

Multilevel mixed effects survival analysis: Estimation, simulation and application

Michael J. Crowther

University of Leicester

michael.crowther@le.ac.uk

Abstract

In this article, I present the user written `stmixed` command for the fitting of multilevel survival models, which serves as both an alternative to Stata's official `mestreg`, and a complimentary program with substantial extensions. `stmixed` can fit multilevel survival models with any number of levels and random effects at each level, including flexible spline-based approaches (such as Royston-Parmar and the log hazard equivalent) or user-defined hazard models. Simple or complex time-dependent effects can be included, as well as the addition of expected mortality for a relative survival model. Left-truncation/delayed entry can be used and t -distributed random effects are provided as an alternative to Gaussian random effects. The methods are illustrated with a commonly used dataset of patients with kidney disease suffering recurrent infections, and a simulated example, illustrating a simple approach to simulating clustered survival data using `survsim` (Crowther and Lambert, 2012, 2013). `stmixed` is part of the `merlin` family (Crowther, 2017, 2018).

June 29, 2018

1 Introduction

Clustered survival data is often observed in a variety of settings. Within medical research, a common example is the analysis of recurrent event data, where individual patients can experience the event of interest multiple times throughout the follow-up period, and the inherent correlation within patients can be accounted for using a frailty term (Gutierrez, 2002).

In the field of meta-analysis, the individual patient data (IPD) meta-analysis of survival data is growing in use, as this form of analysis is recognised as the gold standard approach (Simmonds et al., 2005). Analysing the IPD simultaneously within a hierarchical structure, allows direct adjustment for confounders and incorporation of non-proportional hazards in covariate effects (Tudur-Smith et al., 2005; Crowther et al., 2012, 2014). Often a random treatment effect is assumed to account for heterogeneity present in treatment effects across the pooled trials.

A further area of interest is relative survival. Particularly prevalent in cancer survival studies, relative survival allows the modelling of excess mortality associated with a diseased population compared to that of the general population (Dickman et al., 2004). Such data often exhibits a hierarchical structure, with patients nested within geographical regions such as counties. Patients living in the same area may share unobserved characteristics, such as environmental aspects or medical care access (Charvat et al., 2016).

With the release of Stata 14 came the `mestreg` command to fit multilevel mixed effects parametric survival models, assuming normally distributed random effects, estimated with maximum likelihood utilising Gaussian quadrature. In this article, I present the user written `stmixed` command for the

fitting of multilevel survival models, which serves as both an alternative to Stata’s official `mestreg`, and a complimentary program with substantial extensions. `stmixed` can fit multilevel survival models with any number of levels and random effects at each level, including flexible spline-based approaches (such as Royston-Parmar and the log hazard equivalent) or user-defined hazard models. Simple or complex time-dependent effects can be included, as well as the addition of expected mortality for a relative survival model. Left-truncation/delayed entry can be used and t -distributed random effects are provided as an alternative to Gaussian random effects. `stmixed` has the ability to estimate a multilevel survival model, with any of the aforementioned extensions, combined with a user-defined hazard model, providing a platform for methods development within a survival context.

In essence, `stmixed` is now a shell file for the recently introduced `merlin` command (Crowther, 2017, 2018), which provides a general framework for estimating multivariate mixed effects models. Multilevel survival models are one of the many classes of models of which `merlin` can fit, but to make the methods more accessible to researchers, I provide `stmixed` as a convenience wrapper function, with a far simpler syntax, yet still providing the power and flexibility of `merlin`.

The article is arranged as follows. Section 2 describes the multilevel parametric survival framework, and derives the likelihood used to estimate the models, including the extension to relative survival models and delayed entry/left-truncation. Section 3 details the model syntax of `stmixed`, describing the available options, and Section 4 describes the post-estimation tools available. I illustrate the command in Section 5 with a dataset of patients with kidney disease who are followed up for recurrent infection at the catheter insertion point, and show how to simulate clustered survival data using the `survsim` command, representing an IPD meta-analysis scenario, with a random treatment effect. I conclude the paper in Section 6.

2 Multilevel mixed effects survival models

For ease of exposition, I describe the methods in the context of a two-level model, but `stmixed` can handle any number of levels. I begin with some notation. Define $i = 1, \dots, N$ clusters (e.g. trials or centres), with each cluster having $j = 1, \dots, n_i$ patients. Let S_{ij} be the true survival time of the j^{th} patient in the i^{th} cluster, $T_{ij} = \min(S_{ij}, C_{ij})$ the observed survival time, with C_{ij} the censoring time. I define an event indicator d_{ij} , which takes the value of 1 if $S_{ij} \leq C_{ij}$ and 0 otherwise.

2.1 Proportional hazards parametric survival models

The proportional hazards mixed effect survival model can be written as follows,

$$h_{ij}(t) = h_0(t) \exp \left[x_{ij}^T \beta + z_{ij}^T b_j \right] \quad (1)$$

where $h_0(t)$ is the baseline hazard function of either a standard parametric model, such as the exponential, Weibull or Gompertz distributions, or a more general spline based approach, such as using restricted cubic splines on the log hazard scale (Bower et al., 2016), or even a user-defined function. I define design matrices x_{ij} and z_{ij} for the fixed (β) and random (b_j) effects, respectively. I assume the random effects follow a multivariate normal distribution, with $b_j \sim N(0, \Sigma)$ (`stmixed` also allowed multivariate t -distributed random effects). If $z_{ij} = 1$ for all i and j , then Equation (1)

reduces to a shared frailty model, such as those available in `streg`, albeit with a different choice of frailty distribution.

2.2 Flexible parametric models

An alternative to the standard proportional hazards distributions is the flexible parametric model of Royston and Parmar (2002), modelled on the cumulative hazard scale, which has recently been extended to incorporate random effects by Crowther et al. (2014). Therefore,

$$H_{ij}(t) = H_0(t) \exp \left[x_{ij}^T \beta + z_{ij}^T b_j \right] \quad (2)$$

where $H_0(t)$ is the cumulative baseline hazard function. The spline basis for this specification is derived from the log cumulative hazard function of a Weibull proportional hazards model. The linear relationship with log time is relaxed through the use of restricted cubic splines. Further details can be found in (Royston and Parmar, 2002) and (Royston and Lambert, 2011). On the log cumulative hazard scale, we have,

$$\log\{H_{ij}(t)\} = \eta_{ij}(t) = s\{\log(t)|\gamma, \mathbf{k}_0\} + x_{ij}^T \beta + z_{ij}^T b_j \quad (3)$$

where $s(\cdot)$ are our basis functions, with knot vector \mathbf{k}_0 . Transforming to the hazard and survival scales, gives

$$h_{ij}(t) = \left[\frac{1}{t} \frac{ds\{\log(t)|\gamma, \mathbf{k}_0\}}{d\log(t)} \right] \exp(\eta_{ij}(t)), \quad S_{ij}(t) = \exp[-\exp(\eta_{ij}(t))] \quad (4)$$

In this framework I am assuming proportional cumulative hazards; however, this in fact implies proportional hazards, as in the models described in Section 2.1.

2.2.1 Non-proportional (cumulative) hazards

Relaxing the assumption of proportional hazards allows the investigation of whether the effect of a covariate changes with time. Termed non-proportional hazards, or time-dependent effects, the occurrence of which is commonplace in the analysis of survival data. Examples include treatment effects which vary over time (Mok et al., 2009), and in registry based studies, where follow-up can be substantial, covariate effects have been found to vary (Lambert et al., 2011).

Non-proportional cumulative hazards have been incorporated into the flexible parametric framework by Royston and Parmar (2002), achieved by interacting covariates with spline functions of log time and including them in the linear predictor (Lambert and Royston, 2009). This provides even greater flexibility in capturing complex effects, not restricted to linear functions of time. Equation (3) becomes

$$\log\{H_{ij}(t)\} = \eta_{ij}(t) = s\{\log(t)|\gamma, k_0\} + x_{ij}^T \beta + z_{ij}^T b_j + \sum_{r=1}^R s\{\log(t)|\delta_r, k_r\} x_{ijr} \quad (5)$$

Each time-dependent effect can have varying number of spline terms, depending on the number of knots, \mathbf{k}_r .

Within `stmixed`, time-dependent effects using restricted cubic splines can be used with all available models.

2.3 User-defined survival models

`stmixed` also allows the user to provide their own definitions for the hazard function, with or without also defining a cumulative hazard function, to allow the use of bespoke survival models with general hazard functions. For estimation, both the hazard and cumulative hazard functions are required (see Section 2.5 for further details), so when only the hazard is provided, the cumulative hazard function is calculated using numerical integration. Such a general implementation allows the user complete flexibility, whilst still being able to make use of the random effects engine within `stmixed`, which also syncs with the relative survival extension and delayed entry. More details on writing a user-defined function can be found in Crowther (2018).

2.4 Relative survival

Relative survival allows us to model the excess mortality associated with a diseased population compared to that of the general population, matched appropriately on the main factors associated with patient survival, such as age and gender (Dickman et al., 2004). For a recent extensive description and implementation of the tools available for relative survival analysis in Stata, along with a description of the differing approaches, I refer the reader to Dickman and Coviello (2015) and references therein.

Concentrating on applications of relative survival to cancer settings, the data generally comes from population based registries. Such data often exhibits a hierarchical structure, with patients nested within geographical regions such as counties. Patients living in the same area may share unobserved characteristics, such as environmental aspects or medical care access. Charvat et al. (2016) recently described a flexible relative survival model allowing a random intercept, with the baseline log hazard function modelled with B-splines, or restricted cubic splines. In this article, I extend the multilevel Royston-Parmar survival model described in Crowther et al. (2014), and essentially any other hazard based survival model, to the relative survival setting, further allowing any number of random effects, including random coefficients. Modelling on the log cumulative hazard scale avoids the need for numerical integration which is required when modeling on the log hazard scale with splines, and will generally require fewer spline terms than when modelling on the log hazard scale.

Within a multilevel modeling framework, I therefore define the total hazard at the time since diagnosis, t , for the j^{th} patient in the i^{th} cluster (area) to be $h_{ij}(t)$, with

$$h_{ij}(t) = h_{ij}^*(t) + \lambda_{ij}(t)$$

where

- $h_{ij}^*(t)$ is the expected mortality for the j^{th} patient in the i^{th} cluster (area)
- $\lambda_{ij}(t)$ is the excess mortality for the j^{th} patient in the i^{th} cluster (area)

and our model is

$$\lambda_{ij}(t) = \lambda_0(t) \exp(X_{ij}^T \beta + Z_{ij}^T b_i)$$

where $\lambda_0(t)$ is the baseline hazard function, with choices available including the exponential, Weibull, Gompertz, spline-based or user-defined.

Alternatively, I could model on the (log) cumulative excess hazard scale, using the flexible parametric model of Royston and Parmar (2002), where I define the total cumulative hazard at the time since diagnosis, t , for the j^{th} patient in the i^{th} cluster (area) to be $h_{ij}(t)$, with

$$H_{ij}(t) = H_{ij}^*(t) + \Lambda_{ij}(t)$$

where

- $H_{ij}^*(t)$ is the expected cumulative mortality for the j^{th} patient in the i^{th} cluster (area)
- $\Lambda_{ij}(t)$ is the excess cumulative mortality for the j^{th} patient in the i^{th} cluster (area)

and our model is

$$\Lambda_{ij}(t) = \Lambda_0(t) \exp(X_{ij}^T \beta + Z_{ij}^T b_i)$$

where $\Lambda_0(t)$ is the baseline cumulative hazard function, modeled with restricted cubic splines.

2.5 Likelihood and estimation

Defining the likelihood for the i^{th} cluster under the mixed effects framework, I define

$$L_i(\theta) = \int_{-\infty}^{\infty} \left[\prod_{j=1}^{n_i} p(T_{ij}, d_{ij} | b_i, \theta) \right] p(b_i | \theta) db_i \quad (6)$$

with parameter vector θ . Under a hazard scale model

$$p(T_{ij}, d_{ij} | b_i, \theta) = h(T_{ij})^{d_{ij}} \exp \left[- \int_0^{T_{ij}} h(u) du \right] \quad (7)$$

with $h(T_{ij})$ defined in Equation (1). Under the flexible parametric survival model

$$p(T_{ij}, d_{ij} | b_i, \theta) = \left[\left\{ \frac{1}{T_{ij}} \frac{ds\{\log(T_{ij})|\gamma, k_0\}}{d \log(T_{ij})} \right\} \exp(\eta_{ij}) \right]^{d_{ij}} \exp \{- \exp(\eta_{ij})\} \quad (8)$$

Finally, I assume the random effects follow a multivariate normal distribution

$$p(b_i | \theta) = (2\pi|\Sigma|)^{-q/2} \exp \left\{ - \frac{b_i' \Sigma^{-1} b_i}{2} \right\}, \quad (9)$$

with variance-covariance matrix, Σ , and q the number of random effects. The (possibly multi-dimensional) integral in Equation (6) is analytically intractable, requiring numerical techniques to evaluate. `stmixed` uses either m -point mean-variance adaptive or non-adaptive Gauss-Hermite

quadrature (Pinheiro and Bates, 1995; Rabe-Hesketh et al., 2002; Liu and Huang, 2008), or Monte-Carlo integration. The default estimation method first fits the appropriate fixed effects model, followed by the full model with variance and covariance parameters given starting values of 0 and 1, respectively. `stmixed` also allows multivariate t -distributed random effects, with specified degrees of freedom, in which case only Monte-Carlo integration is supported.

2.6 Relative survival likelihood

The adaption to the likelihood in Equation (6) to turn it into a relative survival model is relatively simple. All that is needed is the expected mortality rate at each event time, which are usually obtained from national/regional life tables. Under a hazard scale model, Equation (7) becomes,

$$p(T_{ij}, d_{ij}|b_i, \theta) = [h^*(T_{ij}) + \lambda(T_{ij})]^{d_{ij}} \exp \left[- \int_0^{T_{ij}} \lambda(u) \, du \right] \quad (10)$$

and under a cumulative hazard scale model, Equation (11) becomes,

$$p(T_{ij}, d_{ij}|b_i, \theta) = \left[\left\{ \frac{1}{T_{ij}} \frac{ds\{\log(T_{ij})|\gamma, k_0\}}{d \log(T_{ij})} \right\} \exp(\eta_{ij}) \right]^{d_{ij}} \exp \{-\exp(\eta_{ij})\} \quad (11)$$

which provides substantial extensions to the relative survival literature.

2.7 Left-truncation/delayed entry

The addition of left-truncation/delayed entry within a random effects survival setting raises a particular extra level of complexity. The random effects distributions are defined at $t = 0$, and as such, the left truncation time point is conditional on on each patient's subject-specific random effect contributions. For more details I refer the reader to van den Berg and Drepper (2014). As such, our likelihood function becomes,

$$L_i(\theta|T_{0i}) = \frac{\int_{-\infty}^{\infty} \left[\prod_{j=1}^{n_i} p(T_{ij}, d_{ij}|b_i, \theta) \right] p(b_i|\theta) \, db_i}{S(T_{0i}|\theta)} \quad (12)$$

where $S(T_{0i}|\theta)$ is the marginal survival function at the entry time T_{0i} , defined as

$$S(T_{0i}|\theta) = \int_{-\infty}^{\infty} S(T_{0i}|b_i, \theta) p(b_i|\theta) \, db_i$$

as such, there are two sets of analytically intractable integrals to evaluate in Equation (12), which increases computation time.

3 The `stmixed` command

3.1 Syntax

```
stmixed [fe_equation] || re_equation [|| re_equation] [, options ]
```

where the syntax of *fe_equation* is

```
[varlist][if][in]
```

and the syntax of *re_equation* is

```
levelvar: [varlist][, noconstant ]
```

`stmixed` requires that your data is `stset`.

3.2 Options

3.2.1 Model

`noconstant` suppresses the constant (intercept) term and may be specified for the fixed effects equation and for the random effects equations.

`distribution(string)` specifies the survival distribution.

`distribution(exponential)` fits an exponential survival model.

`distribution(weibull)` fits a Weibull survival model.

`distribution(gompertz)` fits a Gompertz survival model.

`distribution(rp)` fits a Royston-Parmar survival model. This is a highly flexible fully parametric alternative to the Cox model, modelled on the log cumulative hazard scale using restricted cubic splines.

`distribution(rcs)` fits a log hazard scale flexible parametric survival model. This is a highly flexible fully parametric alternative to the Cox model, modelled on the log hazard scale using restricted cubic splines.

`distribution(user)` specify a user-defined survival model; see options below and `help merlin user-defined functions`.

`df(#)` specifies the degrees of freedom for the restricted cubic spline function used for the baseline function under a `rp` or `rcs` survival model. `#` must be between 1 and 10, but usually a value between 1 and 5 is sufficient. The `knots()` option is not applicable if the `df()` option is specified. The knots are placed at the evenly spaced centiles of the distribution of the uncensored log survival times. Note that these are `interior knots` and there are also boundary knots placed at the minimum and maximum of the distribution of uncensored survival times.

`knots(numlist)` specifies knot locations for the baseline distribution function under a `rp` or `rcs` survival model, as opposed to the default locations set by `df()`. Note that the locations of the

knots are placed on the standard time scale. However, the scale used by the restricted cubic spline function is always log time. Default knot positions are determined by the `df()` option.

`tvc(varlist)` gives the name of the variables that have time-varying coefficients. Time-dependent effects are fitted using restricted cubic splines. The degrees of freedom are specified using the `dftvc()` option.

`dftvc(df_list)` gives the degrees of freedom for time-dependent effects in `df_list`. The potential degrees of freedom are listed under the `df()` option. With 1 degree of freedom a linear effect of log time is fitted. If there is more than one time-dependent effect and different degrees of freedom are requested for each time-dependent effect then the following syntax applies:

`knotstvc(knotslist)` defines numlist `knotslist` as the location of the interior knots for time-dependent effects.

`bhazard(varname)` specifies the variable which contains the expected mortality rate, which invokes a relative survival model.

`covariance(vartype_list)`, where each `vartype` is

`diagonal | exchangeable | identity | unstructured`

specifies the structure of the covariance matrix for the random effects. An `diagonal` covariance structure allows a distinct variance for each random effect within a random-effects equation and assumes that all covariances are zero. `exchangeable` covariances have common variances and one common pairwise covariance. `identity` is short for “multiple of the identity”; that is, all variances are equal and all covariances are zero. `unstructured` allows for all variances and covariances to be distinct. If an equation consists of p random-effects terms, the `unstructured` covariance matrix will have $p(p + 1)/2$ unique parameters. `covariance(diagonal)` is the default.

3.2.2 Integration

`intmethod(intmethod)`, `intpoints(#)`, and `adaptopts(adaptopts)` affect how integration for the latent variables is numerically calculated.

`intmethod(intmethod)` specifies the method and defaults to `intmethod(mvaghermite)`. The current implementation uses mean-variance adaptive quadrature at the highest level, and non-adaptive at lower levels. Sometimes it is useful to fall back on the less computationally intensive and less accurate `intmethod(ghermite)` and then perhaps use one of the other more accurate methods.

`intmethod(mcarlo)` tells `stmixed` to use Monte-Carlo integration, which either uses Halton sequences with normally-distributed random effects, or anti-thetic random draws with t -distributed random effects.

`intpoints(#)` specifies the number of integration points to use and defaults to `intpoints(7)` with `intmethod(mvaghermite)` or `intmethod(ghermite)`, and `intpoints(150)` with `intmethod(mcarlo)`. Increasing the number increases accuracy but also increases computational time. Computational time is roughly proportional to the number specified.

`adaptopts(adaptopts)` affects the adaptive part of adaptive quadrature (another term for numerical integration) and thus is relevant only for `intmethod(mvaghermite)`.

`adaptopts()` defaults to `adaptopts(nolog iterate(1001) tolerance(1e-8))`.

[no]log specifies whether iteration logs are shown each time a numerical integral is calculated.

3.2.3 Estimation

from(matname) allows you to specify starting values.

restartvalues(sv_list) allows you to specify starting values for specific random effect variances. See merlin estimation for further details.

apstartvalues(#) allows you to specify a starting value for all ancillary parameters, i.e those defined by using the nap() option.

zeros tells stmixed to use 0 for all parameters starting values, rather than fit the fixed effect model. Both restartvalues() and apstartvalues() can be used with zeros.

maximization_options: difficult, technique(algorithm_spec), iterate(#), [no]log, trace, gradient, showstep, hessian, shownrtolerance, tolerance(#), ltolerance(#), gtolerance(#), nrtolerance(#), nonrtolerance, from(init_specs); see [R] **maximize**. These options are seldom used, but the difficult option may be useful if there are convergence problems.

3.2.4 Reporting

showmerlin displays the merlin syntax used to fit the model.

level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is level(95) or as set by set level.

4 stmixed postestimation

4.1 Syntax for obtaining predictions

```
predict newvarname [if] [in] [, eta hazard survival chazard cif rmst timelost fixedonly  
marginal at(varname # [varname # ...]) ci timevar(varname) level(#) ]
```

4.2 Options

4.2.1 Predictions

eta calculates the expected value of the linear predictor

hazard calculates the predicted hazard.

survival calculates each observation's predicted survival probability.

chazard calculates the predicted cumulative hazard.

cif calculates the predicted cumulative incidence function.

`rmst` calculates the restricted mean survival time, i.e. the integral of the survival function up to time t .

`timelost` calculates the time lost due to the event occurring, i.e. the integral of the cumulative incidence function up to time t .

4.2.2 Subsidiary

`fixedonly` specifies predictions based on the fixed portion of the model.

`marginal` specifies predictions calculated marginally with respect to the random effects, i.e. population-averaged predictions.

`at(varname #[varname #...])` requests that the covariates specified by the listed `varname(s)` be set to the listed `#` values. For example, `at(x1 1 x3 50)` would evaluate predictions at $x_1 = 1$ and $x_3 = 50$. This is a useful way to obtain out of sample predictions. Note that if `at()` is used together with zeros all covariates not listed in `at()` are set to zero. If `at()` is used without zeros then all covariates not listed in `at()` are set to their sample values. See also `zeros`.

`ci` calculate confidence interval and store in `newvarname_lci` and `newvarname_uci`. The delta-method is used in all calculations using `predictnl`.

`timevar(varlist)` defines the variable used as time in the predictions. Default is `_t`.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is as set by `set level`.

5 Examples

In this section, I illustrate the command in two areas of research, namely, recurrent events analysis, and the individual participant data meta-analysis of survival data.

5.1 Recurrent event data

I consider the commonly used `catheter` dataset consisting of 38 patients with kidney disease (McGilchrist and Aisbett, 1991). The outcome of interest is infection at the catheter insertion point, with our baseline being time of initial catheter insertion. Patients can experience up to two recurrences of infection, resulting in a total of 58 events. In the examples I use the Royston-Parmar model for illustration. I initially fit a null model, i.e. no covariates and no random effects, to select the degrees of freedom for the baseline cumulative hazard function, using the Akaike Information Criterion to guide the choice. This selected 3 degrees of freedom (not shown), clearly indicating the need for a flexible spline based model to capture the complex hazard function. I now fit a Royston-Parmar proportional hazards model with a normally distributed frailty, adjusting for age and gender,

```
. webuse catheter, clear  
(Kidney data, McGilchrist and Aisbett, Biometrics, 1991)
```

```
. stset time, fail(infect)
```

```
failure event: infect != 0 & infect < .  
obs. time interval: (0, time]  
exit on or before: failure
```

```
-----  
76 total observations  
0 exclusions  
-----
```

```
76 observations remaining, representing  
58 failures in single-record/single-failure data  
7,424 total analysis time at risk and under observation  
at risk from t = 0  
earliest observed entry t = 0  
last observed exit t = 562
```

```
. stmixed age female || patient:, distribution(rp) df(3)  
Random effect M1: Intercept at level patient  
variables created: _rcs1_1 to _rcs1_3
```

Fitting fixed effects model:

Fitting full model:

```
Iteration 0: log likelihood = -100.8752  
Iteration 1: log likelihood = -95.540155  
Iteration 2: log likelihood = -94.989854  
Iteration 3: log likelihood = -94.921004  
Iteration 4: log likelihood = -94.9208  
Iteration 5: log likelihood = -94.920802
```

```
Mixed effects survival model Number of obs = 76  
Log likelihood = -94.920802
```

```
-----  
| Coef. Std. Err. z P>|z| [95% Conf. Interval]  
-----+-----  
_t: |  
age | .0071588 .0129147 0.55 0.579 -.0181536 .0324712  
female | -1.467426 .4928409 -2.98 0.003 -2.433376 -.5014755  
M1[patient] | 1 . . . . .  
_cons | -.3469469 .6710707 -0.52 0.605 -1.662221 .9683276  
-----+-----  
patient: |  
sd(M1) | .801154 .2732038 .410625 1.563099  
-----
```

Warning: Baseline spline coefficients not shown - use ml display

As `stmixed` calls `merlin`, the naming of random effects in the results table follows `merlin` conventions, in that random effects are named with a `M` and a number. Therefore, `stmixed` first provides some text ensuring the user can understand which random effects correspond to what. It also reports creating some spline variables, named `_rcs#_1`, which are the baseline splines for the Royston-Parmar model. The estimation procedure by default fits the fixed effect only model, to obtain starting values for the full model. Random effect variances are given a starting value of 1, with any covariances given a starting value of 0.

The model estimates a hazard ratio of 0.231 (95% CI: 0.088, 0.606) for a female compared to a male of the same age, and a non-statistically significant age effect. The estimated frailty standard deviation is 0.801 (95% CI: 0.411, 1.563), indicating a highly heterogenous baseline hazard function.

5.1.1 Predictions

A variety of predictions can be obtained following the fitting of a model. I can obtain the predicted survival function, shown in Figure 1 with 95% confidence interval, based on the fixed portion of the model, for a female aged 45, through use of the `at()` option, as follows

```
. predict s1, survival ci at(age 45 female 1)
note: confidence intervals calculated using Z critical values
```

Which can be plotted by

```
. twoway rarea s1_lci s1_uci _t, sort || line s1 _t, sort    ///
>   ylabel(,angle(h) format(%2.1f))                        ///
>   xtitle("Follow-up time (days)")                       ///
>   ytitle("Survival probability")                         ///
>   legend(order(2 "Predicted survival" 1 "95% CI")      ///
>   ring(0) pos(1) cols(1))
```

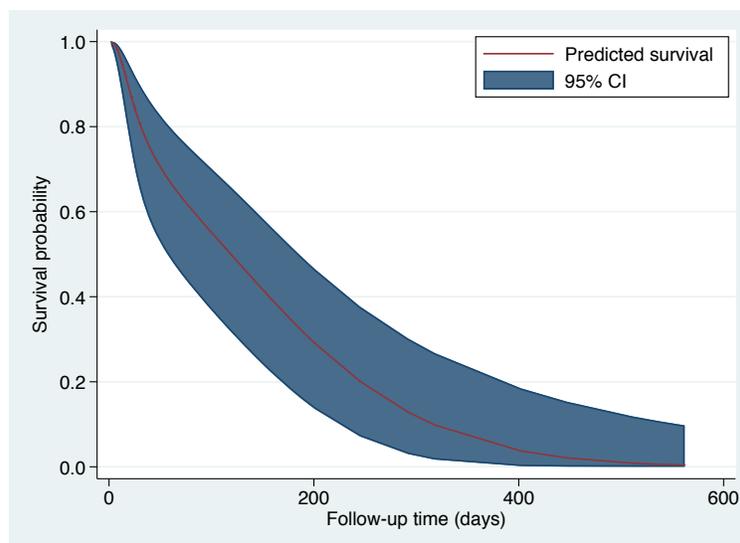


Figure 1: Predicted survival for a female, aged 45, based on the fixed portion of the model.

To compare across covariate patterns, for example, to assess the impact of gender, we can predict restricted mean survival as follows,

```

. predict rmst_male , rmst marginal ci at(age 45 female 0)
note: confidence intervals calculated using Z critical values

. predict rmst_female, rmst marginal ci at(age 45 female 1)
note: confidence intervals calculated using Z critical values

and plotting,

. twoway (rarea rmst_male_lci rmst_male_uci _t, sort) ///
> (line rmst_male _t, sort) ///
> (rarea rmst_female_lci rmst_female_uci _t, sort) ///
> (line rmst_female _t, sort) ///
> , ylabel(,angle(h) format(%2.1f)) ///
> xtitle("Follow-up time (days)") ///
> ytitle("Restricted mean survival time") ///
> legend(order(2 "Male" 4 "Female"))

```

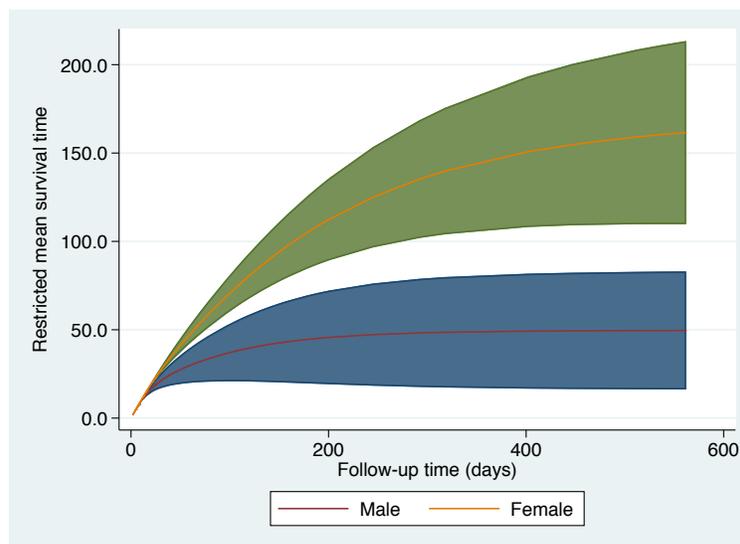


Figure 2: Restricted mean survival for a male or female aged 45.

where 2 shows restricted mean survival as a function of time, for a male or female patient with the same age of 45. The impact of being female is shown clearly, indicating females live substantially longer than a male of the same age.

5.2 Individual participant data meta-analysis of survival data

In this example, I will illustrate a simple approach to simulating clustered survival data, in the setting of an IPD meta-analysis of survival data, through the use of the `survsim` command (Crowther and Lambert, 2012, 2013). I assume a scenario where I have data from 30 trials, each with 100 patients. Each trial compared a treatment to a control, with the probability of being assigned to each arm being 50%. I assume that the treatment effect for each trial comes from a normal distribution, $N(-0.5, 0.5^2)$, i.e. an average log hazard ratio of -0.5 (hazard ratio = $\exp(-0.5) = 0.607$), with heterogeneity standard deviation of 0.5. I assume a Weibull baseline hazard function, with scale

and shape parameter values of $\lambda = 0.1$ and $\gamma = 1.2$, indicating 50.2% survival in the control group after 5 years, at which time administrative censoring is assumed.

```
. clear

. // Set seed for reproducibility
. set seed 278945

. // Assume we have 30 trials
. set obs 30
number of observations (_N) was 0, now 30

. // Generate a trial id variable
. gen trialid = _n

. // Generate trial specific treatments effect (log hazard ratio) from a
. // normal distribution with mean 0.5, and std. dev. 0.5
. gen trteffect = rnormal(-0.5,0.5)

. // Assume 100 patients in each trial
. expand 100
(2,970 observations created)

. // Generate 0/1 patient level treatment group indicator
. gen trt = runiform()>0.5

. // Generate a variable containing patient specific treatment
. // effects for use in simulation
. gen trteffectsim = trt*trteffect

. // Simulate survival times from a Weibull distribution,
. // incorporating the random treatment effect
. survsim stime died, dist(weibull) lambda(0.1) gamma(1.2) ///
> covariates(trteffectsim 1) maxt(5)

. stset stime, f(died)

      failure event:  died != 0 & died < .
obs. time interval:  (0, stime]
exit on or before:   failure

-----
      3,000 total observations
           0 exclusions
-----

      3,000 observations remaining, representing
      1,239 failures in single-record/single-failure data
11,877.683 total analysis time at risk and under observation
```

```

                                at risk from t =      0
                    earliest observed entry t =      0
                                last observed exit t =    5

```

```

. stmixed trt || trialid: trt, nocons distribution(weibull)
Random effect M1: trt at level trialid

```

Fitting fixed effects model:

Fitting full model:

```

Iteration 0:  log likelihood = -3945.0537
Iteration 1:  log likelihood = -3943.6472
Iteration 2:  log likelihood = -3940.1832
Iteration 3:  log likelihood = -3939.8849
Iteration 4:  log likelihood = -3939.8835
Iteration 5:  log likelihood = -3939.8835

```

```

Mixed effects survival model                    Number of obs    =      3,000
Log likelihood = -3939.8835

```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	

_t:						
trt	-.4622045	.1240911	-3.72	0.000	-.7054186	-.2189904
trt#M1[tri~]	1
_cons	-2.415182	.0608592	-39.68	0.000	-2.534464	-2.2959
log(gamma)	.2017346	.0264974	7.61	0.000	.1498006	.2536686

trialid:						
sd(M1)	.5914661	.0934644			.4339319	.8061913

A key trick to note here is in the `survsim` command, I included the variable `trteffectsim` and assigned it a coefficient value of 1. This allows you to seamlessly incorporate random effects on covariates which are included in the linear predictor, multiplied by a coefficient of 1. Then when the model is fitted using `stmixed`, the `trt` variable is used which indicates treatment group.

In the `stmixed` model fit, I enter `trt` as both a fixed and random effect, but using the `nocons` option to indicate no random intercept. This is a rather restrictive model as it assumes that each trial has the same baseline hazard function. In practice, you may include the trial id variable in the linear predictor, to allow proportional trials effects, or indeed stratify by trial membership to allow separate trials effects (Crowther et al., 2012, 2014), or allow a random intercept at the trial level.

6 Conclusion

In this paper, I have introduced the `stmixed` command for multilevel mixed effects survival analysis. `stmixed` provides substantial extensions to `mestreg`, including flexible spline-based survival models, user-defined survival models, the extension to relative multilevel survival, simple or complex time-dependent effects, and t -distributed random effects. I hope the wide range of survival models available will be found useful in applied research. The most up-to-date version of `stmixed` can be installed using `ssc install stmixed`.

Given that `stmixed` is essentially a shell file for `merlin`, any improvements that are implemented in `merlin` will filter up to `stmixed`.

About the author

Michael J. Crowther is a Lecturer in Biostatistics at the University of Leicester. He works heavily in methods and software development, particularly in the field of survival analysis. He is currently part funded by a MRC New Investigator Research Grant (MR/P015433/1).

References

- van den Berg, G. J., and B. Drepper. 2014. Inference for shared-frailty survival models with left-truncated data. *Econometric Reviews* .
- Bower, H., M. J. Crowther, and P. C. Lambert. 2016. `strcs`: A command for fitting flexible parametric survival models on the log-hazard scale. *Stata Journal* 16(4): 989–1012. URL <http://www.stata-journal.com/article.html?article=st0462>.
- Charvat, H., L. Remontet, N. Bossard, L. Roche, O. Dejardin, B. Rachet, G. Launoy, and A. Belot. 2016. A multilevel excess hazard model to estimate net survival on hierarchical data allowing for non-linear and non-proportional effects of covariates. *Statistics in medicine* .
- Crowther, M. J. 2017. Extended multivariate generalised linear and non-linear mixed effects models. *arXiv preprint arXiv:1710.02223* .
- . 2018. `merlin` - a unified framework for data analysis and methods development in Stata. *Submitted* .
- Crowther, M. J., and P. C. Lambert. 2012. Simulating complex survival data. *Stata J* 12(4): 674–687.
- . 2013. Simulating biologically plausible complex survival data. *Stat Med* 32(23): 4118–4134. URL <http://dx.doi.org/10.1002/sim.5823>.
- Crowther, M. J., M. P. Look, and R. D. Riley. 2014. Multilevel mixed effects parametric survival models using adaptive Gauss-Hermite quadrature with application to recurrent events and individual participant data meta-analysis. *Stat Med* 33(22): 3844–3858. URL <http://dx.doi.org/10.1002/sim.6191>.

- Crowther, M. J., R. D. Riley, J. A. Staessen, J. Wang, F. Gueyffier, and P. C. Lambert. 2012. Individual patient data meta-analysis of survival data using Poisson regression models. *BMC Med Res Methodol* 12(34). URL <http://dx.doi.org/10.1186/1471-2288-12-34>.
- Dickman, P. W., and E. Coviello. 2015. Estimating and modeling relative survival. *Stata Journal* 15(1): 186–215(30). URL <http://www.stata-journal.com/article.html?article=st0376>.
- Dickman, P. W., A. Sloggett, M. Hills, and T. Hakulinen. 2004. Regression models for relative survival. *Stat Med* 23(1): 51–64. URL <http://dx.doi.org/10.1002/sim.1597>.
- Gutierrez, R. G. 2002. Parametric frailty and shared frailty survival models. *The Stata Journal* 2(1): 22–44.
- Lambert, P. C., L. Holmberg, F. Sandin, F. Bray, K. M. Linklater, A. Purushotham, D. Robinson, and H. Møller. 2011. Quantifying differences in breast cancer survival between England and Norway. *Cancer Epidemiol* 35(6): 526–533. URL <http://dx.doi.org/10.1016/j.canep.2011.04.003>.
- Lambert, P. C., and P. Royston. 2009. Further development of flexible parametric models for survival analysis. *Stata J* 9: 265–290.
- Liu, L., and X. Huang. 2008. The use of Gaussian quadrature for estimation in frailty proportional hazards models. *Stat Med* 27(14): 2665–2683. URL <http://dx.doi.org/10.1002/sim.3077>.
- McGilchrist, C. A., and C. W. Aisbett. 1991. Regression with frailty in survival analysis. *Biometrics* 47(2): 461–466.
- Mok, T. S., Y.-L. Wu, S. Thongprasert, C.-H. Yang, D.-T. Chu, N. Saijo, P. Sunpaweravong, B. Han, B. Margono, Y. Ichinose, Y. Nishiwaki, Y. Ohe, J.-J. Yang, B. Chewaskulyong, H. Jiang, E. L. Duffield, C. L. Watkins, A. A. Armour, and M. Fukuoka. 2009. Gefitinib or carboplatin-paclitaxel in pulmonary adenocarcinoma. *N Engl J Med* 361(10): 947–957. URL <http://dx.doi.org/10.1056/NEJMoa0810699>.
- Pinheiro, J. C., and D. M. Bates. 1995. Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model. *J Comput Graph Statist* 4(1): pp. 12–35.
- Rabe-Hesketh, S., A. Skrondal, and A. Pickles. 2002. Reliable estimation of generalized linear mixed models using adaptive quadrature. *Stata J* 2: 1–21.
- Royston, P., and P. C. Lambert. 2011. *Flexible Parametric Survival Analysis Using Stata: Beyond the Cox Model*. Stata Press.
- Royston, P., and M. K. B. Parmar. 2002. Flexible Parametric Proportional Hazards and Proportional Odds Models for Censored Survival Data, with Application to Prognostic Modelling and Estimation of Treatment Effects. *Stat Med* 21(15): 2175–2197.
- Simmonds, M. C., J. P. T. Higgins, L. A. Stewart, J. F. Tierney, M. J. Clarke, and S. G. Thompson. 2005. Meta-analysis of individual patient data from randomized trials: a review of methods used in practice. *Clin Trials* 2(3): 209–217.
- Tudur-Smith, C., P. R. Williamson, and A. G. Marson. 2005. Investigating heterogeneity in an individual patient data meta-analysis of time to event outcomes. *Stat Med* 24(9): 1307–1319.