Multi-state survival analysis in Stata

Michael J. Crowther

Biostatistics Research Group
Department of Health Sciences
University of Leicester, UK
michael.crowther@le.ac.uk

Italian Stata Users Group Meeting
Bologna, Italy,
15th November 2018

Plan

- I will give a broad overview of multistate survival analysis
- I will focus on (flexible) parametric models
- All the way through I will show example Stata code using the `multistate` package [1]
- I'll discuss some recent extensions, and what I'm working on now

Background

- In survival analysis, we often concentrate on the time to a single event of interest
- In practice, there are many clinical examples of where a patient may experience a variety of intermediate events
  - Cancer
  - Cardiovascular disease
- This can create complex disease pathways

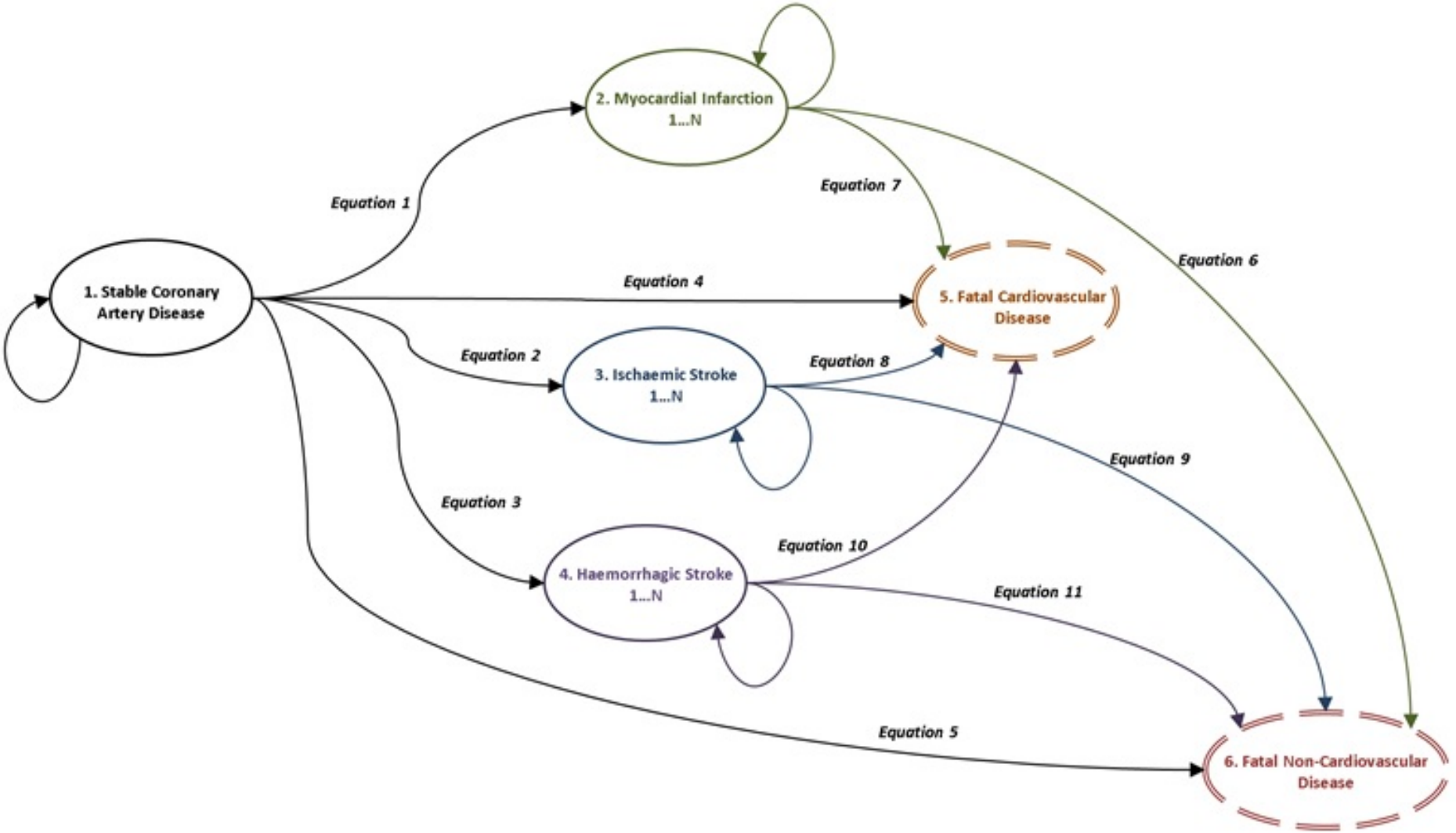


Figure 1: An example from stable coronary disease [2]

- Each transition between any two states is a survival model
- We want to investigate covariate effects for each specific transition between two states
- What if where I've been impacts where I might go?
- With the drive towards personalised medicine, and expanded availability of registry-based data sources, including data-linkage, there are substantial opportunities to gain greater understanding of disease processes, and how they change over time

Primary breast cancer [3]

- To illustrate, I use data from 2,982 patients with primary breast cancer, where we have information on the time to relapse and the time to death.
- All patients begin in the initial post-surgery state, which is defined as the time of primary surgery, and can then move to a relapse state, or a dead state, and can also die after relapse.

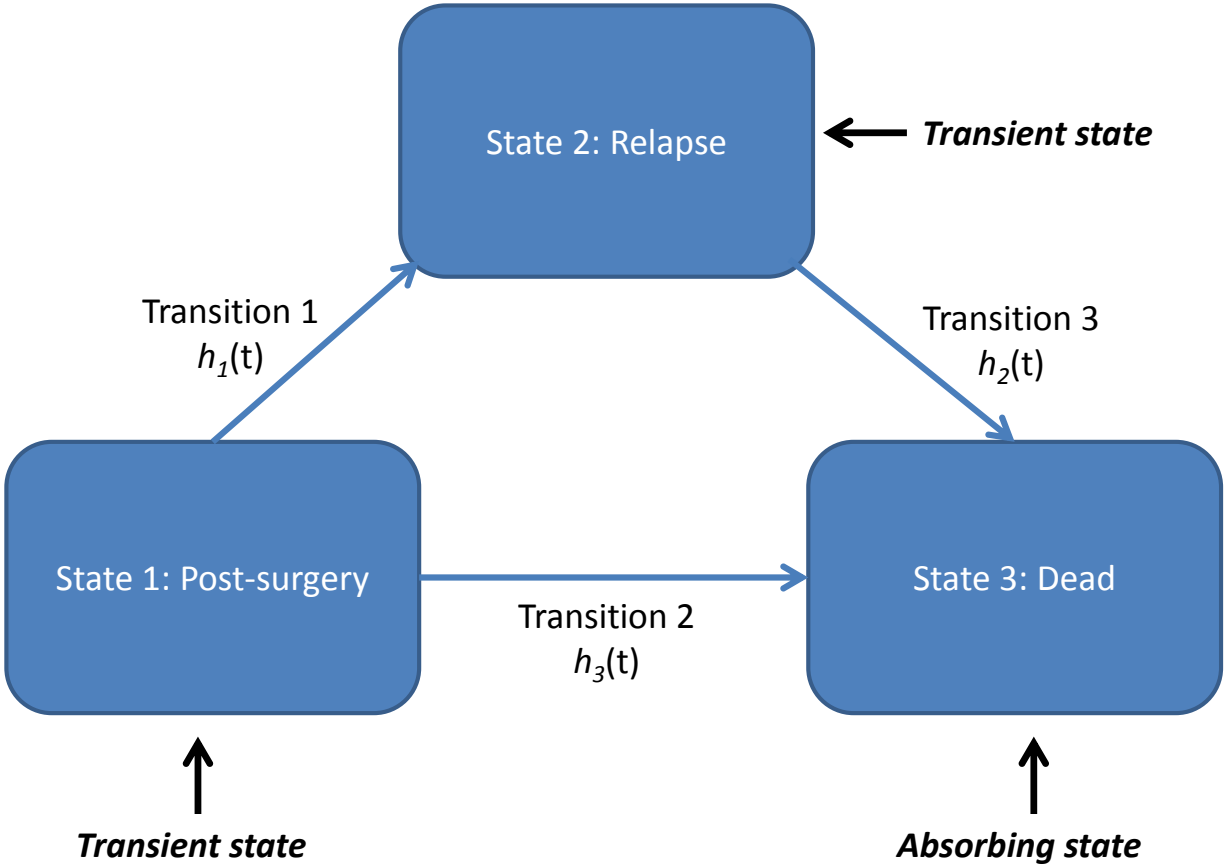


Figure 2: Illness-death model for primary breast cancer example.

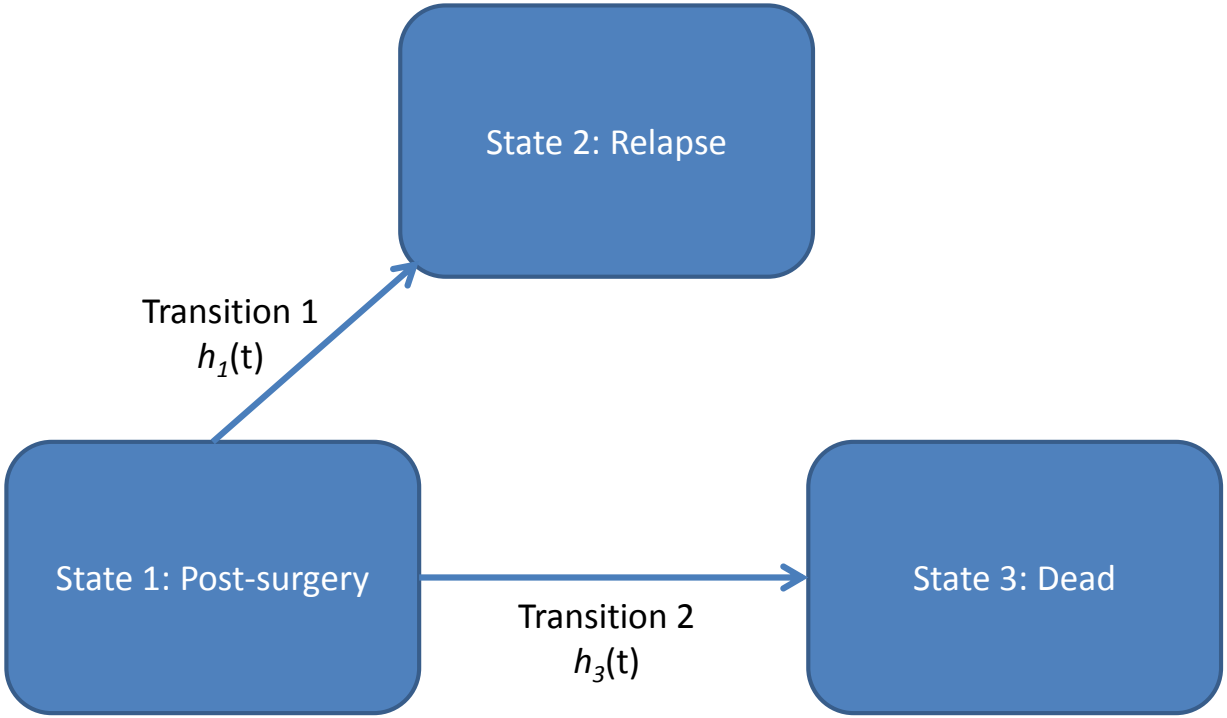


Figure 3: Illness-death model for primary breast cancer example.

Covariates of interest

- age at primary surgery
- tumour size (three classes; $\leq$ 20mm, 20-50mm, $>$ 50mm)
- number of positive nodes
- progesterone level (fmol/l) - in all analyses we use a transformation of progesterone level ($\log(pgr + 1)$)
- whether patients were on hormonal therapy (binary, yes/no)

Markov multi-state models

Consider a random process $\{Y(t), t \geq 0\}$ which takes the values in the finite state space $\mathcal{S} = \{1, \ldots, S\}$. We define the history of the process until time s, to be $\mathcal{H}_s = \{Y(u); 0 \leq u \leq s\}$. The transition probability can then be defined as,

$$P(Y(t) = b | Y(s) = a, \mathcal{H}_{s-})$$

where $a, b \in \mathcal{S}$. This is the **probability of being in state b at time t, given that it was in state a at time s and conditional on the past trajectory until time s.**

Markov multi-state models

A **Markov** multi-state model makes the following assumption,

$$P(Y(t) = b | Y(s) = a, \mathcal{H}_{s-}) = P(Y(t) = b | Y(s) = a)$$

which implies that the future behaviour of the process is only dependent on the present.

- This simplifies things for us later
- It is an assumption! We can conduct an informal test by including time spent in previous states in our model for a transition

Markov multi-state models

The transition intensity is then defined as,

$$h_{ab}(t) = \lim_{\delta t \to 0} \frac{P(Y(t+\delta t) = b | Y(t) = a)}{\delta t}$$

Or, for the kth transition from state a_k to state b_k, we have

$$h_k(t) = \lim_{\delta t \to 0} \frac{P(Y(t+\delta t) = b_k | Y(t) = a_k)}{\delta t}$$

which represents the instantaneous risk of moving from state a_k to state b_k. Our collection of transitions intensities governs the multi-state model.

This is simply a collection of survival models!

Estimating a multi-state models

- There are a variety of challenges in estimating transition probabilities in multi-state models, within both non-/semi-parametric and parametric frameworks [4], which I'm not going to go into today
- Essentially, a multi-state model can be specified by a combination of transition-specific survival models
- The most convenient way to do this is through the stacked data notation, where each patient has a row of data for each transition that they are at risk for, using start and stop notation (standard delayed entry setup)

Consider the breast cancer dataset, with recurrence-free and overall survival

```
. use http://fmwww.bc.edu/repec/bocode/m/multistate_example,clear
(Rotterdam breast cancer data, truncated at 10 years)
. list pid rf rfi os osi age if pid==1 | pid==1371, sepby(pid) noobs
```

pid	rf	rfi	os	osi	age
1	59.1	0	59.1	alive	74
1371	16.6	1	24.3	deceased	79

We can restructure using `msset`

Title

msset — data preparation for multi-state and competing risks analysis

Syntax

msset [*if*] [*in*] , **id(***varname***) states(***varlist***) times(***varlist***)** [*options*]

options	Description
id(*varname***)**	identification variable
states(*varlist***)**	indicator variables for each state
times(*varlist***)**	time variables for each state
transmatrix(*matname***)**	transition matrix
covariates(*varlist***)**	variables to expand into transition specific covariates

msset creates the following variables:

_from	starting state
_to	receiving state
_trans	transition number
_start	starting time for each transition
_stop	stopping time for each transition
_status	status variable, indicating a transition (coded 1) or censoring (coded 0)
_flag	indicator variable to show observations where changes to the original data have been made

Saved results

msset returns the following in r():

Matrices:

r(Nnextstates)	number of possible next states from starting state (row number)
r(transmatrix)	transition matrix
r(freqmatrix)	frequencies of transitions

```
. use http://fmwww.bc.edu/repec/bocode/m/multistate_example,clear
(Rotterdam breast cancer data, truncated at 10 years)
. list pid rf rfi os osi age if pid==1 | pid==1371, sepby(pid) noobs
```

pid	rf	rfi	os	osi	age
1	59.1	0	59.1	alive	74
1371	16.6	1	24.3	deceased	79

```
. msset, id(pid) states(rfi osi) times(rf os) covariates(age)
variables age_trans1 to age_trans3 created

. mat tmat = r(transmatrix)
. mat list tmat
tmat[3,3]
              to:     to:     to:
           start     rfi     osi
from:start     .       1       2
  from:rfi     .       .       3
  from:osi     .       .       .
```

```
. //wide (before msset)
. list pid rf rfi os osi age if pid==1 | pid==1371, sepby(pid)
```

pid	rf	rfi	os	osi	age
1	59.1	0	59.1	alive	74
1371	16.6	1	24.3	deceased	79

```
. //long (after msset)
. list pid _from  _to _start _stop _status _trans if pid==1 | pid==1371, noobs
```

pid	_from	_to	_start	_stop	_status	_trans
1	1	2	0	59.104721	0	1
1	1	3	0	59.104721	0	2
1371	1	2	0	16.558521	1	1
1371	1	3	0	16.558521	0	2
1371	2	3	16.558521	24.344969	1	3

```
. use http://fmwww.bc.edu/repec/bocode/m/multistate_example,clear
(Rotterdam breast cancer data, truncated at 10 years)

. msset, id(pid) states(rfi osi) times(rf os) covariates(age)
variables age_trans1 to age_trans3 created

. mat tmat = r(transmatrix)

. stset _stop, enter(_start) failure(_status=1) scale(12)

     failure event:  _status == 1
obs. time interval:  (0, _stop]
 enter on or after:  time _start
 exit on or before:  failure
    t for analysis:  time/12
--------------------------------------------------------------------------------
      7,482  total observations
          0  exclusions
--------------------------------------------------------------------------------
      7,482  observations remaining, representing
      2,790  failures in single-record/single-failure data
 38,474.539  total analysis time at risk and under observation
                                                at risk from t =         0
                                     earliest observed entry t =         0
                                          last observed exit t =  19.28268
```

- Now our data is restructured and declared as survival data, we can use any standard survival model available within Stata
  - Proportional baselines across transitions
  - Stratified baselines
  - Shared or separate covariate effects across transitions
- This is all easy to do in Stata; however, calculating transition probabilities (what we are generally most interested in!) is not so easy. We'll come back to this later...

Examples

Proportional Weibull baseline hazards

```
. streg _trans2 _trans3, dist(weibull) nohr nolog
        failure _d:  _status == 1
  analysis time _t:  _stop/12
 enter on or after:  time _start
Weibull PH regression
No. of subjects =        7,482                 Number of obs    =        7,482
No. of failures =        2,790
Time at risk    =  38474.53852
                                               LR chi2(2)       =      2701.63
Log likelihood  =   -5725.5272                 Prob > chi2      =      0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_trans2	-2.052149	.0760721	-26.98	0.000	-2.201248	-1.903051
_trans3	1.17378	.0416742	28.17	0.000	1.0921	1.25546
_cons	-2.19644	.0425356	-51.64	0.000	-2.279808	-2.113072
/ln_p	-.1248857	.0197188	-6.33	0.000	-.1635337	-.0862376
p	.8825978	.0174037			.8491379	.9173763
1/p	1.133019	.0223417			1.090065	1.177665

Examples

Separate (stratified) Weibull baselines

```
. streg _trans2 _trans3, dist(weibull) anc(_trans2 _trans3) nohr nolog
        failure _d:  _status == 1
  analysis time _t:  _stop/12
 enter on or after:  time _start
Weibull PH regression
No. of subjects =        7,482                 Number of obs    =       7,482
No. of failures =        2,790
Time at risk    =  38474.53852
                                               LR chi2(2)       =      935.32
Log likelihood  =   -5656.1627                 Prob > chi2      =      0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_t						
_trans2	-3.168605	.2013437	-15.74	0.000	-3.563232	-2.773979
_trans3	2.352642	.1522638	15.45	0.000	2.05421	2.651073
_cons	-2.256615	.0477455	-47.26	0.000	-2.350194	-2.163035
ln_p						
_trans2	.4686402	.063075	7.43	0.000	.3450155	.592265
_trans3	-.6043193	.087695	-6.89	0.000	-.7761984	-.4324403
_cons	-.0906001	.0224852	-4.03	0.000	-.1346702	-.0465299

Examples

Separate (stratified) Weibull baselines and age

```
. streg age _trans2 _trans3, dist(weibull) anc(_trans2 _trans3) nohr nolog
        failure _d:  _status == 1
  analysis time _t:  _stop/12
 enter on or after:  time _start
Weibull PH regression
No. of subjects =        7,482                 Number of obs    =       7,482
No. of failures =        2,790
Time at risk    =  38474.53852
                                               LR chi2(3)       =      968.10
Log likelihood  =   -5639.7693                 Prob > chi2      =      0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_t						
age	.0085662	.0014941	5.73	0.000	.0056379	.0114946
_trans2	-3.173808	.2017164	-15.73	0.000	-3.569165	-2.778451
_trans3	2.324363	.1505177	15.44	0.000	2.029354	2.619373
_cons	-2.7353	.0971366	-28.16	0.000	-2.925684	-2.544916
ln_p						
_trans2	.4697586	.0630304	7.45	0.000	.3462214	.5932959
_trans3	-.5827026	.0858211	-6.79	0.000	-.7509089	-.4144963
_cons	-.0873818	.0224793	-3.89	0.000	-.1314404	-.0433231

Examples

Separate (stratified) Weibull baselines and age

```
. streg age_* _trans2 _trans3, dist(weibull) anc(_trans2 _trans3) nohr nolog noshow
Weibull PH regression
No. of subjects =        7,482                 Number of obs    =       7,482
No. of failures =        2,790
Time at risk    =  38474.53852
                                               LR chi2(5)       =     1314.91
Log likelihood  =   -5466.3633                 Prob > chi2      =      0.0000
```

_t	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
_t						
age_trans1	-.0021734	.002071	-1.05	0.294	-.0062325	.0018857
age_trans2	.1289129	.0078069	16.51	0.000	.1136116	.1442142
age_trans3	.0063063	.0023447	2.69	0.007	.0017107	.0109019
_trans2	-11.78602	.623599	-18.90	0.000	-13.00825	-10.56379
_trans3	1.861322	.2348573	7.93	0.000	1.40101	2.321634
_cons	-2.13714	.1230997	-17.36	0.000	-2.378411	-1.895869
ln_p						
_trans2	.5773103	.0617153	9.35	0.000	.4563505	.6982701
_trans3	-.585393	.0865301	-6.77	0.000	-.7549889	-.415797
_cons	-.0913214	.0224979	-4.06	0.000	-.1354165	-.0472262

Fitting one model to the stacked data

- The previous examples all fit 'one' model to the full stacked dataset
- This is convenient
  - Data setup is nice and clean
  - We can share effects across transitions
- This is not convenient
  - Syntax can get tricky with lots of interactions
  - We are restricted to the same distributional form for all transition models

Fitting separate models to the stacked data

Before we had:

Separate (stratified) Weibull baselines and age

```
streg age_* _trans2 _trans3, dist(weibull) anc(_trans2 _trans3)
```

We can fit the same model with:

Separate (stratified) Weibull baselines and age

```
streg age if _trans1==1, dist(weibull)

streg age if _trans2==1, dist(weibull)

streg age if _trans3==1, dist(weibull)
```

Fitting transition-specific models to the stacked data

- We gain substantially more flexibility
- No longer restricted to one distribution
- Much easier in terms of model specification/syntax
- Transition models could come from different datasets!

Building our transition models

Returning to the breast cancer dataset

- Choose the best fitting parametric survival model, using AIC and BIC
- Comparing:
  - exponential
  - Weibull
  - Gompertz
  - Royston-Parmar
  - Splines on the log hazard scale
  - ...

Building our transition models

We find...

- Transition 1 - RP model with 3 degrees of freedom

```
stpm2 if _trans1==1, scale(h) df(3)
```

- Transition 2 - Weibull

```
streg if _trans2==1, distribution(weibull)
```

- Transition 3 - RP model with 3 degrees of freedom

```
stpm2 if _trans3==1, scale(h) df(3)
```

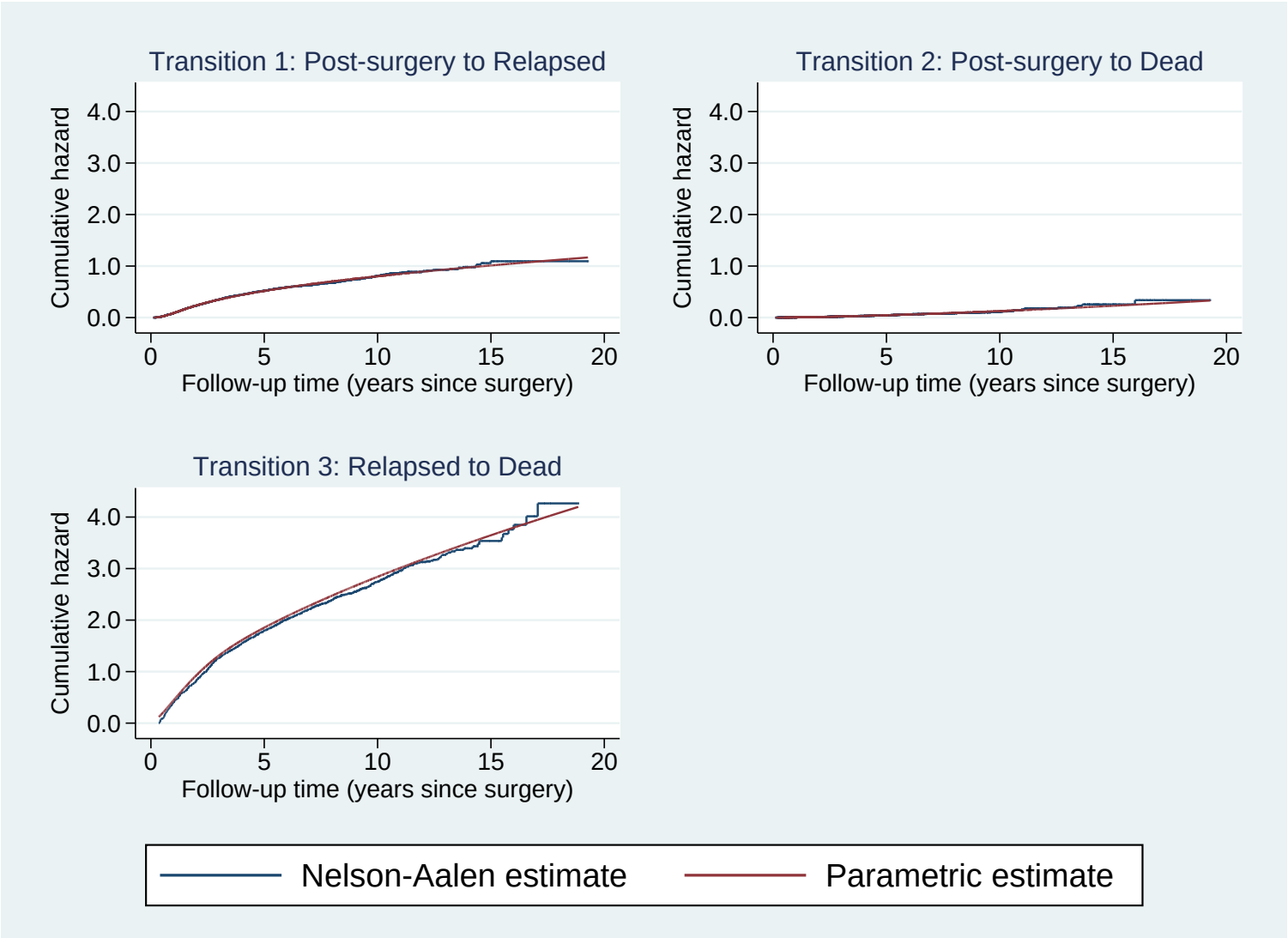


Figure 4: Best fitting parametric cumulative hazard curves overlaid on the Nelson-Aalen estimate for each transition.

Building our transition models

Next:

- Adjust for important covariates; age, tumour size, number of nodes, progesterone level
- Check proportional hazards assumption

Final models

- Transition 1: Royston-Parmar baseline with df=3. Non-PH in tumour size (both levels) and progesterone level, modelled with interaction with log time.

```
. stpm2 age sz2 sz3 nodes hormon pr_1 if _trans1==1, scale(h) df(3) ///
> tvc(sz2 sz3 pr_1) dftvc(1) nolog
Log likelihood = -3476.6455                     Number of obs     =      2,982
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
xb						
age	-.0062709	.0021004	-2.99	0.003	-.0103875	-.0021543
sz2	.4777289	.0634816	7.53	0.000	.3533073	.6021505
sz3	.744544	.0904352	8.23	0.000	.5672943	.9217937
nodes	.0784025	.0045454	17.25	0.000	.0694937	.0873113
hormon	-.0797426	.0824504	-0.97	0.333	-.2413424	.0818572
pr_1	-.0783066	.0122404	-6.40	0.000	-.1022973	-.0543159
_rcs1	.9703563	.0472652	20.53	0.000	.8777182	1.062994
_rcs2	.3104222	.0218912	14.18	0.000	.2675162	.3533282
_rcs3	-.0176099	.0114839	-1.53	0.125	-.0401179	.0048982
_rcs_sz21	-.1740546	.0446893	-3.89	0.000	-.261644	-.0864652
_rcs_sz31	-.2669255	.0616161	-4.33	0.000	-.3876909	-.1461601
_rcs_pr_11	.072824	.0086399	8.43	0.000	.0558901	.0897578
_cons	-.9480559	.1266088	-7.49	0.000	-1.196205	-.6999071

Final models

- Transition 2: Weibull baseline.

```
. streg age sz2 sz3 nodes hormon pr_1 if _trans2==1, distribution(weibull) ///
> nolog noshow noheader
```

_t	Haz. Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	1.133232	.0090317	15.69	0.000	1.115668	1.151073
sz2	1.175333	.1897555	1.00	0.317	.8565166	1.61282
sz3	1.514838	.3533698	1.78	0.075	.9589683	2.392919
nodes	1.044921	.0190746	2.41	0.016	1.008197	1.082984
hormon	.8694367	.1992656	-0.61	0.542	.5548194	1.362462
pr_1	1.022602	.0341792	0.67	0.504	.9577593	1.091835
_cons	8.13e-07	5.06e-07	-22.55	0.000	2.40e-07	2.75e-06
/ln_p	.5106518	.0572511	8.92	0.000	.3984416	.622862
p	1.666377	.095402			1.489502	1.864256
1/p	.6001043	.0343567			.5364071	.6713655

Note: Estimates are transformed only in the first equation.
Note: _cons estimates baseline hazard.

Final models

- Transition 3: Royston-Parmar with df=3. Non-PH found in progesterone level, modelled with interaction with log time.

```
. stpm2 age sz2 sz3 nodes hormon pr_1 if _trans3==1, scale(h) df(3) ///
> tvc(pr_1) dftvc(1) nolog
note: delayed entry models are being fitted
Log likelihood = -929.11658                     Number of obs     =      1,518
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
xb						
age	.0049441	.0024217	2.04	0.041	.0001977	.0096906
sz2	.1653563	.0712326	2.32	0.020	.025743	.3049696
sz3	.3243048	.0992351	3.27	0.001	.1298075	.5188021
nodes	.0297031	.0057735	5.14	0.000	.0183873	.0410189
hormon	.0315634	.0976384	0.32	0.746	-.1598045	.2229312
pr_1	-.1843876	.0211383	-8.72	0.000	-.225818	-.1429572
_rcs1	.5057489	.0581187	8.70	0.000	.3918383	.6196595
_rcs2	.1035699	.03143	3.30	0.001	.0419681	.1651716
_rcs3	-.0100584	.0117741	-0.85	0.393	-.0331352	.0130185
_rcs_pr_11	.0636225	.0121503	5.24	0.000	.0398085	.0874366
_cons	.391217	.1659763	2.36	0.018	.0659094	.7165246

Calculating transition probabilities

Transition probabilities

$$P(Y(t) = b | Y(s) = a)$$

Or even simpler, we define **state occupation probabilities** as

$$P(Y(t) = b) = \sum_a P(Y(0) = a) P(Y(t) = b | Y(0) = a)$$

which is the *probability of being in state b at time t* [5].

When $s = 0$ and everyone starts in state a, transition probabilities are the same as state occupation probabilities.

Calculating transition probabilities

$$P(Y(t) = b | Y(s) = a)$$

There are a variety of approaches within a parametric framework

- Exponential distribution is convenient [6]
- Numerical integration [7, 8] - computationally intensive, dimension of the integration grows exponentially
- Ordinary differential equations [9] - appealing but difficult to generalise
- Simulation [10, 11, 12] - my favoured approach!

Simulation

- Given our estimated transition intensities, we simulate n patients through the transition matrix
- At specified time points, we simply count how many people are in each state, and divide by the total to get our transition probabilities
- To get confidence intervals, we draw from a multivariate normal distribution, with mean vector the estimated coefficients from the intensity models, and associated variance-covariance matrix, and repeated M times
- Some details come next...remember that the software does it all for you!

Simulating survival times

Under a general hazard model

$$h(t) = h_0(t)\exp(X(t)\beta(t))$$

$$H(t) = \int_0^t h(u)\ \mathrm{d}u, \quad S(t) = \exp[-H(t)]$$

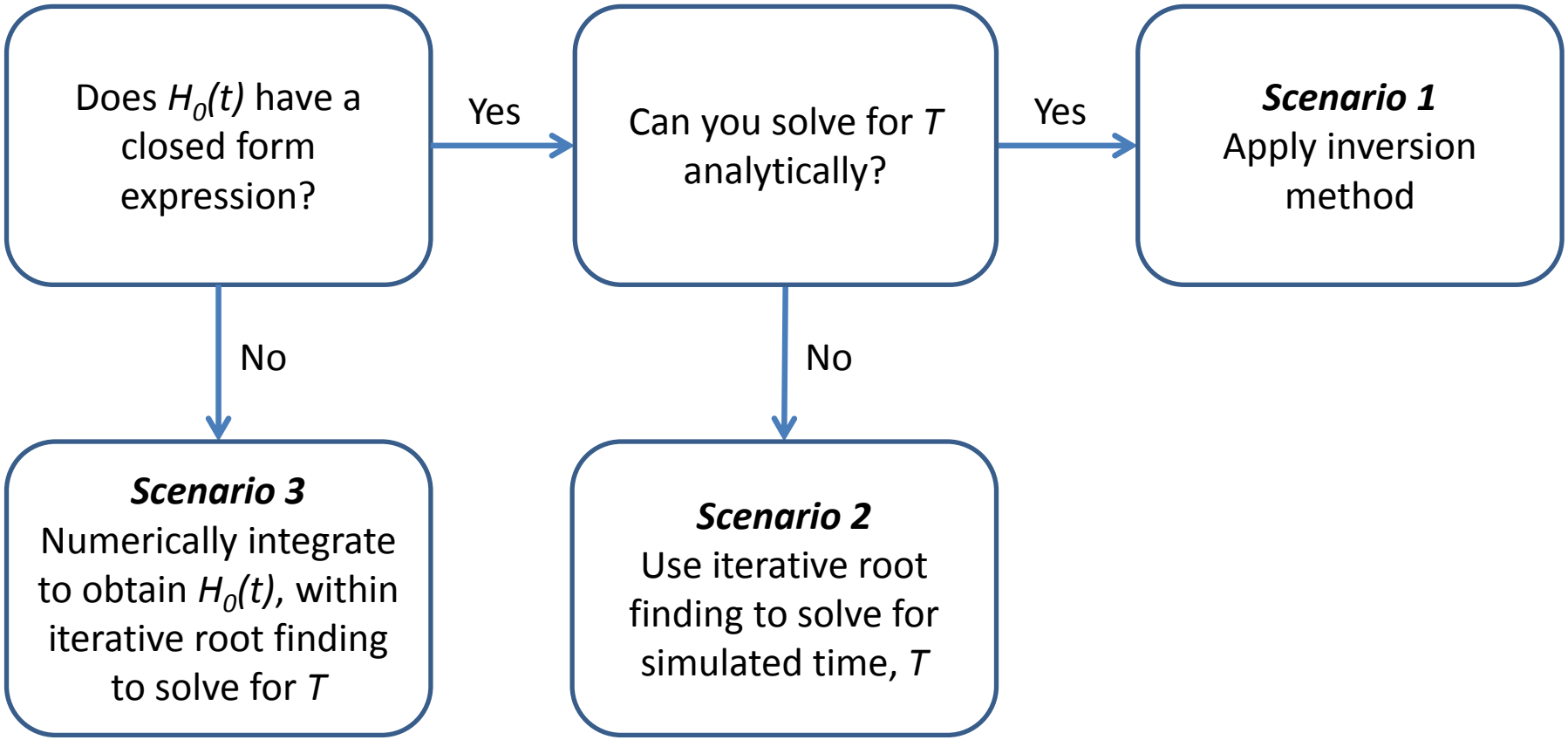
$$F(t) = 1 - \exp[-H(t)]$$

$$U = \exp[-H(t)] \sim \mathrm{U}(0,1)$$

Solve for t... Under a standard parametric PH model,

$$T = H_0^{-1}[-\log(U)\exp(-X\beta)]$$

Simulation methods [13]



Simulation methods

- Standard parametric models (Weibull, Gompertz, etc.) - closed form $H(t)$ and can invert -> extremely efficient
- Royston-Parmar model - closed form $H(t)$ but can't invert -> Brent's univariate root finder
- Splines on the log hazard scale - intractable $H(t)$ and can't invert -> numerical integration and root finding

The last two are not as computationally intensive as you would expect...

predictms

Title

predictms — predictions from a multi-state survival model

Syntax

predictms , transmatrix(*varname***)** [*options*]

options	Description
transmatrix(*matname***)**	transition matrix
models(*namelist***)**	list of estimates stored for # transition
reset	use clock-reset approach
from(*numlist***)**	starting state(s) for predictions
obs(#)	number of time points to calculate predictions at between **mint()** and **maxt()**
mint(#)	minimum time at which to calculate predictions
maxt(#)	maximum time at which to calculate predictions
timevar(*varname***)**	time points at which to calculate predictions
enter(#)	time that observations enter model, default 0, for forward predictions
exit(#)	time that observations exit the model, for fixed horizon predictions

Many more options...

Computation time in Stata with `predictms`

- Predicting transition probabilities at 20 evenly spaced points in time across follow-up
- Starting in state 1 at time 0
- Times are in seconds
- Tolerance of $<$1E-08

n	Weibulls	Royston-Parmar (df=1,5,5)	Log-hazard splines (df=1,5,5)
10,000	0.05	0.31	3.23
100,000	0.30	2.60	32.10
1,000,000	2.50	29.70	302.04
10,000,000	22.35	300.46	3010.30

Baseline only models fit to ebmt3 data

Examples

Separate baselines, transition specific age effects

```
. quietly streg age_trans1 age_trans2 age_trans3  _trans2 _trans3, ///
> dist(weibull) anc(_trans2 _trans3)
```

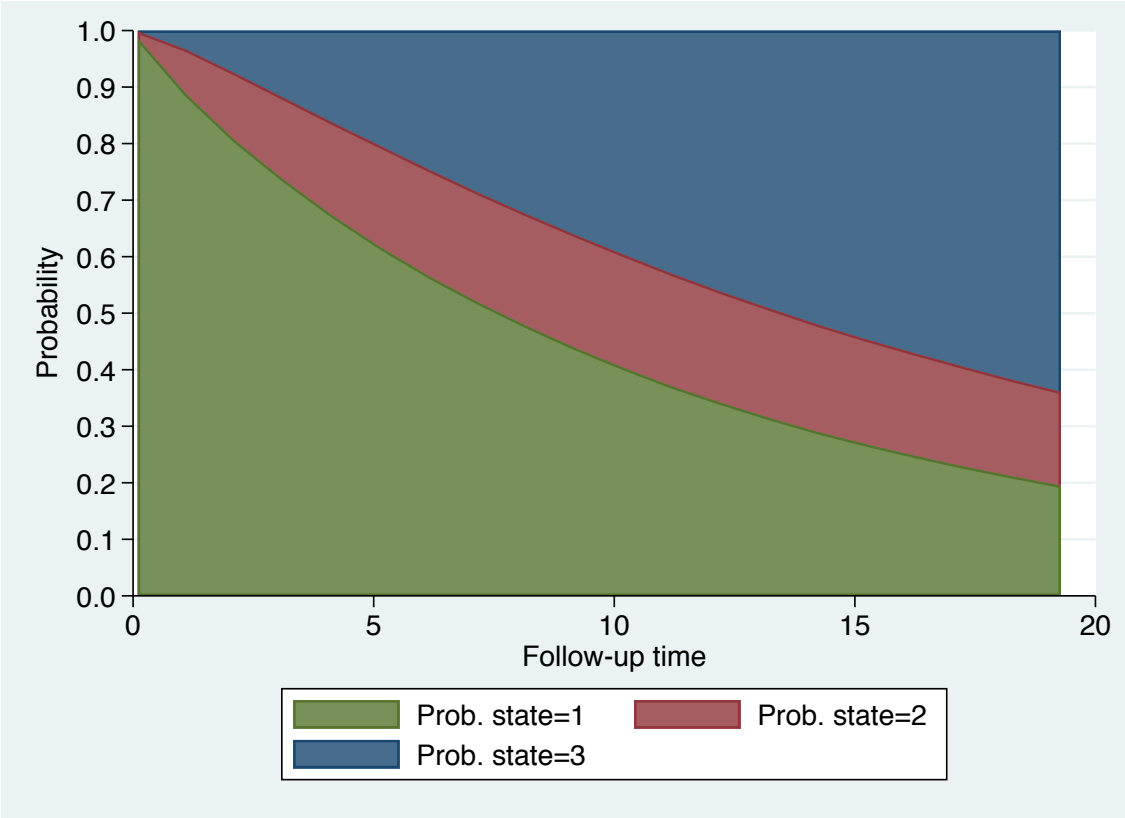
```
. predictms , transmat(tmat) at1(age 45)
```

```
. list _prob* _time in 1/10, noobs ab(15)
```

_prob_at1_1_1	_prob_at1_1_2	_prob_at1_1_3	_time
.98678	.01179	.00143	.09856263
.88871	.07766	.03363	1.1082532
.80736	.11835	.07429	2.1179437
.73707	.14444	.11849	3.1276343
.67506	.16351	.16143	4.1373248
.6189	.17816	.20294	5.1470154
.56723	.1882	.24457	6.1567059
.5207	.1943	.285	7.1663965
.47889	.19847	.32264	8.176087
.44077	.20048	.35875	9.1857776

predictms

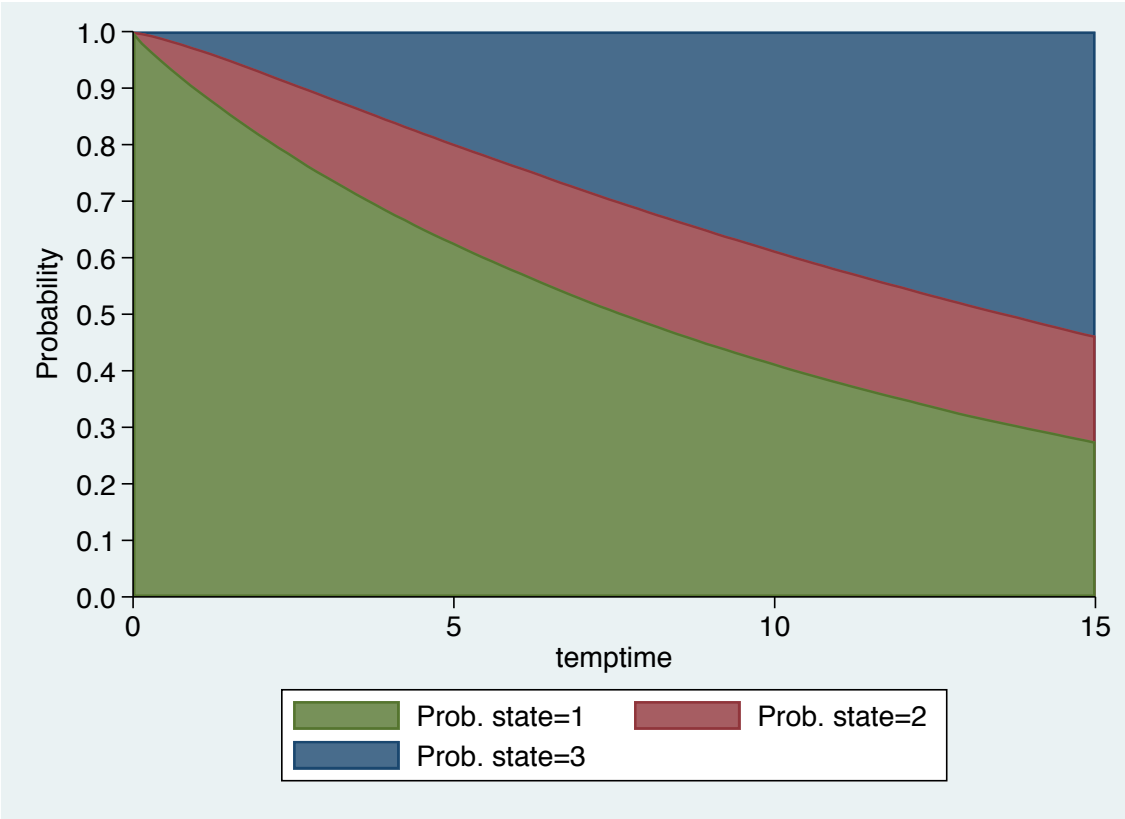
```
. predictms , transmat(tmat) at1(age 45) graph
```



We can tidy it up a bit...

predictms

```
. range temptime 0 15 100
(7,382 missing values generated)
. predictms , transmat(tmat) at1(age 45) graph timevar(temptime)
```



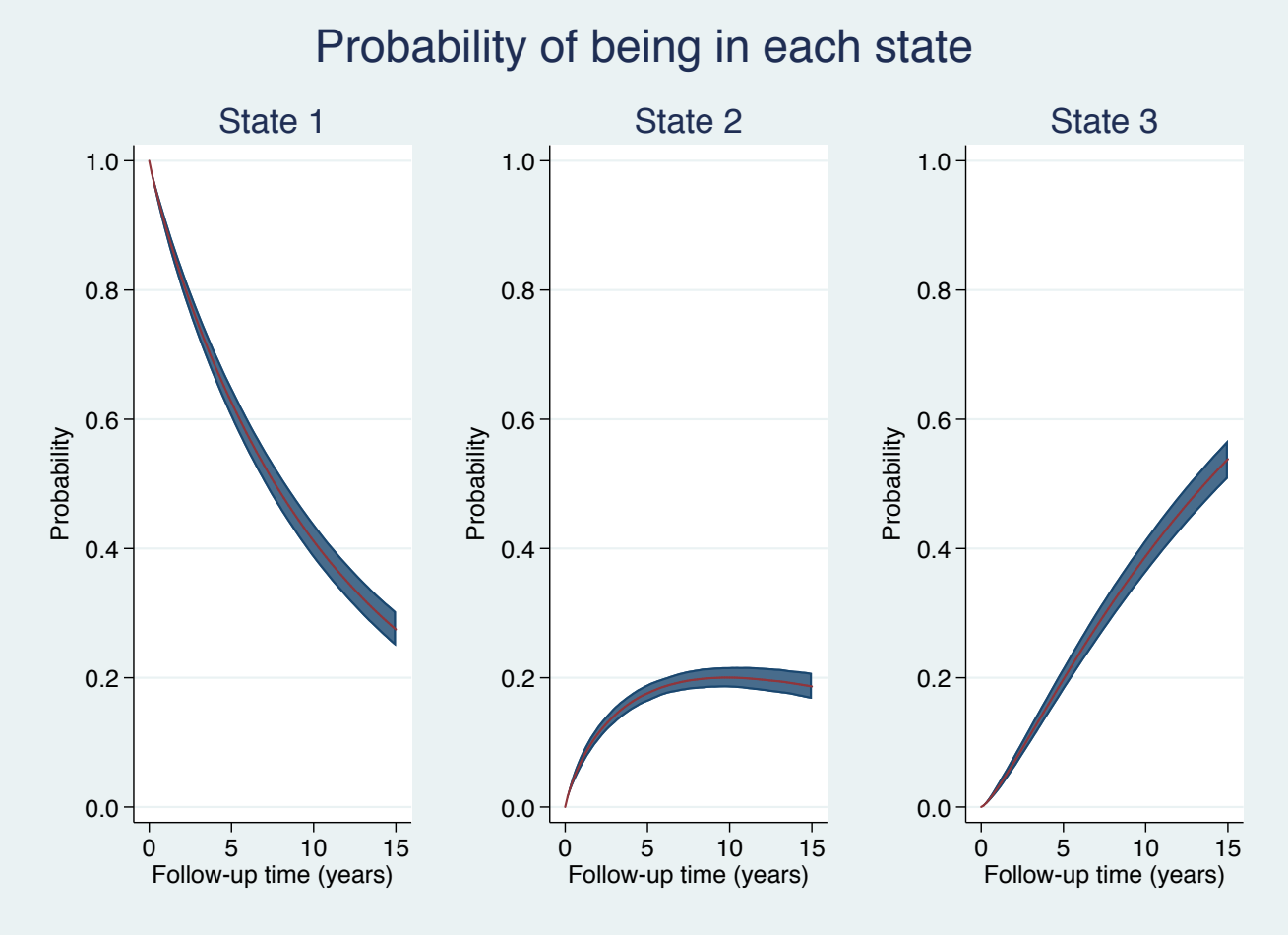
Uncertainty...

```
. predictms , transmat(tmat) at1(age 45) timevar(temptime) ci
. list _prob_at1_1_1* temptime in 1/10, noobs ab(15)
```

_prob_at1_1_1	_prob_at1~1_lci	_prob_at1~1_uci	temptime
1	1	1	0
.98098483	.97647768	.98464194	.1515152
.96469169	.95788723	.97043065	.3030303
.94927773	.94101558	.95643615	.4545455
.93442814	.92525291	.94254704	.6060606
.92019291	.91009883	.92924175	.7575758
.90642898	.89544508	.91636675	.9090909
.89311497	.88136687	.90382663	1.060606
.88018498	.86774232	.89160327	1.212121
.86739877	.85438362	.87941482	1.363636

Uncertainty...

```
. predictms , transmat(tmat) at1(age 45) timevar(temptime) ci
```



Getting predictions for multiple covariate patterns

```
. predictms , transmat(tmat) timevar(temptime) ///
>         at1(age 45) at2(age 80)
. list _prob_at1_1_3 _prob_at2_1_3 temptime in 1/10, noobs ab(15)
```

_prob_at1_1_3	_prob_at2_1_3	temptime
0	0	0
.00231	.00387	.1515152
.00577	.01048	.3030303
.01	.0192	.4545455
.01502	.02904	.6060606
.0203	.03961	.7575758
.02603	.05084	.9090909
.03143	.06292	1.060606
.03713	.07552	1.212121
.04324	.08852	1.363636

predictms

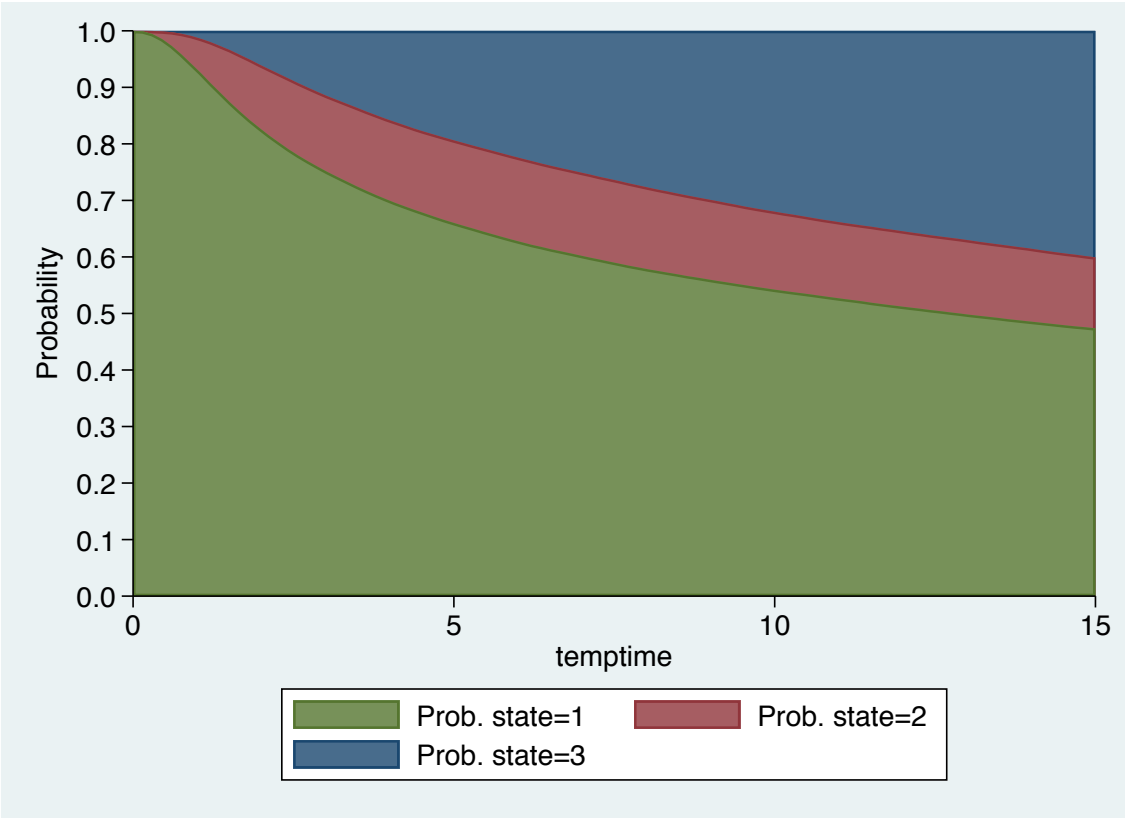
Now let's go back to our final models that we had before

Getting predictions from separate models

```
.
. qui stpm2 age sz2 sz3 nodes hormon pr_1 if _trans1==1, scale(h) df(3) ///
>                 tvc(sz2 sz3 pr_1) dftvc(1)
. estimates store m1
. qui streg age sz2 sz3 nodes hormon pr_1 if _trans2==1, distribution(weibull)
. estimates store m2
. qui stpm2 age sz2 sz3 nodes hormon pr_1 if _trans3==1, scale(h) df(3) ///
>                 tvc(pr_1) dftvc(1) nolog
. estimates store m3
```

Getting predictions from separate models

```
. predictms , transmat(tmat) at1(age 45) timevar(temptime) graph  ///
>                 models(m1 m2 m3)
```



predictms

- Everything available within `predictms` works on either the stacked or separate modelling format
- We tend to favour the separate modelling approach
- This gives us a very powerful tool to model each transition as simply or as complex as needed...yet still get easily interpreted probabilities (and more...) with a single line of code!

```
predictms, transmat(tmat) at(age 54 pr_1 3 sz2 1)
    models(m1 m2 m3)
```

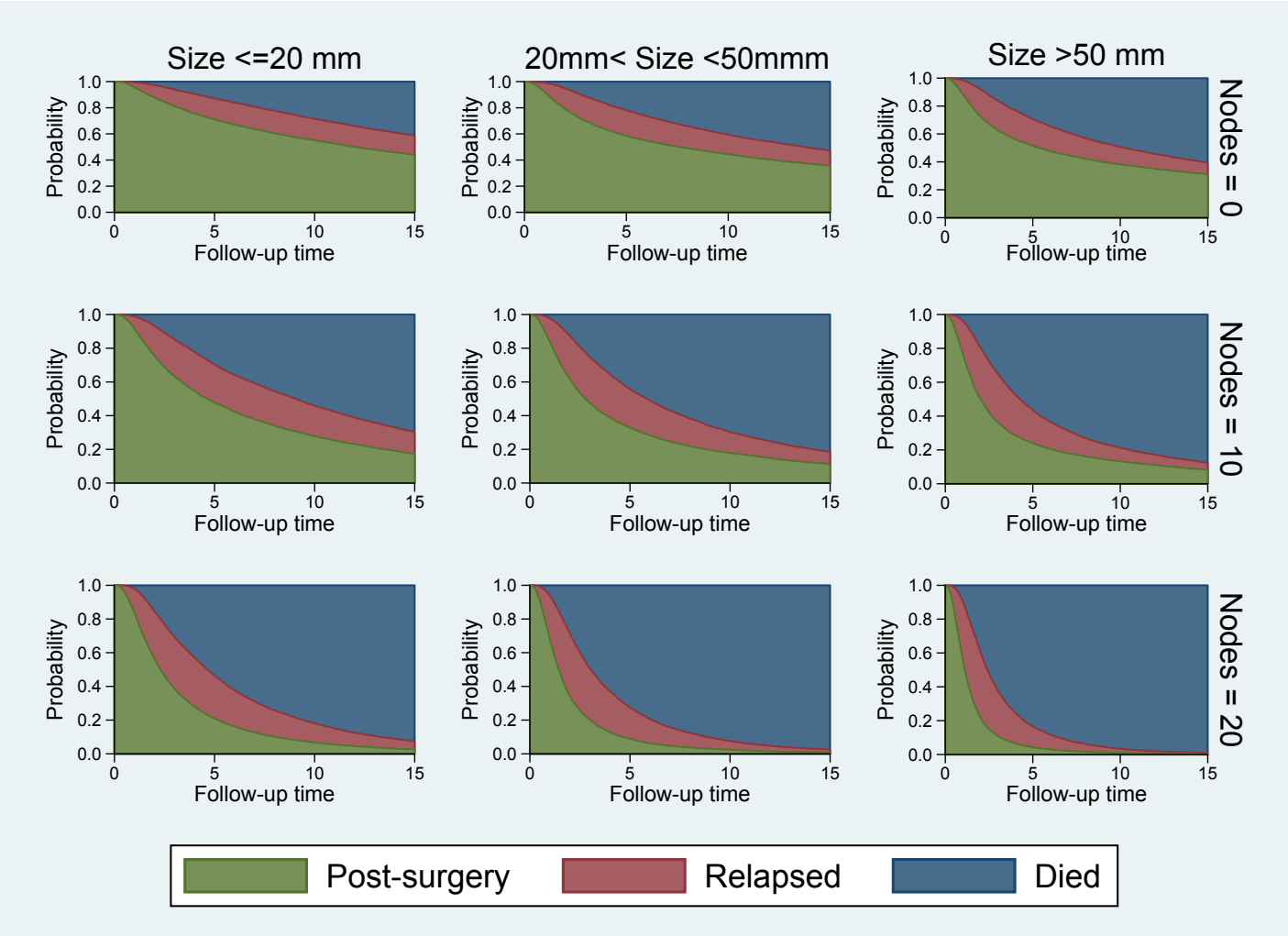


Figure 5: Probability of being in each state for a patient aged 54, with progesterone level (transformed scale) of 3.

```
predictms, transmat(tmat) at(age 54 pr_1 3 sz2 1)
    models(m1 m2 m3) ci
```

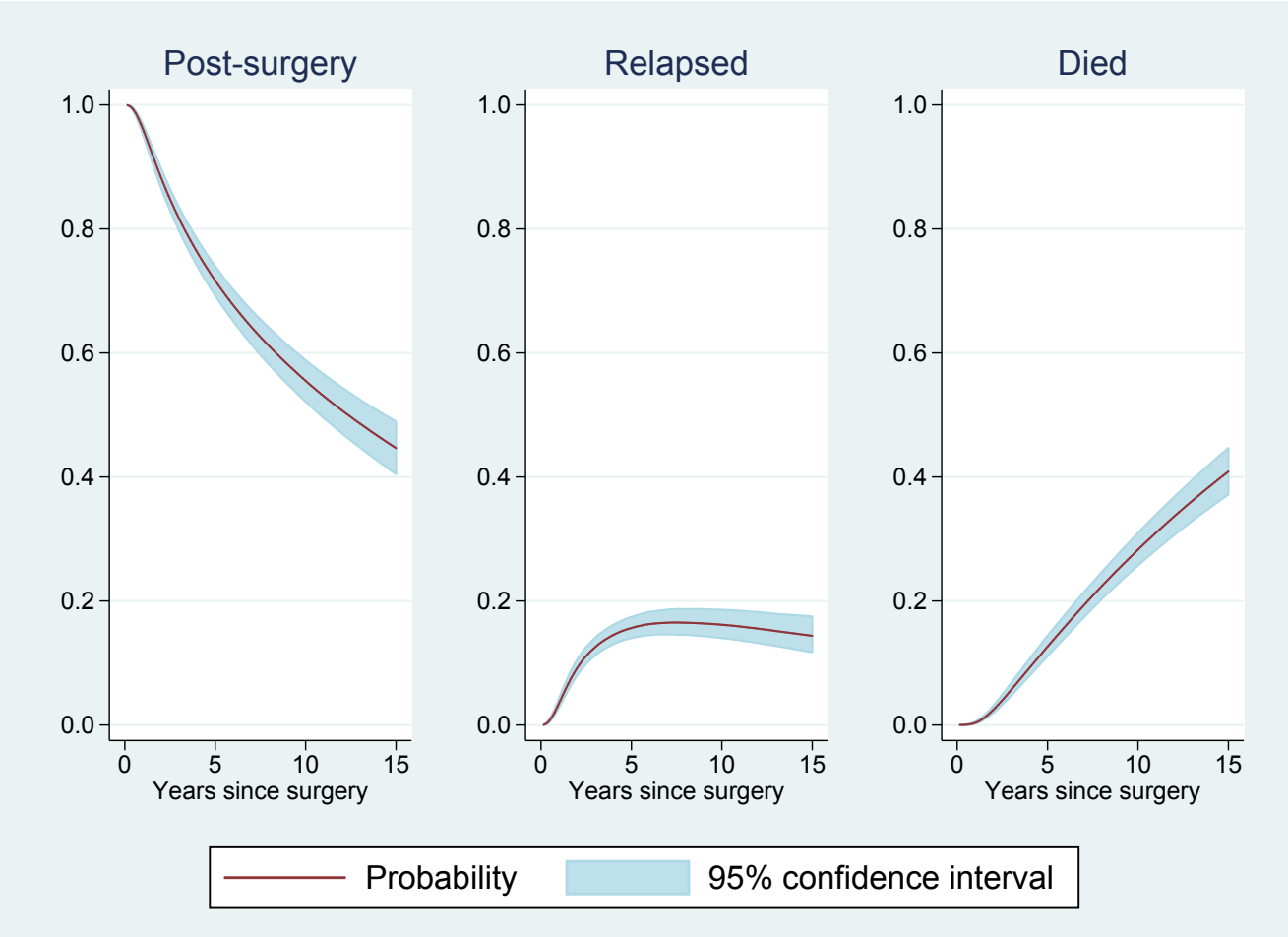


Figure 6: Probability of being in each state for a patient aged 54, 50$>$ size $\geq$20 mm, with progesterone level (transformed scale) of 3, and associated confidence intervals.

Contrasts

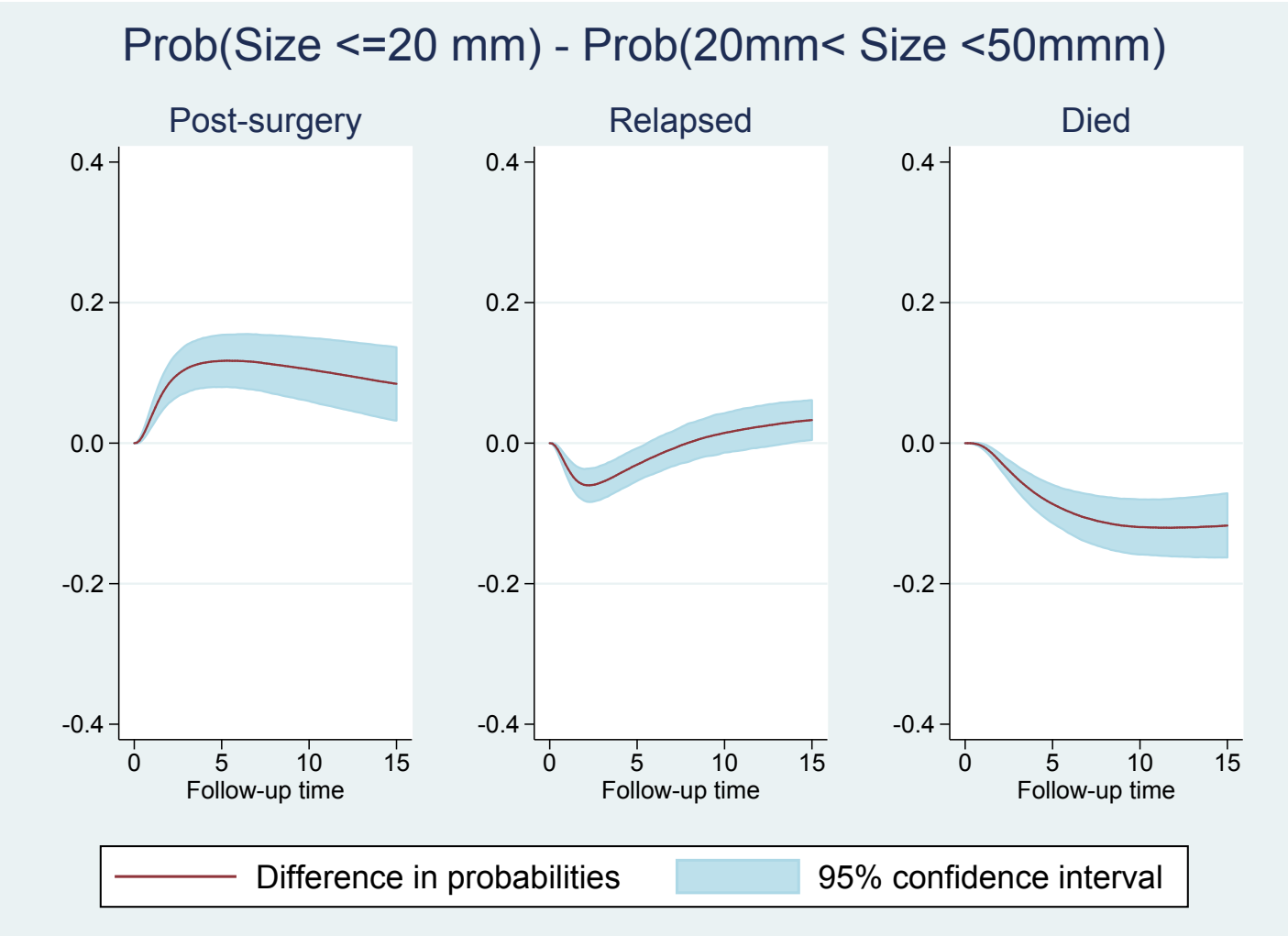
- It's easy to show predictions for a particular covariate pattern, but what about showing the impact of differences in covariate patterns?
- How does treatment change the probability if being in each state?
- How does tumour size at diagnosis influence these probabilities?
- We can use contrasts - differences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

Contrasts - `difference`

$$P(Y(t) = b | Y(s) = a, X = 1) - P(Y(t) = b | Y(s) = a, X = 0)$$

The difference in transition probabilities for $X = 1$ compared to $X = 0$

Differences in transition probabilities



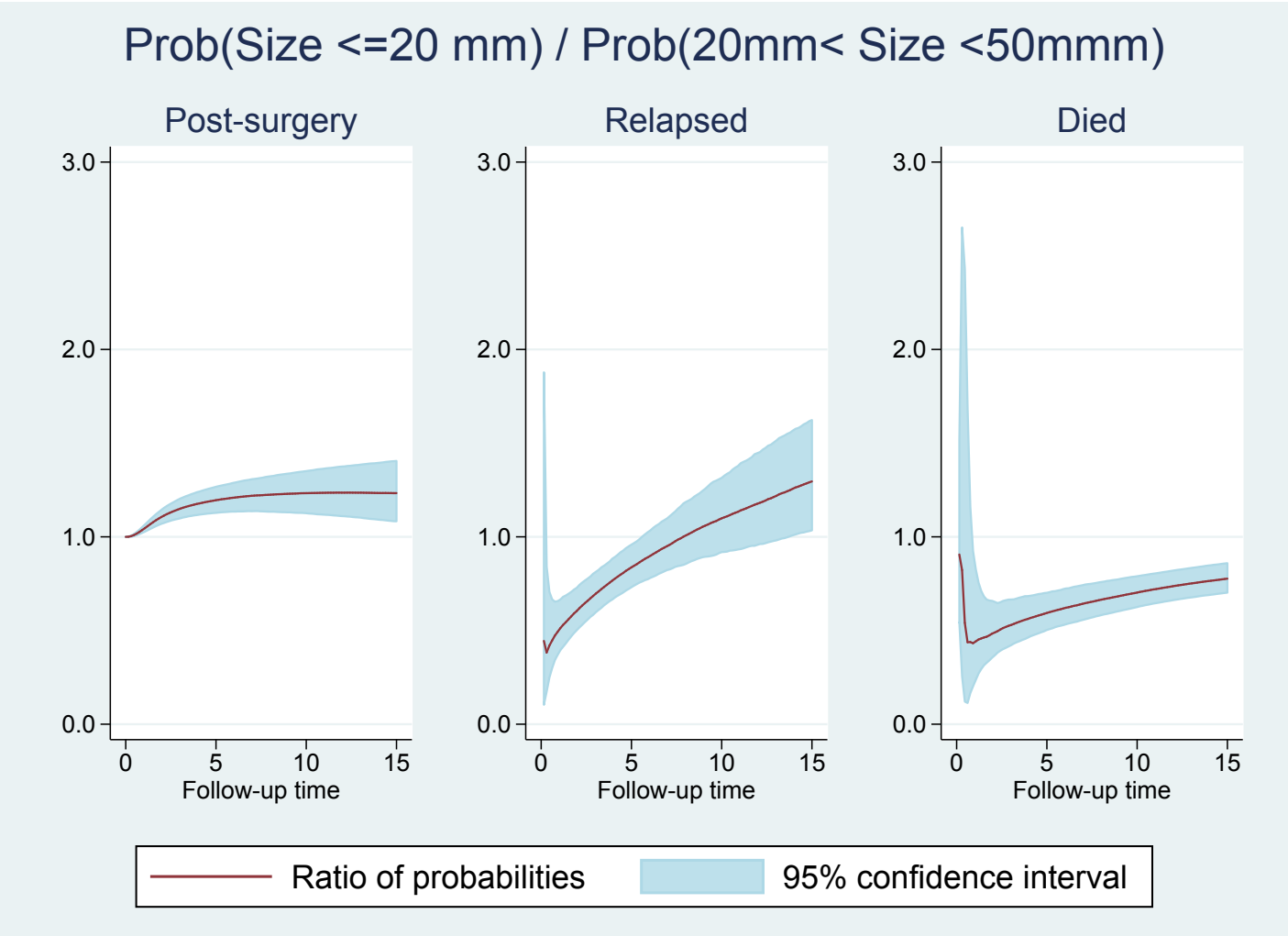

```
. predictms, transmat(tmat) models(m1 m2 m3) ///
.   at1(age 54 pgr 3 size1 1) at2(age 54 pgr 3 size2 1) difference ci
```

Contrasts - `ratio`

$$\frac{P(Y(t) = b | Y(s) = a, X = 1)}{P(Y(t) = b | Y(s) = a, X = 0)}$$

The ratio of transition probabilities for $X = 1$ compared to $X = 0$

Ratios of transition probabilities




```
. predictms, transmat(tmat) models(m1 m2 m3) ///
.   at1(age 54 pgr 3 size1 1) at2(age 54 pgr 3 size2 1) ci ratio
```

Contrasts

- `predictms` gives you the transition probabilities for each `at#()` pattern, in variables called `_prob_at#*`
- `predictms` gives you the `difference` between transition probabilities for each `at#()` pattern compared to the reference `atref(1)`, in variables called `_diff_prob_at#*`
- `predictms` gives you the `ratio` between transition probabilities for each `at#()` pattern compared to the reference `atref(1)`, in variables called `_ratio_prob_at#*`
- You can all these predictions in one call to `predictms`

```
. predictms, transmat(tmat) models(m1 m2 m3) ///
.   at1(age 54 pgr 3 size1 1) at2(age 54 pgr 3 size2 1)
.   difference ratio 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. This is the restricted mean survival equivalent in a multi-state model.

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

This is the multi-state equivalent of restricted mean survival time [11]

Length of stay in a state

Such a quantity allows us to ask questions such as

- How much time would you spend in hospital over a ten year period?
- How much time would you spend relapse-free?
- Does treatment influence the time spent in hospital?
- What is my life expectancy?

Thanks to the simulation approach, we can calculate such things extremely easily.

Example - breast cancer

In our breast cancer example, we may be interested in

- the amount of time a patient spends relapse-free
- how does tumour size influence length of stay?

Example - breast cancer

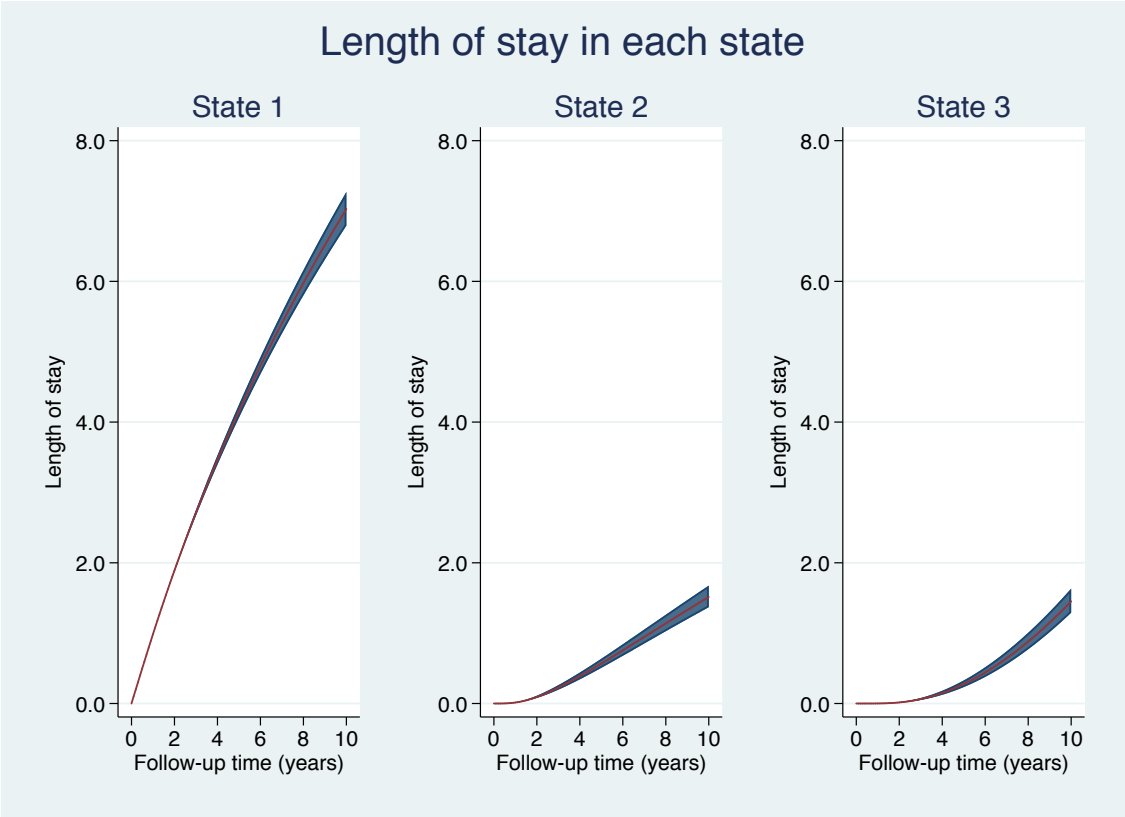
predictms

```
 . range temptime 0 10 101
(7,381 missing values generated)
. predictms , transmat(tmat) at1(age 45 pr_1 3 nodes 2) timevar(temptime)        ///
>                 models(m1 m2 m3) los
. list _los_at1_1_* temptime if _n==51 | _n==101, noobs ab(15)
```

_los_at1_1_1	_los_at1_1_2	_los_at1_1_3	temptime
4.157891	.56545628	.27665273	5
7.0421219	1.5039284	1.4539497	10

Example - breast cancer

```
. predictms , transmat(tmat) at1(age 45 pr_1 3 nodes 2) timevar(temptime)          ///
>                 models(m1 m2 m3) los ci
```



Example - breast cancer

So after 10 years, a patient aged 45 with progesterone of 3 and 2 positive nodes, spends

- 7 years alive and relapse-free
- 1.5 years alive post-relapse
- 1.5 years dead...does that make sense?

Length of stay should only be reported for **transient** states

Example - breast cancer

How about restricted mean survival? This is the total time spent in the initial state and the relapse state

```
. gen rmst = _los_at1_1_1 + _los_at1_1_2
(7,381 missing values generated)
. list _los_at1_1_1 _los_at1_1_2 rmst temptime if _n==51 | _n==101, noobs ab(15)
```

_los_at1_1_1	_los_at1_1_2	rmst	temptime
4.1537604	.56775277	4.721513	5
7.0281965	1.5145309	8.542727	10

What about confidence intervals?

predictms

We can use the `userfunction()` ability of `predictms`, which let's us pass our own function of transition probabilities and/or length of stays, to calculate bespoke predictions

predictms

userfunction()

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: real matrix ufunc(M)
> {
>         los1 = ms_user_los(M,1)
>         los2 = ms_user_los(M,2)
>         return(los1:+los2)
> }
: end
-------------------------------------------------------------------------------

.
. predictms , transmat(tmat) at1(age 45 pr_1 3 nodes 2) timevar(temptime)      ///
>                 models(m1 m2 m3) los ci userfunction(ufunc)
. list rmst _user_at1_1* temptime if _n==51 | _n==101, noobs ab(15)
```

rmst	_user_at1_1_1	_user_at1_1~lci	_user_at1_1~uci	temptime
4.721513	4.7231721	4.6753368	4.7710075	5
8.542727	8.5454766	8.3664569	8.7244962	10

Example - breast cancer

All of our contrasts are available as well, so we can easily assess the impact of covariates, through differences,

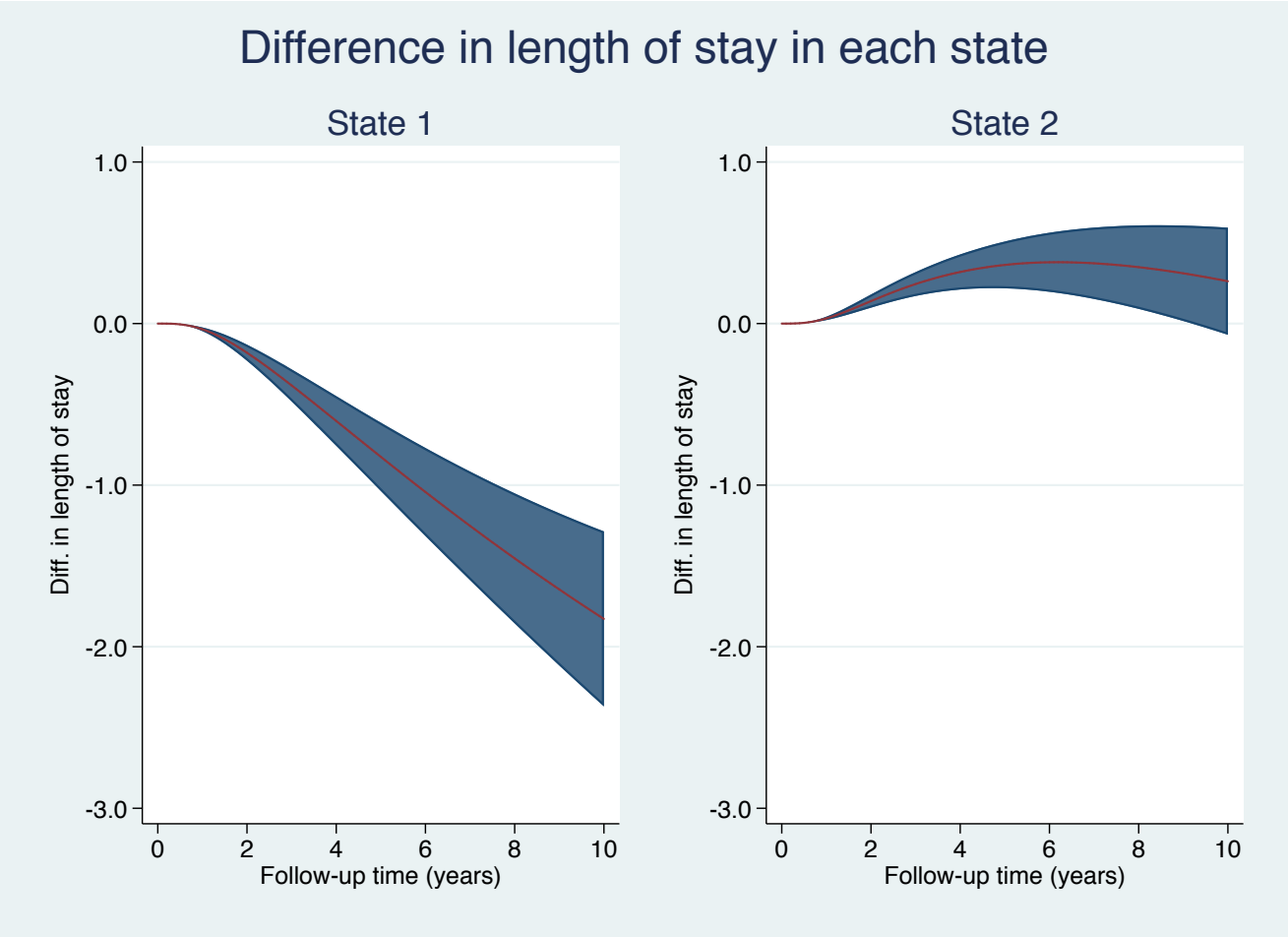
$$LoS(t|X = 1) - LoS(t|X = 0)$$

or ratios,

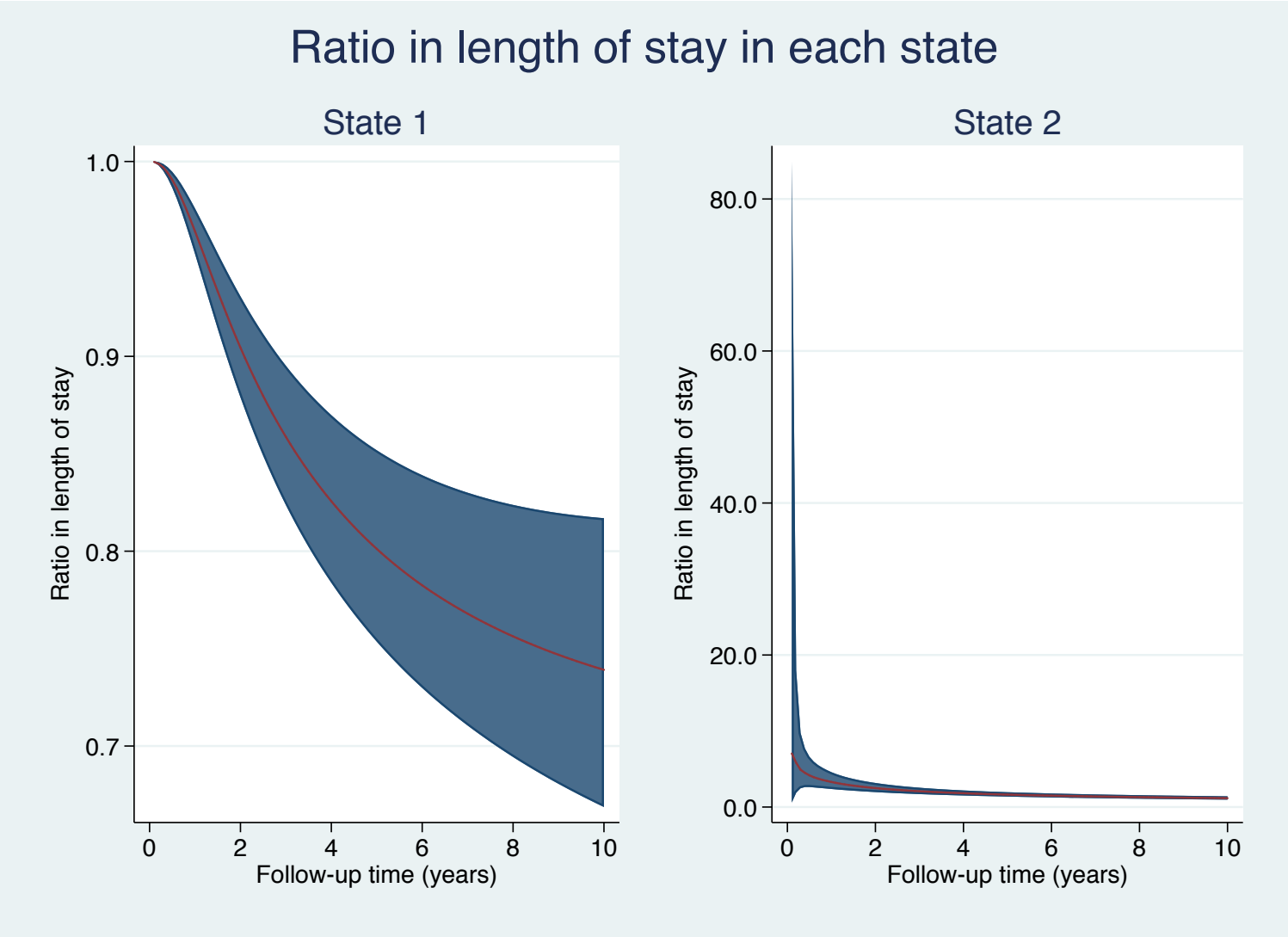
$$\frac{LoS(t|X = 1)}{LoS(t|X = 0)}$$

Example - breast cancer

```
. predictms , transmat(tmat) at1(age 45 pr_1 3 nodes 2) timevar(temptime)          ///
>                 at2(age 45 pr_1 3 nodes 2 sz3 1) models(m1 m2 m3) los ci
>                 difference ratio
```



Example - breast cancer



Markov models - reminder

- All the multistate models we have discussed so far have been Markov models
- Remember, this means that where you are going is not influenced by where you have been
- We can relax this assumption in a number of ways

Semi-Markov multi-state models I

- The Markov assumption can be considered restrictive
- We can relax it by allowing the transition intensities to depend on the time at which earlier states were entered - multiple timescales [10]
- This is commonly simplified further, by defining the transition hazards/intensities to be dependent only on the time spent in the current state - clock-reset approach [4]

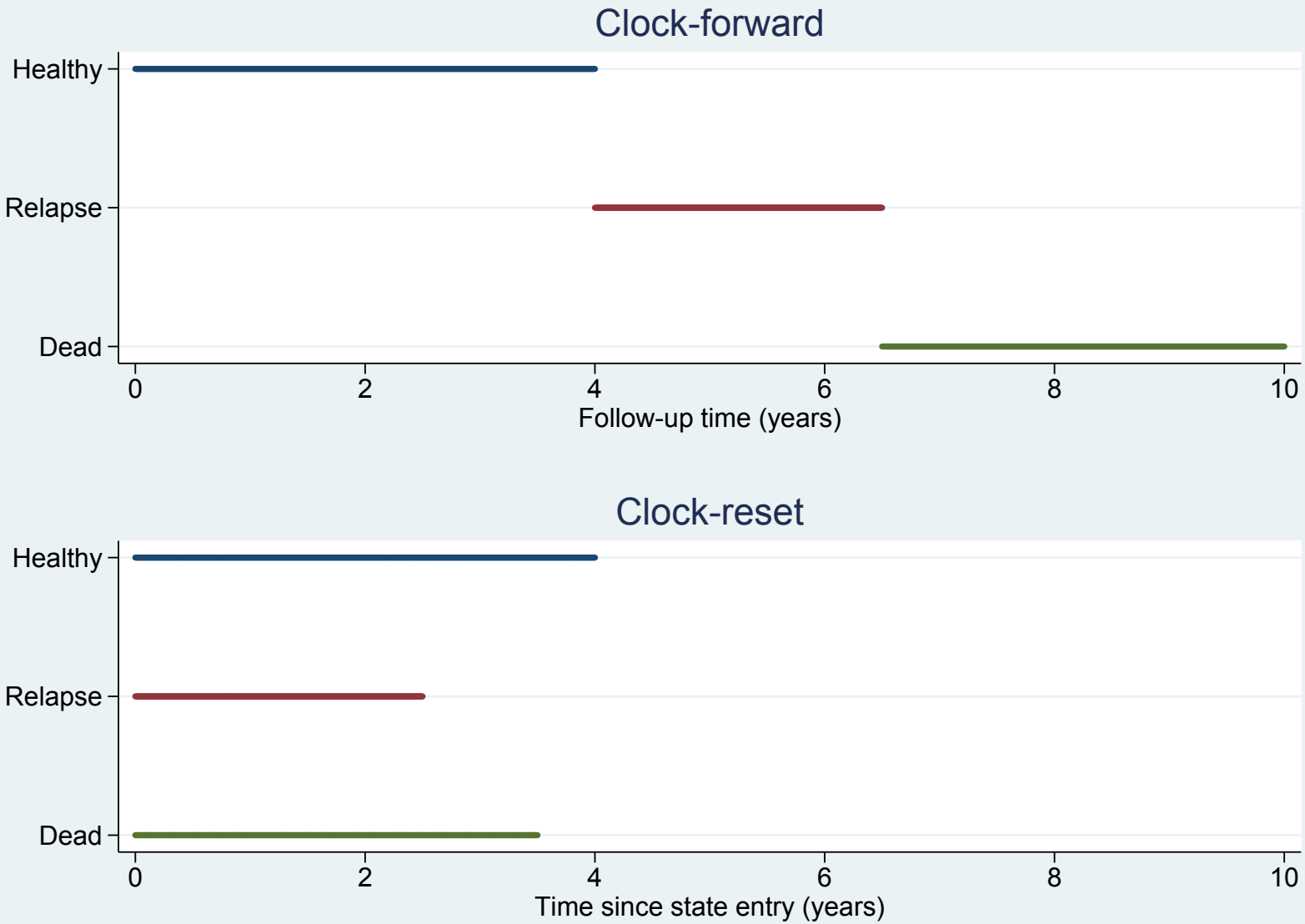


Figure 7: The impact of timescale.

Clock reset approach

- If the Markov assumption does not hold we may consider the clock-reset approach
- The transition from relapse to death may be a function of time since entry into the relapse state
- Timescale is set to zero after each new state entry

Clock reset approach - estimation

- Just as easy as the clock forward approach

```
. gen _newt = _stop - _start
. stset _newt , failure(_status=1)
```

- Before we had

```
. stset _stop , enter(_start) failure(_status=1)
```

- Given we've `stset` our data, we can now fit any models we like!

`predictms` with clock-reset models

- We've seen that the only thing you have to change is how you `stset` your data
- It's equally simple to use `predictms` after fitting a clock-reset model
- Add the `reset` option...yes that's it!

A clock reset model

predictms and reset

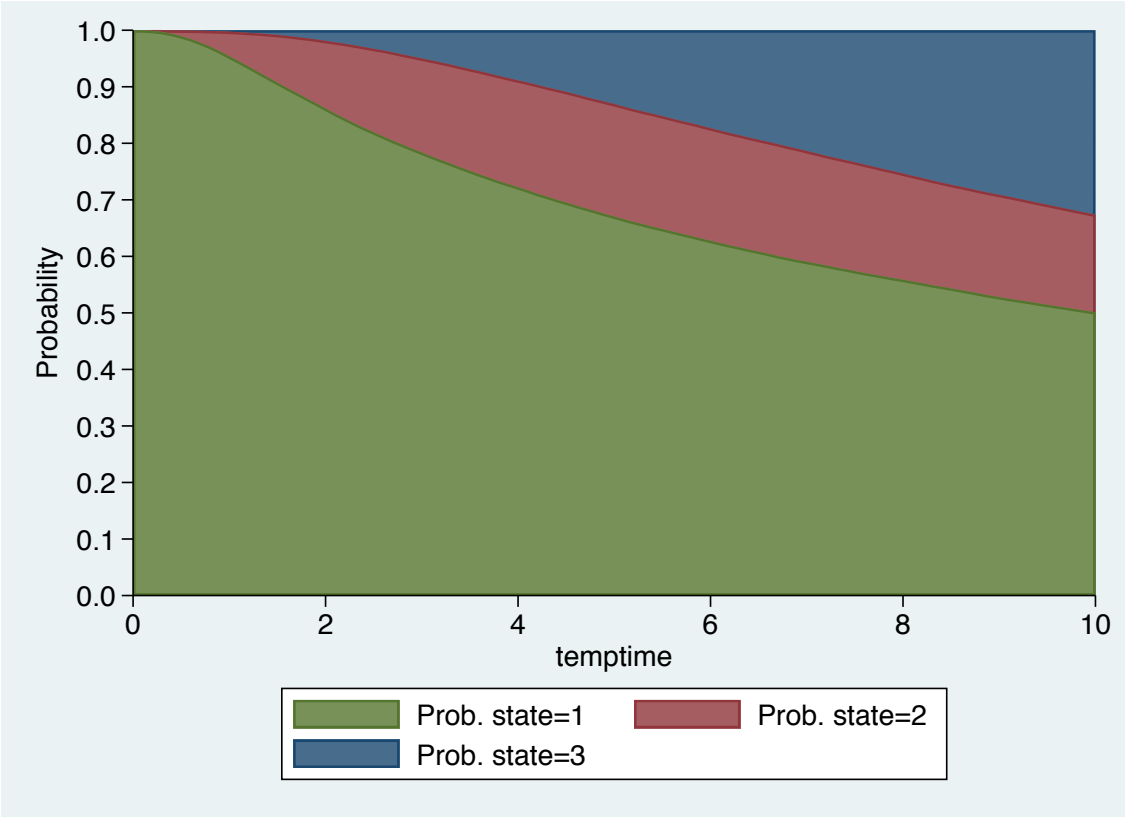
```
 . range temptime 0 10 101
(7,381 missing values generated)
. predictms , transmat(tmat) at1(age 45 pr_1 3 nodes 2) timevar(temptime)        ///
>                 models(m1 m2 m3) reset
. list _prob_at1_1_* temptime if _n==51 | _n==101, noobs ab(15)
```

_prob_at1_1_1	_prob_at1_1_2	_prob_at1_1_3	temptime
.66881	.19877	.13242	5
.49783	.17259	.32958	10

A clock reset model

predictms and reset

```
. predictms , transmat(tmat) at1(age 45 pr_1 3 nodes 2) timevar(temptime)         ///
>                 models(m1 m2 m3) reset graph
```



Choice of timescale

- It's setting specific
- Clock reset models would generally be more appropriate when an intermediate event is 'substantial', for example a heart attack
- A useful property of state occupation probabilities is that they are robust to deviations of the Markov assumption

Current and future plans

- The `multistate` package is actively being developed
- Some future projects will include:
  - Reversible transitions
    - There's no restriction on the transition matrix
  - Frailties for clustered data
    - I've begun syncing `predictms` with `merlin`
    - Find out more on mjcrowther.co.uk/software/merlin
  - Multiple timescales
    - Fitting survival models with multiple timescales is challenging
    - `merlin` can do this simply and flexibly, e.g.:

`merlin`

```
merlin (stime                                  /// response
               trt sex                         /// baseline covariates
               trt#rcs(stime, df(3))           /// complex time-dependent effect
               rcs(stime, df(2) offset(age))   /// second timescale
       , family(rp, failure(died) df(5))       /// survival model
       )
```

References

[1] Crowther MJ, Lambert PC. Parametric multistate 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.

[2] Asaria M, Walker S, Palmer S, Gale CP, Shah AD, Abrams KR, *et al.*. Using electronic health records to predict costs and outcomes in stable coronary artery disease. *Heart* 2016;**102**:755–762.

[3] Sauerbrei W, Royston P, Look M. A new proposal for multivariable modelling of time-varying effects in survival data based on fractional polynomial time-transformation. *Biometrical Journal* 2007;**49**:453–473.

[4] Putter H, Fiocco M, Geskus RB. Tutorial in biostatistics: competing risks and multi-state models. *Stat Med* 2007;**26**:2389–2430.

[5] Andersen PK, Perme MP. Inference for outcome probabilities in multi-state models. *Lifetime data analysis* 2008;**14**:405.

References 2

[6] Jackson CH. Multi-state models for panel data: the msm package for r. *Journal of Statistical Software* 2011;**38**:1–29.

[7] Hsieh HJ, Chen THH, Chang SH. Assessing chronic disease progression using non-homogeneous exponential regression markov models: an illustration using a selective breast cancer screening in taiwan. *Statistics in medicine* 2002;**21**:3369–3382.

[8] Hinchliffe SR, Abrams KR, Lambert PC. The impact of under and over-recording of cancer on death certificates in a competing risks analysis: a simulation study. *Cancer Epidemiol* 2013;**37**:11–19.

[9] Titman AC. Flexible nonhomogeneous markov models for panel observed data. *Biometrics* 2011;**67**:780–787.

[10] Iacobelli S, Carstensen B. Multiple time scales in multi-state models. *Statistics in Medicine* 2013;**32**:5315–5327.

References 3

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

[12] Jackson C. flexsurv: A platform for parametric survival modeling in r. *Journal of Statistical Software* 2016;**70**:1–33.

[13] Crowther MJ, Lambert PC. Simulating biologically plausible complex survival data. *Stat Med* 2013;**32**:4118–4134.

[14] Sjölander A. Regression standardization with the r package stdreg. *European Journal of Epidemiology* 2016;**31**:563–574.

[15] Gran JM, Lie SA, Ã˜yeflaten I, Borgan Ã, Aalen OO. Causal inference in multi-state models-sickness absence and work for 1145 participants after work rehabilitation. *BMC Public Health* 2015;**15**:1082.