Extended multivariate mixed effect models with merlin in Stata: Joint models and beyond

IBC 2022, Riga, Latvia

Michael J. Crowther

Founder & CEO, Red Door Analytics AB
Biostatistician, Karolinska Institutet
Honorary Senior Lecturer, University of Bristol

reddooranalytics.se michael@reddooranalytics.se @RDAnalyticsAB

About me

- I was for many years an academic biostatistician, becoming Associate Professor at the University of Leicester
- Moved to Stockholm in 2020 to pursue some new challenges
- Now running my own consultancy company, working as a software developer and biostatistician
- Still working 20% at Karolinska Institutet as a biostatistician

Methods development + software

- `stjm` - joint longitudinal-survival models [1, 2, 3, 4]
- `stmixed` - multilevel survival models [5, 6]
- `stgenreg` - general parametric survival models [7, 8]
- `survsim` - simulating complex survival data [9, 10]
- `multistate` - multi-state survival analysis [11]
- ...

Methods development + software

- `stjm` - joint longitudinal-survival models [1, 2, 3, 4]
- `stmixed` - multilevel survival models [5, 6]
- `stgenreg` - general parametric survival models [7, 8]
- `survsim` - simulating complex survival data [9, 10]
- `multistate` - multi-state survival analysis [11]
- ...

Each new project brings a new code base to maintain...could I make my life easier?

Mixed Effects Regression for LInear, Non-linear and user-defined models

`merlin`

The goal

- multiple outcomes of varying types
- measurement schedule can vary across outcomes
- any number of levels and random effects
- sharing and linking random effects between outcomes
- sharing functions of the expected value of other outcomes
- a reliable estimation engine
- easily extendable by the user
- ...

a unified framework for data analysis and methods development

The goal

- multiple outcomes of varying types
- measurement schedule can vary across outcomes
- any number of levels and random effects
- sharing and linking random effects between outcomes
- sharing functions of the expected value of other outcomes
- a reliable estimation engine
- easily extendable by the user
- ...

a unified framework for data analysis and methods development

(I think I made my life more difficult)

Some maths

For a one-level model with n response variables:

$$p(y|x,b,\beta) = \prod_{i=1}^{n} p_i(y_i|x,b,\beta)$$

For a two-level model:

$$p(y|x,b,\beta) = \prod_{i=1}^{n}\prod_{j=1}^{t} p_i(y_{ij}|x,b,\beta)$$

Some maths

The log likelihood is obtained by integrating out the unobserved random effects

$$ll(\beta) = \log \int_{\mathcal{R}^r} p(y|x, b, \beta)\phi(b|\Sigma_b)\ \mathrm{d}b$$

we assume $\phi()$ is the multivariate normal density for b, with mean vector 0 and variance-covariance matrix Σ_b. We have Σ_b becoming block diagonal with further levels, with a block for each level

Some maths

Alternatively, exploiting conditional independence amongst level $l-1$ units, given the random effects at higher levels,

$$ll(\beta) = \log \int \phi(b^{(L)}|\Sigma^{(L)}) \prod p^{(L-1)}(y|x, b^{L}, \beta) \, \mathrm{d}b^{(L)}$$

where, for $l = 2, \ldots, L$

$$p^{(l)}(y|x, B^{l+1}, \beta) = \int \phi(b^{(l)}|\Sigma^{(l)}) \prod p^{(l-1)}(y|x, B^{l}, \beta) \, \mathrm{d}b^{(l)}$$

Estimation challenges

- At each level, we need to integrate out our normally distributed random effects
- Generally this is done using Gauss-Hermite numerical quadrature

  `intmethod(mvaghermite | ghermite)`
- Issue with GH quadrature is it doesn't scale up well
- Monte-Carlo integration can help

  `intmethod(mcarlo)`

Standard linear predictor

The standard linear predictor for a general level model can be written as follows,

$$\eta = X\beta + \sum_{l=2}^{L} X^l b^l$$

where subscripts are omitted. We have X our vector of covariates, which could vary at any level, with associated fixed effect coefficient vector β, and X^l the vector of covariates with random effects b^l at level l.

Extended linear predictor

$$\eta_i = g_i(E[y_i|X, b]) = \sum_{r=1}^{R_i} \prod_{s=1}^{S_{ir}} \psi_{irs}$$

where $g_i()$ is the link function for the ith outcome. To maintain generality, $\psi_{irs}(t)$ can take many forms, including,

$$\psi_{irs}(t) = X$$
$$\psi_{irs}(t) = \beta$$
$$\psi_{irs}(t) = b$$
$$\psi_{irs}(t) = q(t)$$
$$\psi_{irs}(t) = d_{rs}(E[y_j]), \qquad \text{where } j = 1, \ldots, k, j \neq i$$

Extended linear predictor

$$\eta_i = g_i(E[y_i|X, b]) = \sum_{r=1}^{R_i} \prod_{s=1}^{S_{ir}} \psi_{irs}$$

where $g_i()$ is the link function for the ith outcome. To maintain generality, $\psi_{irs}(t)$ can take many forms, including,

$$\begin{aligned}
&\psi_{irs}(t) = X \\
&\psi_{irs}(t) = \beta \\
&\psi_{irs}(t) = b \\
&\psi_{irs}(t) = q(t) \\
&\psi_{irs}(t) = d_{rs}(E[y_j]), \qquad \text{where } j = 1, \ldots, k, j \neq i
\end{aligned}$$

And who says each outcome model can only have one complex predictor?

ssc install merlin

Title

merlin —— Mixed effects regression for linear, non-linear and user-defined models

See **merlin - a unified framework for data analysis and methods development in Stata**, for an introduction.

Syntax

merlin *models* [*if*] [*in*] [, *options*]

where *models* are the model specifications; see **merlin models**.

options	Description
model_description_options	fully define, along with *models*, the model to be fit
estimation_options	method used to obtain estimation results, including specifying initial values
reporting_options	reporting of estimation results

Also see **merlin postestimation** for features available after estimation.

Distributional choices

- Gaussian, Poisson, binomial, beta, negative binomial, ordinal (logit or probit link)
- exponential, Weibull, Gompertz, log-normal, log-logistic, gamma, Royston-Parmar, general log hazard and log cumulative hazard models
- non-linear outcome models
- user-defined hazard functions
- many (many) more to add...

An example

- data from 312 patients with PBC collected at the Mayo Clinic 1974-1984 (Murtaugh et al. (1994))
- 158 randomised to receive D-penicillamine and 154 to placebo
- survival outcome is all-cause death, with 140 events observed
  - we're going to pretend we have competing causes of death - cancer and other causes
- 1,945 measurements of serum bilirubin, among other things

data

id	time	logb	prothr~n	trt	stime	cancer	other
1	0	2.674149	12.2	D-penicil	1.09517	1	0
1	.525682	3.058707	11.2	D-penicil	.	.	.
2	0	.0953102	10.6	D-penicil	14.1523	0	1
2	.498302	-.2231435	11	D-penicil	.	.	.
2	.999343	0	11.6	D-penicil	.	.	.
2	2.10273	.6418539	10.6	D-penicil	.	.	.
2	4.90089	.9555114	11.3	D-penicil	.	.	.
2	5.88928	1.280934	11.5	D-penicil	.	.	.
2	6.88588	1.435084	.	D-penicil	.	.	.
2	7.8907	1.280934	.	D-penicil	.	.	.
2	8.83255	1.526056	.	D-penicil	.	.	.

data

id	time	logb	prothr~n	trt	stime	cancer	other
1	0	2.674149	12.2	D-penicil	1.09517	1	0
1	.525682	3.058707	11.2	D-penicil	.	.	.
2	0	.0953102	10.6	D-penicil	14.1523	0	1
2	.498302	-.2231435	11	D-penicil	.	.	.
2	.999343	0	11.6	D-penicil	.	.	.
2	2.10273	.6418539	10.6	D-penicil	.	.	.
2	4.90089	.9555114	11.3	D-penicil	.	.	.
2	5.88928	1.280934	11.5	D-penicil	.	.	.
2	6.88588	1.435084	.	D-penicil	.	.	.
2	7.8907	1.280934	.	D-penicil	.	.	.
2	8.83255	1.526056	.	D-penicil	.	.	.

Let's fit 12 different models, without changing the dataset

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                ,                           /// options
                family(gaussian)            /// distribution
        )
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                ,                           /// options
                family(gaussian)            /// distribution
        )                                   ///
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                ,                           /// options
                family(gaussian)            /// distribution
        )                                   ///
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                time#M2[id]@1               /// random slope
                ,                           /// options
                family(gaussian)            /// distribution
       )
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                time#M2[id]@1               /// random slope
                ,                           /// options
                family(gaussian)            /// distribution
        )                                   ///
        (pro                                /// prothrombin index
                rcs(time, df(3))            /// covariate
                , family(gamma)             /// distribution
        )
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                time#M2[id]@1               /// random slope
                ,                           /// options
                family(gaussian)            /// distribution
        )                                   ///
        (pro                                /// prothrombin index
                rcs(time, df(3))            /// covariate
                M3[id]@1                    /// random effect
                , family(gamma)             /// distribution
        )
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                time#M2[id]@1               /// random slope
                ,                           /// options
                family(gaussian)            /// distribution
        )                                   ///
        (pro                                /// prothrombin index
                rcs(time, df(3))            /// covariate
                M3[id]@1                    /// random effect
                , family(gamma)             /// distribution
        )                                   ///
        ,                                   /// main options
        covariance(unstructured)            //  vcv
```

```
merlin (logb                                /// log serum bilirubin
                time                        /// covariate
                time#trt                    /// interaction
                M1[id]@1                    /// random intercept
                time#M2[id]@1               /// random slope
                ,                           /// options
                family(gaussian)            /// distribution
        )                                   ///
        (pro                                /// prothrombin index
                rcs(time, df(3))            /// covariate
                M3[id]@1                    /// random effect
                , family(gamma)             /// distribution
        )                                   ///
        ,                                   /// main options
        covariance(unstructured)            /// vcv
        redistribution(t) df(5)             //  re dist.
```

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               M3[id]@1                     /// random effect
               , family(gamma)              /// distribution
       )                                    ///
       (stime  trt                          /// resp. + covariate
              , family(rp, df(3)            /// distribution
                       failure(other))      /// event indicator
       )                                    ///
       ,                                    /// main options
       covariance(unstructured)             /// vcv
       redistribution(t) df(5)              //  re dist.
```

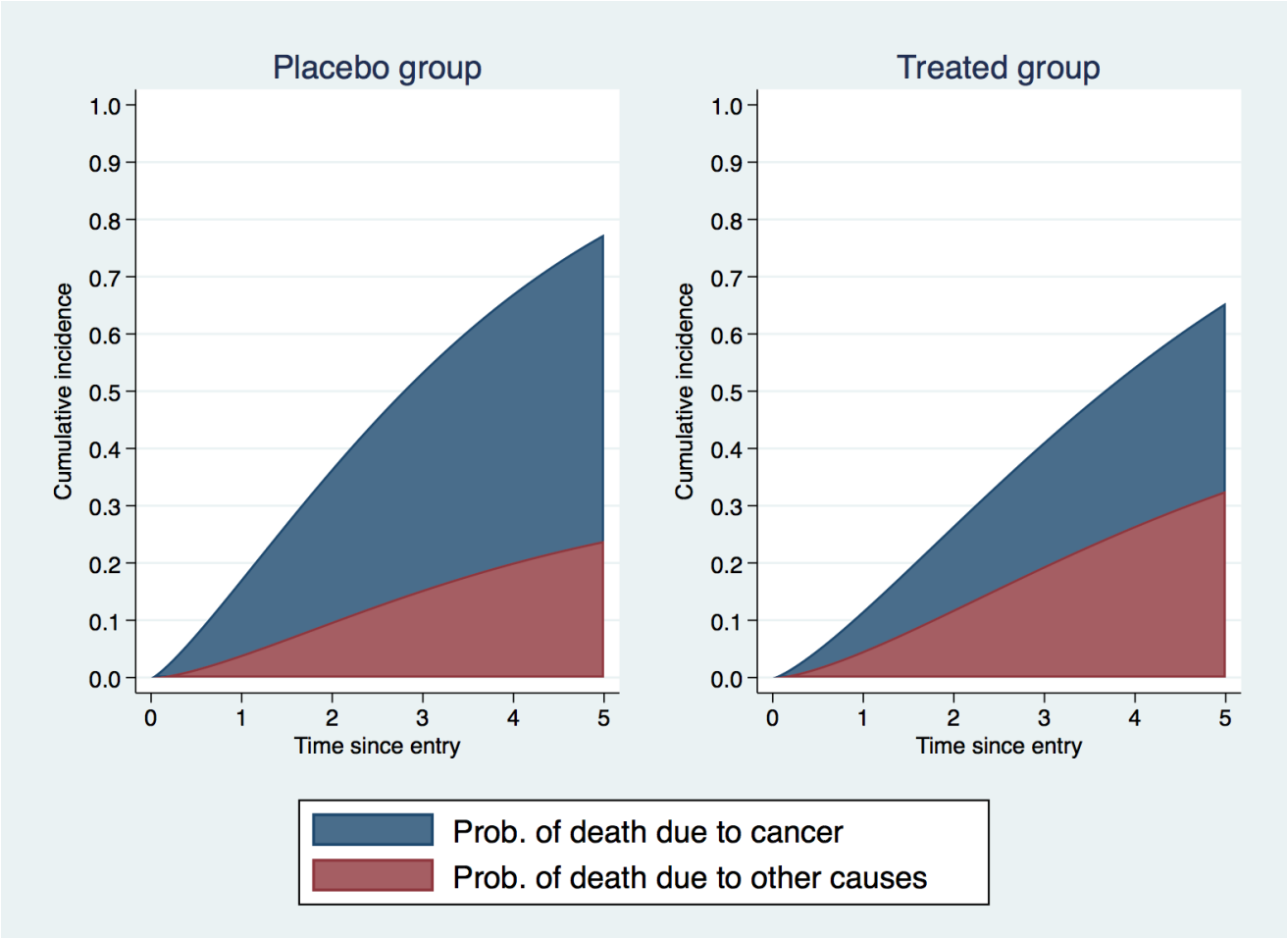
```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               M3[id]@1                     /// random effect
               , family(gamma)              /// distribution
       )                                    ///
       (stime  trt                          /// resp. + covariate
               dEV[logb] EV[pro]            /// associations
               , family(rp, df(3)           /// distribution
                       failure(other))      /// event indicator
       )                                    ///
       ,                                    /// main options
       covariance(unstructured)             /// vcv
       redistribution(t) df(5)              //  re dist.
```

```
merlin (logb                                /// log serum bilirubin
               time                         /// covariate
               time#trt                     /// interaction
               M1[id]@1                     /// random intercept
               time#M2[id]@1                /// random slope
               ,                            /// options
               family(gaussian)             /// distribution
       )                                    ///
       (pro                                 /// prothrombin index
               rcs(time, df(3))             /// covariate
               M3[id]@1                     /// random effect
               , family(gamma)              /// distribution
       )                                    ///
       (stime  trt                          /// resp. + covariate
               trt#fp(stime, power(0))      /// tde
               dEV[logb] EV[pro]            /// associations
               , family(rp, df(3)           /// distribution
                       failure(other))      /// event indicator
       )                                    ///
       ,                                    /// main options
       covariance(unstructured)             /// vcv
       redistribution(t) df(5)              //  re dist.
```

```
merlin (logb time time#trt M1[id]@1                  /// model 1
                time#M2[id]@1 ,                      ///
                family(gaussian)                     ///
       )                                             ///
       (pro  rcs(time, df(3)) M3[id]@1               /// model 2
                , family(gamma)                      ///
       )                                             ///
       (stime  trt                                   ///
                trt#fp(stime, power(0))              /// model 3: cause 1
                dEV[logb] EV[pro]                    /// tde
                , family(rp, df(3)                   /// distribution
                         failure(other))             /// event indicator
       )                                             ///
       (stime  trt                                   /// model 4: cause 2
                trt#rcs(stime, df(3) log)            /// tde
                EV[logb] iEV[pro]                    /// associations
                , family(weibull,                    /// distribution
                         failure(cancer))            /// event indicator
       )                                             ///
       ,                                             ///
       covariance(unstructured)
```

predictions

```
predict cif1, cif marginal outcome(3) at(trt 0)
predict cif1, cif marginal outcome(4) at(trt 0)
```



a user-defined model

```
real matrix gauss_logl(gml)
{
     y         = merlin_util_depvar(gml)          // dep. var.
     linpred   = merlin_util_xzb(gml)             // lin. pred.
     sdre      = exp(merlin_util_ap(gml,1))       // anc. param.
     return(lnnormalden(y,linpred,sdre))         // logl
}

merlin (logb ... , family(user, llfunction(gauss_logl) nap(1)))
       ...
       ...
       ...
```

a user-defined model

```
real matrix gauss_logl(gml)
{
     y        = merlin_util_depvar(gml)          // dep. var.
     linpred  = merlin_util_xzb(gml)             // lin. pred.
     sdre     = exp(merlin_util_xzb_mod(gml,2)) // anc. param.
     return(lnnormalden(y,linpred,sdre))        // logl
}

merlin (logb ...     , family(user, llfunction(gauss_logl)))
       (age M1[id]@1, family(null))
       ...
       ...
```

Things I didn't show

- random effects at arbitrary levels - `M4[centre>id]@1`
- B-splines - `bs(time, df(3) order(4))`
- `d2EV[]`, `?XB[]`
- `ltruncated(varname)` - left-truncation
- 9 (so far) other inbuilt families, e.g. `beta`, `ologit`
- `bhazard(varname)` - relative survival
- `mf(func_name)` - user-defined element function

Part 2: Multi-state modelling

Survival analysis with merlin

merlin can fit very simple models

```
. merlin (stime trt age, family(weibull, failure(died)))
```

Survival analysis with merlin

merlin can fit very simple models

```
. merlin (stime trt age, family(weibull, failure(died)))
```

merlin can fit very complex models

```
. merlin (stime trt age dEV[sbp] EV[sbp] trt#stime,
>                 family(rp, failure(cause1) df(3)) timevar(stime))
>        (stime trt age EV[sbp] iEV[dbp],
>                 family(weibull, failure(cause2)) timevar(stime))
>        (sbp fp(time, powers(1 2) M2[id]@1, family(gaussian) timevar(time))
>        (dbp rcs(time, df(3)) M1[id]@1, family(gaussian) timevar(time))
```

Survival analysis with merlin

merlin can fit very simple models

```
. merlin (stime trt age, family(weibull, failure(died)))
```

merlin can fit very complex models

```
. merlin (stime trt age dEV[sbp] EV[sbp] trt#stime,
>                 family(rp, failure(cause1) df(3)) timevar(stime))
>        (stime trt age EV[sbp] iEV[dbp],
>                 family(weibull, failure(cause2)) timevar(stime))
>        (sbp fp(time, powers(1 2) M2[id]@1, family(gaussian) timevar(time))
>        (dbp rcs(time, df(3)) M1[id]@1, family(gaussian) timevar(time))
```

stmerlin provides an easier interface

```
. stset _stop, fail(_status=1) enter(_start)
. stmerlin hormon age if _trans==1, distribution(weibull)
```

Example model - Transition 1

Royston-Parmar model with non-PH

```
. stmerlin hormon age sz2 sz3 nodes pr_1 if _trans==1, dist(rp) df(3)
>          tvc(sz2 sz3 pr_1) dftvc(1)

Survival model                                  Number of obs     =      2,982
Log likelihood = -4790.4569
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_t:          |
      hormon |   -.079763   .0824503    -0.97   0.333    -.2413627    .0818367
         age |  -.0062709   .0021004    -2.99   0.003    -.0103875   -.0021542
         sz2 |   .4777274   .0634815     7.53   0.000     .3533059     .602149
         sz3 |   .7445507   .0904353     8.23   0.000     .5673008    .9218006
       nodes |   .0784041   .0045454    17.25   0.000     .0694953    .0873129
        pr_1 |  -.0783096   .0122403    -6.40   0.000    -.1023002    -.054319
   sz2#rcs() |  -.1740493   .0446888    -3.89   0.000    -.2616378   -.0864608
   sz3#rcs() |  -.2669211   .0616148    -4.33   0.000    -.3876839   -.1461583
  pr_1#rcs() |   .0728241   .0086396     8.43   0.000     .0558907    .0897574
       _cons |  -.9480283    .126609    -7.49   0.000    -1.196177   -.6998791
------------------------------------------------------------------------------
    Warning: Baseline spline coefficients not shown - use ml display

. estimates store m1
```

Example model - Transition 2

Weibull PH model

```
. stmerlin hormon age sz2 sz3 nodes pr_1 if _trans==2, dist(weibull)

Survival model                                  Number of obs     =      2,982
Log likelihood = -4962.3641
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_t:          |
      hormon |  -.0014575   .0821299    -0.02   0.986    -.1624291    .1595142
         age |  -.0062153   .0021012    -2.96   0.003    -.0103335    -.002097
         sz2 |   .3739363   .0580319     6.44   0.000     .2601959    .4876767
         sz3 |   .6799465   .0868836     7.83   0.000     .5096577    .8502353
       nodes |   .0811535   .0044792    18.12   0.000     .0723744    .0899326
        pr_1 |  -.0408655   .0115458    -3.54   0.000    -.0634949   -.0182362
       _cons |   -2.35664   .1321145   -17.84   0.000    -2.615579     -2.0977
  log(gamma) |  -.0211339   .0218543    -0.97   0.334    -.0639676    .0216997
------------------------------------------------------------------------------

. estimates store m2
```

Example model - Transition 3

Royston-Parmar model with NPH

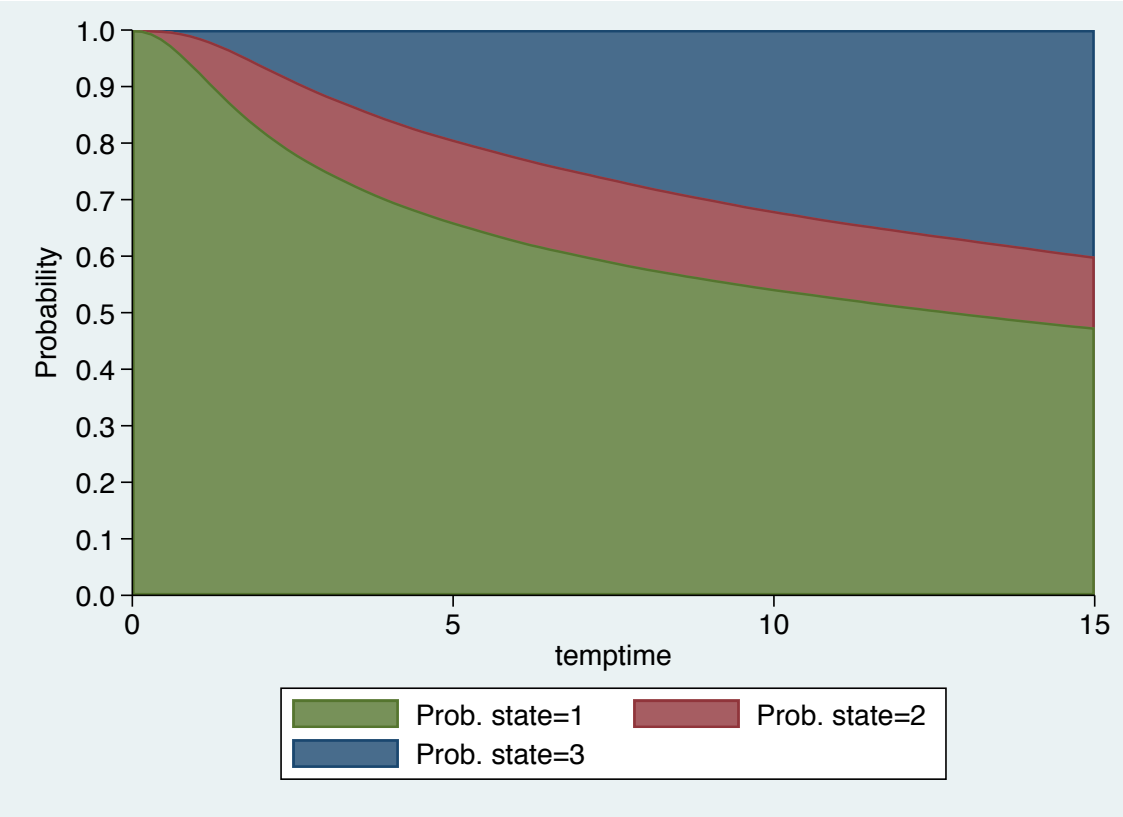
```
. stmerlin hormon age sz2 sz3 nodes pr_1 if _trans==3,
>          dist(rp) df(3) tvc(pr_1) dftvc(1)

Survival model                                  Number of obs     =      2,982
Log likelihood = -4802.1916
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_t:          |
      hormon |  -.0765467    .082381    -0.93   0.353    -.2380104     .084917
         age |  -.0062749   .0020982    -2.99   0.003    -.0103874   -.0021624
         sz2 |   .3755491   .0580081     6.47   0.000     .2618553    .4892428
         sz3 |   .6515838   .0873452     7.46   0.000     .4803903    .8227773
       nodes |   .0794874   .0045112    17.62   0.000     .0706456    .0883292
        pr_1 |  -.0795757   .0122651    -6.49   0.000    -.1036149   -.0555366
  pr_1#rcs() |   .0756141   .0086727     8.72   0.000     .0586159    .0926122
       _cons |  -.8641675   .1247649    -6.93   0.000    -1.108702   -.6196327
------------------------------------------------------------------------------
    Warning: Baseline spline coefficients not shown - use ml display

. estimates store m3
```

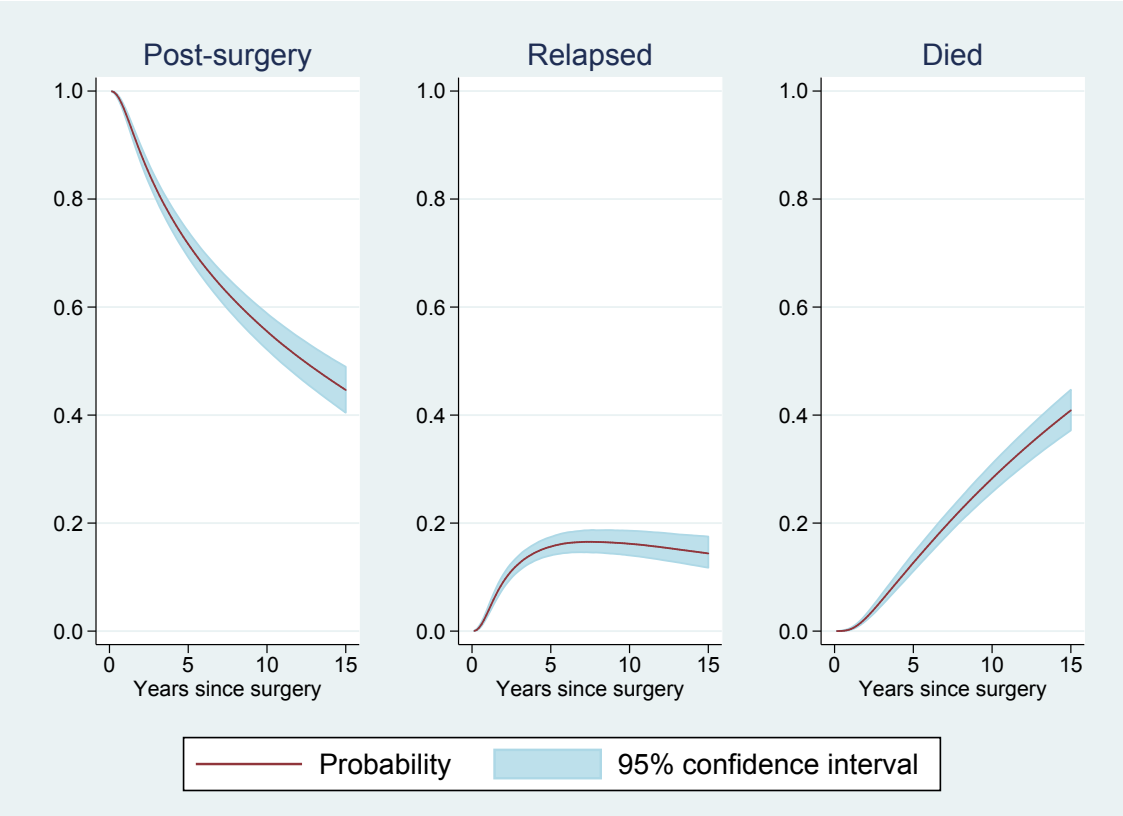
Predicting transition probabilities

```
. predictms , transmat(tmat)      /// transition matrix
              models(m1 m2 m3)    /// fitted objects
              probability         /// request trans. probs.
              at1(age 55)         //  conditional prediction
```



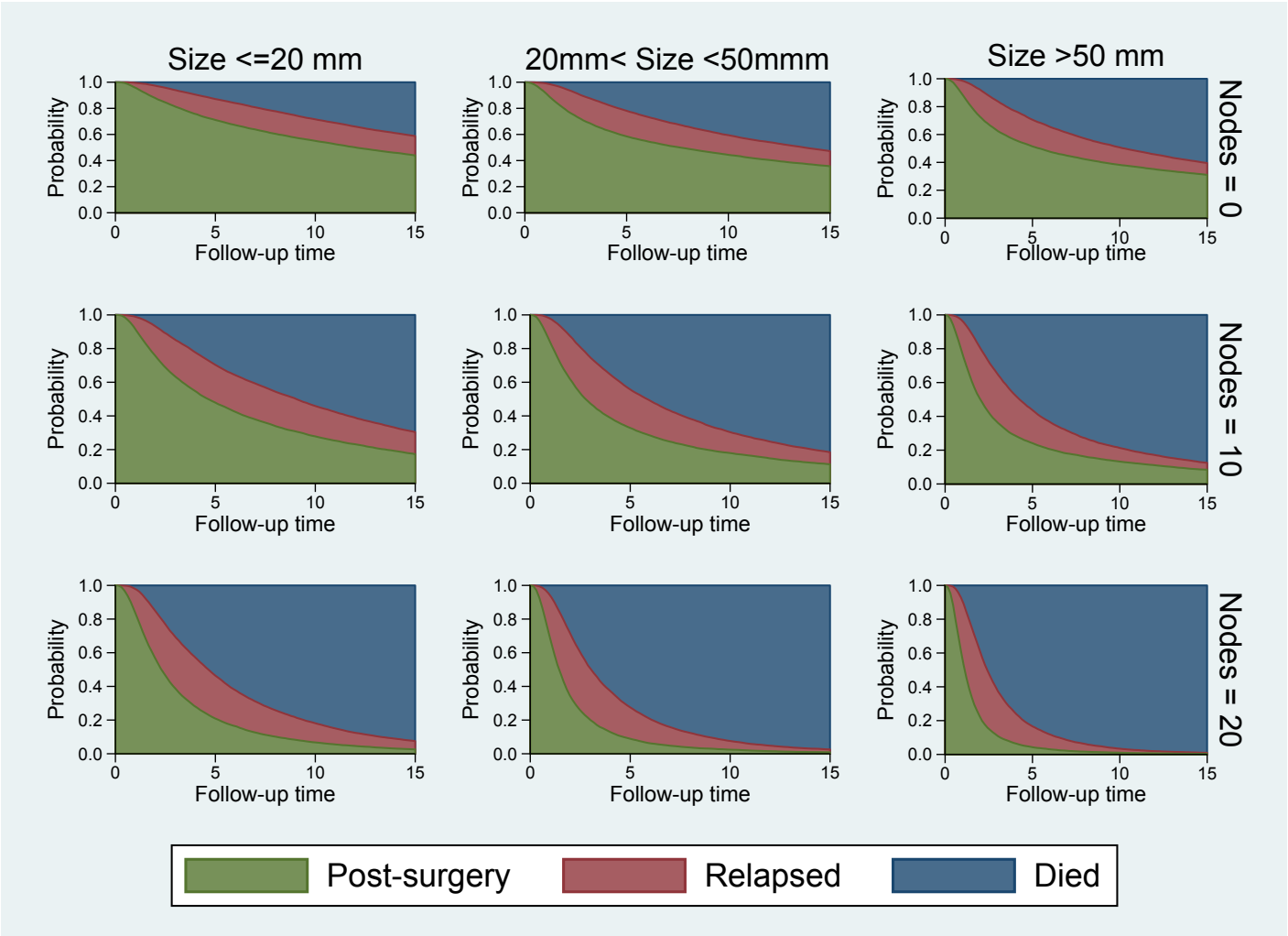
Predicting transition probabilities with CIs

```
. predictms , transmat(tmat) models(m1 m2 m3)   ///
              probability                       /// request trans. probs.
              at1(age 55)                       /// conditional prediction
              ci                                //  confidence intervals
```



Multiple at#()s

```
. predictms , transmat(tmat) models(m1 m2 m3) probability  ///
              at1(age 55 nodes 10)                         ///
              at2(age 55 nodes 10 sz2 1)                   ///
              at3(age 55 nodes 10 sz3 1)                   ///
```

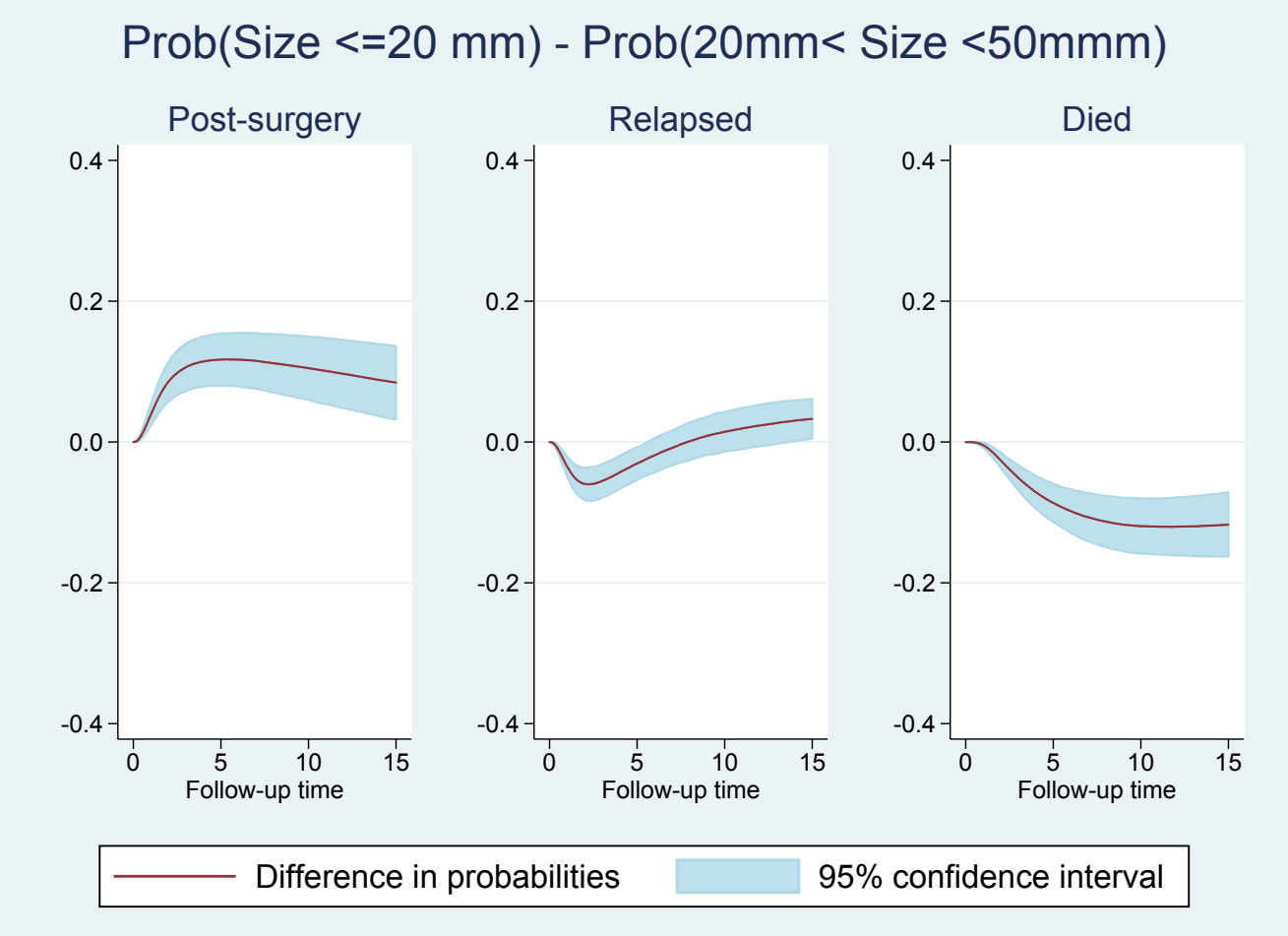


Contrasts

- It's easy to show predictions for a particular covariate pattern, but what about showing the impact of differences in covariate patterns?
- How does treatment change the probability of being in each state?
- How does tumour size at diagnosis influence these probabilities?
- We can use contrasts - differences and ratios

Differences across at#()s

```
. predictms , transmat(tmat) models(m1 m2 m3) probability  ///
              at1(age 55 nodes 10)                         ///
              at2(age 55 nodes 10 sz2 1)                   ///
              at3(age 55 nodes 10 sz3 1)                   ///
              difference atref(2) ci
```



Length of stay in a state

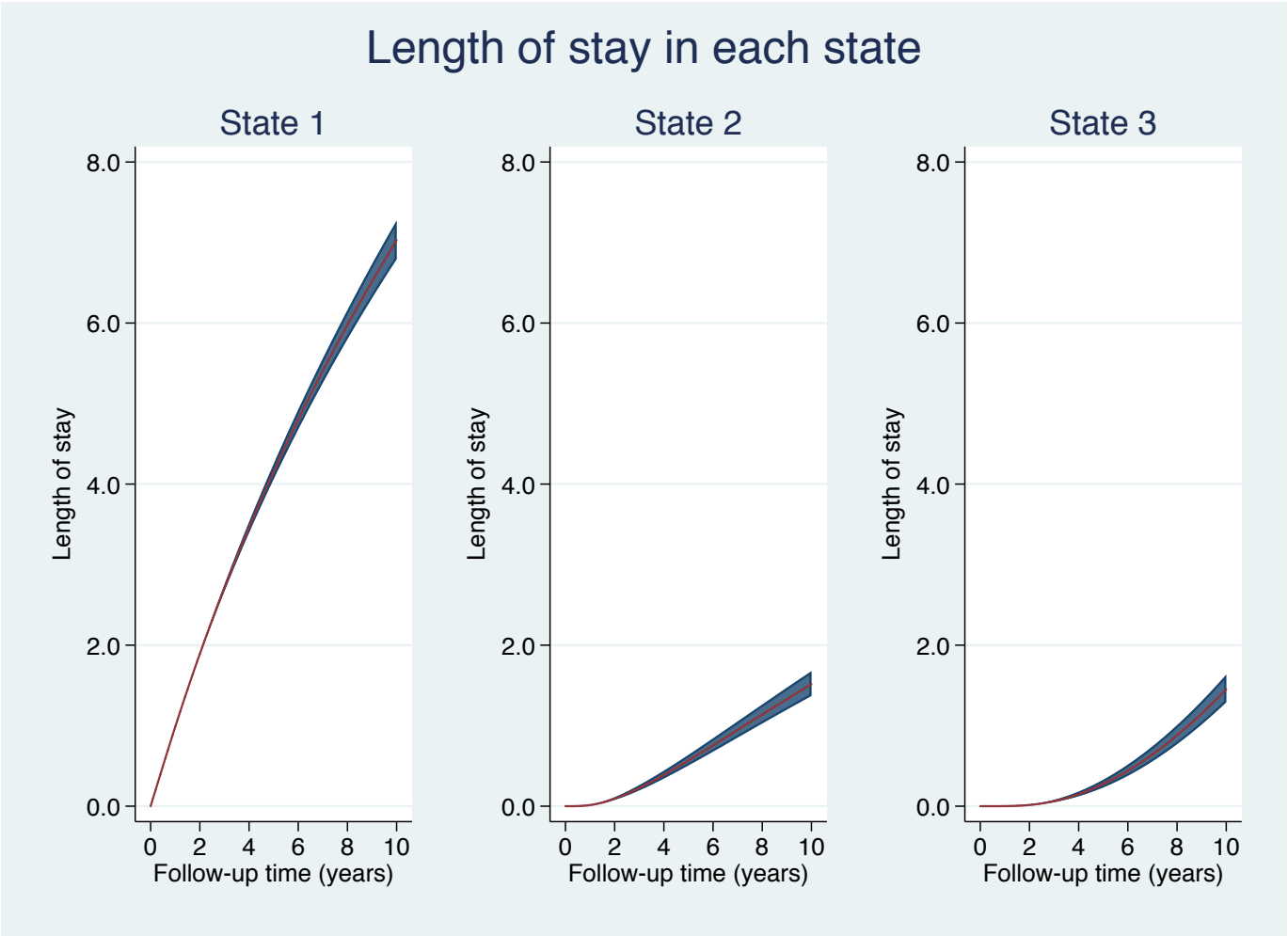
A clinically useful measure is called length of stay, which defines the amount of time spent in a particular state,

$$\int_s^t P(Y(u) = b | Y(s) = a) du.$$

This is the multi-state equivalent of restricted mean survival time [12]

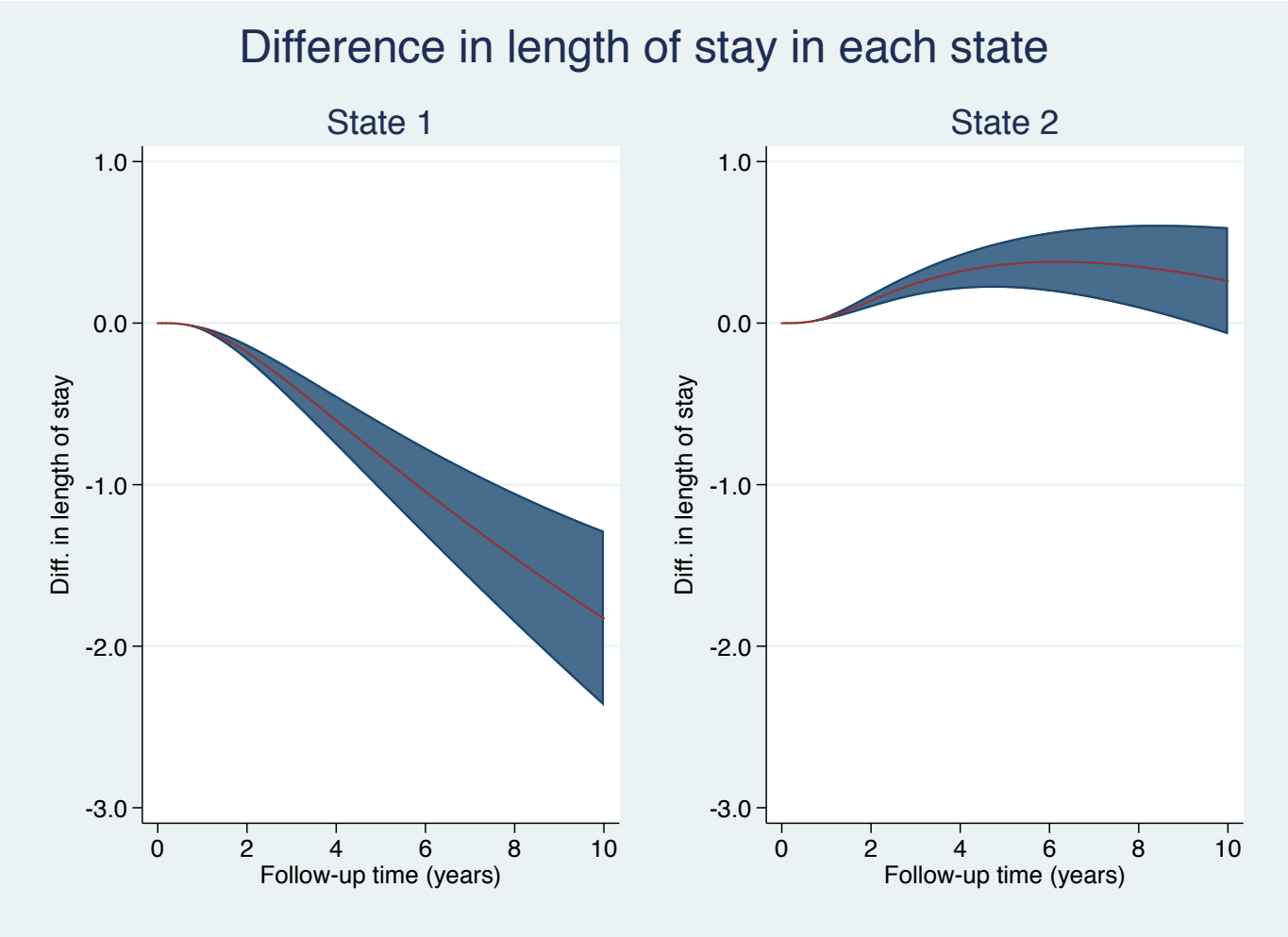
Length of stay

```
. predictms , transmat(tmat) models(m1 m2 m3) probability        ///
              at1(age 55 nodes 10) at2(age 55 nodes 10 sz2 1)    ///
              at3(age 55 nodes 10 sz3 1)                         ///
              difference ci                                      ///
              los
```



Differences in length of stay

```
. predictms , transmat(tmat) models(m1 m2 m3) probability       ///
              at1(age 55 nodes 10) at2(age 55 nodes 10 sz2 1)   ///
              at3(age 55 nodes 10 sz3 1)                        ///
              difference ratio ci                               ///
              los
```



Further topics

- Markov assumption
- Multiple timescales:
  - Time since diagnosis
  - Attained age
  - Calendar time
  - Time since intermediate events
- Interval censoring
- Frailties and random effects
- Marginal/standardised predictions

Future work

- `merlin` can do a lot of things, hopefully in a usable way
- `merlin` is easily extended - adding one thing usually opens up a lot more
- It's general, and hence it can be slow(er)
- Much more on:

reddooranalytics.se

References

[1] Crowther MJ, Abrams KR, Lambert PC. Flexible parametric joint modelling of longitudinal and survival data. *Stat Med* 2012; **31**(30):4456–4471, doi:10.1002/sim.5644. URL http://dx.doi.org/10.1002/sim.5644.

[2] Crowther MJ, Abrams KR, Lambert PC. Joint modeling of longitudinal and survival data. *Stata J* 2013; **13**(1):165–184.

[3] Crowther MJ, Lambert PC, Abrams KR. Adjusting for measurement error in baseline prognostic biomarkers included in a time-to-event analysis: A joint modelling approach. *BMC Med Res Methodol* 2013; **13**(146).

[4] Crowther MJ, Andersson TML, Lambert PC, Abrams KR, Humphreys K. Joint modelling of longitudinal and survival data: incorporating delayed entry and an assessment of model misspecification. *Statistics in medicine* 2016; **35**(7):1193–1209.

[5] Crowther MJ, Look MP, Riley RD. Multilevel mixed effects parametric survival models using adaptive gauss-hermite quadrature with application to recurrent events and individual participant data meta-analysis. *Stat Med* Sep 2014; **33**(22):3844–3858, doi:10.1002/sim.6191. URL http://dx.doi.org/10.1002/sim.6191.

[6] Crowther MJ. Multilevel mixed-effects parametric survival analysis: Estimation, simulation, and application ; **19**:931–949, doi:10.1177/1536867x19893639.

[7] Crowther MJ, Lambert PC. A general framework for parametric survival analysis. *Stat Med* Dec 2014; **33**(30):5280–5297, doi:10.1002/sim.6300. URL http://dx.doi.org/10.1002/sim.6300.

[8] Crowther MJ, Lambert PC. stgenreg: A Stata package for the general parametric analysis of survival data. *J Stat Softw* 2013; **53**(12).

[9] Crowther MJ, Lambert PC. Simulating complex survival data. *Stata J* 2012; **12**(4):674–687.

[10] Crowther MJ, Lambert PC. Simulating biologically plausible complex survival data. *Stat Med* 2013; **32**(23):4118–4134, doi:10.1002/sim.5823. URL http://dx.doi.org/10.1002/sim.5823.

[11] Crowther MJ. Extended multivariate generalised linear and non-linear mixed effects models ; .

[12] Touraine C, Helmer C, Joly P. Predictions in an illness-death model. *Statistical methods in medical research* 2013; .