# Multi-state survival analysis in Stata

## Stata UK Meeting
## 8th-9th September 2016

Michael J. Crowther and Paul C. Lambert

Department of Health Sciences
University of Leicester
and
Department of Medical Epidemiology and Biostatistics
Karolinska Institutet
michael.crowther@le.ac.uk
paul.lambert@le.ac.uk

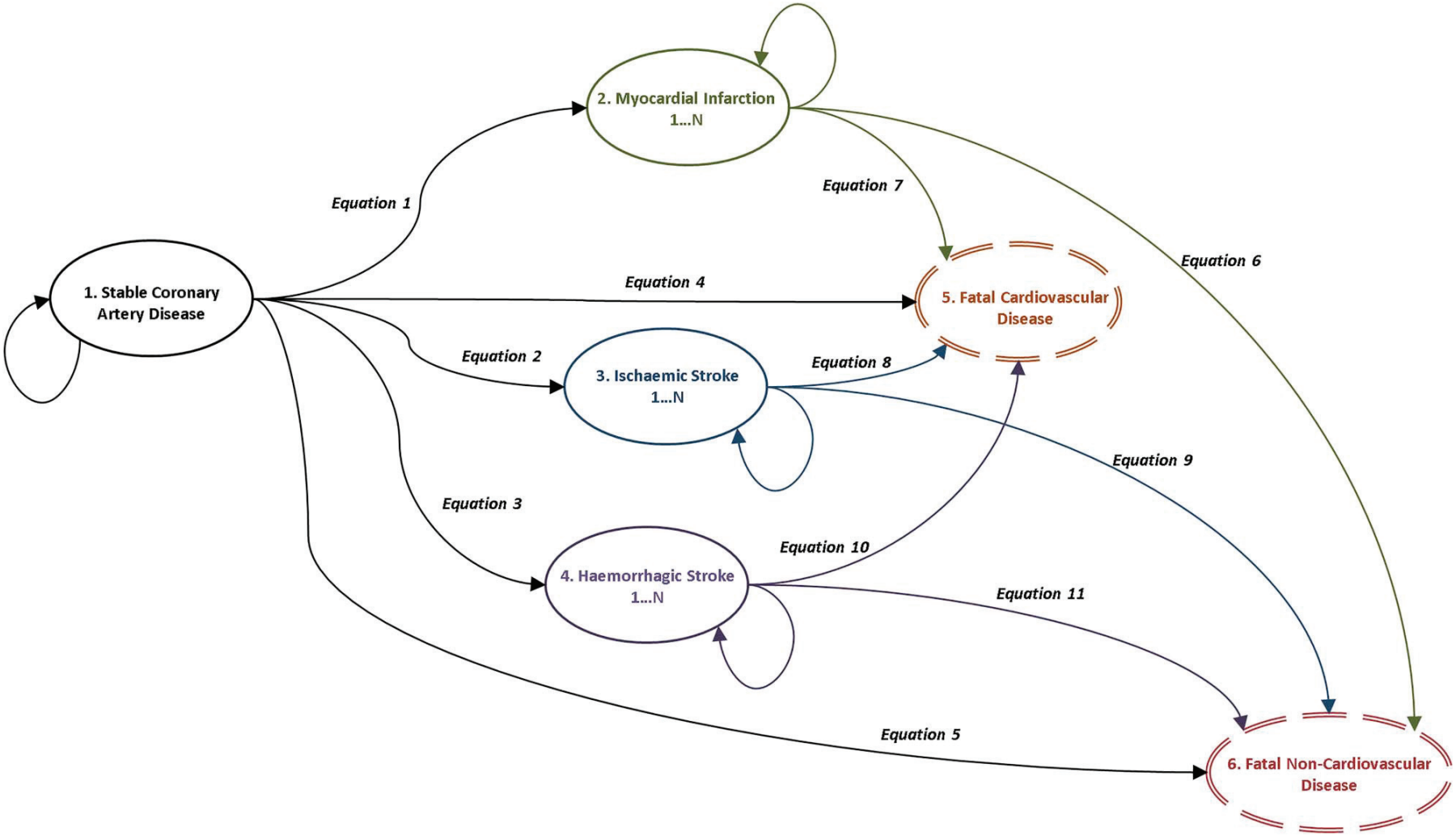
# Plan

- Background
- Primary breast cancer example
- Multi-state survival models
  - Common approaches
  - Some extensions
  - Clinically useful measures of absolute risk
- New Stata `multistate` package
- Future research

# 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

# An example from stable coronary diseaseAsaria et al. (2016)



# Primary breast cancer (Sauerbrei et al., 2007)

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

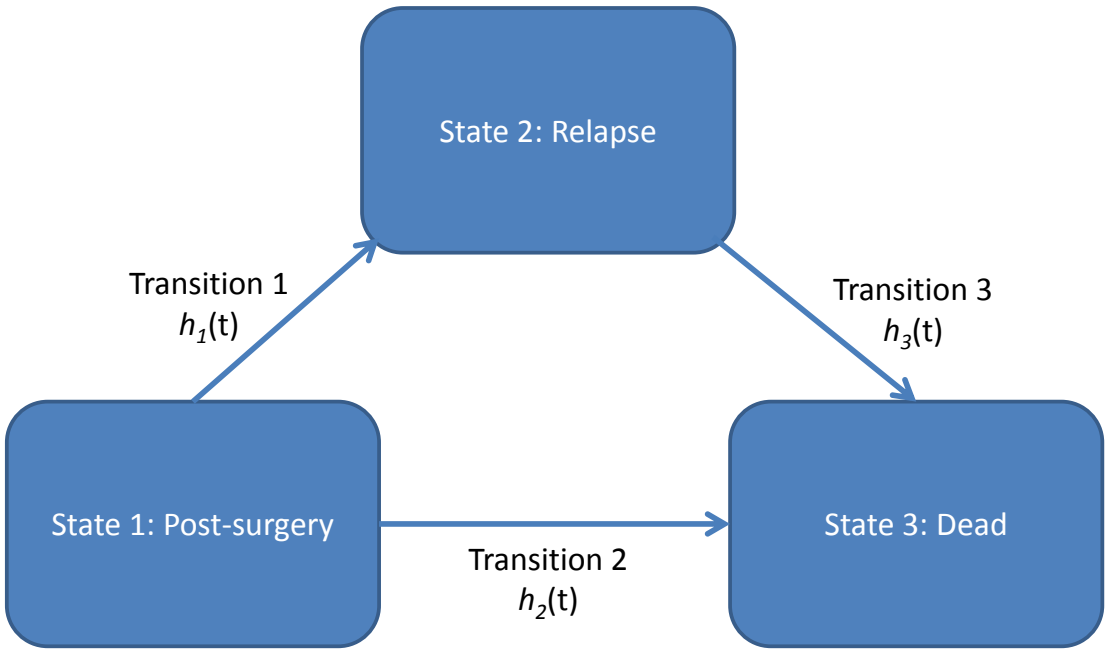



Figure: Illness-death model for primary breast cancer example.

# 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.

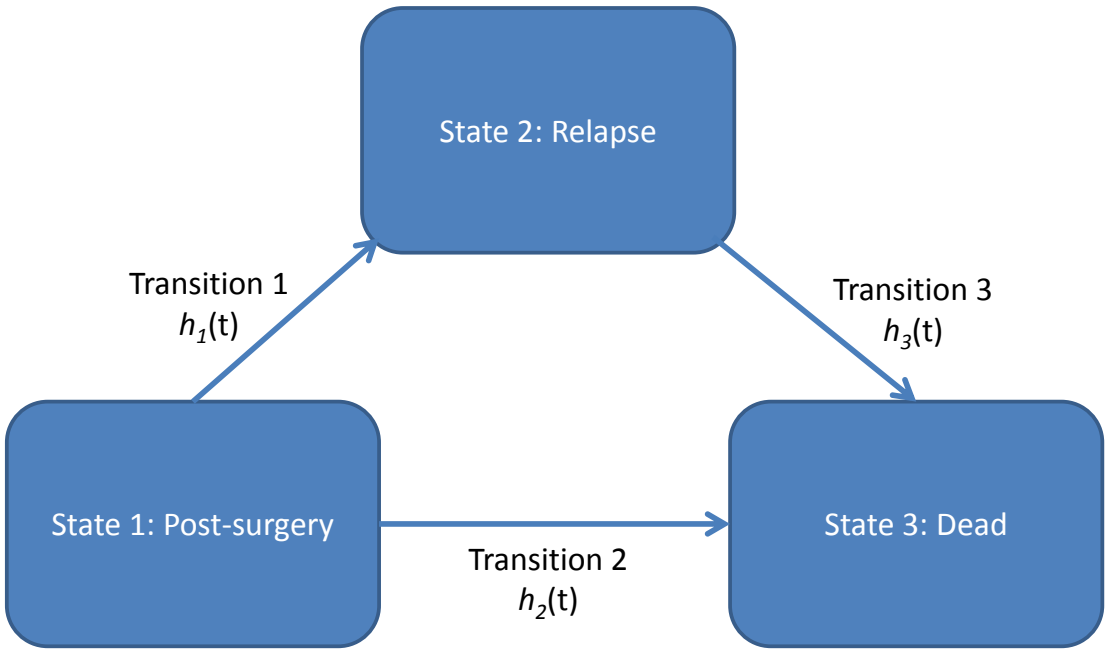



Figure: Illness-death model for primary breast cancer example.

# Markov multi-state models

The transition intensity is then defined as, For the $k$th transition from state $a_k$ to state $b_k$, the transition intensity (hazard function) is

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

which represents the transition rate from state $a_k$ to state $b_k$ at time $t$. Our collection of transitions intensities (hazard rates) governs the multi-state model.

# Estimating a multi-state models

- 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

```
. list pid rf rfi os osi if pid==1 | pid==1371, sepby(pid) noobs
```

| pid | rf | rfi | os | osi |
|---|---|---|---|---|
| 1 | 59.1 | 0 | 59.1 | alive |
| 1371 | 16.6 | 1 | 24.3 | deceased |

Time is recorded in months.

# 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 covar |

**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) |

```
. list pid rf rfi os osi if pid==1 | pid==1371, sepby(pid) noobs
```

| pid | rf | rfi | os | osi |
|---|---|---|---|---|
| 1 | 59.1 | 0 | 59.1 | alive |
| 1371 | 16.6 | 1 | 24.3 | deceased |

```
. list pid rf rfi os osi if pid==1 | pid==1371, sepby(pid) noobs
```

| pid | rf | rfi | os | osi |
|---|---|---|---|---|
| 1 | 59.1 | 0 | 59.1 | alive |
| 1371 | 16.6 | 1 | 24.3 | deceased |

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

```
. list pid rf rfi os osi if pid==1 | pid==1371, sepby(pid) noobs
```

| pid | rf | rfi | os | osi |
|---|---|---|---|---|
| 1 | 59.1 | 0 | 59.1 | alive |
| 1371 | 16.6 | 1 | 24.3 | deceased |

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

```
. list pid rf rfi os osi if pid==1 | pid==1371, sepby(pid) noobs
```

| pid | rf | rfi | os | osi |
|---|---|---|---|---|
| 1 | 59.1 | 0 | 59.1 | alive |
| 1371 | 16.6 | 1 | 24.3 | deceased |

```
. msset, id(pid) states(rfi osi) times(rf os) covariates(age)
  variables age_trans1 to age_trans3 created
. matrix tmat = r(transmatrix)
. list pid _start _stop _from _to _status _trans if pid==1 | pid==1371
```

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

```
. list pid rf rfi os osi if pid==1 | pid==1371, sepby(pid) noobs
```

| pid | rf | rfi | os | osi |
|---|---|---|---|---|
| 1 | 59.1 | 0 | 59.1 | alive |
| 1371 | 16.6 | 1 | 24.3 | deceased |

```
. msset, id(pid) states(rfi osi) times(rf os) covariates(age)
  variables age_trans1 to age_trans3 created
. matrix tmat = r(transmatrix)
. list pid _start _stop _from _to _status _trans if pid==1 | pid==1371
```

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

```
.
```

- 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

- 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

# Calculating transition probabilities

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

There are a variety of approaches

- Exponential distribution is convenient (Jackson, 2011)
- Numerical integration (Hsieh et al., 2002; Hinchliffe et al., 2013)
- Ordinary differential equations (Titman, 2011)
- Simulation (Iacobelli and Carstensen, 2013; Touraine et al., 2013; Jackson, 2016)

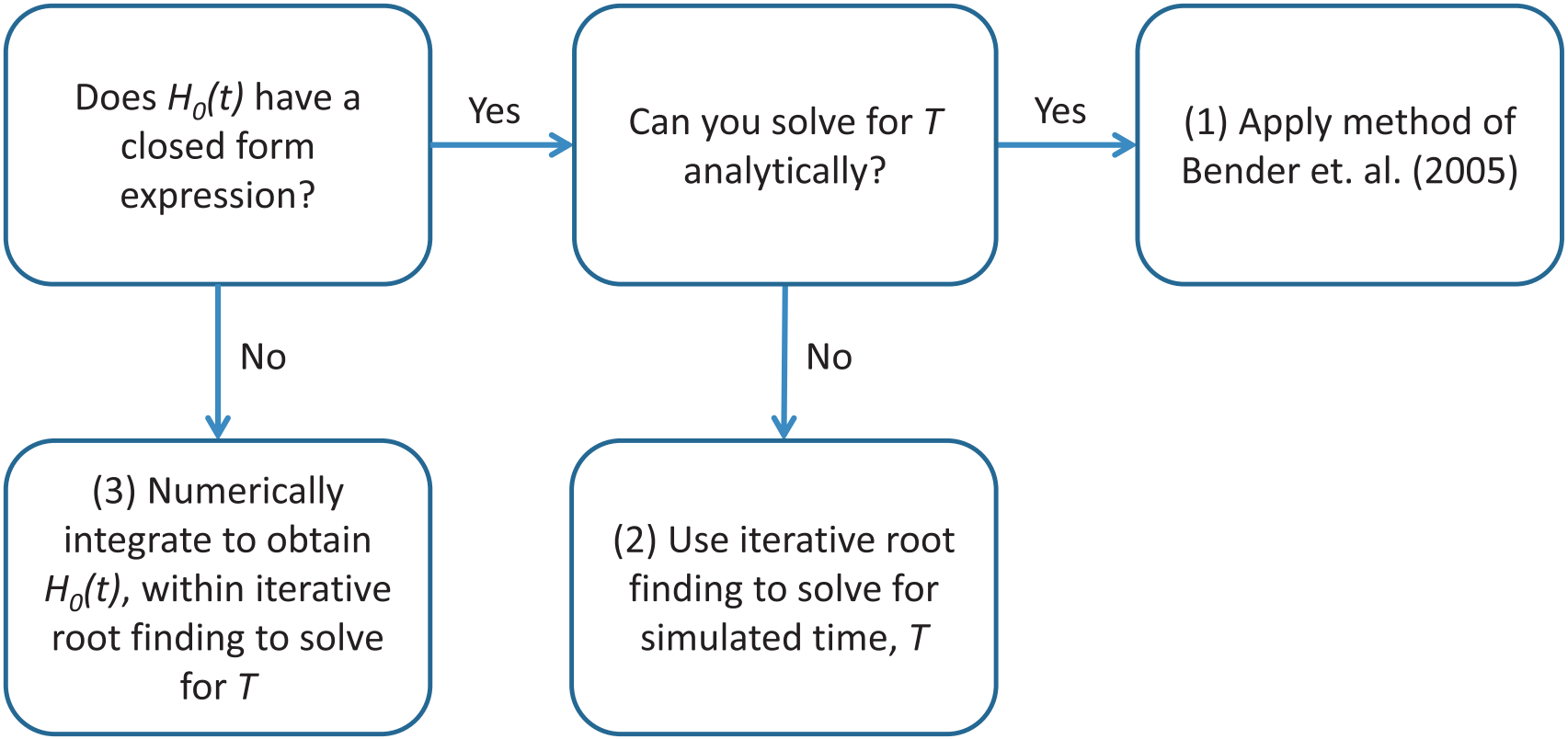
# Simulation

After fitting our model we can estimate the transition intensity (hazard rate) for all transitions.

1. Define a large sample of $N$ subjects (e.g. 100,000) and simulate through different states.
2. The model is a series of competing risk scenarios.
3. Continue until all patients in an absorbing state (or maximum follow-up time is reached).
4. At specified time points, we simply count how many people are in each state, and divide by the total to get our transition probabilities.
5. Other summaries e.g. mean time in each state.
6. Confidence intervals obtained by sampling, from MVN distribution, with mean vector, $\beta$, and variance-covariance matrix, $\boldsymbol{V}$, and repeated $M$ times.
7. Applicable to both Markov and non-Markov models.

# Can simulate from complex survival functions

We have shown how it is possible to simulate from complex survival distributions(Crowther and Lambert, 2013). See `survsim` command.



# Proportional baseline, transition specific age effect

```
. streg age_trans1 age_trans2 age_trans3 _trans2 _trans3, dist(weibull)
Weibull regression -- log relative-hazard form
No. of subjects =        7,482                 Number of obs    =        7,482
No. of failures =        2,790
Time at risk    =  38474.53852
                                               LR chi2(5)       =      3057.11
Log likelihood  =   -5547.7893                 Prob > chi2      =       0.0000
```

| _t | Haz. Ratio | Std. Err. | z | P>\|z\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| age_trans1 | .9977633 | .0020646 | -1.08 | 0.279 | .993725 | 1.001818 |
| age_trans2 | 1.127599 | .0084241 | 16.07 | 0.000 | 1.111208 | 1.144231 |
| age_trans3 | 1.007975 | .0023694 | 3.38 | 0.001 | 1.003342 | 1.01263 |
| _trans2 | .0000569 | .000031 | -17.95 | 0.000 | .0000196 | .0001653 |
| _trans3 | 1.85405 | .325532 | 3.52 | 0.000 | 1.314221 | 2.615619 |
| _cons | .1236137 | .0149401 | -17.30 | 0.000 | .0975415 | .1566547 |
| /ln_p | -.1156762 | .0196771 | -5.88 | 0.000 | -.1542426 | -.0771098 |
| p | .8907636 | .0175276 | | | .8570641 | .9257882 |
| 1/p | 1.122632 | .0220901 | | | 1.080161 | 1.166774 |

# predictms

```
. predictms, transmat(tmat) at(age 50)
```

# predictms

```
. predictms, transmat(tmat) at(age 50) graph
```

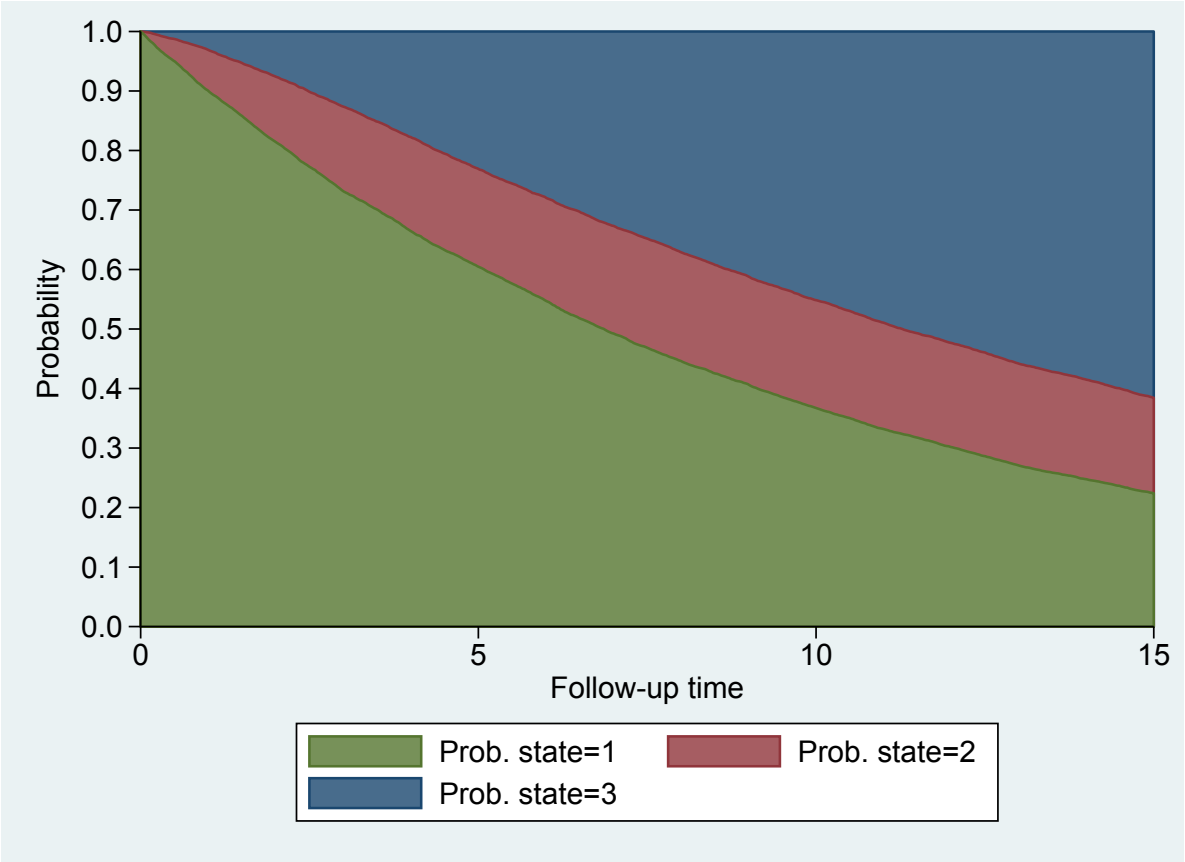



Figure: Predicted transition probabilities.

# Extending multi-state models

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

// Is equivalent to...

. streg age if _trans==1, dist(weibull)
. est store m1

. streg age if _trans==2, dist(weibull)

. est store m2

. streg age if _trans==3, dist(weibull)

. est store m3
```

# Extending multi-state models

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

// Is equivalent to...

. streg age if _trans==1, dist(weibull)
. est store m1

. streg age if _trans==2, dist(weibull)

. est store m2

. streg age if _trans==3, dist(weibull)

. est store m3


//Predict transition probabilities

. predictms, transmat(tmat) models(m1 m2 m3) at(age 50)
```

Separate models...we can now use *different* distributions

# Building our model

Returning to the breast cancer dataset

- Choose the best fitting parametric survival model, using AIC and BIC
- We find that the best fitting model for transitions 1 and 3 is the Royston-Parmar model with 3 degrees of freedom, and the Weibull model for transition 2.
- Adjust for important covariates; age, tumour size, number of nodes, progesterone level
- Check proportional hazards assumption

# Final model

- Transition 1: Royston-Parmar baseline with df=3, age, tumour size, number of positive nodes, hormonal therapy. Non-PH in tumour size (both levels) and progesterone level, modelled with interaction with log time.
- Transition 2: Weibull baseline, age, tumour size, number of positive nodes, hormonal therapy.
- Transition 3: Royston-Parmar with df=3, age, tumour size, number of positive nodes, hormonal therapy. Non-PH found in progesterone level, modelled with interaction with log time.

# Final model

- Transition 1: Royston-Parmar baseline with df=3, age, tumour size, number of positive nodes, hormonal therapy. Non-PH in tumour size (both levels) and progesterone level, modelled with interaction with log time.
- Transition 2: Weibull baseline, age, tumour size, number of positive nodes, hormonal therapy.
- Transition 3: Royston-Parmar with df=3, age, tumour size, number of positive nodes, hormonal therapy. Non-PH found in progesterone level, modelled with interaction with log time.

# Three separate models

```
. stpm2 age sz2 sz3 enodes pr_1 if _trans==1, ///
    scale(hazard)  df(3) tvc(sz2 sz3 pr_1) dftvc(1)
. estimates store m1

. streg age sz2 sz3 enodes pr_1 hormon if _trans==2, dist(weibull)
. estimates store m2

. stpm2 age sz2 sz3 enodes pr_1 if _trans==3, ///
    scale(hazard) df(3) tvc(pr_1) dftvc(1)
. estimates store m3
```

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

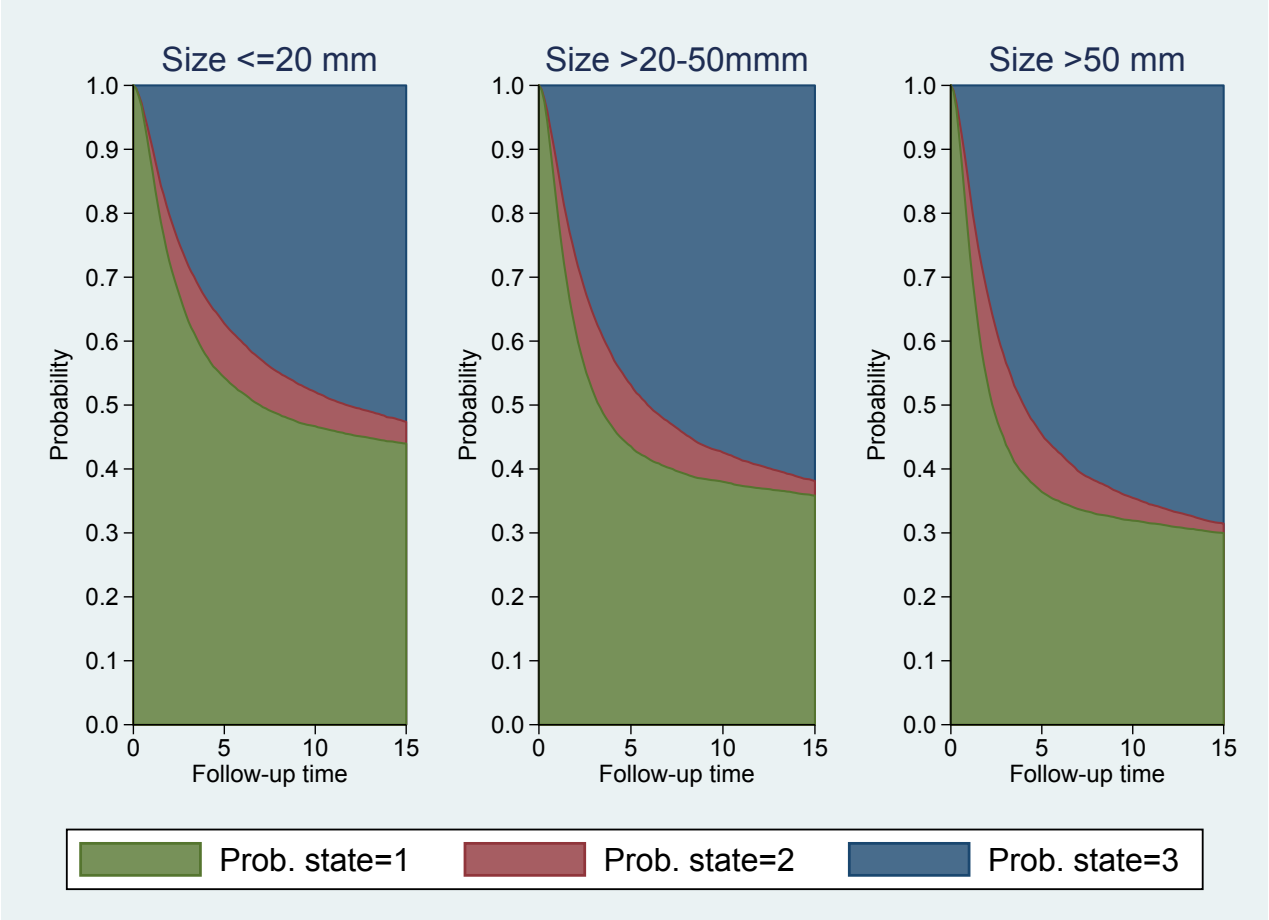



Figure: Probability of being in each state for a patient aged 54, with progesterone level (transformed scale) of 3.

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

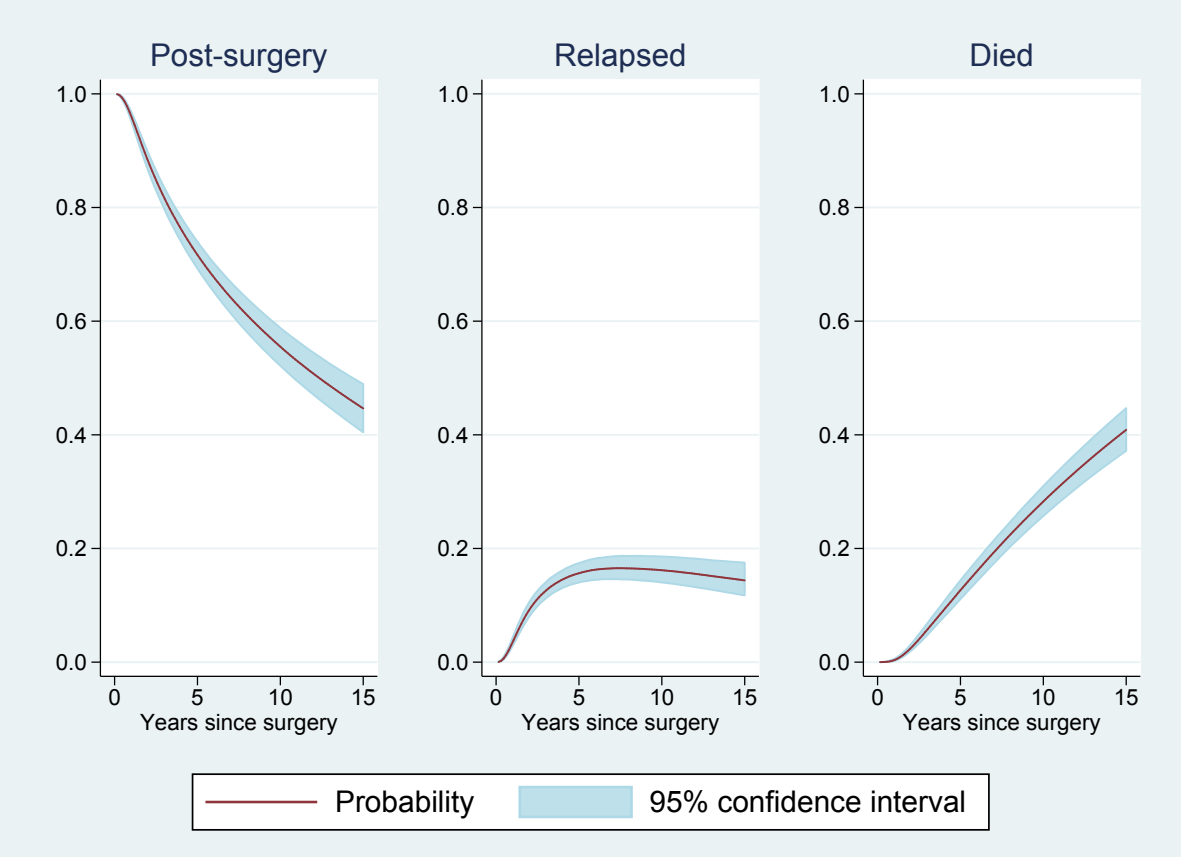
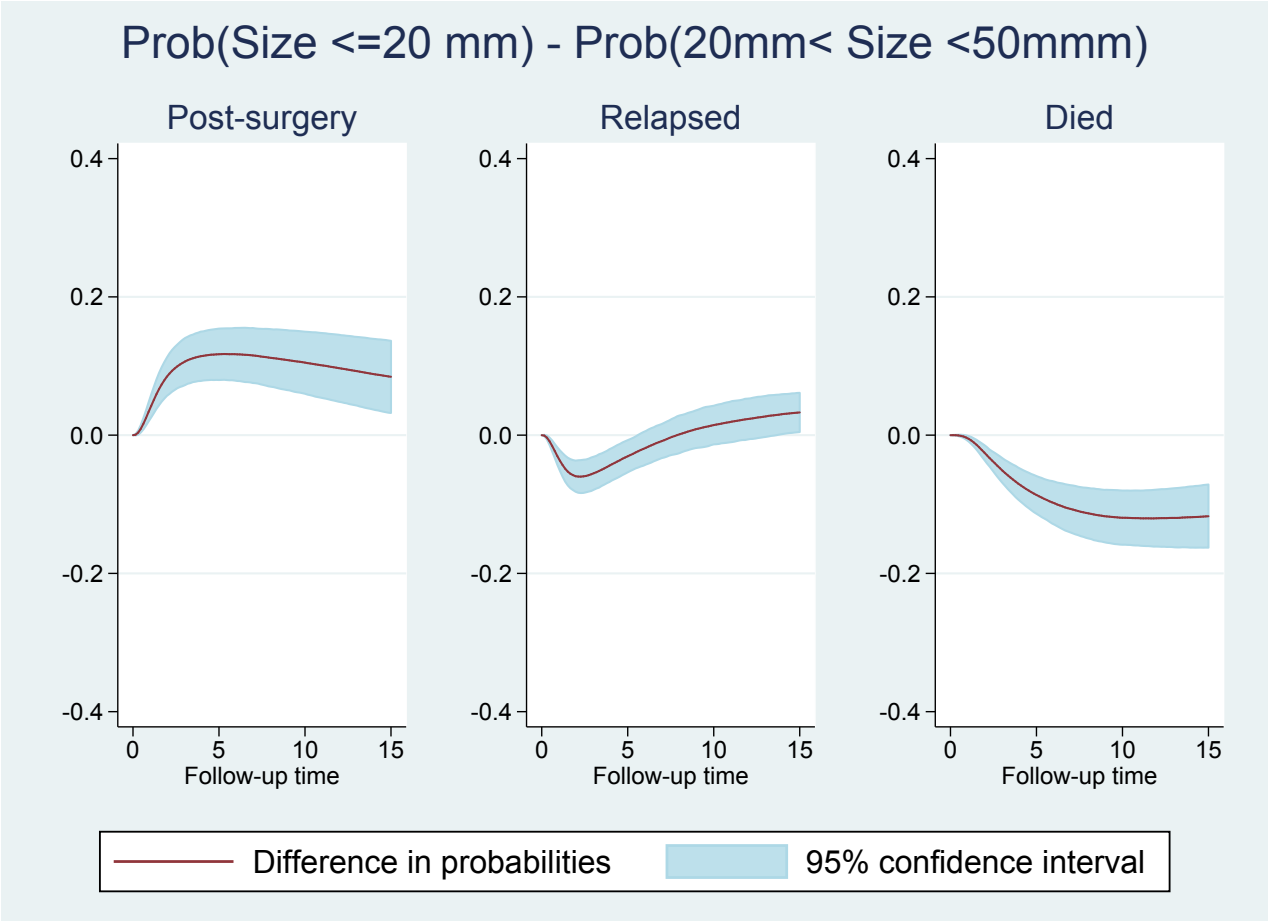


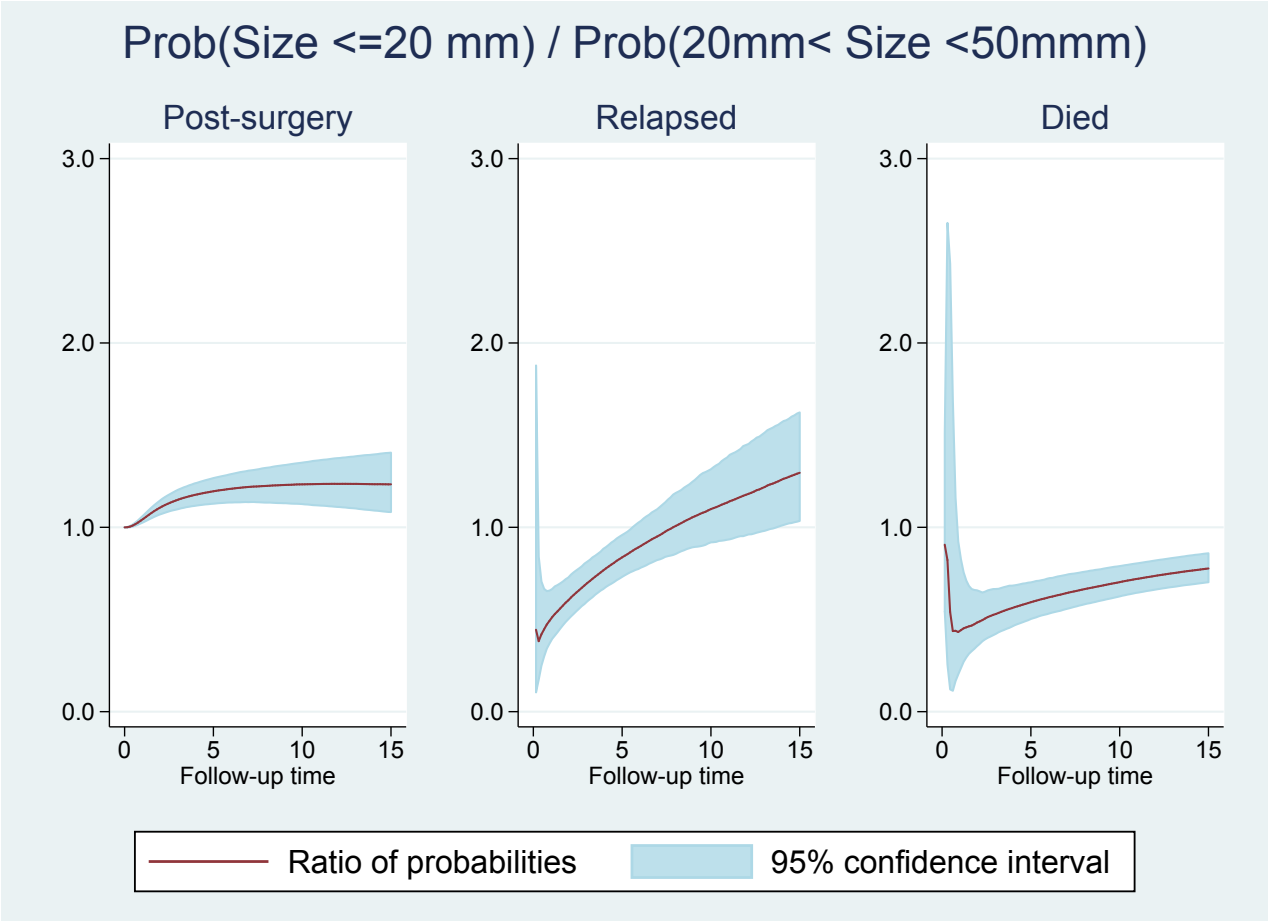

Figure: Probability of being in each state for a patient aged 54, 50> size ≥20 mm, with progesterone level (transformed scale) of 3, and associated confidence intervals.

# Differences in transition probabilities




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

# Ratios of transition probabilities




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

# Length of stay

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

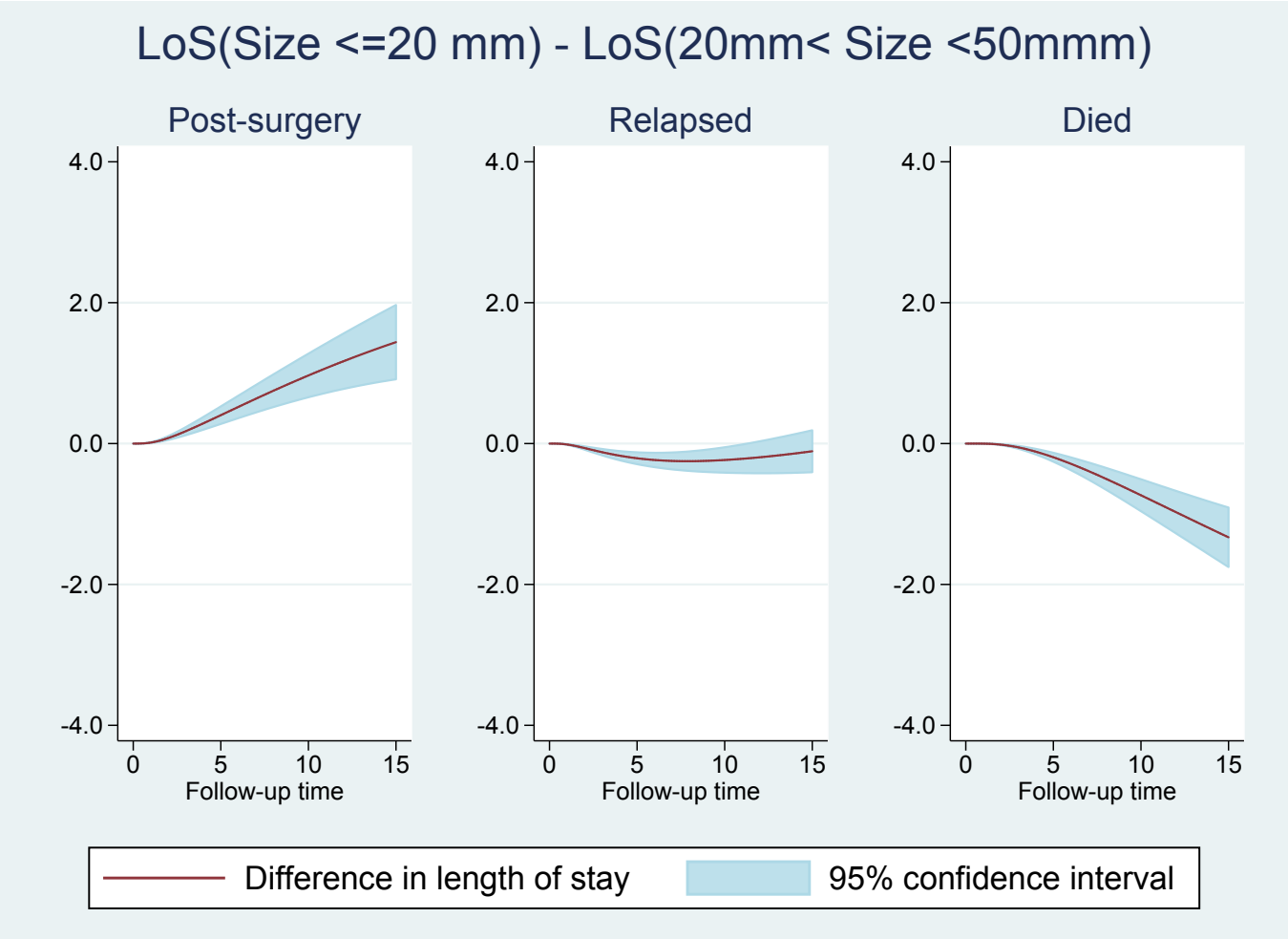
$$\int_s^t P(Y(u) = b | Y(s) = a) du$$

Using this we could calculate life expectancy if t $= \infty$, and $a = b = 1$ (Touraine et al., 2013). Thanks to the simulation approach, we can calculate such things extremely easily.

# Length of stay

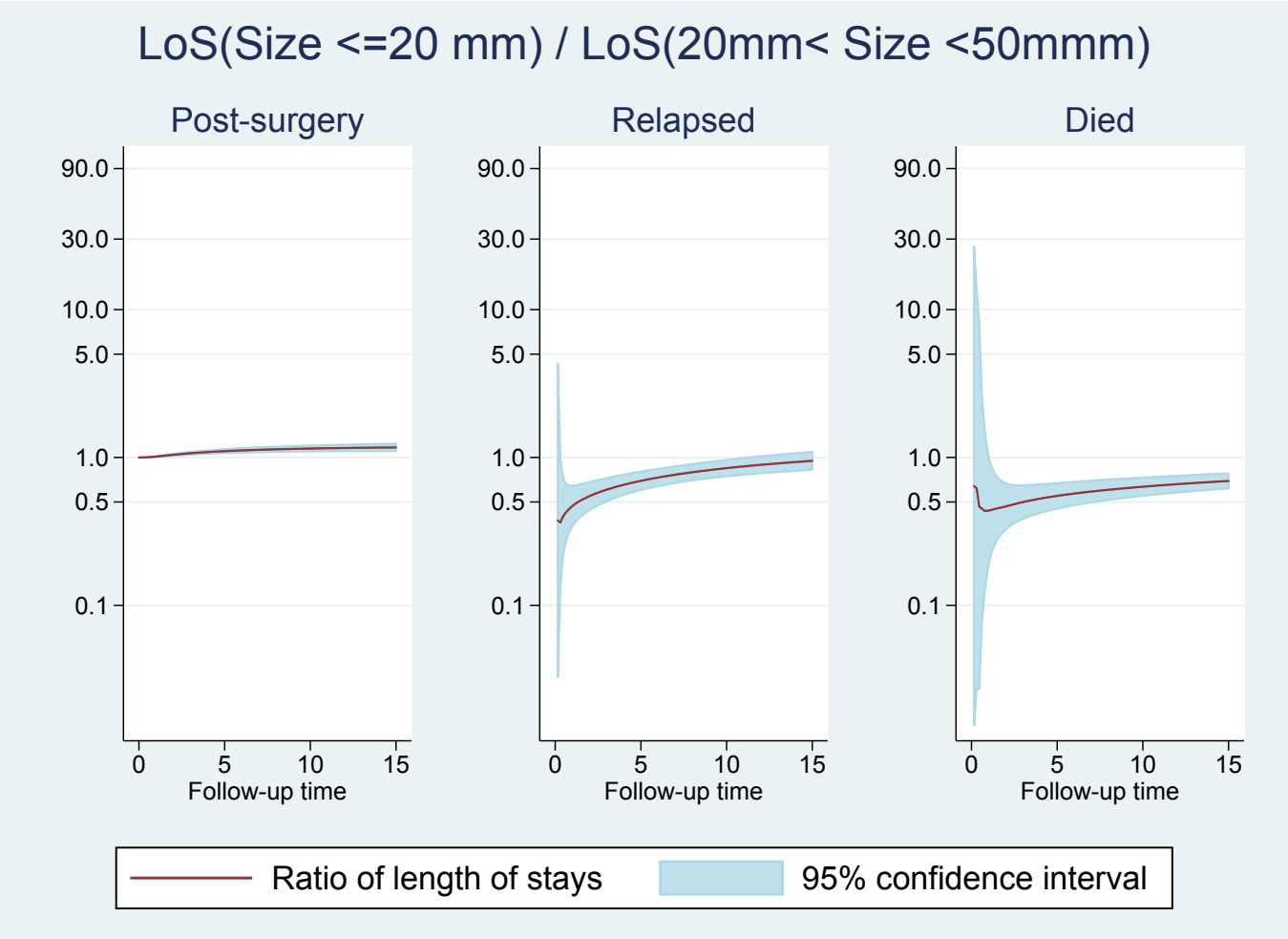

```
. predictms, transmat(tmat) models(m1 m2 m3) ///
  at(age 54 pgr 3 size1 1) ci los
```

# Differences in length of stay




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

# Ratios in length of stay




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

# Sharing covariate effects

- Fitting models separately to each transition means we can no longer share covariate effects - one of the benefits of fitting to the stacked data
- We therefore want to fit different distributions, but jointly, to the stacked data, which will allow us to constrain parameters to be equal across transitions

# Transition-specific distributions, estimated jointly

Jointly fit models with different distributions. Can constrain parameters to be equal for specified transitions.

```
. stms (age sz2 sz3 nodes pr_1 hormon, model(rp) df(3) scale(h)) ///
     (age sz2 sz3 nodes pr_1 hormon, model(weib)) ///
     (age sz2 sz3 nodes pr_1 hormon, model(rp) df(3) scale(h)) ///
     , transvar(_trans)
```

# Transition-specific distributions, estimated jointly

Jointly fit models with different distributions. Can constrain parameters to be equal for specified transitions.

```
. stms (age sz2 sz3 nodes pr_1 hormon, model(rp) df(3) scale(h)) ///
     (age sz2 sz3 nodes pr_1 hormon, model(weib)) ///
     (age sz2 sz3 nodes pr_1 hormon, model(rp) df(3) scale(h)) ///
     , transvar(_trans) constrain(age 1 3 nodes 2 3)
```

# Transition-specific distributions, estimated jointly

Jointly fit models with different distributions. Can constrain parameters to be equal for specified transitions.

```
. stms (age sz2 sz3 nodes pr_1 hormon, model(rp) df(3) scale(h)) ///
     (age sz2 sz3 nodes pr_1 hormon, model(weib)) ///
     (age sz2 sz3 nodes pr_1 hormon, model(rp) df(3) scale(h)) ///
     , transvar(_trans) constrain(age 1 3 nodes 2 3)
```

```
. predictms, transmat(tmat) at(age 34 sz2 1 nodes 5) ci
```

# Summary

- Multi-state survival models are increasingly being used to gain much greater insights into complex disease pathways

# Summary

- Multi-state survival models are increasingly being used to gain much greater insights into complex disease pathways
- The transition-specific distribution approach I've described provides substantial flexibility

# Summary

- Multi-state survival models are increasingly being used to gain much greater insights into complex disease pathways
- The transition-specific distribution approach I've described provides substantial flexibility
- We can fit a very complex model, but immediately obtain interpretable measures of absolute and relative risk

# Summary

- Multi-state survival models are increasingly being used to gain much greater insights into complex disease pathways
- The transition-specific distribution approach I've described provides substantial flexibility
- We can fit a very complex model, but immediately obtain interpretable measures of absolute and relative risk
- Software now makes them accessible
  - `ssc install multistate`

# Summary

- Multi-state survival models are increasingly being used to gain much greater insights into complex disease pathways
- The transition-specific distribution approach I've described provides substantial flexibility
- We can fit a very complex model, but immediately obtain interpretable measures of absolute and relative risk
- Software now makes them accessible
  - `ssc install multistate`
- Extensions:
  - Semi-Markov - `reset` with `predictms`
  - Cox model will also be available (`mstate` in R)
  - Reversible transition matrix
  - Standardised predictions - `std` (Gran et al., 2015; Sjölander, 2016)

# References I


Asaria, M., Walker, S., Palmer, S., Gale, C. P., Shah, A. D., Abrams, K. R., Crowther, M., Manca, A., Timmis, A., Hemingway, H., et al. Using electronic health records to predict costs and outcomes in stable coronary artery disease. *Heart*, 102(10):755–762, 2016.

Crowther, M. J. and Lambert, P. C. Simulating biologically plausible complex survival data. *Stat Med*, 32(23): 4118–4134, 2013.

Gran, J. M., Lie, S. A., Øyeflaten, I., Borgan, Ø., and Aalen, O. O. Causal inference in multi-state models–sickness absence and work for 1145 participants after work rehabilitation. *BMC Public Health*, 15(1):1–16, 2015.

Hinchliffe, S. R., Scott, D. A., and Lambert, P. C. Flexible parametric illness-death models. *Stata Journal*, 13(4): 759–775, 2013.

Hsieh, H.-J., Chen, T. H.-H., and Chang, S.-H. Assessing chronic disease progression using non-homogeneous exponential regression Markov models: an illustration using a selective breast cancer screening in Taiwan. *Statistics in medicine*, 21(22):3369–3382, 2002.

Iacobelli, S. and Carstensen, B. Multiple time scales in multi-state models. *Stat Med*, 32(30):5315–5327, Dec 2013.

Jackson, C. flexsurv: A platform for parametric survival modeling in r. *Journal of Statistical Software*, 70(1):1–33, 2016.

Jackson, C. H. Multi-state models for panel data: the msm package for R. *Journal of Statistical Software*, 38(8): 1–29, 2011.

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

Sjölander, A. Regression standardization with the r package stdreg. *European Journal of Epidemiology*, 31(6): 563–574, 2016.

Titman, A. C. Flexible nonhomogeneous Markov models for panel observed data. *Biometrics*, 67(3):780–787, Sep 2011.

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