

Advanced Bayesian survival analysis with merlin and morgana

Stata UK Conference 2024

Michael J. Crowther, PhD

Red Door Analytics

13th September 2024

About me

- Background in mathematics and statistics
- PhD in survival analysis and joint modelling from University of Leicester.
- 10+ years experience working in survival analysis methods development.
- CEO & Director of Statistical Methodology at Red Door Analytics
- Main research and teaching interests: survival methods, multi-state models and joint models

Why Bayesian survival analysis?

- Stata released `bayes` and `bayesmh`, but the latter passed me by for awhile
 - `bayes` : ... is possibly my favourite example of well designed syntax
- Lots of previous work syncing with Stan and WinBUGS

Why Bayesian survival analysis?

- Stata released `bayes` and `bayesmh`, but the latter passed me by for awhile
 - `bayes` : ... is possibly my favourite example of well designed syntax
- Lots of previous work syncing with Stan and WinBUGS

`bayesmh` allows user-defined likelihood evaluators – could I play around with it?

- I started last year at the Nordic meeting with
 - `morgana : stmerlin ... , distribution(rp) df(3)`
- This year I moved the goalpost
 - `morgana : merlin (...)(...)(...) ...`
 - `predict newvarname, statistic`

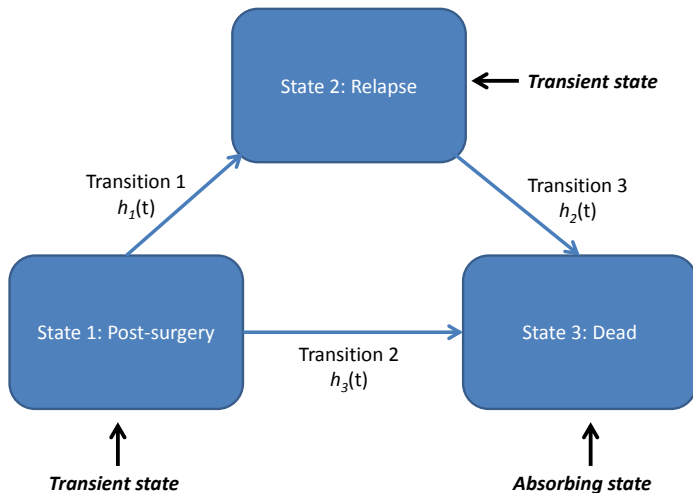
example

Primary breast cancer

- 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.

Primary breast cancer

- 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.



Illness–death model for primary breast cancer example.

Covariates of interest

- age at primary surgery
- tumour size (three classes; $\leq 20\text{mm}$, $20\text{--}50\text{mm}$, $> 50\text{mm}$)
- 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)

Data setup

```
. use multistate_example, clear
. mat tmat = (.,1,2\.,.,3\.,.,.)
. msset, id(pid) states(rfi osi osi) times(rf os os) ///
    transmatrix(tmat)
. stset _stop, enter(_start) failure(_status==1) scale(12)
. list _start _stop _from_to _status _trans if pid==3007
```

```
-----+-----+
|   _start   _stop  _from  _to  _status  _trans |
|-----+-----+
7480. |         0  12.451745      1    2         1      1 |
7481. |         0  12.451745      1    3         0      2 |
7482. | 12.451745  12.485079      2    3         1      3 |
-----+-----+
```

merlin

Building a merlin model

```
merlin (_t                /// response
        hormon pr_1       /// covariates
        if _trans==1     /// transition 1
        , family(rp, df(3) /// distribution
          failure(_d))    // event indicator
    )
```

Building a merlin model

```
merlin (_t                                /// response
      hormon rcs(pr_1, df(3))             /// covariates
      if _trans==1                         /// transition 1
      , family(rp, df(3))                 /// distribution
      failure(_d))                        // event indicator
)
```

Building a merlin model

```
merlin (_t                                /// response
      hormon rcs(pr_1, df(3))             /// covariates
      rcs(_t , df(2) offset(age))         /// second timescale
      if _trans==1                        /// transition 1
      , family(rp, df(3))                 /// distribution
      failure(_d))                        // event indicator
)
```

Building a merlin model

```
merlin (_t                                ///  
      hormon rcs(pr_1, df(3))             ///  
      rcs(_t , df(2) offset(age))         ///  
      rcs(_t, ...)#rcs(pr_1, df(1))       ///  
      if _trans==1                        ///  
      , family(rp, df(3))                 ///  
      failure(_d))                        ///  
)
```

Building a merlin model

```
merlin (_t hormon rcs(pr_1, df(3))           /// covariates
       rcs(_t , df(2) offset(age))          /// second timescale
       rcs(_t, ...)#rcs(pr_1, df(1))        /// NPH on timesc. 2
       if _trans==1                          /// transition 1
         , family(rp, df(3) failure(_d)))    /// distribution
(_t age hormon                               /// covariates
 if _trans==2                                /// transition 2
   , family(rp, df(5) failure(_d)))        // distribution
```

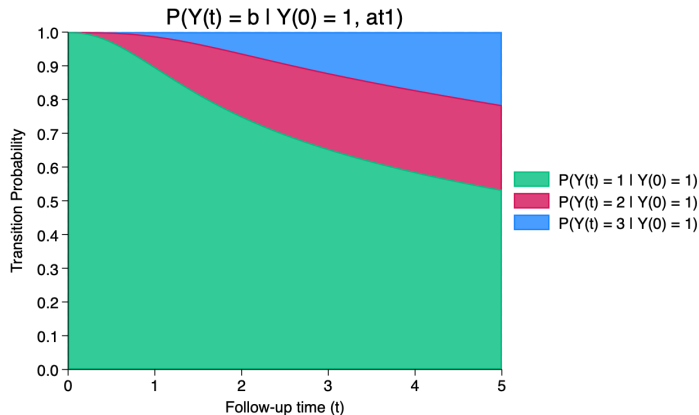

Building a merlin model

```
merlin (_t hormon rcs(pr_1, df(3))           /// covariates
        rcs(_t , df(2) offset(age))         /// second timescale
        rcs(_t, ...)#rcs(pr_1, df(1))       /// NPH on timesc. 2
        if _trans==1                         /// transition 1
        , family(rp, df(3) failure(_d)))    /// distribution
(_t age hormon                               /// covariates
  if _trans==2                               /// transition 2
  , family(rp, df(5) failure(_d)))         /// distribution
(_t hormon fp(pr_1, powers(1 2))           /// covariates
  if _trans==3                               /// transition 3
  , family(rp, df(5) failure(_d)          /// distribution
            ltruncated(_t0)))              // left truncation
```

Predictions

Probability of being in state 1 at time t

```
predict p1, transprob(1) transmatrix(tmat) at(hormon 1) zeroes
```



morgana

Bayesian flexible survival analysis

- Bayesian methods are hugely popular in HTA
- Particularly in meta-analysis and survival analysis
- Incorporating prior information is a strength, particularly in rare diseases
- In survival analysis, a huge issue is extrapolation to a lifetime horizon

Bayesian flexible survival analysis – morgana

Title

`morgana` — prefix command for estimating a Bayesian `stmerlin` survival model

Syntax

```
morgana [ , bayesmh_options ] : stmerlin_model
```

Description

The `morgana` prefix command fits Bayesian versions of survival models available with the `stmerlin` command.

[morgana's homepage](#)

`stmerlin` fits survival models, including a range of parametric distributions, flexible spline-based models, and the Cox model. It is a convenience wrapper of the more powerful `merlin` command, but with a much more user-friendly syntax. Time-dependent effects can be specified using restricted cubic splines.

[stmerlin's homepage](#)

The `merlin` command fits an extremely broad class of mixed effects regression models for linear, non-linear and user-defined outcomes. For full details and many tutorials, take a look at the accompanying website:

[merlin's homepage](#)

Examples

Setup

```
. webuse brcancer, clear  
. stset rectime, failure(censrec)
```

Estimate a Bayesian flexible parametric Royston–Parmar model:

```
. morgana : stmerlin hormon , dist(rp) df(3)
```

Syncing bayesmh with a likelihood evaluator

```
// fill up the current samples
gml.myb = newb
merlin_xb(gml,gml.myb)

// calculate the total log likelihood
alllnl = 0
for (i=1;i<=gml.Nmodels;i++) {
    gml.model = gml.modtoind = i
    alllnl = alllnl + quadcolsum>(*Plnl[i])(gml)
}
st_numscalar(st_local("lden"),alllnl)
```

Building a merlin model

```
morgana :                               ///  
merlin (_t hormon rcs(pr_1, df(3))      ///  
       rcs(_t , df(2) offset(age))      ///  
       rcs(_t, ...)#rcs(pr_1, df(1))    ///  
       if _trans==1                    ///  
       , family(rp, df(3) failure(_d))) ///  
(_t age hormon                          ///  
  if _trans==2                          ///  
  , family(rp, df(5) failure(_d))) ///  
(_t hormon fp(pr_1, powers(1 2))       ///  
  if _trans==3                          ///  
  , family(rp, df(5) failure(_d)      ///  
    ltruncated(_t0)))                 ///  
                                       // left truncation
```

Bayesian flexible survival modelling

```

Bayesian survival regression          MCMC iterations =    12,500
Random-walk Metropolis-Hastings sampling  Burn-in          =     2,500
                                          MCMC sample size =   10,000
                                          Number of obs    =    7,482
                                          Acceptance rate  =    .2178
                                          Efficiency: min  =    .005139
                                          avg             =    .05556
                                          max             =    .09127
Log marginal-likelihood = -8495.3885
  
```

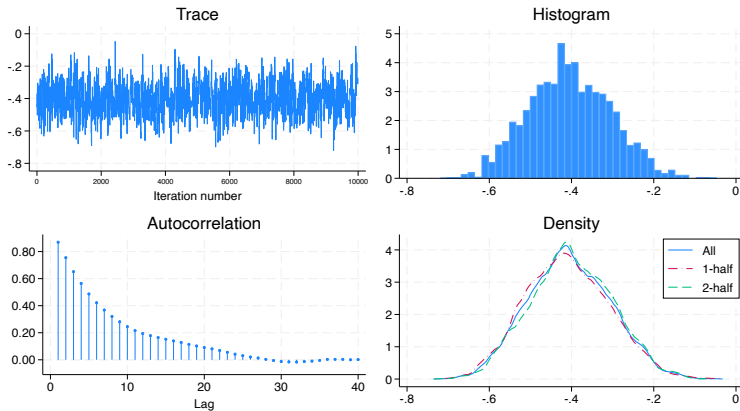
	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred. interval]	
m_1_chemo	.1708921	.0679421	.002249	.1737327	.0352579	.2992006
m_1_age	-.0008333	.0022796	.000076	-.0007661	-.005252	.0035402
_cons1	-.9754964	.1340637	.004445	-.9744596	-1.234694	-.7158145
_rcs1_1	.9921607	.02529	.000988	.9919211	.9433598	1.041425
_rcs1_2	.2822802	.0215711	.000924	.2826694	.2387542	.3253255
_rcs1_3	-.0252952	.0110329	.000434	-.0258134	-.0467949	-.0026522
m_2_chemo	.1724838	.4334772	.01641	.1727081	-.7478419	1.002909
m_2_age	.1329851	.0086941	.000343	.1333643	.1155756	.1492067
_cons2	-11.87702	.6344833	.025249	-11.90373	-13.11052	-10.67364
_rcs2_1	1.422209	.0951004	.0049	1.420971	1.244006	1.619145
_rcs2_2	-.1064825	.0603199	.002495	-.1091108	-.2120814	.0192805
_rcs2_3	-.0534212	.0354124	.001436	-.0528529	-.1230249	.0150708
m_3_chemo	.3283317	.0844886	.003559	.3303777	.1579553	.4939602
m_3_age	.0103573	.0025383	.000084	.0103202	.0054004	.0151745
_cons3	-.059857	.1807235	.013859	-.0649057	-.4057074	.2981892
_rcs3_1	.5350637	.0572586	.007987	.5392806	.4165171	.6398975
_rcs3_2	.0761114	.0292631	.003475	.0761524	.0195288	.1328764
_rcs3_3	-.0167223	.0099446	.000858	-.0167482	-.0359778	.0021036

Adding an informative prior

```
morgana , prior({m_1_hormon}, normal(-0.2,0.1)) : ///  
merlin (_t hormon rcs(pr_1, df(3))          ///  
        rcs(_t , df(2) offset(age))         ///  
        rcs(_t, ...) # rcs(pr_1, df(1))     ///  
        if _trans==1                        ///  
        , family(rp, df(3) failure(_d)))    ///  
(_t age hormon                             ///  
        if _trans==2                        ///  
        , family(rp, df(5) failure(_d)))   ///  
(_t hormon fp(pr_1, powers(1 2))           ///  
        if _trans==3                        ///  
        , family(rp, df(5) failure(_d)     ///  
              ltruncated(_t0)))           ///  
                                          //
```

bayesgraph diagnostic {m_1_hormon}

hormon



Predictions

Utilising the saved samples

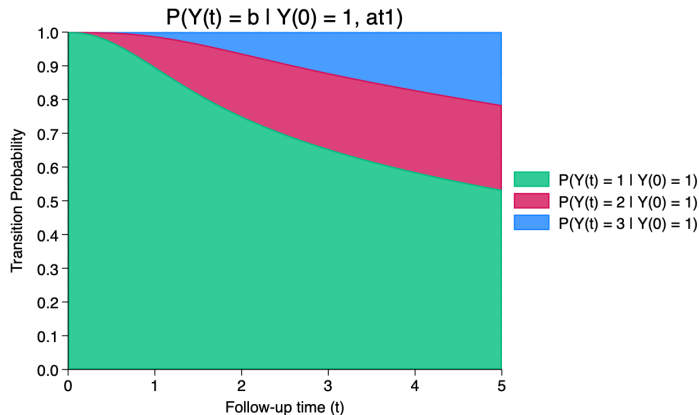
```
// grab the saved samples
curfr = st_framecurrent()
st_framecurrent(st_local("sframe"))
st_view(b=.,., "eq0_*")
st_view(freq=.,., "_frequency")
st_framecurrent(curfr)

// calculate the prediction and store
for (i=1;i<=Nsamps;i++) {
  gml.myb = b[i,]
  merlin_xb(gml,gml.myb)
  pred[,i] = (*pf)(gml)
}
```

Predictions

Probability of being in state 1 at time t

```
predict p1, transprob(1) transmatrix(tmat) at(horizon 1) zeroes
```



Next steps with bayesmh

Multilevel & multiple outcomes

```
morgana : merlin (y ... M1[id]@1, family(gaussian)) ///  
              (...)(...)
```

User-defined models with bayesmh

```
bayesmh y x U0[id], llevaluator(l1, parameters({var})) ///  
      block(U0[id], reffects) ...
```

Random effects not supported;

Parameters corresponding to a random-effects variable may not be used in user-defined evaluators.

Next steps with merlin

- merlin's current release is v2.4.5
- version 3.0.0 is under development, and:
 - cuts computation time by anywhere from 40% to 70% in testing
 - supports full factor variable specification

Meet...



Merlin & Morgana

Meet...



Merlin & Morgana
Ronaldo & Messi

Meet...



Merlin & Morgana

Ronaldo & Messi

Stata & Mata

Meet...



Merlin & Morgana

Ronaldo & Messi

Stata & Mata

Yes, I'm aware the cats are the highlight of my talk