 60 dep 1)     /// 1st prediction
>         at2(agediag 70 dep 1)     /// 2nd prediction
>         at3(agediag 80 dep 1)     /// 3rd prediction
>         timevar(0 10, step(0.1))  /// times to predict at
>         frame(surv, replace)      //  save in frame
Predictions are stored in frame - surv
```

```
. frame surv: format %5.4f S*
. frame surv: list tt S60_d1 S70_d1 S80_d1 if inlist(tt,1,5,10), noobs
```

| tt | S60_d1 | S70_d1 | S80_d1 |
|---|---|---|---|
| 1 | 0.8985 | 0.8623 | 0.7684 |
| 5 | 0.6785 | 0.5846 | 0.3850 |
| 10 | 0.5188 | 0.4032 | 0.1989 |

# Merge predictions with existing frame

```
. predict S60_d5 S70_d5 S80_d5,     /// new variables
>         survival  ci              /// predict survival and CI
>         at1(agediag 60 dep 5)     /// 1st prediction
>         at2(agediag 70 dep 5)     /// 2nd prediction
>         at3(agediag 80 dep 5)     /// 3rd prediction
>         frame(surv, merge)        //  merge to frame surv
Predictions are stored in frame - surv
```

# Merge predictions with existing frame

```
. predict S60_d5 S70_d5 S80_d5,     /// new variables
>         survival  ci              /// predict survival and CI
>         at1(agediag 60 dep 5)     /// 1st prediction
>         at2(agediag 70 dep 5)     /// 2nd prediction
>         at3(agediag 80 dep 5)     /// 3rd prediction
>         frame(surv, merge)        //  merge to frame surv
Predictions are stored in frame - surv
```

```
. frame surv: format %5.4f S*
. frame surv: list tt S60_d1 S70_d1 S80_d1 S60_d5 S70_d5 S80_d5 ///
>                 if inlist(tt,1,5,10), noobs
```

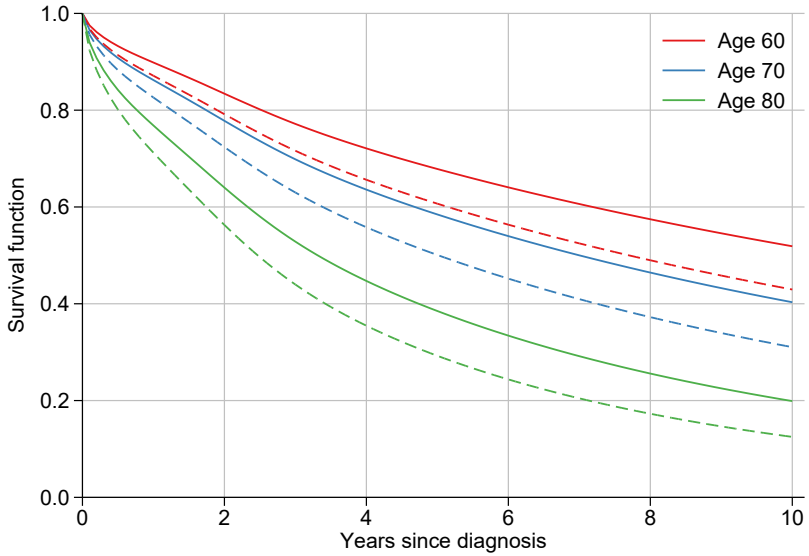
| tt | S60_d1 | S70_d1 | S80_d1 | S60_d5 | S70_d5 | S80_d5 |
|---|---|---|---|---|---|---|
| 1 | 0.8985 | 0.8623 | 0.7684 | 0.8712 | 0.8262 | 0.7123 |
| 5 | 0.6785 | 0.5846 | 0.3850 | 0.6068 | 0.5008 | 0.2925 |
| 10 | 0.5188 | 0.4032 | 0.1989 | 0.4295 | 0.3104 | 0.1250 |

# Plot predictions

```
. frame surv {
.   twoway (line S60_d1 S70_d1 S80_d1 tt)                    ///
>          (line S60_d5 S70_d5 S80_d5 tt, lpattern(dash..)   ///
>               pstyle(p1line p2line p3line)),               ///
>          ytitle("Survival function")                       ///
>          xtitle("Years since diagnosis")                   ///
>          ylabel(0(0.2)1, format(%3.1f))                    ///
>          legend(order(1 "Age 60" 2 "Age 70" 3 "Age 80")    ///
>                 ring(0) pos(1) cols(1))                    ///
>          note("Sold lines: Least deprived, Dashed lines: Most deprived")
. }
```

# Plot predictions

```
. frame surv {
.   twoway (line S60_d1 S70_d1 S80_d1 tt)                    ///
>          (line S60_d5 S70_d5 S80_d5 tt, lpattern(dash..)   ///
>               pstyle(p1line p2line p3line)),               ///
>          ytitle("Survival function")                       ///
>          xtitle("Years since diagnosis")                   ///
>          ylabel(0(0.2)1, format(%3.1f))                    ///
>          legend(order(1 "Age 60" 2 "Age 70" 3 "Age 80")    ///
>                 ring(0) pos(1) cols(1))                    ///
>          note("Sold lines: Least deprived, Dashed lines: Most deprived")
. }
```



# The `predict` command with complex models

- Consider the following two models

## Main effects and proportional hazards

```
stpm3 i.dep agediag, scale(lncumhazard) df(4)
```

# The `predict` command with complex models

- Consider the following two models

## Main effects and proportional hazards

```
stpm3 i.dep agediag, scale(lncumhazard) df(4)
```

## Non-linearity, interactions and non-proportional hazards

```
stpm3 i.dep##@ns(agediag, df(3)), scale(lncumhazard) df(4) ///
      tvc(i.dep @ns(agediag, df(2))) dftvc(3)
```

# The predict command with complex models

- Consider the following two models

## Main effects and proportional hazards

```
stpm3 i.dep agediag, scale(lncumhazard) df(4)
```

## Non-linearity, interactions and non-proportional hazards

```
stpm3 i.dep##@ns(agediag, df(3)), scale(lncumhazard) df(4) ///
      tvc(i.dep @ns(agediag, df(2))) dftvc(3)
```

- The predict command remains the same

```
predict S60_d1 S70_d1 S80_d1,     /// new variables
        survival ci               /// predict survival and CI
        at1(agediag 60 dep 1)     /// 1st prediction
        at2(agediag 70 dep 1)     /// 2nd prediction
        at3(agediag 80 dep 1)     /// 3rd prediction
        timevar(0 10, step(0.1))  /// times to predict at
        frame(surv, replace)      //  save in frame
```

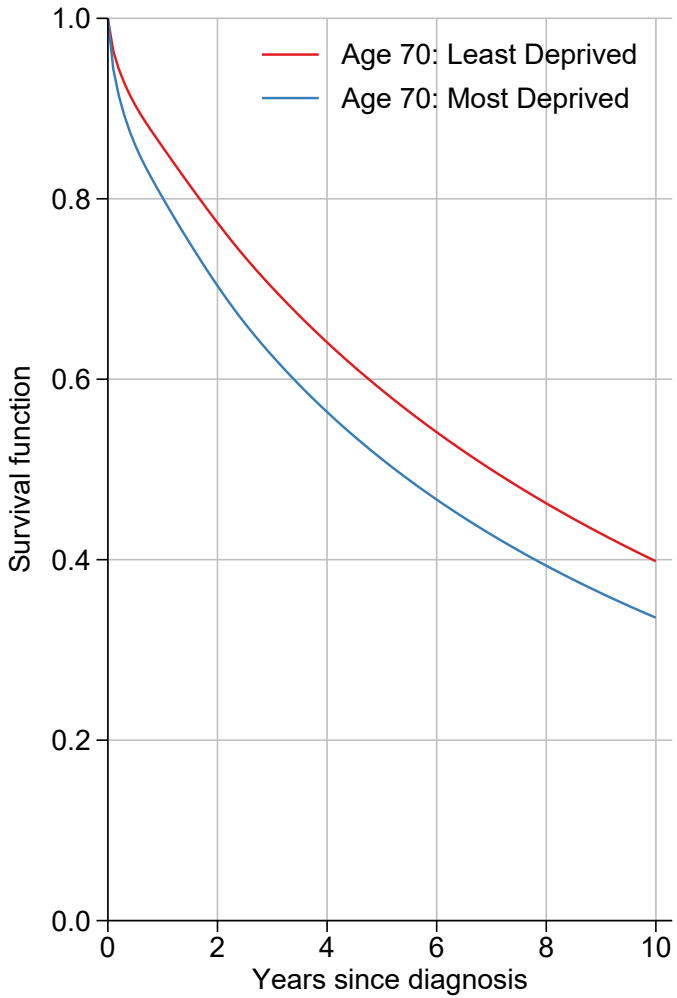
# Contrasts

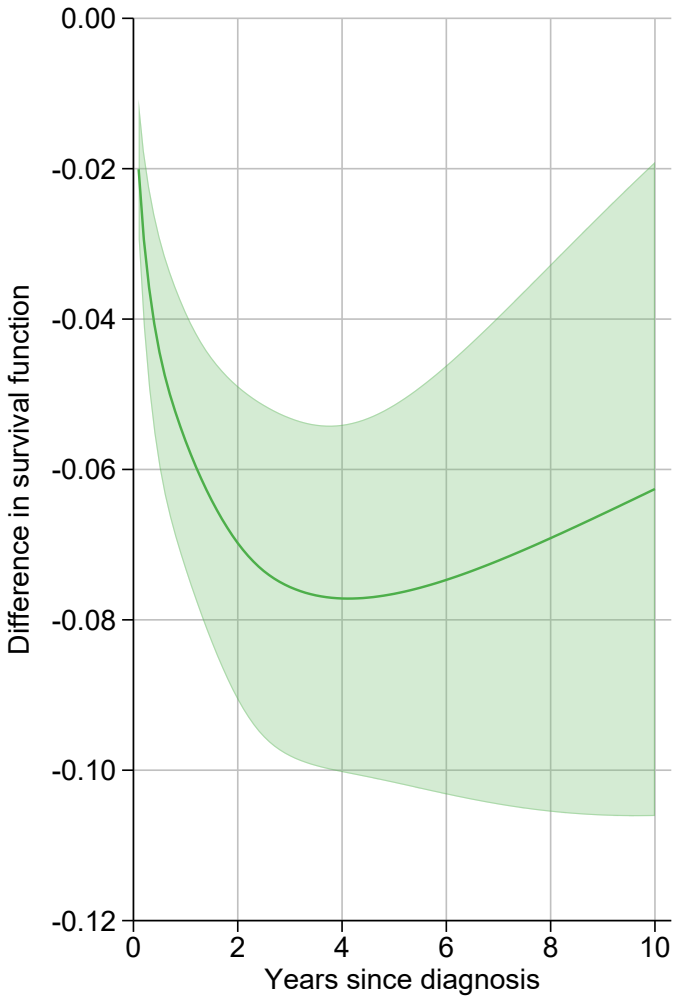
- We want to obtain contrasts between different covariate patterns.
- Just add `contrast()` and `contrastvar()` options.

```
. predict S70_d1  S70_d5,              /// new variables
>         survival ci                  /// predict survival and CI
>         at1(agediag 70 dep 1)        /// 1st prediction
>         at2(agediag 70 dep 5)        /// 2nd prediction
>         timevar(0 10, step(0.1))     /// times to predict at
>         contrast(difference)         /// contrast type
>         contrastvar(Sdiff)           /// contrast variable name
>         frame(survdiff, replace)     //  save in frame
Predictions are stored in frame - survdiff
```

Use `atreference()` to set reference `at()` option.

# Plot contrasts



# Marginal predictions (standardization).

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

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

- 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.
- `standsurv` will take averages of predicted survival curves.
- Like `margins`, but for survival models.

# Counterfactual marginal survival probabilities

- Let $X = 1$ correspond to exposed and $X = 0$ to unexposed.
- Let $S^x(t)$ denote the marginal survival probability had all individuals in the population, possibly contrary to fact, been assigned $X = x$.
- We can also form contrasts between different levels of exposure

$$S^1(t) - S^0(t)$$

- Difference in survival probabilities *"had all patients been exposed"* and *"had all patients been unexposed"*.
- Different to simply comparing the observed probabilities of exposed and unexposed.

# Assumptions for identifiability

- These hypothetical quantities are estimated using observed data under various assumptions [4].
- Assuming that covariates $\boldsymbol{Z}$ are sufficient to control for confounding control,

$$S^x(t) = E[S(t|X = x, \boldsymbol{Z})]$$

with the expectation taken over the marginal distribution of $\boldsymbol{Z}$.

- The difference between exposed and unexposed is,

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

- Under further assumptions (consistency, positivity, well-defined interventions) the marginal survival probability under $X = x$ can be estimated by the standardised survival probability using regression standardisation (G-formula) [5].

# Standardised survival probabilities

The average (causal) difference in marginal survival probabilities can be estimated as

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

1. Fit a statistical model that contains exposure, $X$, and potential confounders, $Z$.
2. Predict outcome for all individuals assuming they are all exposed ($X=1$).
3. Take mean to give marginal estimate of outcome under $X=1$.
4. Repeat for unexposed ($X=0$).
5. Take the difference in means to form contrasts.

Distribution of confounders, $Z$, is the same for both groups.

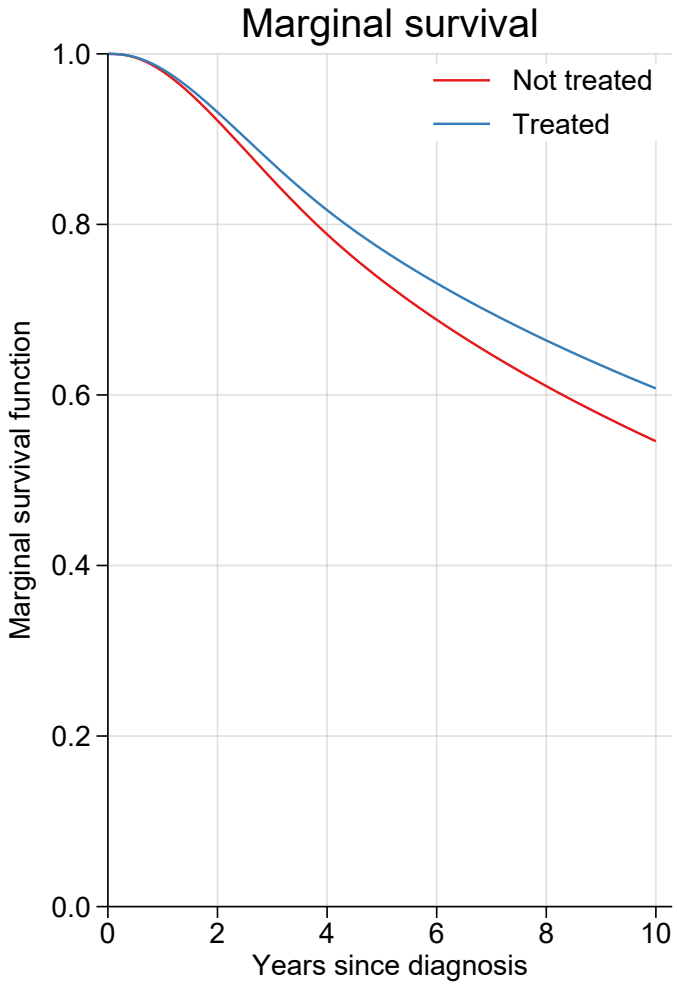
# Example: Rotterdam Breast Cancer Data

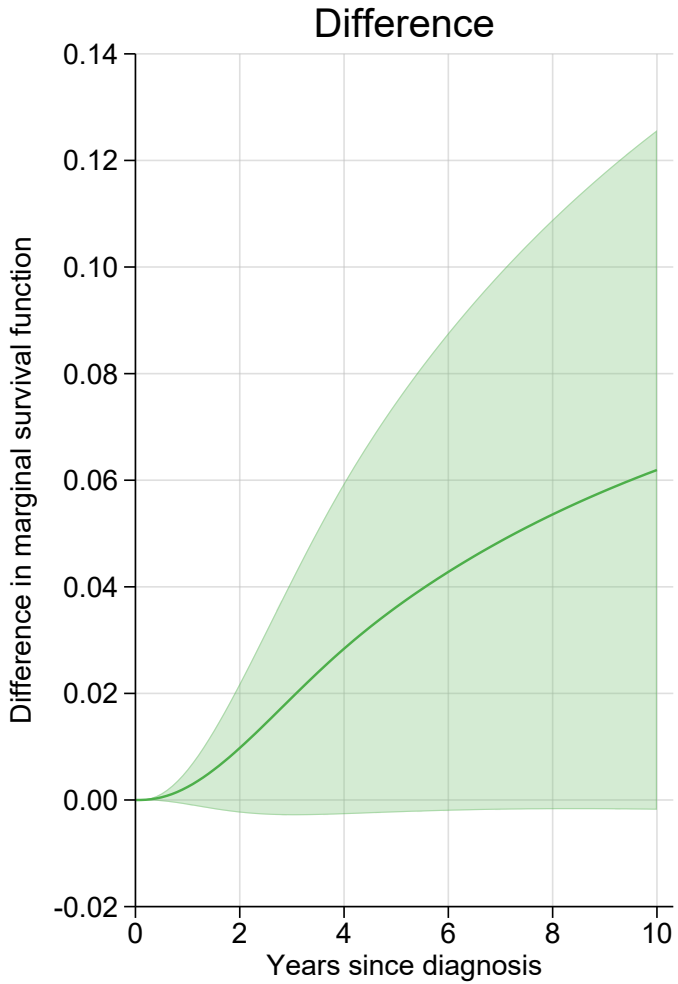
- Exposure is hormonal treatment (hormon)

```
stpm3 i.hormon##(@fn(exp(-0.12 * nodes),stub(enodes)) ///
                 @ns(age, df(3))                      ///
                 i.size i.grade c.pr_1),              ///
                 scale(lncumhazard) df(4)

range tt 0 10 101
standsurv S0 S1, surv ci timevar(tt) frame(mar_surv)  ///
                 at1(hormon 0) at2(hormon 1)          ///
                 contrast(difference)                 ///
                 contrastvar(Sdiff)
```

- Difference between average of
  - 2982 survival curves where everyone treated (hormon=1) and
  - 2982 survival curves where everyone untreated (hormon=0)

# Counterfactual marginal survival probabilities



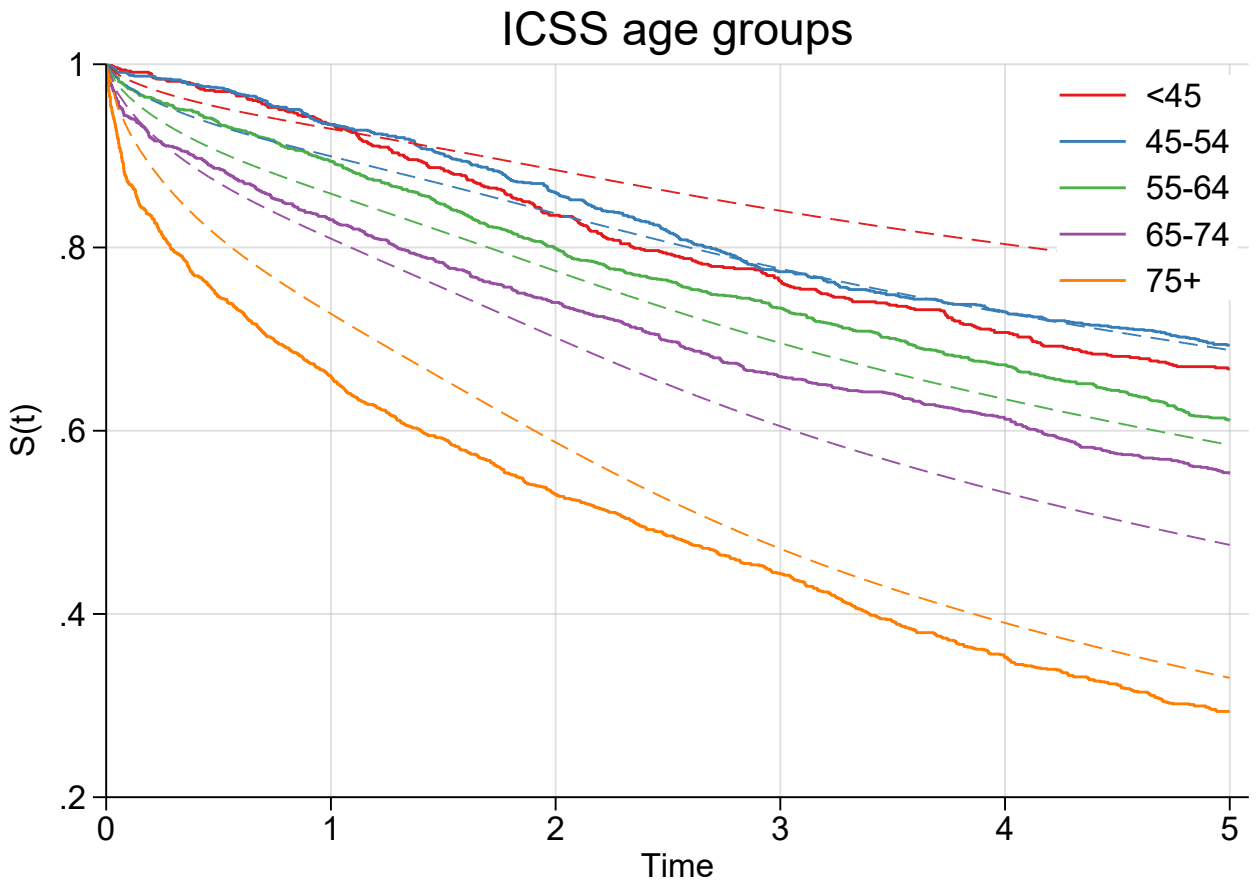
# Assessing the fit of models

- If our model is a good fit to the data then obtaining the average survival of the model predictions in a subgroup should be similar to the corresponding non-parametric Kaplan-Meier curve.
- Subgroups can be based on a prognostic index or a covariates in, or not in, the model.
- The `stpm3km` commands makes this easy.
- `stpm3km` calls `standsurv` with the `over` option.

# Breast Cancer: Assessing the fit of models

## Linear age and proportional hazards

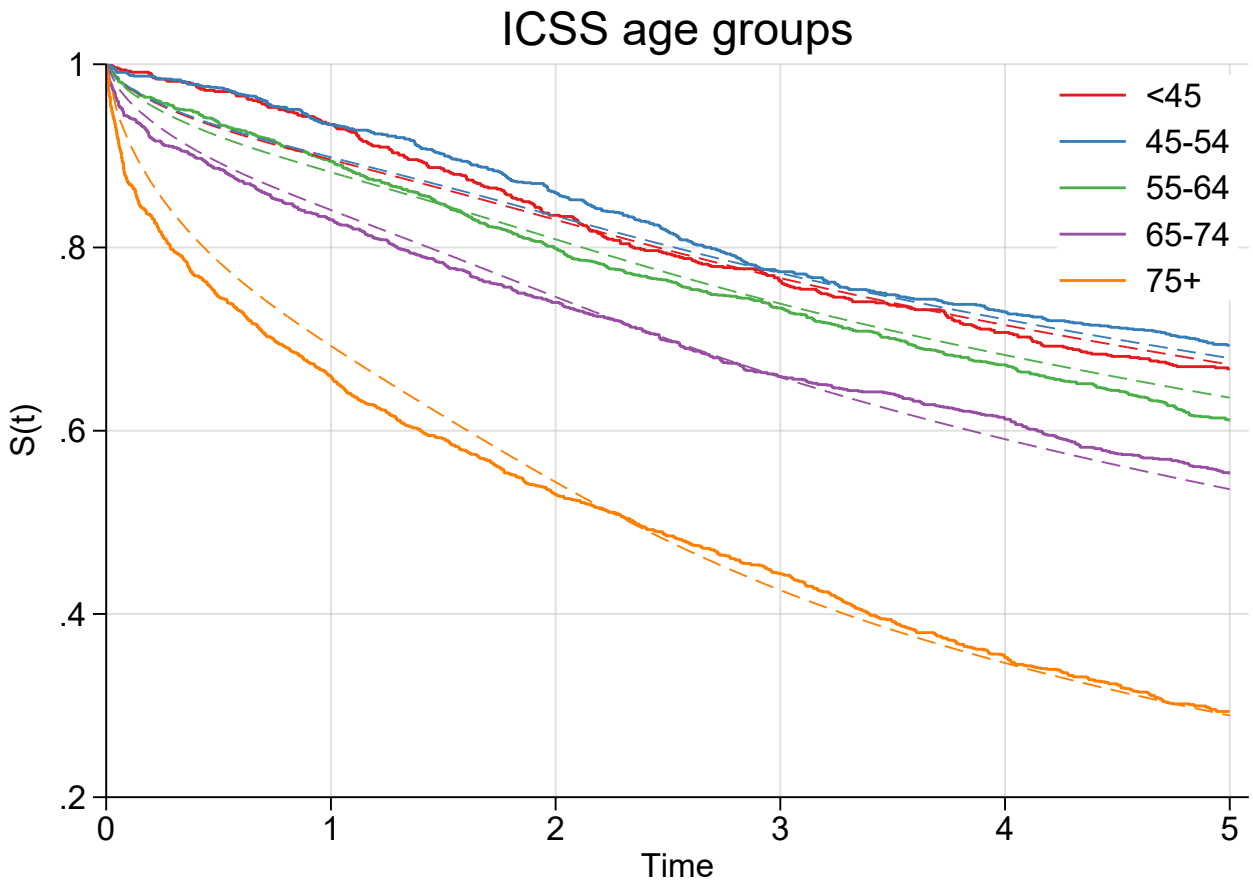
```
. stpm3 agediag i.dep, scale(lncumhazard) df(4)

. stpm3km i.agegrp, name(linear, replace)
```



# Breast Cancer: Assessing the fit of models

## Natural splines for age and proportional hazards

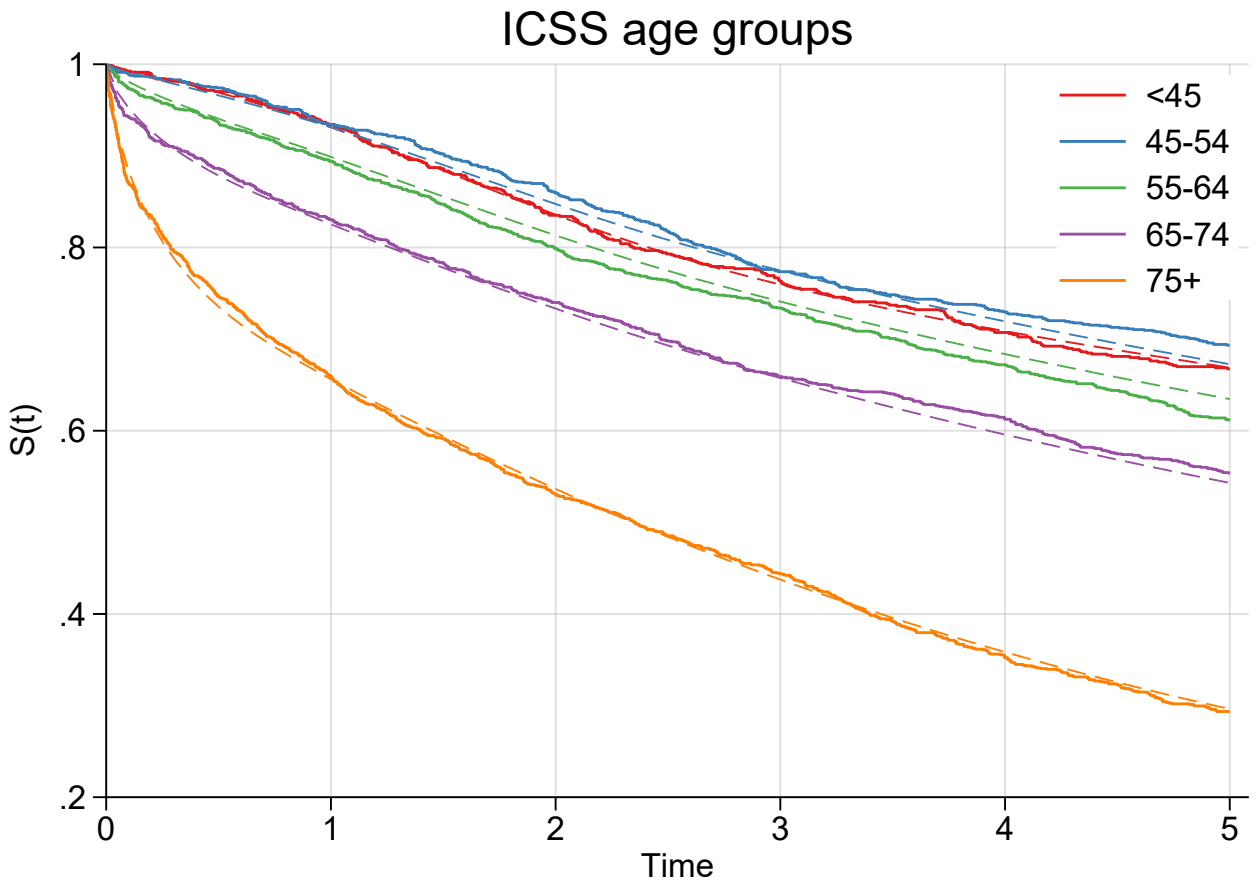
```
. stpm3 @ns(agediag,df(3)) i.dep, scale(lncumhazard) df(4)

. stpm3km i.agegrp, name(ns, replace)
```



# Breast Cancer: Assessing the fit of models

## Natural splines for age and non-proportional hazards

```
. stpm3 @ns(agediag,df(3)) i.dep, scale(lncumhazard) df(4) ///
        tvc(@ns(agediag,df(3))) dftvc(3)
. stpm3km i.agegrp, name(ns, replace)
```



# 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[6].
- 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.

# 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
```

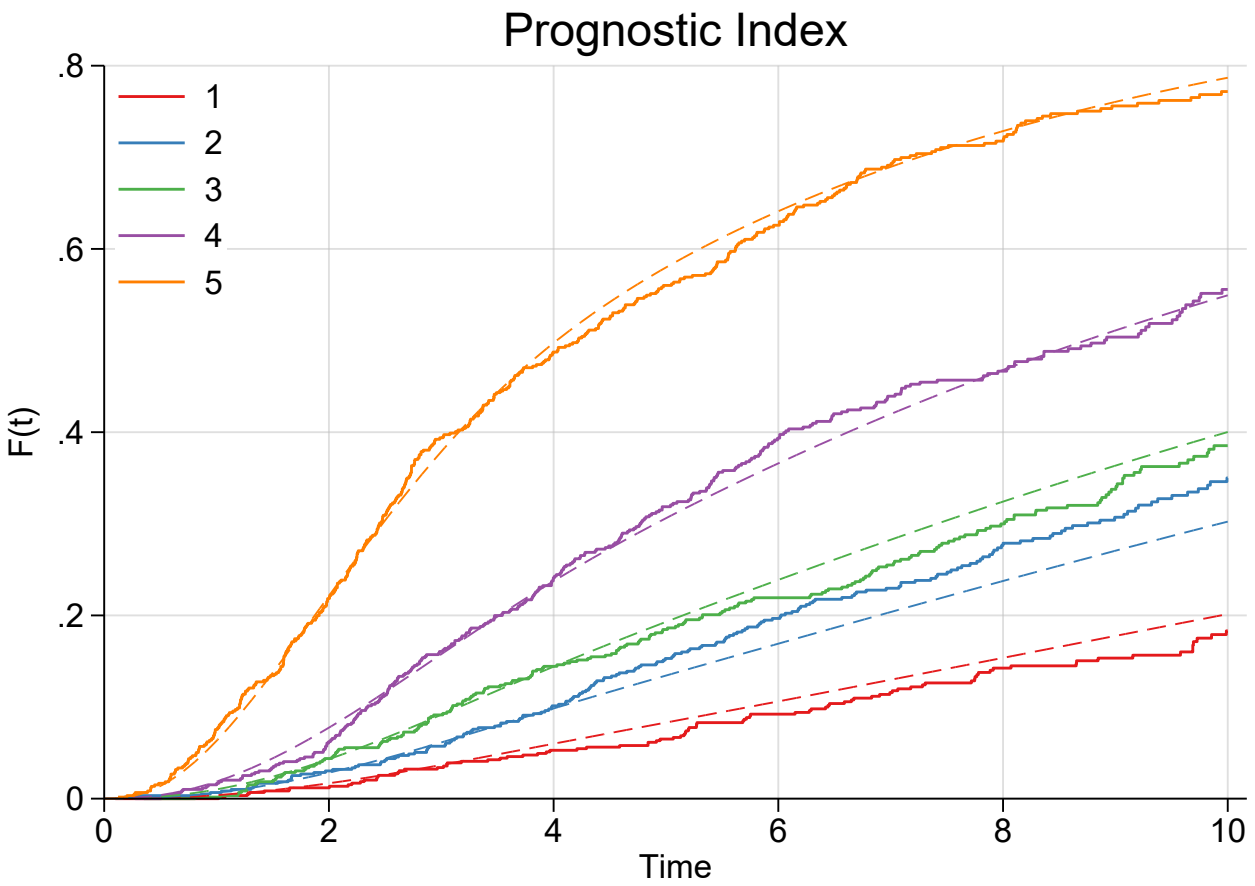
| | Coefficient | Std. err. | z | P>\|z\| | [95% conf. | interval] |
|---|---|---|---|---|---|---|
| xb | | | | | | |
| age | .0148001 | .0029896 | 4.95 | 0.000 | .0089405 | .0206596 |
| _fn_enodes | -2.664496 | .1550357 | -17.19 | 0.000 | -2.96836 | -2.360631 |
| size | | | | | | |
| >20-50mmm | .4698654 | .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

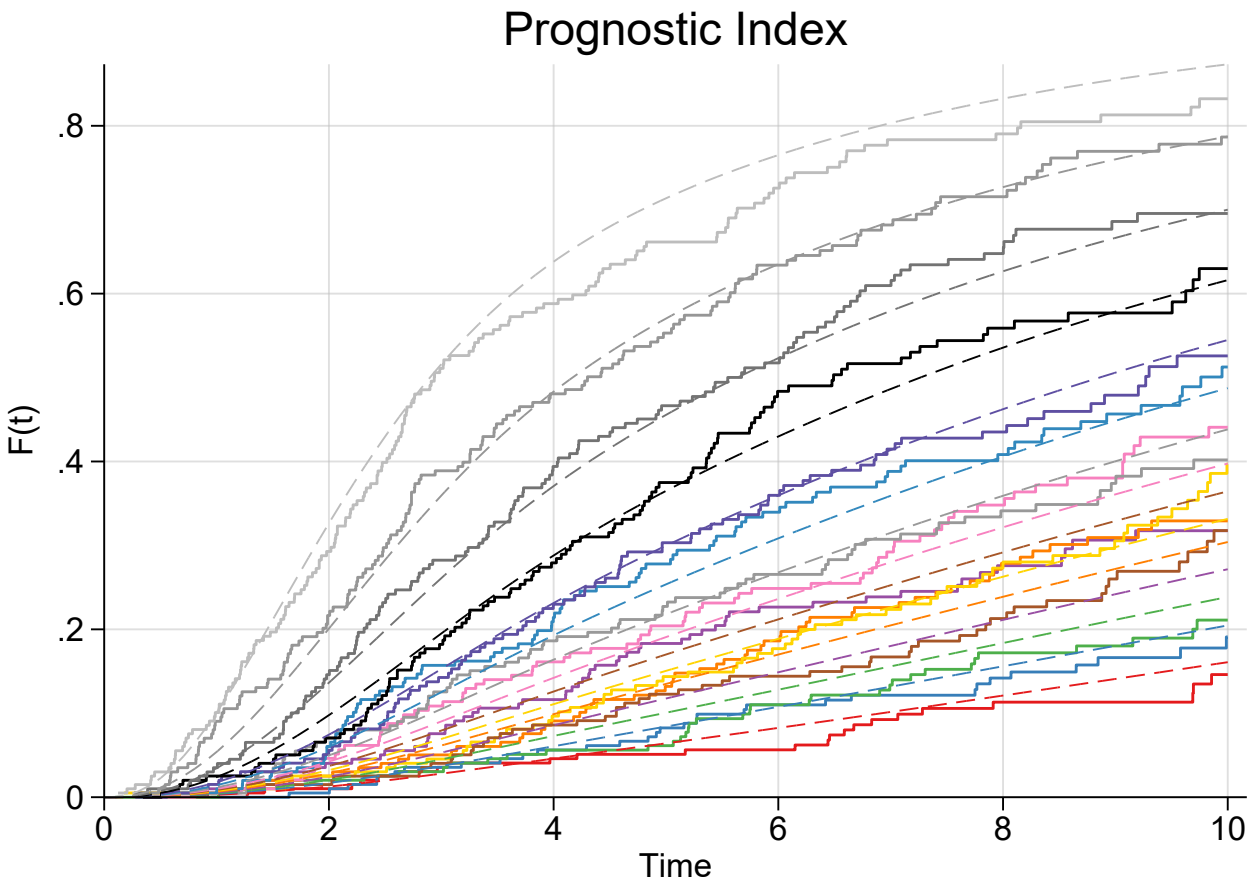
```
. stpm3km, failure
```



# Calibration

## stpm3km with lots of groups

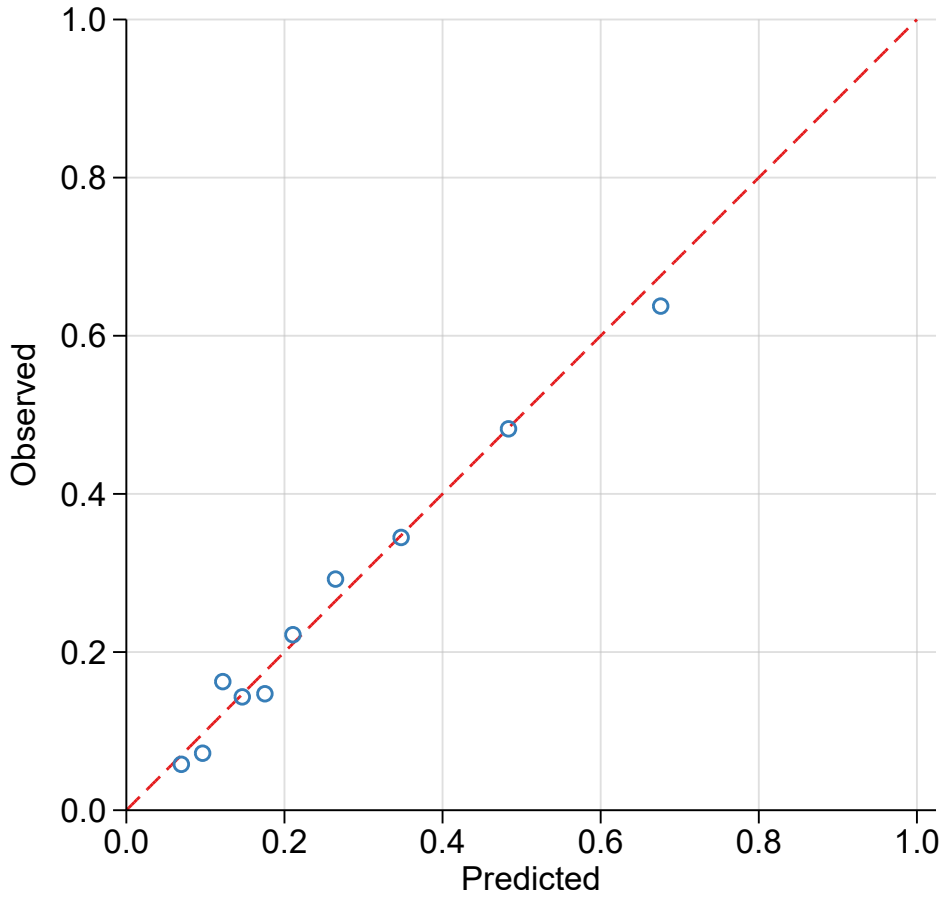
```
. stpm3km, groups(15) legend(off)
```



# Calibration

## stpm3calplot at 5 years

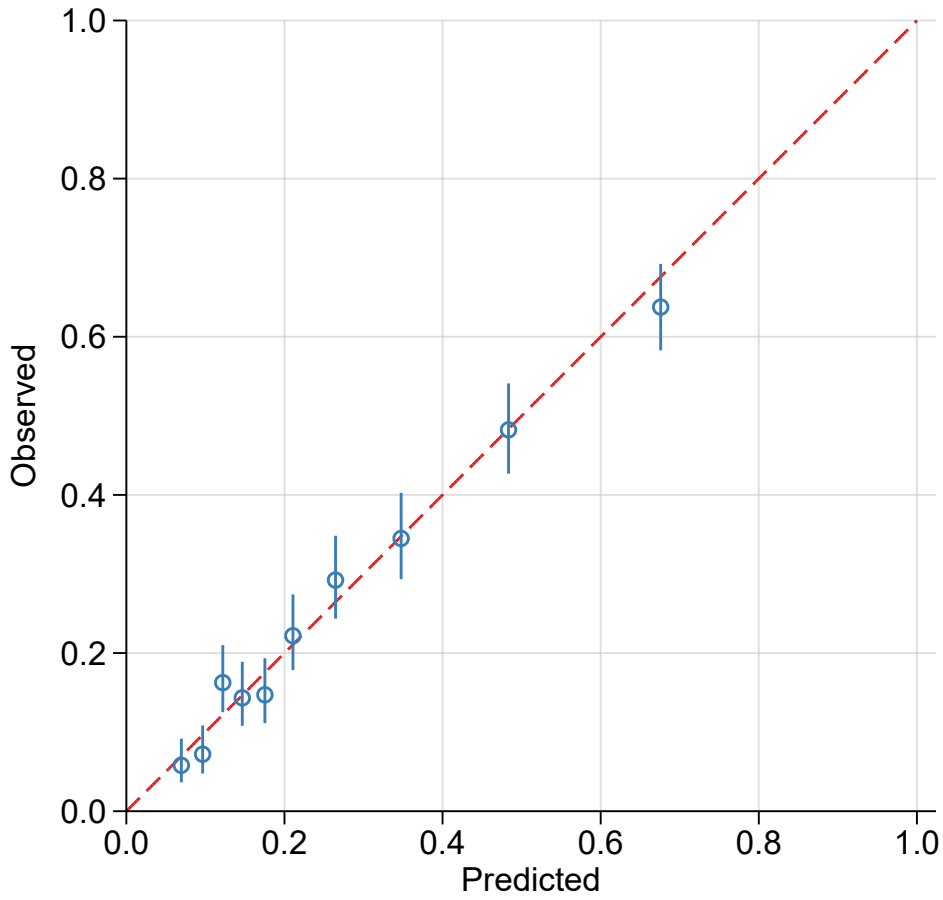
```
. stpm3calplot, time(5)
```



# Calibration

## stpm3calplot with Observed CIs

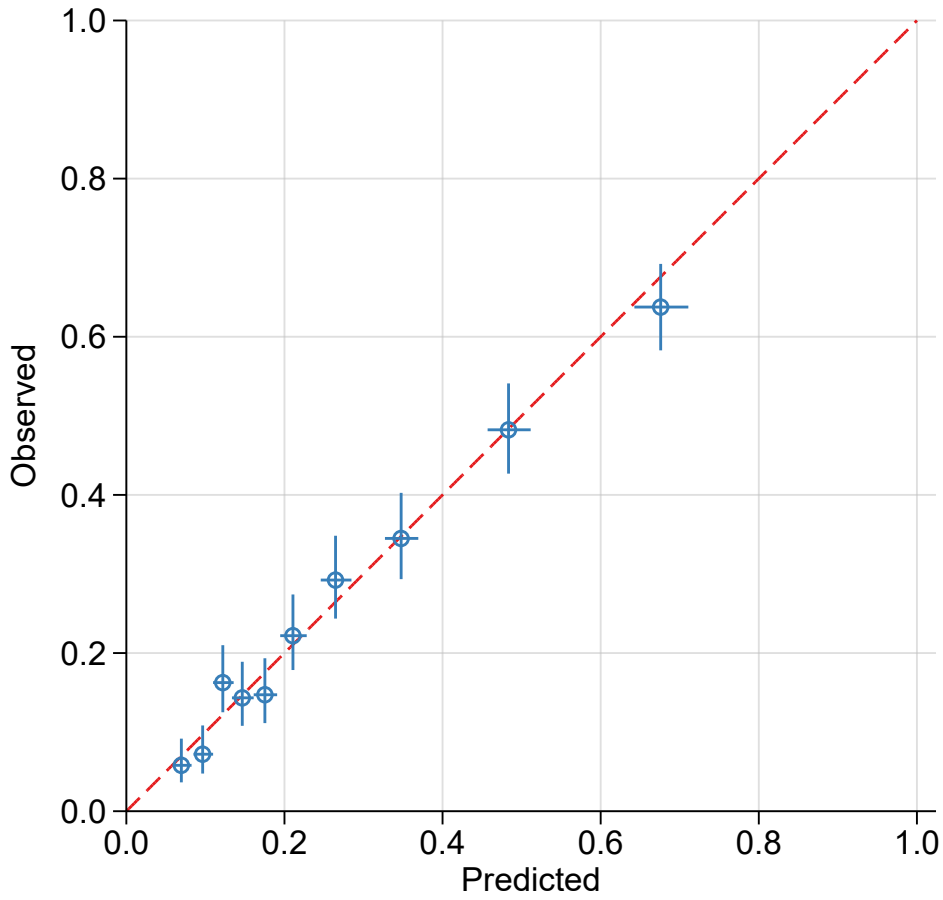
```
. stpm3calplot, time(5) ciobs
```



# Calibration

## stpm3calplot with Expected CIs

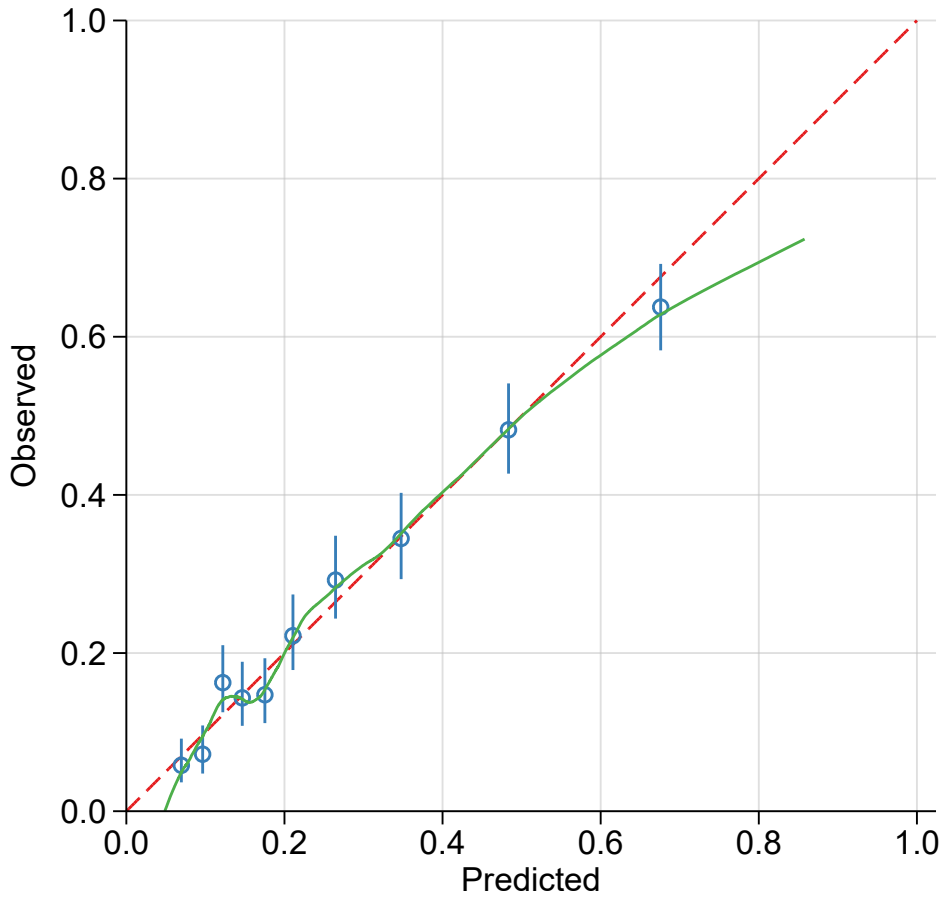
```
. stpm3calplot, time(5) cipred
```



# Calibration

## stpm3calplot with pseudo observations smoother

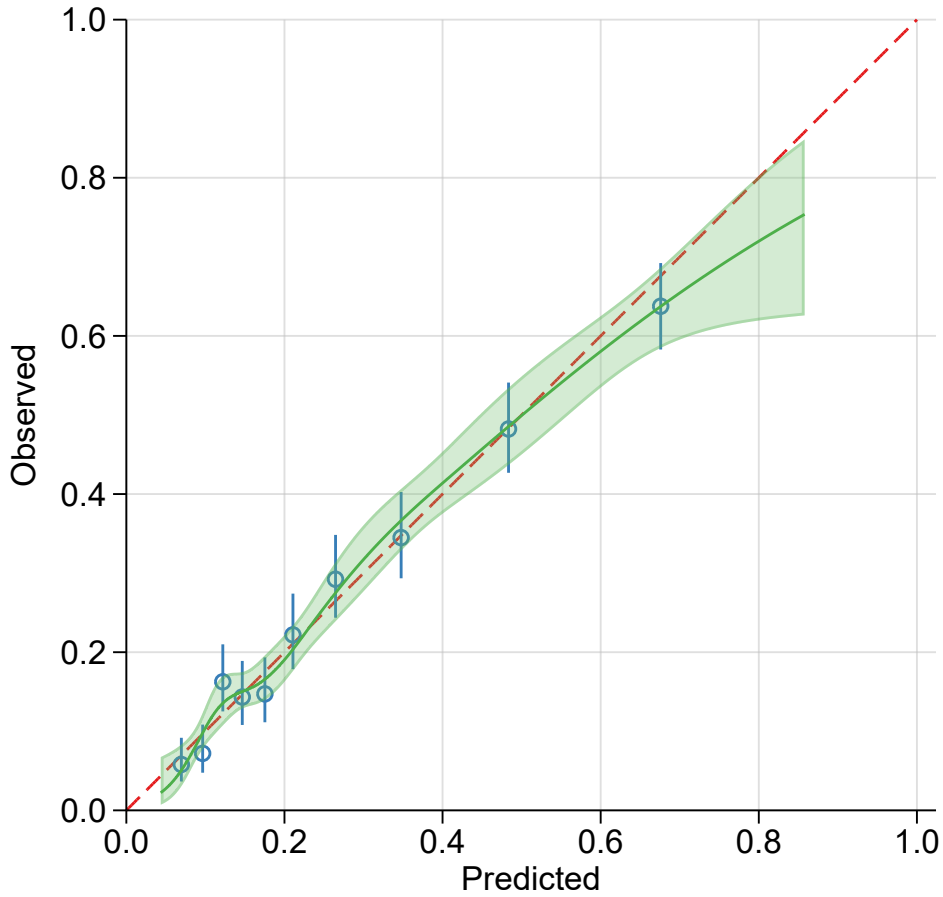
```
. stpm3calplot, time(5) ciobs pseudo
```



# Calibration

## stpm3calplot with pseudo observations smoother (splines)

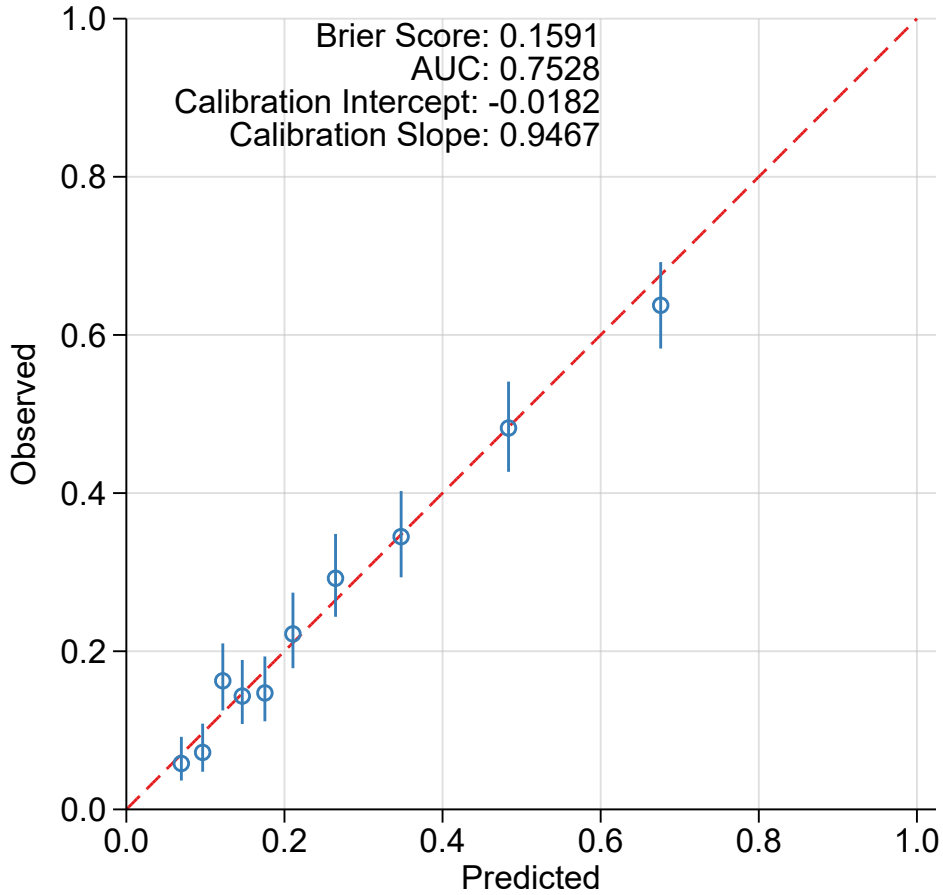
```
. stpm3calplot, time(5) pseudo smoother(ns) smootherci
```



# Calibration

## stpm3calplot at 5 years with performance statistics

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

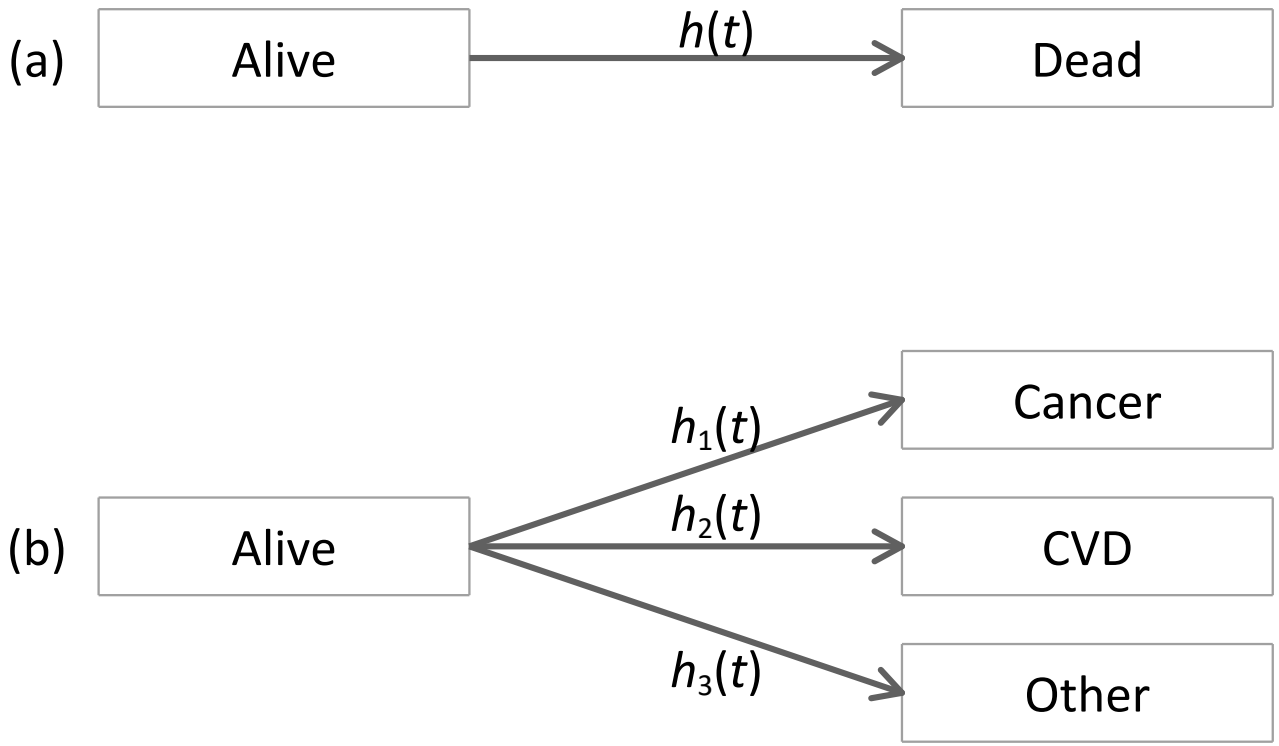


# Competing risks

- 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




Use a separate model for each hazard function, $h_k(t)$

## 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

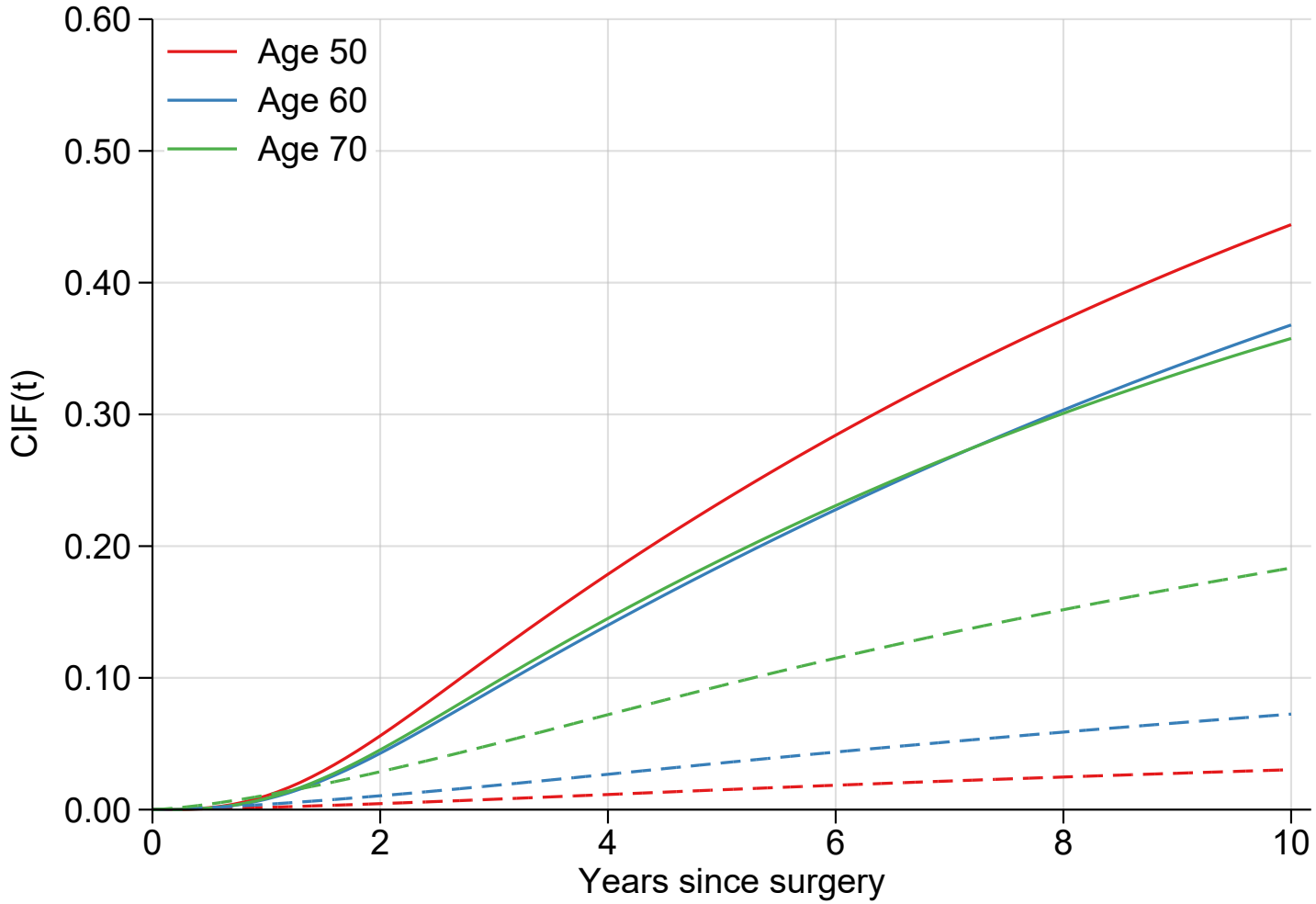
```
. stset os, failure(cause=2) exit(time 120) scale(12)
. stpm3 @ns(age,df(3)), scale(lncumhazard) df(3)
. estimates store other
```

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

# Predictions

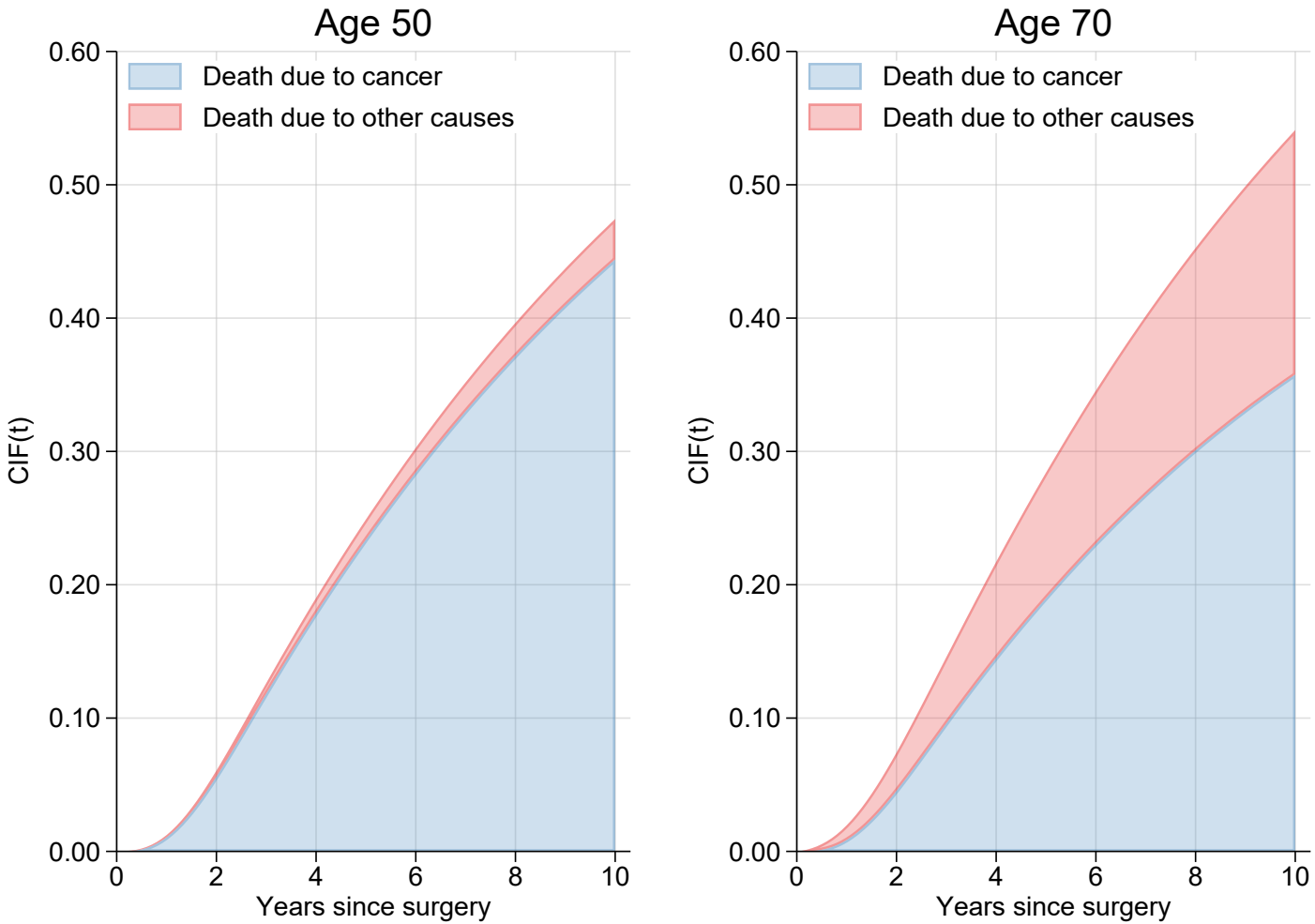
```
. // Conditional predictions
. predict cif50 cif60 cif70, cif crmodels(cancer other) ci              ///
>                            timevar(0 10, step(0.1))                   ///
>                            at1(age 50 size 1 grade 2 nodes 3 pr_1 0) ///
>                            at2(age 60 size 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




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

# Competing Risks: Predictions



# Competing Risks Extensions

- Causal Inference and competing risks using `standsurv` [7].
- Competing risks and prognostic models [8].
- Parametric version of Fine and Gray model [9, 10].

# Log hazard models

- Most models I have presented on log cumulative hazard scale

$$\ln(H(t)) = s(\ln(t)|\gamma) + \mathbf{x}\beta$$

- Sometimes useful to change to log hazard scale.

$$\ln(h(t)) = s(\ln(t)|\gamma) + \mathbf{x}\beta$$

- log hazard models useful for
  - Multiple time scales.
  - Multiple time-dependent effects (sometimes).
  - Standardized incidence/mortality rate ratios.
  - Extraploation (sometimes).

# Likelihood

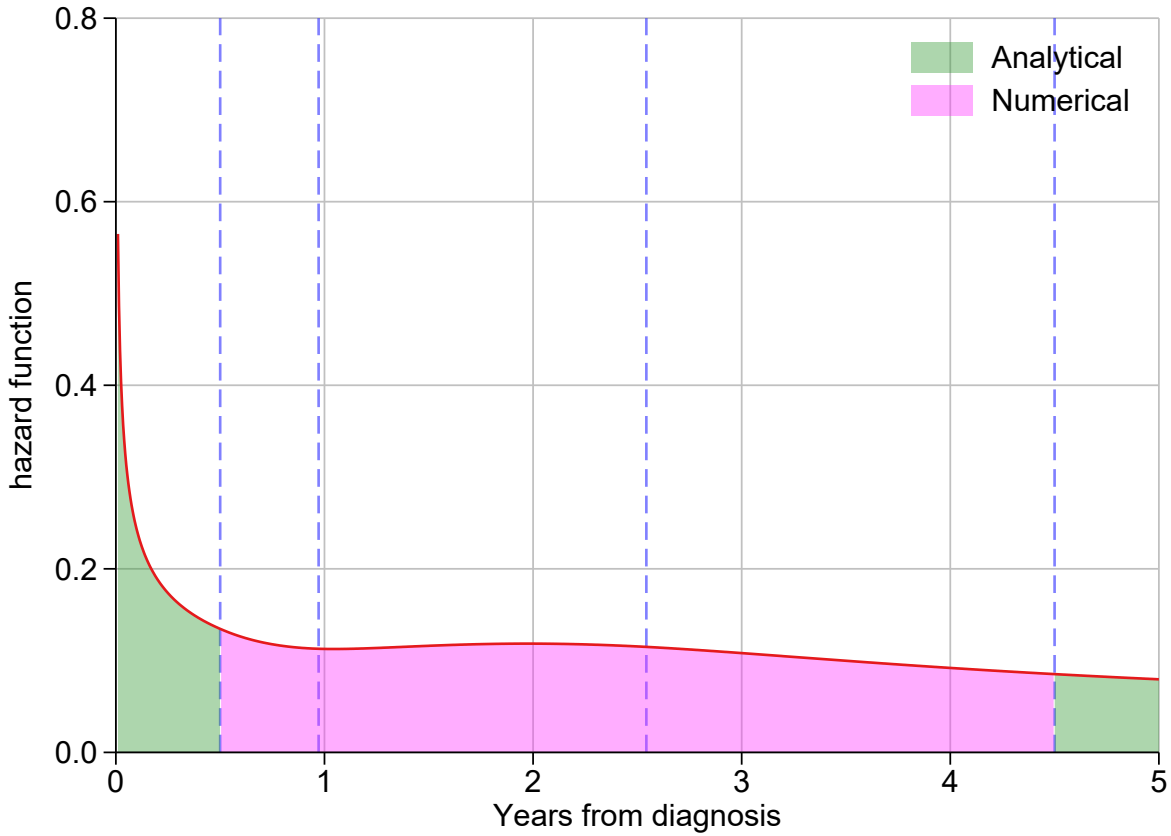
- Likelihood contribution of $i^{th}$ individual is,

$$l_i = d_i \ln(h(t_i)) + \int_{t_{0i}}^{t_i} h(u)du$$

- The integral needs to be calculated *numerically*.
  - For all individuals and for each call to likelihood/gradient/Hessian functions.
  - Computationally intensive in large datasets.
  - Usually use Gauss-Legendre quadrature, but some precision issues as can have singularity in hazard function at $t = 0$.

# Solutions to precision issue

- (**3-part integration:**) Use analytic integral before and after last knots[11].




- tanh-sinh quadrature when hazard has singularity[12].

# Faster models with large data sets

- For large datasets can send heavy computations to Python.
- Just add `python` option.
- The `mlad` program is used to maximize the likelihood.
- Calls `mlad`
  - Maximum Likelihood using Automatic Differentiation.
  - Calls Python Jax module.
  - Scores and Hessian automatically created
  - Just-In-Time (JIT) compilation
  - Efficient use of multiple processors.

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

See `mlad` talk at Stata Conference
https://www.stata.com/meeting/us21/slides/US21_Lambert.pdf

# Times in seconds

| | Sample Size | | | |
|---|---|---|---|---|
| | 500,000 | | 1,000,000 | |
| *3 part integration* | | | | |
| `strcs` | 2930 | | 4807 | |
| `stpm3` | 493 | (83.1%) | 981 | (79.6%) |
| `stpm3 (python option)` | 46 | (98.4%) | 83 | (98.3%) |
| *All numerical integration* | | | | |
| `stmerlin` | 1950 | | 3996 | |
| `stpm3` | 464 | (76.2%) | 917 | (77.0%) |
| `stpm3 (python option)` | 34 | (98.3%) | 69 | (98.3%) |

# Discussion

- Will get very similar estimates (and standard errors) to a Cox proportional hazards model.
- Including the baseline in the model makes (complex) predictions much easier.
- Many alternative ways to present data than hazard ratios.
- Lots of extensions
  - Relative survival[13, 14]
  - Extrapolation [15, 16]
  - Multistate models [17]
  - Loss in life expectancy [18, 19]
  - Multiple timescales [20]

# Want to find out more?

2024 Northern European Stata Conference Oslo, Norway, Tuesday 10 September 2024

Pre-conference course Monday 9 September 2024

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

# References


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

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

[3] P Royston PL. *Flexible parametric survival analysis in Stata: Beyond the Cox model*. Stata Press, 2011.

[4] Hernán MA, Robins JM. Estimating causal effects from epidemiological data. *Journal of Epidemiology Community Health* 2006;**60**:578–586.

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

[6] Gerds TA, Andersen PK, Kattan MW. Calibration plots for risk prediction models in the presence of competing risks. *Statistics in medicine* 2014;**33**:3191–3203.

[7] 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.

# References 2


[8] 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.

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

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

[11] 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.

[12] M. M, M. S. The double-exponential transformation in numerical analysis. *Journal of Computational and Applied Mathematics* 2001;**127**:287–296.

[13] Lambert PC, Dickman PW, Nelson CP, Royston P. Estimating the crude probability of death due to cancer and other causes using relative survival models. *Statistics in Medicine* 2010;**29**:885 – 895.

[14] Lambert PC, Dickman PW, Rutherford MJ. Comparison of approaches to estimating age-standardized net survival. *BMC Med Res Methodol* 2015;**15**:64.

# References 3


[15] Lee S, Lambert PC, Sweeting MJ, Latimer NR, Rutherford MJ. Evaluation of flexible parametric relative survival approaches for enforcing long-term constraints when extrapolating all-cause survival. *Value in Health* 2024;**27**:51–60.

[16] Rutherford MJ, Lambert PC, Sweeting MJ, Pennington B, Crowther MJ, Abrams KR, Latime NR. NICE DSU technical support document 21: Flexible methods for survival analysis. *Decision Support Unit University of Sheffield* 2021;.

[17] Crowther MJ, Lambert PC. Parametric multi-state survival models: flexible modelling allowing transition-specific distributions with application to estimating clinically useful measures of effect differences. *Statistics in Medicine* 2017;**36**:4719–4742.

[18] 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.

[19] Andersson TML, Rutherford MJ, Lambert PC. Illustration of different modelling assumptions for estimation of loss in expectation of life due to cancer. *BMC Medical Research Methodology* 2019;**19**:19.

[20] Batyrbekova N, Bower H, Dickman PW, Ravn Landtblom A, Hultcrantz M, Szulkin R, *et al.*. Modelling multiple time-scales with flexible parametric survival models. *BMC medical research methodology* 2022;**22**:290.