Health technology assessment and Stata

Reviewing the old and coding the new

Michael J. Crowther

Red Door Analytics AB
Stockholm, Sweden

 reddooranalytics.se  contact@reddooranalytics.se  @RDAnalyticsAB

Introduction

Aim

- This talk was an excuse to review my own Stata packages, with the aim to improve them, motivated by some of our work at RDA

Aim

- This talk was an excuse to review my own Stata packages, with the aim to improve them, motivated by some of our work at RDA
- This talk was also an excuse to force me to finish two new packages, again, because we're using them at RDA

Aim

- This talk was an excuse to review my own Stata packages, with the aim to improve them, motivated by some of our work at RDA
- This talk was also an excuse to force me to finish two new packages, again, because we're using them at RDA

Unsurprisingly, I mostly failed at both

Health Technology Assessment

- Health technology assessment (HTA) is a systematic and multidisciplinary evaluation of the properties of health technologies and interventions covering both their direct and indirect consequences

Health Technology Assessment

- Health technology assessment (HTA) is a systematic and multidisciplinary evaluation of the properties of health technologies and interventions covering both their direct and indirect consequences
- It is a multidisciplinary process that aims to determine the value of a health technology and to inform guidance on how these technologies can be used in health systems around the world.

Health Technology Assessment

- Health technology assessment (HTA) is a systematic and multidisciplinary evaluation of the properties of health technologies and interventions covering both their direct and indirect consequences
- It is a multidisciplinary process that aims to determine the value of a health technology and to inform guidance on how these technologies can be used in health systems around the world.
- It has been described as a bridge that connects the world of research to that of policy making.

https://www.who.int/health-topics/health-technology-assessment

Health Technology Assessment

A framework for HTA

Health Technology Assessment

A framework for HTA

- Analysis of a clinical trial with a survival outcome
  - `stmerlin`
  - We might need `bayes` if we have prior information

Health Technology Assessment

A framework for HTA

- Analysis of a clinical trial with a survival outcome
  - `stmerlin`
  - We might need `bayes` if we have prior information
- Combine and compare with existing evidence
  - (Network) meta-analysis `network`, `meta`
  - Matching adjusted indirect comparisons ?

Health Technology Assessment

A framework for HTA

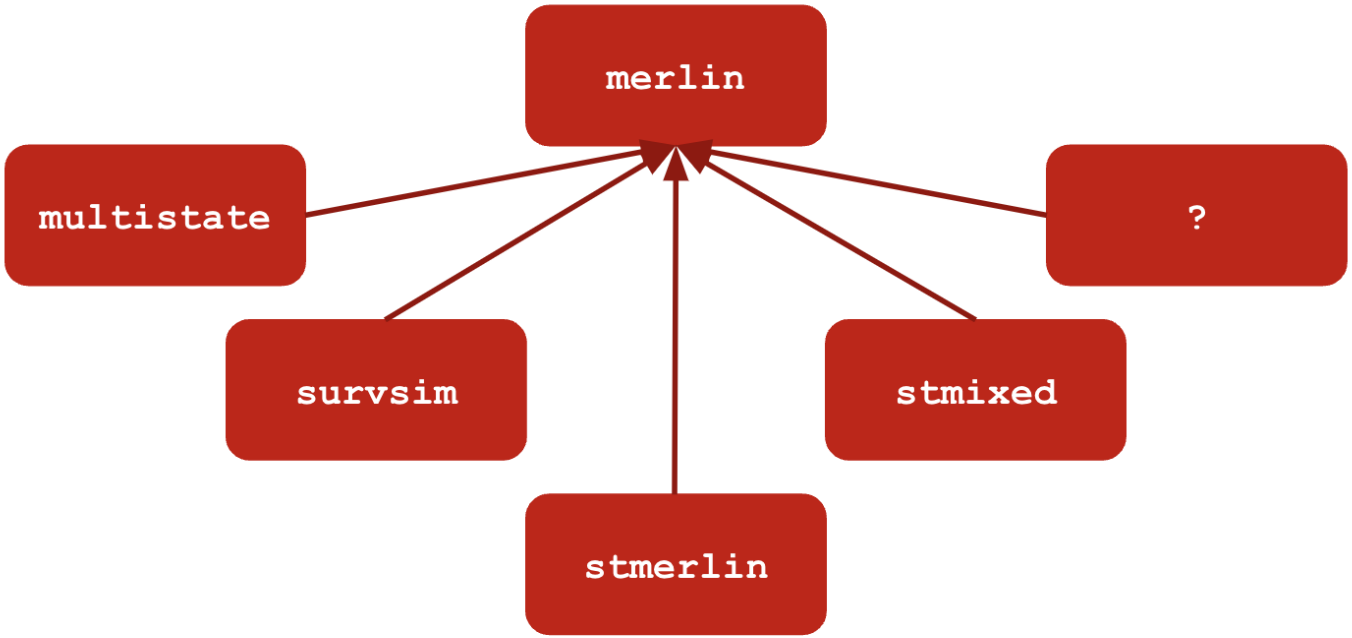
- Analysis of a clinical trial with a survival outcome
  - `stmerlin`
  - We might need `bayes` if we have prior information
- Combine and compare with existing evidence
  - (Network) meta-analysis `network`, `meta`
  - Matching adjusted indirect comparisons ?
- Build a natural history model
  - Required for assessing lifetime costs - `multistate`
  - Markov models are the default

Health Technology Assessment

A framework for HTA

- Analysis of a clinical trial with a survival outcome
  - `stmerlin`
  - We might need `bayes` if we have prior information
- Combine and compare with existing evidence
  - (Network) meta-analysis `network`, `meta`
  - Matching adjusted indirect comparisons ?
- Build a natural history model
  - Required for assessing lifetime costs - `multistate`
  - Markov models are the default
- Conduct microsimulation & probabilistic sensitivity analysis
  - `survsim`

A framework for HTA in Stata



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.

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)
```

merlin needs a refresh

merlin needs a refresh

- It can do a lot
  - This is great

merlin needs a refresh

- It can do a lot
  - This is great
  - This is not so great

merlin needs a refresh

- It can do a lot
  - This is great
  - This is not so great
- Priorities
  - Making it faster
  - Allowing factor variables

`merlin` needs a refresh

- It can do a lot
  - This is great
  - This is not so great
- Priorities
  - Making it faster
  - Allowing factor variables

Importantly, everything that comes next uses `merlin` under the hood, so if `merlin` gets better, everything else does.

stmerlin

Title

stmerlin —— convenience wrapper for estimating a parametric and semi-parametric survival model with **merlin**, optionally including multiple timescales

Syntax

stmerlin [*indepsyntax*] [*if*] [*in*] , **distribution(***model***)** [, *options display_options*]

where *indepsyntax* is a **merlin** linear predictor, which can be anything from a simple *varlist*, to directly specifying spline or fractional polynomial functions of continuous covariates.

You must **stset** your data before using **stmerlin**; see **[ST] stset**.

options	Description
distribution(addrcs)	hazard scale spline model
distribution(exponential)	exponential model
distribution(cox)	Cox model
distribution(gompertz)	Gompertz model
distribution(ggamma)	generalised gamma
distribution(lognormal)	log normal
distribution(loglogistic)	log logistic
distribution(pwexponential)	piecewise-exponential model
distribution(rp)	Royston-Parmar model
distribution(rcs)	Log-hazard scale spline model
distribution(weibull)	Weibull model

Flexible survival model with stmerlin

```
. webuse brcancer
(German breast cancer data)
. quietly stset rectime, failure(censrec)
. stmerlin hormon, distribution(rp) df(3)
Obtaining initial values
variables created: _rcs1_1 to _rcs1_3
Fitting full model:
Iteration 0:  Log likelihood = -2632.0961
Iteration 1:  Log likelihood = -2612.0924
Iteration 2:  Log likelihood = -2607.9978
Iteration 3:  Log likelihood = -2607.9714
Iteration 4:  Log likelihood = -2607.9714
Survival model                                    Number of obs = 686
Log likelihood = -2607.9714
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_t:						
hormon	-.3613746	.1248801	-2.89	0.004	-.606135	-.1166142
_cons	-1.192909	.0814642	-14.64	0.000	-1.352576	-1.033243

```
    Warning: Baseline spline coefficients not shown - use ml display
```

Cox model with multiple timescales

```
. set obs 10000
Number of observations (_N) was 0, now 10,000.
. gen trt = runiform()>0.5
. gen agec = rnormal(50,5) - 50
. gen yearc = 1990 + floor(20*runiform()) - 2000
. survsim stime died, maxtime(5) cov(trt -0.5)                    ///
>         hazard( 0.1:*1.2:*{t}:^0.2 :*                           ///
>                 exp(                                            ///
>                         0.1 :* (agec :+ {t})                    ///
>                         :+ trt :* 0.1 :* (agec :+ {t})          ///
>                         :- 0.1 :* (yearc :+ {t})                ///
>                 )                                               ///
>         )
. qui stset stime, f(died)
. stmerlin trt, dist(cox) time2(df(2) offset(agec) time noorthog)  ///
>         time3(df(2) offset(yearc) time noorthog)
Obtaining initial values
Fitting full model:
Iteration 0:  Log likelihood = -43558.366
Iteration 1:  Log likelihood = -43558.364
Survival model                                    Number of obs = 10,000
Log likelihood = -43558.364
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
_t:						
trt	-.1031376	.0282546	-3.65	0.000	-.1585155	-.0477596
rcs():1	.1652705	.0101485	16.29	0.000	.1453797	.1851613
rcs():2	.0000487	.0000226	2.16	0.031	4.47e-06	.000093
rcs():1	-.0949912	.0064554	-14.72	0.000	-.1076435	-.0823389
rcs():2	-4.70e-06	.0000326	-0.14	0.885	-.0000686	.0000592

multistate

Title

multistate — Multi-state survival analysis

Description

multistate provides a set of commands, described below, for multi-state survival analysis. This includes data preparation tools, obtaining predictions from general continuous time multi-state survival models, both Markov and semi-Markov, and plotting utilities. Transition hazard models must be estimated using the **stmerlin** or **merlin** commands.

There are a number of commands in the **multistate** package, including:

msset is a data preparation tool which converts a dataset from wide (one observation per subject, multiple time and status variables) to long (one observation for each transition of which a subject is at risk).

msboxes creates a descriptive plot of the multi-state process through the transition matrix and numbers at risk.

msaj calculates the non-parametric Aalen-Johansen estimates of transition probabilities, and the length of stay in each state.

predictms calculates a variety of predictions from a Markov or semi-Markov multi-state survival model, including transition probabilities, length of stay (restricted mean time in each state), the probability of ever visiting each state and transition specific hazard and survival functions. Predictions are made at user-specified covariate patterns. Differences and ratios of predictions across covariate patterns can also be calculated. Standardised (study population-averaged) predictions can be obtained. Confidence intervals for all quantities are available. User-defined predictions can also be calculated by providing a user-written Mata function, to provide complete flexibility. **predictms** can be used with a general transition matrix (cyclic or acyclic), and allows the use of transition-specific timescales.

graphms creates stacked transition probability plots, following a **predictms** call.

multistate and HTA

multistate and HTA

- Markov models are ubiquitous in HTA

multistate and HTA

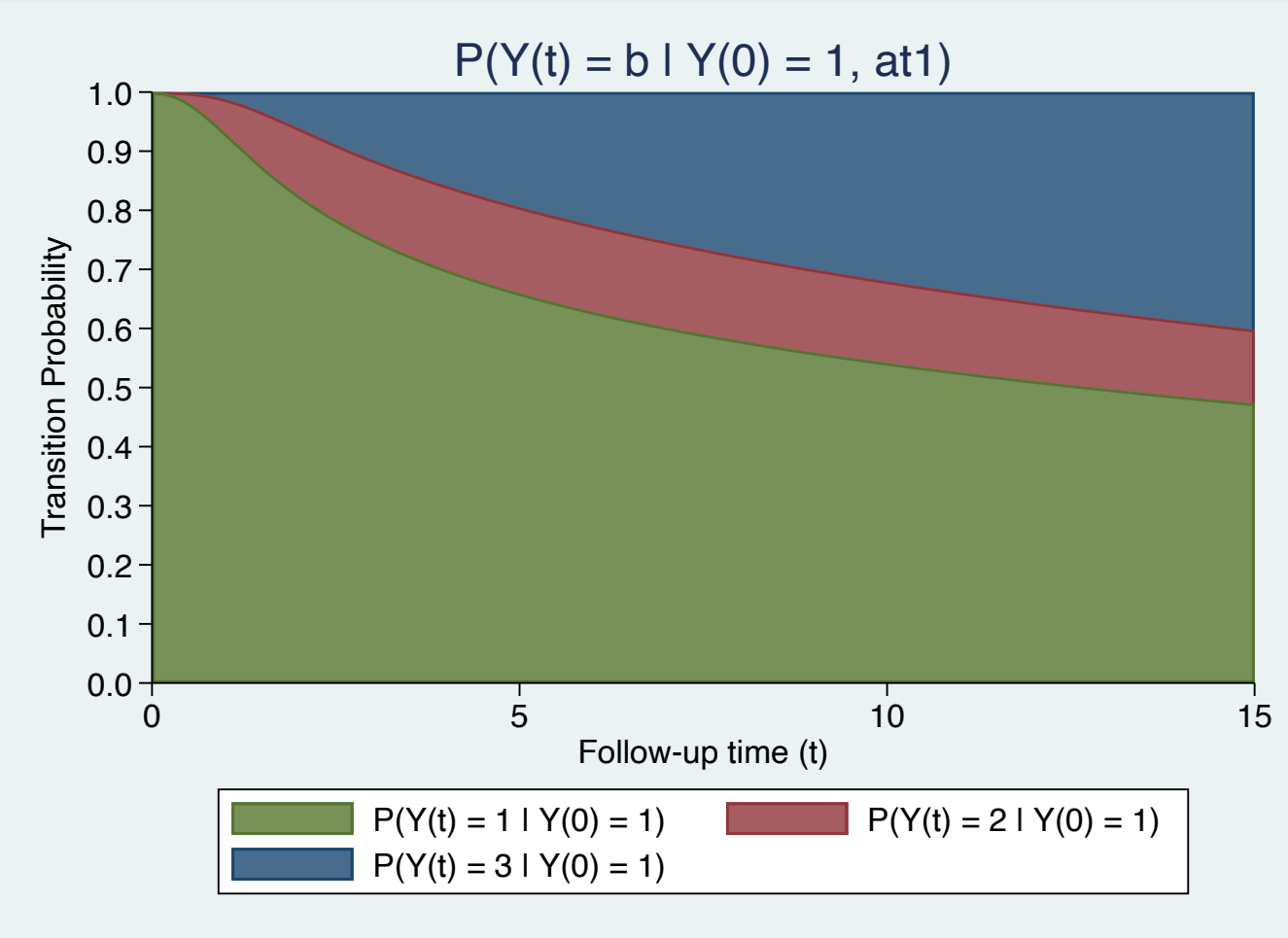
- Markov models are ubiquitous in HTA
- This assumption is extremely strong, and extremely unlikely

multistate and HTA

- Markov models are ubiquitous in HTA
- This assumption is extremely strong, and extremely unlikely
- Estimation, prediction and simulation of a non-Markov is not easy in the slightest

multistate and HTA

predictms, models(m1 m2 m3) at1(trt 1) probability



survsim

Title

survsim — Simulate survival data from a parametric distribution, a user-defined distribution, from a fitted **merlin** model, from a cause-specific hazards competing risks model, or from a general multi-state model

Description

survsim simulates survival data from:

help survsim parametric - a parametric distribution including the exponential, Gompertz and Weibull, and 2-component mixtures of them. Baseline covariates can be included, with specified associated log hazard ratios. Non-proportional hazards can also be included with all models; under an exponential or Weibull model covariates are interacted with log time, under a Gompertz model covariates are interacted with time. See **Crowther and Lambert (2012)** for more details.

help survsim user - a user-defined distribution. Survival times can be simulated from bespoke, user-defined [log] [cumulative] hazard functions. The function must be specified in Mata code (using colon operators), with survival times generated using a combination of numerical integration and root finding techniques. Time-dependent effects can also be specified with a user-defined function of time. See **Crowther and Lambert (2013)** for more details.

help survsim model - a fitted **merlin** model. **merlin** fits a broad class of survival models, including standard parametric models, spline-based survival models, and user-defined survival models.

help survsim msm - a competing risks or general multi-state model. Event times can be simulated from transition-specific hazards, where each transition hazard function can be a standard parametric distribution, or a user-defined complex hazard function. Covariates and time-dependent effects can be specified for each transition-specific hazard independently.

```
. survsim, model(m1) maxtime(15)
```

Bayesian flexible survival analysis

- Bayesian methods are hugely popular in HTA
- Particularly in meta-analysis
- Incorporating prior information is a huge strength, particularly in rare diseases

Bayesian flexible survival analysis - morgana

Title

morgana — prefix commmand for estimating a Bayesian **stmerlin** survival model

Syntax

morgana [, *bayesmh_options*] : *stmerlin_model*

Description

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

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.

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:

reddooranalytics.se/products/merlin

Syncing bayesmh with a likelihood evaluator

```
*! version 1.0.0  30aug2023

program morgana_ll
        version 18
        args lden ${eqns}
        mata: morgana_ll()
end


version 18
mata:

void morgana_ll()
{
        struct merlin_struct scalar gml

        gml     = *findexternal(st_global("object"))
        eqns    = tokens(st_global("eqns"))
        Neqns   = cols(eqns)
        newb    = J(1,Neqns,.)
        for (i=1;i<=Neqns;i++) newb[1,i] = st_numscalar(st_local(eqns[1,i]))

        gml.myb = newb
        merlin_xb(gml,gml.myb)

        gml.survind = gml.todo = 0
        if      (gml.familys=="rp") {
                lnl = quadcolsum(merlin_logl_rp(gml,G,H),1)
        }
        /*
                other distributions
        */

        st_numscalar(st_local("lden"),lnl)
}

end
```

Bayesian flexible survival model with morgana & stmerlin

```
. morgana : stmerlin hormon, distribution(rp) df(3)
Obtaining initial values
variables created: _rcs1_1 to _rcs1_3

Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  _t ~ morgana_ll({hormon},{_cons1},{_rcs_1_1},{_rcs_1_2},{_rcs_1_3})
Prior:
  {hormon _cons1 _rcs_1_1 _rcs_1_2 _rcs_1_3} ~ normal(0,10000)
------------------------------------------------------------------------------

Bayesian survival regression                    MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling        Burn-in          =      2,500
                                                MCMC sample size =     10,000
                                                Number of obs    =        686
                                                Acceptance rate  =      .2217
                                                Efficiency:  min =     .05387
                                                             avg =     .07593
Log marginal-likelihood = -2644.8553                         max =     .09965

------------------------------------------------------------------------------
             |                                                 Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
      hormon |  -.364192   .1251727   .003965  -.3624191  -.6164516  -.1251256
       _cons1 | -1.200934   .0787249   .002947  -1.199913  -1.361028  -1.056648
    _rcs_1_1 |   1.61066   .1247576   .005375    1.61113   1.374546   1.865117
    _rcs_1_2 |  .5887768   .1131379   .004268   .5908103   .3721069   .8272459
    _rcs_1_3 |  -.051528     .03358   .001155  -.0512411  -.1216544   .0132672
------------------------------------------------------------------------------
```

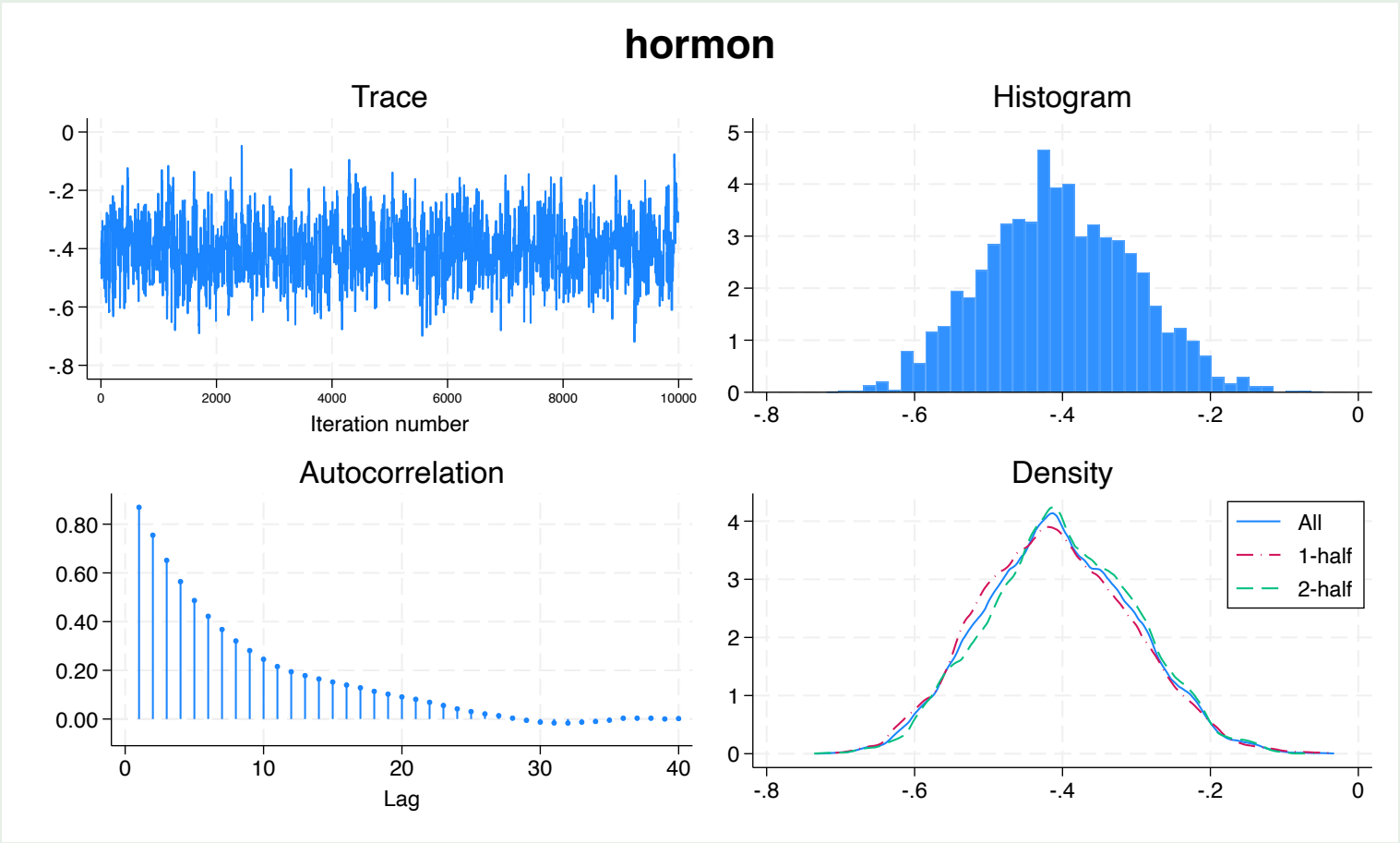
Adding an informative prior on {hormon}

```
. morgana, prior({hormon}, normal(-0.5,0.03)) :   ///
>         stmerlin hormon, distribution(rp) df(3)
Obtaining initial values
variables created: _rcs1_1 to _rcs1_3

Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  _t ~ morgana_ll({hormon},{_cons1},{_rcs_1_1},{_rcs_1_2},{_rcs_1_3})
Priors:
                              {hormon} ~ normal(-0.5,0.03)
  {_cons1 _rcs_1_1 _rcs_1_2 _rcs_1_3} ~ normal(0,10000)
------------------------------------------------------------------------------
Bayesian survival regression                    MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling        Burn-in          =      2,500
                                                MCMC sample size =     10,000
                                                Number of obs    =        686
                                                Acceptance rate  =      .1886
                                                Efficiency:  min =      .0246
                                                             avg =     .04944
Log marginal-likelihood = -2638.9597                         max =     .06889

------------------------------------------------------------------------------
             |                                                 Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
      hormon | -.4049185   .1003096   .003822  -.4089641  -.5980976  -.2106219
       _cons1 |  -1.18509   .0775748   .003037  -1.183455  -1.340632  -1.039873
    _rcs_1_1 |  1.604934   .1259947   .005818    1.59776   1.369986   1.872174
    _rcs_1_2 |  .5846417   .1146586   .005626   .5770516   .3746932   .8246799
    _rcs_1_3 | -.0520245   .0357107   .002277   -.051204  -.1203669   .0189753
------------------------------------------------------------------------------
```

bayesgraph diagnostic {hormon}



Next steps with bayesmh

What I would like

```
. predict s, survival time(timevar) at(hormon 1)
```

- Not so easy to generalise
- Feasible for one specific timepoint

Next steps with bayesmh

What I would like

```
. predict s, survival time(timevar) at(hormon 1)
```

- Not so easy to generalise
- Feasible for one specific timepoint

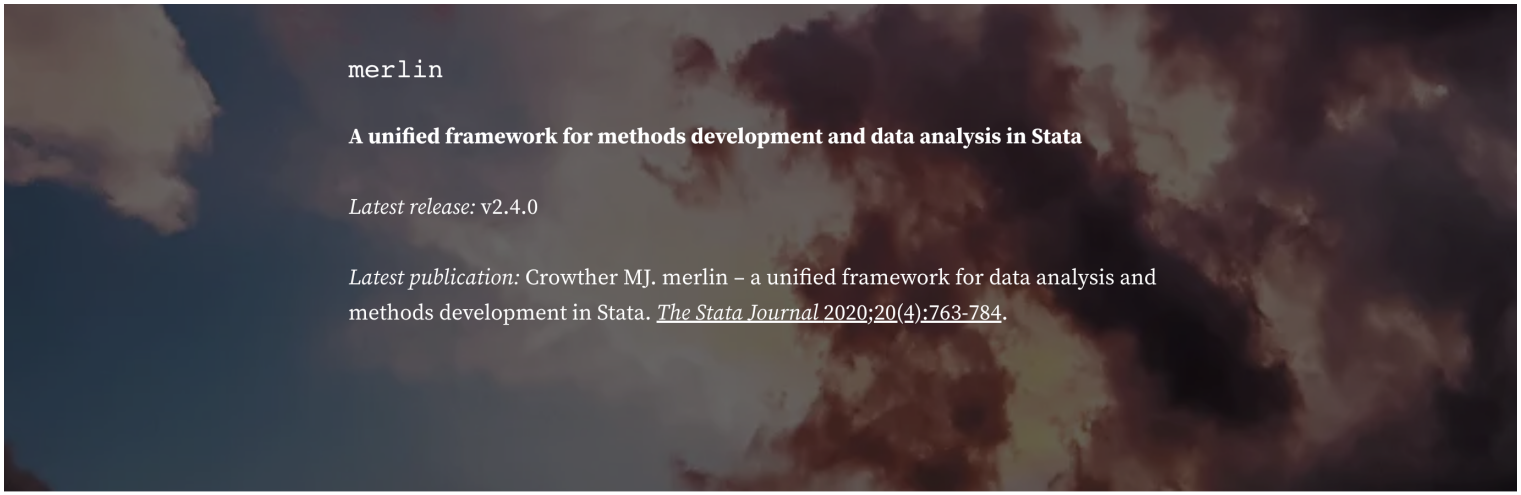
Multilevel & multiple outcomes

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

Summary & future work

- The ecosystem around `merlin` is growing, along with its user base
- There's a lot to do to make it more usable
- I promised a new command called `maic` for Matching-Adjusted Indirect Comparison

References



`merlin` stands for ***Mixed Effects Regression for LInear, Non-linear and user-defined models***. `merlin` has the capabilities to fit a linear regression or a Weibull survival model, a three-level logistic mixed effects model, or a multivariate joint model of multiple longitudinal outcomes (of different types) and a recurrent event and survival with non-linear effects... the list is rather endless.

reddooranalytics.se/products/merlin